Foundations of Semantics III: Quantification and Raising

Rick Nouwen | R.W.F.Nouwen@uu.nl

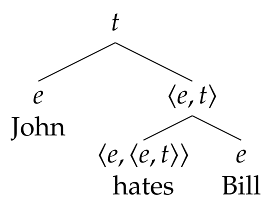
February/March 2011, Gent

---

1 Quantification and type-logical composition

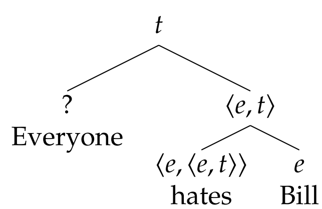
So far:

(1)



What happens when the verb takes a quantificational argument like *everyone*?

(2)



1.1. Are quantificational expressions of type e?

(3) I see nobody on the road.
('I only wish I had such eyes,' the King remarked in a fretful tone. 'To be able to see Nobody! And at that distance too!')

(4) Somebody told me that nothing works faster than aspirin. So, I went out immediately to try and by nothing. (example due to Hodges)

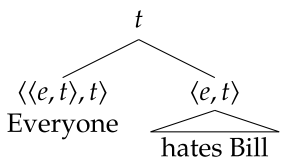
(5) No boy hates his mother. *Which type e object would* no boy *refer to?*

(6) Desideratum:
a. Everyone hates Bill
b. $\forall x[person(x) \rightarrow hate(x, b)]$

1.2. The proper treatment of quantification in natural language

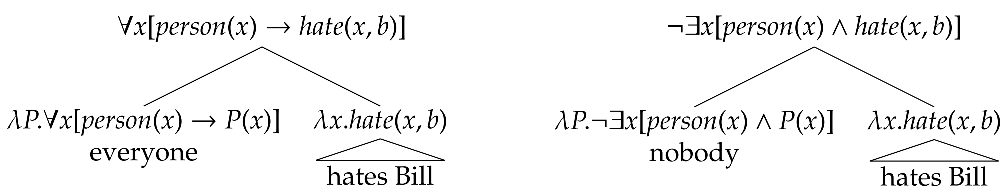
If quantificational subjects are not of type e, then there is no analysis of (2) analogous to (1). The solution, due to Montague (1973), is that subjects are not the type-logical argument of the verb phrase. Rather, the verb phrase is the argument of the subject. That is, quantificational subjects are of type $\langle\langle e,t\rangle,t\rangle$.

(7)



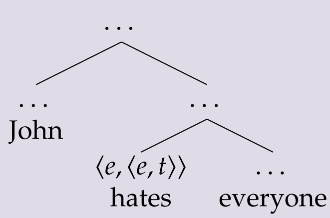
(8) a. Everyone: $\lambda P_{\langle e,t\rangle}.\forall x[person(x) \rightarrow P(x)]$
b. Nobody: $\lambda P_{\langle e,t\rangle}.\neg\exists x[person(x) \wedge P(x)]$

(9)



The strategy we explained above works well for quantificational subjects, but as it stands now it does not work for objects. Explain the problem in detail.

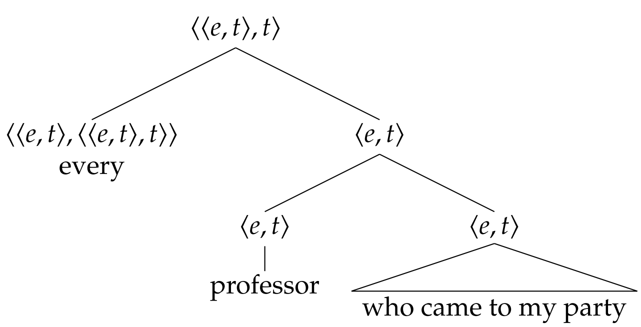
(10)



1.3. Determiner semantics

Given that we know that quantifiers are of type $\langle\langle e,t\rangle,t\rangle$, we can now find out what is the type of determiners. Determiners combine with a noun phrase of type $\langle e,t\rangle$ to return a $\langle\langle e,t\rangle,t\rangle$, and their type is therefore quite complex:

(11)



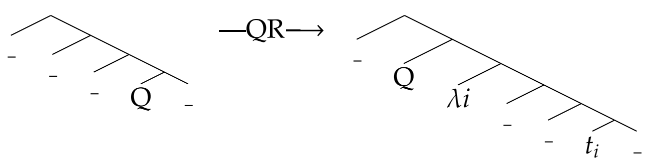
The type $\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$ is the type of functions that take a property, then take a second property, to return a proposition. What this means is that these functions are *relations between properties*. For instance, *every* is the relation between properties such that all entities that have the first one, have the second one too. *No* is the relation between properties such that no individual who has the first one has the second one. The lexical entries are as follows:

(12)
a. $[\![\text{every}]\!] = \lambda P.\lambda P'.\forall x[P(x) \rightarrow P'(x)]$
b. $[\![\text{no}]\!] = \lambda P.\lambda P'.\neg\exists x[P(x) \wedge P'(x)]$
c. $[\![\text{some}]\!] = \lambda P.\lambda P'.\exists x[P(x) \wedge P'(x)]$

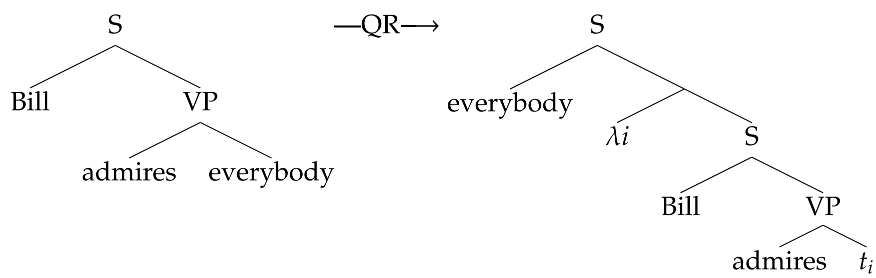
2 Quantificational objects: Quantifier Raising

One way of extending the $\langle\langle e,t\rangle,t\rangle$ strategy for quantificational noun phrases to cover quantificational objects is to use movement at LF. As we will see below, this operation can be independently motivated.

(13) **Quantifier Raising (QR)**: at LF quantifiers can raise to adjoin to any propositional node, leaving behind a trace and introducing a lambda abstraction over the trace's variable

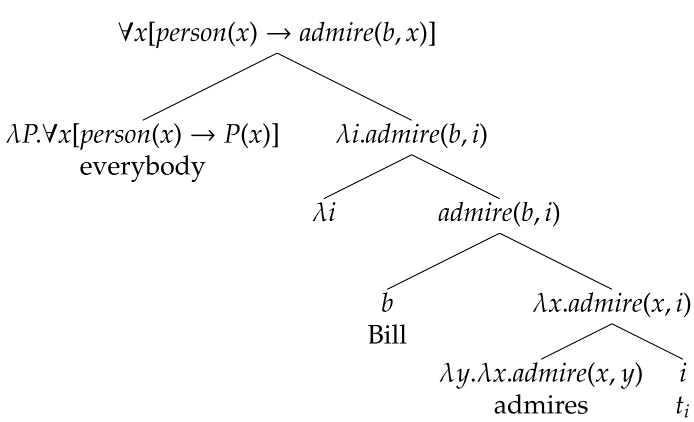


(14)



Note: The inserted lambda operator has no syntactic significance, but is a prerequisite for interpretation

(15)



3 Type shifting

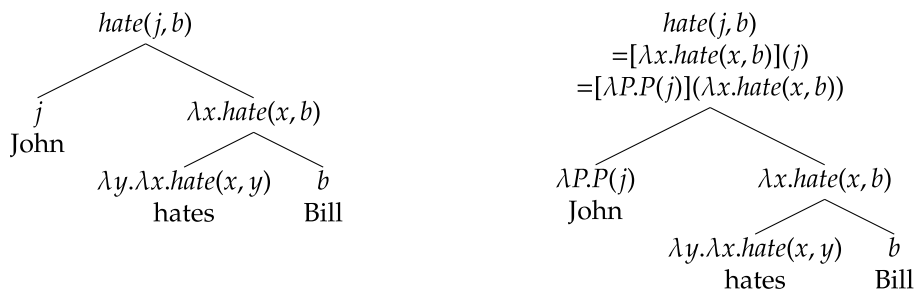
3.1. *John* as a quantifier

An interesting aspect of the now standard approach to the interpretation of quantificational arguments is that this approach is available for *all kinds* of syntactic arguments, including proper names.

(16) Two options for *John*

a. j
b. $\lambda P_{\langle e,t\rangle}.P(j)$

(17)



The fact that proper names can have a quantifier type is not so surprising, given the fact that they can be coordinated with quantifiers. (We're assuming here that coordination always involves expressions of the same type.)

(18) John and every other student passed the exam.

There are also reasons to believe that proper names are not always of the $\langle\langle e,t\rangle,t\rangle$-type, for they are in some ways different from quantifiers like *every NP*. Most importantly, proper names are *referential* in the sense that they refer to entities that can be picked up by pronouns in the subsequent discourse. Quantificational DPs lack this property.

(19) Every boy failed the exam. #He was very disappointed.

(20) John failed the exam. He was very disappointed.

3.2. Definite descriptions

Like proper names, definite descriptions also appear to lead a type-logical double life:

(21) The best student in class failed the exam. He was very disappointed.

(22) I failed the student whose name I forgot and most students whose name I never knew.

3.3. $\langle\langle e,t\rangle,t\rangle$, and $\langle e,t\rangle$, and e

So far, we have seen that DPs can be of type $\langle\langle e,t\rangle,t\rangle$ (true quantifiers, proper names and definite descriptions) and of type e (proper names and definite descriptions). Some DPs can be of type $\langle e,t\rangle$. The best example are indefinites. Here is some evidence for this:

(23) Predicative position (complement of *be* / *consider*:

a. John is lazy.
b. John is a student.
c. #These people here are every student.
d. Mary considers John lazy.
e. Mary considers John a nerd.
f. #Mary considers these people many students.

(24) Coordination with adjectives:

Mary consider John lazy and a nerd.

3.4. Partee's BE

One way to deal with the observed type-flexibility is to assume that there are mechanisms available that type-shift meanings from one type into another. Here is one example, Partee's operator that shifts an $\langle\langle e,t\rangle,t\rangle$-type function into a $\langle e,t\rangle$-type one.

(25) $\text{BE} = \lambda\mathcal{P}_{\langle\langle e,t\rangle,t\rangle}.\lambda z.\mathcal{P}(\lambda z'.z = z')$

Here is an example:

(26) $[\![\text{a student}]\!] = \lambda P.\exists x[student(x) \wedge P(x)]$

(27) $[\![[\text{BE [a student] }]]\!] = \lambda\mathcal{P}_{\langle\langle e,t\rangle,t\rangle}.\lambda z.\mathcal{P}(\lambda z'.z = z')\ (\lambda P.\exists x[student(x) \wedge P(x)])$
$= \lambda z.\ [[\lambda P.\exists x[student(x) \wedge P(x)](\lambda z'.z = z')]$
$= \lambda z.\ [\exists x[student(x) \wedge [\lambda z'.z = z'](x)]]$
$= \lambda z.\exists x[student(x) \wedge z = x]$
$= \lambda z.student(z)$

The last step is crucial. The penultimate line presents a function that takes an argument and returns *true* if there is something identical to this argument which is a student. In other words, it returns true for all and only the students. In other words, this function is the property of being a student.

A similar final step is unavailable if you were to apply BE to a universal quantifier.

(28) $[\![\text{every student}]\!] = \lambda P.\forall x[student(x) \rightarrow P(x)]$

(29) $[\![[\text{ BE [every student]] }]\!] = \lambda z.\forall x[student(x) \rightarrow z = x]$

This states the function that takes an argument and returns *true* if and only if every student is identical to this argument. This only makes sense if there is exactly 1 individual in the whole world (in which case this function is either the set containing this individual, if s/he is a student, or the empty set if s/he is not). If there is more than 1 individual, this property will express the empty set. (For instance, if Bob and Bill are both students, then Bob is not identical to every student and neither is Bill.)

The upshot is this. On the assumption that we will never use natural language to express thoughts about a world with just a single individual, applying BE to *every student* will never result in a sensible property: it invariably results in the empty set. This is why the shift from $\langle\langle e,t\rangle,t\rangle$ to $\langle e,t\rangle$ is available for indefinites, but not to true quantifiers like *every NP*. This explains the difference in distribution between the two.

4 Using Logical Form for scope ambiguity

(30) Someone admires everyone

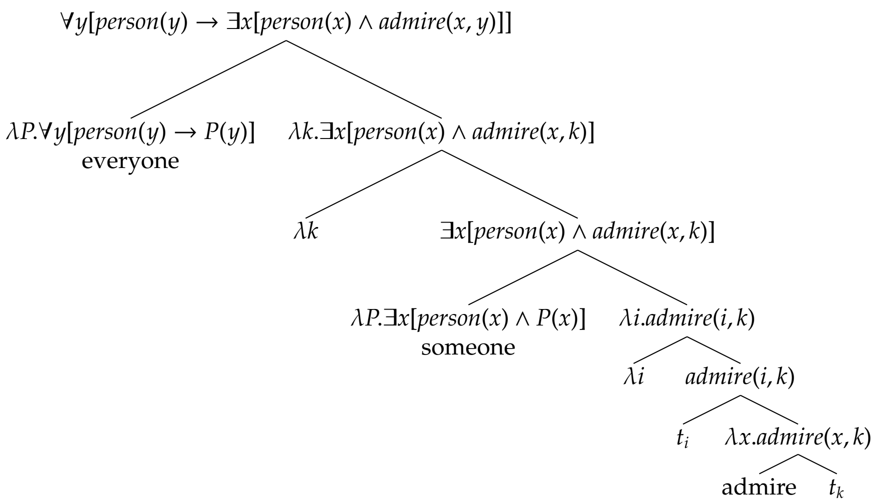
a. $\forall y[person(y) \rightarrow \exists x[person(x) \wedge admire(x,y)]$

b. $\exists x[person(x) \wedge \forall y[person(y) \rightarrow admire(x,y)]]$

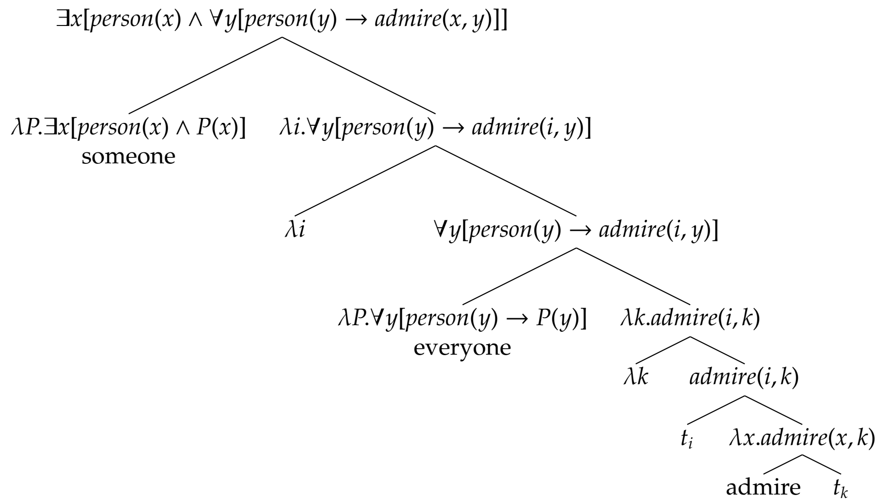
What is the relation between the single surface order (31) and its two readings? One solution: There is a level of syntactic representation that is derived from the surface structure by covert movement.

(31) [Someone [admires everyone]] Surface Structure

(32) Logical Form A



(33) Logical Form B



5 QR and subtypes of quantifiers

As I have presented it, so far, QR is unconstrained and indiscriminate. All DPs can QR and they can all attach to the same positions. There are reasons to believe that this is too simplistic. One reason is that QR seems to be a form of movement that is constrained in pretty much the same ways as overt wh-movement. We will turn to this in the final section of this handout. In this section we will briefly look at different kinds of quantifiers and differences in their scopal behaviour.

(34) Every man read two of the books.

a. ... namely Aspects and Lectures in Government and Binding — two>every
b. ... but none of them read the same two books — every>two

(35) Every man read more than two books.

a. ... # namely Aspects, Lectures in GB and the minimalist program — #more than two>every
b. ... but none of them read the same book — every>two

One way to account for the contrast between (34) and (35) is to assume that different kinds of DPs have different landing sites for movement. Beghelli (1995) assumed a rich hierarchy of functional projections that define potential sites where quantifiers move to. For instance, referential DPs like definites move to the specifier of a high projection, while distributive quantifiers like *every NP* move to a lower position and indefinites move optionally to either a low or a high position. There are many intricate details to such theories. The interested reader is referred to Beghelli (1995), but especially to the volume Szabolcsi (1996).

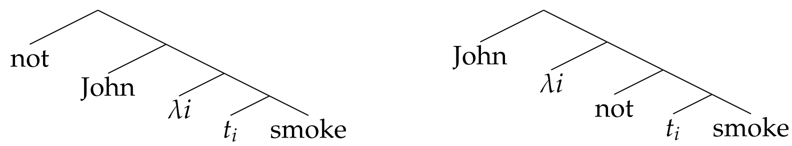
6 Scope economy

6.1. LF scope illusions: the case of $\langle\langle e,t\rangle,t\rangle$-type proper names

Given that a proper name like *John* can be of the $\langle\langle e,t\rangle,t\rangle$-type, we expect it to be able to raise. For instance, (36) would have two LFs, one with the proper name in a position lower than that of negation and one in a position that is higher than that of negation.

(36) John didn't smoke.

(37)



It is important to understand that the two LFs in (37) result in an equivalent interpretation. That is, even though in terms of hierarchical position at LF the scope relation between the proper name and negation differs in the two cases, because of the meaning of *John*, this does not result in any noticeable scope alternation. Proper names are not scope bearing entities, even in their quantifier type they merely contribute an *e*-type argument to a predicate.

Let us see how this works, starting with the case where negation is in a higher position than *John* is. First, the trace combines with the $\langle e,t\rangle$ meaning of *smoke*, resulting in the proposition $smoke(i)$. The next step is to lambda-abstract the *i*-variable, resulting in a property $\lambda i.smoke(i)$. Then, the quantifier meaning of *John* is function applied to this property: $\lambda P.P(j)$ $(\lambda i.smoke(i)) = \lambda i.smoke(i)$ $(j) = smoke(j)$. In the last step we add negation, yielding $\neg smoke(j)$.

Now for the case where negation is in a lower position. Again, we start with combining the trace and the predicate, yielding the proposition $smoke(i)$. It is this proposition that is negated by *not*, yielding $\neg smoke(i)$. Now we lambda-abstract the *i*-variable: $\lambda i.\neg smoke(i)$ and then finally we function apply the quantifier meaning of John to this property: $\lambda P.P(j)$ $(\lambda i.\neg smoke(i)) = \lambda i.\neg smoke(i)$ $(j) = \neg smoke(j)$. This shows you that the relative position at LF of proper name and negation is of no consequence.

6.2. Scope Economy (Fox 1999)

There are many more cases where quantifier movement is semantically vacuous. For instance, (38) is predicted to be structurally ambiguous between (38-a) and (38-b). However, these are truth-conditionally equivalent.

(38) John admires every student.

a. [John [λx [every student [λy [t_x [admires t_y]]]]]]

b. [every student [λy [John [λx [t_x [admires t_y]]]]]]

Also, (39) is predicted ambiguous between (39-a) and (39-b), which are both equivalent to (40-a).

(39) Every student admires every professor

a. $\forall x[student(x) \rightarrow \forall y[professor(y) \rightarrow admire(x,y)]]$

b. $\forall y[professor(y) \rightarrow \forall x[student(x) \rightarrow admire(x,y)]]$.

(40) $\forall x \forall y[student(x) \wedge professor(y) \rightarrow admire(x,y)] = \forall y \forall x[student(x) \wedge professor(y) \rightarrow admire(x,y)]$

Fox (1999) proposes that even though such semantically vacuous movement is theoretically possible, it is prohibited in natural language. That is, he proposed the scope economy principle: quantifier movement cannot be semantically vacuous. This principle is rather surprising, since if what is prohibited is semantically vacuous, how are we going to tell whether the principle is empirically correct?

Fox shows that in certain cases one can observe that vacuous movement has indeed not taken place. For instance, it is commonly assumed that ellipsis involves a form of parallelism. That is, in (41), the elided phrase can only be resolved as being a syntactic structure that is isomorphic to the structure, at LF, of its antecedent.

In (41) there are two readings available. Either the example is about a specific boy and girl (narrow scope for *every professor*), or the example quantifies over different boys and girls (wide scope for *every professor*).

(41) Some boy admires every professor. Some girl does too.

If we now turn to the example in (42), we see that the first sentences receive only one interpretation, namely one in which there is a specific boy who admires every professor. This is because, had the first sentence had a non-specific reading for *some boy*, then *every professor* would have had to scope over the subject. Consequently, in the second sentence a similar structure would have been needed, but there this kind of movement is vacuous. This is why this reading is not observed.

(42) a. Some boy admires every professor. Mary does too.
b. Some boy admires every professor. Every girl does too.

7 Motivation for LF

By itself, scope ambiguity does not justify LF as a level of syntactic representation. There are sophisticated ways of accounting for quantifier scope in a directly compositional setup.

One kind of motivation for LF, however, comes from constraints on movement: QR is limited in a similar way to overt movement. I.e., the argument is based on analogy: what we know about overt movement holds for QR as well.

Specificity conditions: Specific NPs may contain no free variable

(43) a. Who did you see a picture of t?
b. *Who did you see this picture of t?

(44) This picture of everybody is now on sale.
a. [everybody$_i$ [this picture of t$_i$]$_j$ [t$_j$ is now on sale]] UNAVAILABLE
b. [[this picture of everybody$_i$]$_j$ [t$_j$ is now on sale]] AVAILABLE

Coordinate Structure Constraint: Movement out of one of two coordinated conjuncts is prohibited

(45) a. Which professor$_i$ do you think John likes t$_i$?
b. *Which professor$_i$ do you think John likes t$_i$ and hates the dean?

(46) a. A student likes every professor. $\exists > \forall$, $\forall > \exists$
b. A student likes every professor and hates the dean. $\exists > \forall$, $*\forall > \exists$

Weak Crossover

(47) a. ??Who$_i$ does his$_i$ mother love t$_i$?
b. Every student$_i$ told his$_i$ friend that his$_i$ mother disliked his$_i$ cousin. (no movement)

(48) ??His$_i$ mother loves every boy$_i$.

(49) [[every boy]$_i$ his$_i$ mother loves t$_i$] LF

7.1. Problems: Indefinites

There are also cases whether the relation between overt and covert movement is more problematic. We discuss one: indefinites.

(50) Limits of QR: syntactic islands
- a. *Who do you think John likes t and hates the dean. coordination
- b. *Who do you think if Mary kisses t, then John will faint. conditionals
- c. *Who do you think John believes the claim that Mary kissed t. complex NP

(51)
- a. A student likes every professor and hates the dean. $\exists > \forall$, $*\forall > \exists$
- b. If every professor kisses Mary, then John will faint. if-then $> \forall$, $*\forall >$ if-then
- c. John told a rumour about Mary kissing everyone. $\exists > \forall$, $*\forall > \exists$

(52)
- a. Every student likes some professor and hates the dean. $\exists > \forall$, $\forall > \exists$
- b. If a professor kisses Mary, then John will faint. if-then $> \exists$, $\exists >$ if-then
- c. John told several rumours about Mary kissing someone. several $> \exists$, $\exists >$ several

8 Beyond *every, some* and *no*

So far, we have only discussed limited set of quantifiers. In this section, we turn to the great variety of quantifiers that exist. To discuss the semantics of expressions like *most*, *many*, etc. we need to first introduce some more set theory however.

8.1. Set theory revisited

Basic set-theoretic operations and conventions:

(53) $a \in A$ "a is a member of set A"

(54)
- a. $\emptyset$ is the set containing nothing, the *empty set*.
- b. If A is a set and B is a set, then:
  $A \cup B$ is the union of A and B — the set containing the elements of A and B combined
  $A \cap B$ is the intersection of A and B — the set containing only those elements that are in both A and B
  $A \setminus B$, or sometimes $A - B$, is B subtracted from A — the set containing those elements that are in A but not in B
  $\bar{A}$, or sometimes A', is the complement of A — the set of elements not in A

8.2. Relations between sets

Operators like $\cap$ and $\cup$ make a new set out of two existing sets. For instance, if A and B are sets, then $A \cap B$ is the intersection set of A and B.

We now introduce a way of *comparing* sets.

(55) $A \subseteq B$ is true if all the elements of A are also in B, that is, if A is a subset of B.

(56) a. $\{a, b, c\} \subseteq \{a, b, c\}$
b. $\{\{a, b\}, c\} \subseteq \{a, b, \{a, b\}, c\}$
c. $\{a, b\} \nsubseteq \{d, \{a, b\}, c\}$

8.3. Sets as functions

Let us assume that there are just 5 entities, namely a, b, c, d, and e. Now consider the following set:

(57) $S = \{a, b, c\}$

This set can be seen as a particular type of function, *a characteristic function*. That's a function that maps some entities to 1 and other entities to 0. The idea is that S in (57) corresponds to a functions that returns 1 for individuals that are a member of S and 0 for individuals not in S. In other words, S corresponds to the following f:

(58)
$f(a) = 1$
$f(b) = 1$
$f(c) = 1$
$f(d) = 0$
$f(e) = 0$

This function f is of type $\langle e, t\rangle$ (it takes an entity to return a truth-value.) In general, then, we can take $\langle e, t\rangle$-type expressions to correspond to sets. For example:

(59) the verb phrase *hates Mary*
. . . is a function that map individuals who hate Mary to 1 and individuals who do not hate Mary to 0
. . . is the set of people who hate Mary

(60) the noun *student*
. . . is a function that maps students to 1 and individuals who are not students to 0
. . . is the set of students

8.4. The subset relation between $\langle e, t\rangle$-types

(61) a. $[\![$blue sweater$]\!] \subseteq [\![$sweater$]\!]$
b. $[\![$adores and admires Mary$]\!] \subseteq [\![$admire Mary$]\!]$
c. $[\![$student that owns a car$]\!] \subseteq [\![$student$]\!]$
d. $[\![$owns a blue car$]\!] \subseteq [\![$owns a car$]\!]$

8.5. Quantification as set comparison

(62) [[Every student] hates Mary.]
a. is true if the set of students is a subset of the set of people who hate Mary.

(63) the semantics of *every*:

a. $\lambda P.\lambda P'.\forall x[P(x) \rightarrow P'(x)]$
b. $\lambda P.\lambda P'.P \subseteq P'$

The terms in (63-a) and (63-b) are two ways of doing the same thing. In the case of (63-a), you use a statement of predicate logic as your representation of truth-conditions; in (63-b) you express the truth-conditions as a set-theoretic statement.

(64) the semantics of *some*:
a. $\lambda P.\lambda P'.\exists x[P(x) \wedge P'(x)]$
b. $\lambda P.\lambda P'.P \cap P' \neq \emptyset$

(65) the semantics of *no*:
a. $\lambda P.\lambda P'.\neg\exists x[P(x) \wedge P'(x)]$
b. $\lambda P.\lambda P'.P \cap P' = \emptyset$

It is important to be able to express truth-conditions in this set-theoretic way, since not all quantifiers are predicate-logical. In predicate logic, we only have $\forall$ and $\exists$ as logical quantifiers. But natural language has many more determiners above *every* and *some*:

(66) [[$_{\text{QP}}$ [$_{\text{DET}}$ ...] student(s)] are lazy]
many / most / less than half the / more than three / three / exactly three / almost sixteen hundred / at least 10 but no more than 20 / surprisingly few / ...

Some examples, let *HM* be the set of individuals who hate Mary and *S* be the set of students:

(67) a. Most students hate Mary.
b. $|S \cap HM| > |S - HM|$ (there are more students hating Mary than students not hating Mary)

(68) a. Exactly three students hate Mary.
b. $|S \cap HM| = 3$

(69) a. Between four and eight students hate Mary.
b. $4 \leq |S \cap HM| \leq 8$

Fact: there is no predicate-logical form that can express the truth-conditions of *most students are lazy* (or any other sentence with *most*)

(70) a. $Mx[student(x) \rightarrow lazy(x)]$
b. $Mx[student(x) \wedge lazy(x)]$

Say that Mx is to be interpreted as *for most x ....* The problem now is that neither (70-a) nor (70-b) will do. The form in (70-a) tells us that most individuals in the domain are such that if they are a student, then they are lazy. (This is true if most things in the domain are not students.) The form in (70-b) tells us that most things in the domain are both student and lazy, but that's not what the sentence *most students are lazy* means.
The essence of the problem lies in the fact that *most* is two-place. It compare the property of being a student, to the property of being lazy. This is beyond the reach of the predicate language we have defined.

8.6. Determiners as relations

(71) a. Most(A)(B) ⇔ $|A \cap B| > |A - B|$
b. Some(A)(B) ⇔ $A \cap B \neq \emptyset$
c. More-than-a-quarter(A)(B) ⇔ $|A \cap B| \div |A| > 0.25$

(72) Provide the relations expressed by the following determiners
a. more than 8
b. at least 9

Are they equivalent?

9 Properties of Quantifiers

9.1. Universals

The following is an excerpt from von Fintel & Matthewson, 'Universals in Semantics' (von Fintel and Matthewson 2008):

> *3.1.2. Conservativity.* Among formal semanticists, the most celebrated semantic universals are those proposed by Barwise and Cooper in their seminal article on quantifiers in natural language (1981). Their particular concern are quantificational determiners, items such as *every, some, no, most, few, many, ...*, which are treated as denoting second order relations between two sets, the first picked out by the common noun phrase argument of the determiner, the other supplied by the rest of the sentence — this makes them be of semantic type $\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$. For example, *every* A B claims that the A-set is a subset of the B-set, while *some* A B says that the intersection of the two sets is not empty. What Barwise and Cooper noted was that the first argument, the set denoted by the common noun phrase, is special: this set *restricts* the quantifier, or supplies the *domain* of the quantifier. Barwise and Cooper claim that determiner-quantifiers universally "live on" their first argument. This property is now widely known as *conservativity*, so-termed by Keenan and Stavi (1986).[28]
>
> (15) In relational terms: δ is conservative iff for all sets A, B: $\delta(A)(B) \equiv \delta(A)(A \cap B)$.

Examples:

(73) a. Most students hate Mary ⇔ Most students are students who hate Mary
b. Some student is lazy ⇔ Some student is a lazy student

The property of conservativity makes it that the noun argument of the determiner is special in that it determines the domain. Conservativity says that there is no determiner Bla such that the truth-value of *Bla students hate Mary* depends on how many non-students hate Mary.

Exception: *only*

(74) Only students hate Mary.

9.2. Monotonicity

(75) An operator Q of type $\langle\langle e,t\rangle,t\rangle$ is upward monotone if and only if $A \subseteq B$ & $Q(A)$ entails $Q(B)$

(76) An operator Q of type $\langle\langle e,t\rangle,t\rangle$ is downward monotone if and only if $A \subseteq B$ & $Q(B)$ entails $Q(A)$

(77) An operator Q of type $\langle\langle e,t\rangle,t\rangle$ is non-monotone if it is neither upward nor downward monotone.

Example:

(78) a. All students own a blue car $\Rightarrow$ All students own a car — upward
b. No student owns a blue car $\Leftarrow$ No student owns a car — downward
c. Exactly two students own a blue car $\not\Leftrightarrow$ Exactly two students own a car — non-monotone

What generalisation does (79) suggest?

(79) a. *Some of my dogs ever stole a bone.
b. *Every dog of mine ever stole a bone.
c. None of my dogs ever stole a bone.

Why is the following example felicitous?

(80) Every dog that ever stole a bone will go to hell.

10 Further reading

Heim and Kratzer on quantification: chapters 6, 7 and 8.

Type shifting: Partee 1987; de Swart 2001

Indefinites and LF: Fodor and Sag 1982; Kratzer 1997; Reinhart 1997

One can find a particularly deep and rich discussion of scope matters in Ruys and Winter (2010). (See http://www.phil.uu.nl/ yoad/papers/RuysWinterScope.pdf).

For on expressions like *most*, *many*, *few*, *two*, *more than two*, *at least two*, see Nouwen 2009 for an overview of the semantics of these expressions from several perspectives (including a psychological one). (Written for a general linguistic audience.)

A good further reading about the set-theoretic semantics of quantifiers is the chapter on Generalised Quantifiers in de Swart (1998). For a more technical introduction, see the chapter on generalised quantifier theory in Partee et al. (1993). .

References

Beghelli, F. (1995). *The Phrase Structure of Quantifier Scope*. Ph. D. thesis, UCLA.

de Swart, H. (1998). *Introduction to Natural Language Semantics*. CSLI Publications.

de Swart, H. (2001). Weak readings of indefinites: type-shifting and closure. *The Linguistic Review 18*(1), 69–96.

Fodor, J. D. and I. Sag (1982). Referential and quantificational indefinites. *Linguistics and Philosophy 5*, 355—398.

Fox, D. (1999). *Economy and Semantic Interpretation*, Volume 35 of *Linguistic Inquiry Monographs*. MIT press.

Kratzer, A. (1997). Scope or pseudoscope? are there wide scope indefinites? In S. Rothstein (Ed.), *Events and Grammar*. Kluwer.

Montague, R. (1973). The proper treatment of quantification in ordinary english. In K. Hintikka, J. Moravcsik, and P. Suppes (Eds.), *Approaches to Natural Language*, pp. 221–242. Dordrecht: Reidel.

Nouwen, R. (2009). What's in a quantifier? In T. Lentz, H. de Mulder, Ø. Nilsen, and A. Zondervan (Eds.), *Theoretical Validity and Psychological Reality*. Forthcoming, pre-published version available at `http://www.let.uu.nl/~Rick.Nouwen/personal/papers/wiaq.pdf`.

Partee, B. (1987). Noun phrase interpretation and type-shifting principles. In J. Groenendijk, D. de Jongh, and M. Stokhof (Eds.), *Representation Theory and the Theory of Generalized Quantifiers*, Volume 8 of *Studies in Discourse*, pp. 115–143. Dordrecht: Foris.

Partee, B., A. ter Meulen, and R. Wall (1993). *Mathematical methods in linguistics*. Kluwer Academic Publishers.

Reinhart, T. (1997). Quantifier scope: how labor is divided between QR and choice functions. *Linguistics and Philosophy 20*, 335–397.

Ruys, E. and Y. Winter (2010). Quantifier scope in formal linguistics. In D. Gabbay and F. Guenthner (Eds.), *Handbook of philosophical logic*. Amsterdam: Benjamins.

Szabolcsi, A. (1996). *Ways of Scope Taking*. Kluwer Academic Publishers.

von Fintel, K. and L. Matthewson (2008). Universals in semantics. *The linguistic review 25*, 139–201.