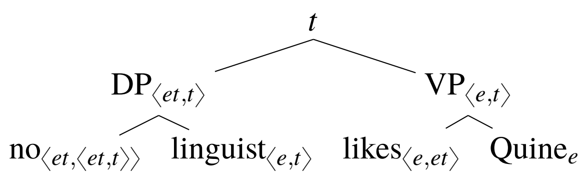
1 The Problem of Quantifiers in Object Position

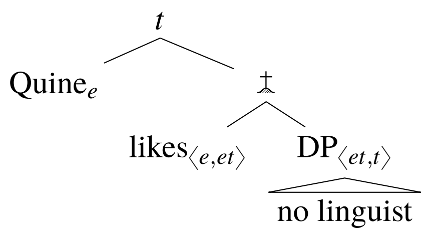
To review, we analyze the denotations of quantificational DPs to be Generealized Quantifiers, i.e. functions of type $\langle et, t\rangle$. This allows us to compute sentences like (1).

(1)


t
$\text{DP}_{\langle et,t\rangle}$
$\text{VP}_{\langle e,t\rangle}$
$\text{no}_{\langle et,\langle et,t\rangle\rangle}$
$\text{linguist}_{\langle e,t\rangle}$
$\text{likes}_{\langle e,et\rangle}$
Quine_e

But a problem arises when we consider sentences containing quantificational DPs in object position.

(2)

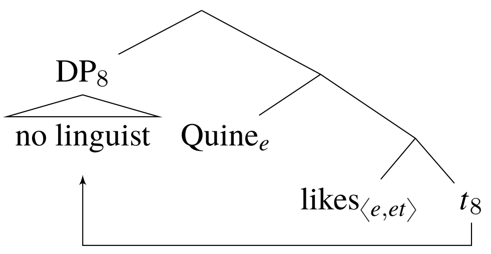

t
Quine_e
$\cancel{t}$
$\text{likes}_{\langle e,et\rangle}$
$\text{DP}_{\langle et,t\rangle}$
no linguist

There is a type-mismatch between the transitive verb of type $\langle e, et\rangle$ and the quantificational DP!

2 Quantifier Raising

In order to solve this problem of object quantificational DPs, we adopt the hypothesis that quantifiers undergo *covert movement* called *Quantifier Raising (QR)*, as illustrated by the following tree diagram.

(3)



This is not the structure that is pronounced, which is to say that it's not the structure that the phonology 'sees', but let's assume that it is what the semantic gets from the syntax. How does the semantics interpret such a structure? We actually have enough machinery to deal with this, but it needs some adjustments. Recall from Lecture 6 that traces are interpreted via Trace Rule.

(4) *Trace Rule*: For any model M and for any assignment function a and for any index $i \in \mathbb{N}$, $[\![t_i]\!]^{a,M} = a(i)$.

Since traces denote individuals by assumption, they can combine with $[\![\text{likes}]\!]^{a,M}$ via Functional Application. So the type-mismatch at that level is resolved.

But this only shifts the problem elsewhere. In particular, how do we deal with the moved quantificational DP? In Lecture 6 we discussed how relative pronouns are interpreted. Relative pronouns undergo overt movement. We assumed that their primary semantic function is to trigger Predicate Abstraction:

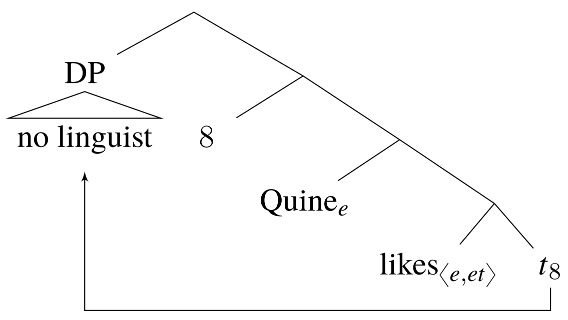
(5) *Predicate Abstraction* (old ver.; to be revised):
for any model M, for any assignment function a, if RP is a relative pronoun with an index $i \in \mathbb{N}$,

$$\text{then } \left[\!\!\left[\; [\, \text{RP}_i \;\; \text{A} \,] \; \right]\!\!\right]^{a,M} = \left[\lambda x \in D_e . \left[\!\!\left[\text{A} \right]\!\!\right]^{a[i \to x],M} \right]$$

This rule captures the semantic dependency between the relative pronoun and its trace.

In order to capture the dependency between the QR'ed quantificational DP and its trace in the structure above, we want to trigger Predicate Abstraction. However, since there is no relative pronoun, we do not have a trigger for it.

In order to fix this, let us assume that it is the index that triggers Predicate Abstraction, rather than the relative pronoun. In particular, we assume that the structure actually looks like (6) with an index on the moved element acting as an independent node.

(6)


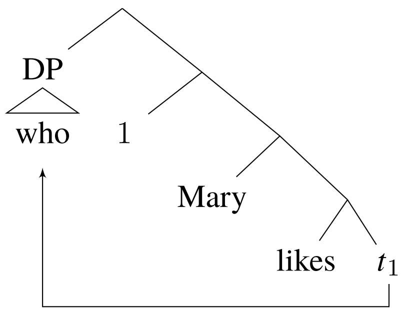
Now we can regard the index node (call it a *binder index*) as the trigger of Predicate Abstraction. Predicate Abstraction is now defined as follows:

(7) *Predicate Abstraction* (new ver.):
for any model M, for any assignment function a and for any index $i \in \mathbb{N}$,

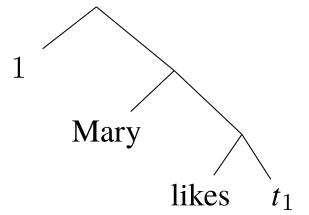
$$\left[\!\!\left[\; [\, i \;\; \text{A} \,] \; \right]\!\!\right]^{a,M} = \left[\lambda x \in D_e . \left[\!\!\left[\text{A} \right]\!\!\right]^{a[i \to x],M} \right]$$

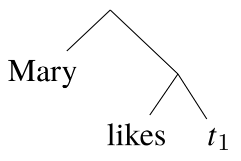
The only difference from the previous version of the rule is that now it is triggered by an index, rather than a relative pronoun with an index. This version works just as well as the old version for relative clauses. We now assume a structure like (8), where the relative pronoun and its index occupy different syntactic positions.

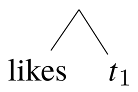
(8)



The denotation of the relative clause should be the same as before, i.e. $[\![(8)]\!]^{a,M} = [\lambda x \in D_e.\ 1 \text{ iff Mary likes } x \text{ in } M]$. According to the new version of Predicate Abstraction, this in fact is the denotation of the part excluding the relative pronoun, as shown by the following computation.

$[\![$  $]\!]^{a,M}$

$= \left[\lambda x \in D_e.\ [\![$  $]\!]^{a[1\to x],M}\right]$ (PA)

$= \left[\lambda x \in D_e.\ [\![$  $]\!]^{a[1\to x],M}([\![\text{Mary}]\!]^{a[1\to x],M})\right]$ (FA)

$$= \Big[\lambda x \in D_e.\ [\![\text{likes}]\!]^{a[1\to x],M}([\![t_1]\!]^{a[1\to x],M})([\![\text{Mary}]\!]^{a[1\to x],M})\Big] \quad \text{(FA)}$$

$$= \Big[\lambda x \in D_e.\ [\![\text{likes}]\!]^{a[1\to x],M}(a[1 \to x](t_1))([\![\text{Mary}]\!]^{a[1\to x],M})\Big] \quad \text{(TR)}$$

$$= \Big[\lambda x \in D_e.\ [\![\text{likes}]\!]^{a[1\to x],M}(x)([\![\text{Mary}]\!]^{a[1\to x],M})\Big]$$

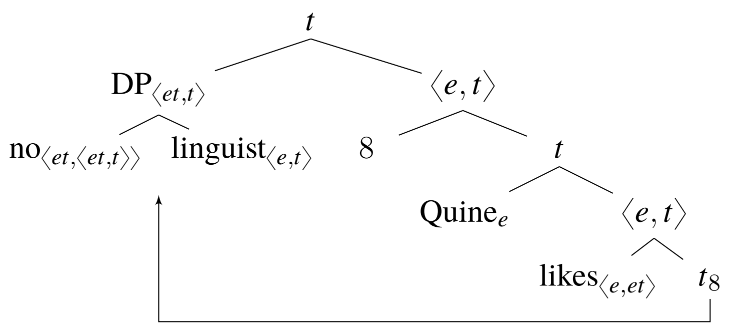
$$= \Big[\lambda x \in D_e.\ [\lambda y \in D_e.[\lambda z \in D_e.\ 1 \text{ iff } z \text{ likes } y \text{ in } M]](x)([\![\text{Mary}]\!]^{a[1\to x],M})\Big] \quad \text{(Lexicon)}$$

$$=[\lambda x \in D_e.\ 1 \text{ iff } [\![\text{Mary}]\!]^{a[1\to x],M} \text{ likes } x \text{ in } M] \quad (\lambda\text{-conv.}\times 2)$$

Given this result, we can just analyze the relative pronoun to be semantically vacuous, i.e. it denotes an identity function. Here, its sister is of type $\langle e,t\rangle$, so its denotation will be: $[\lambda f \in D_{\langle e,t\rangle}.\ f]$.

Now coming back to the structure with a QR'ed quantificational DP, the new version of PA allows us to interpret it, and to get the denotation we want, as we will now see. Firstly, let us make sure that the semantic types work out:

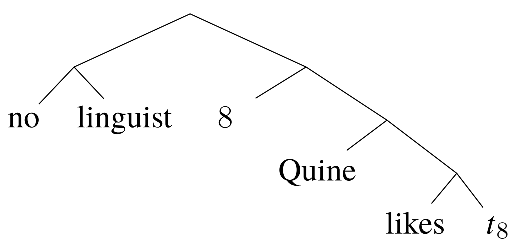
(9)

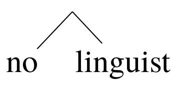
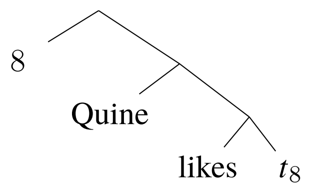


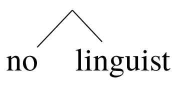
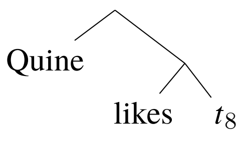
t
$\text{DP}_{\langle et,t\rangle}$
$\langle e,t\rangle$
$\text{no}_{\langle et,\langle et,t\rangle\rangle}$
$\text{linguist}_{\langle e,t\rangle}$
8
t
Quine_e
$\langle e,t\rangle$
$\text{likes}_{\langle e,et\rangle}$
t_8

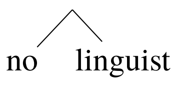
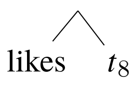
Notice in particular that the sister to the binder index 8 is of type t, but its mother node is of type $\langle e,t\rangle$. This is because Predicate Abstraction adds $\lambda x \in D_e$. And this type-$\langle e,t\rangle$ function can serves as the argument of the moved quantificational DP, which is by assumption of type $\langle et,t\rangle$.

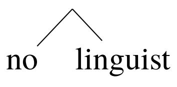
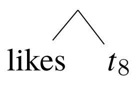
Let us now verify that the denotation is correct. We will do a top-down computation. We assume that $[\![\text{Quine}]\!]^{a,M} = q$, for any assignment a.

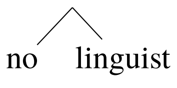
$[\![$  $]\!]^{a,M}$

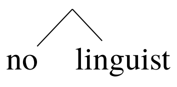
$= [\![$  $]\!]^{a,M}\Bigg([\![$  $]\!]^{a,M}\Bigg)$ (FA)

$= [\![$  $]\!]^{a,M}\Bigg(\Bigg[\lambda x \in D_e.\ [\![$  $]\!]^{a[8\to x],M}\Bigg]\Bigg)$ (PA)

$= [\![$  $]\!]^{a,M}\Bigg(\Bigg[\lambda x \in D_e.\ [\![$  $]\!]^{a[8\to x],M}([\![\text{Quine}]\!]^{a[8\to x],M})\Bigg]\Bigg)$ (FA)

$= [\![$  $]\!]^{a,M}\Bigg(\Bigg[\lambda x \in D_e.\ [\![$  $]\!]^{a[8\to x],M}(q)\Bigg]\Bigg)$ (Lexicon)

$= [\![$  $]\!]^{a,M}\left(\left[\lambda x \in D_e.\ [\![\text{likes}]\!]^{a[8\to x],M}([\![t_8]\!]^{a[8\to x],M})(q)\right]\right)$ (FA)

$= [\![$  $]\!]^{a,M}\left(\left[\lambda x \in D_e.\ [\![\text{likes}]\!]^{a[8\to x],M}(a[8\to x](8))(q)\right]\right)$ (TR)

$$= \left[\!\!\left[\begin{array}{cc} \multicolumn{2}{c}{\diagup\!\diagdown} \\ \text{no} & \text{linguist} \end{array} \right]\!\!\right]^{a,M} \left(\left[\lambda x \in D_e.\ [\![\text{likes}]\!]^{a[8\to x],M}(x)(q)\right]\right)$$

$$= \left[\!\!\left[\begin{array}{cc} \multicolumn{2}{c}{\diagup\!\diagdown} \\ \text{no} & \text{linguist} \end{array} \right]\!\!\right]^{a,M} ([\lambda x \in D_e.\ [\lambda y \in D_e.\ [\lambda z \in D_e.\ 1 \text{ iff } z \text{ likes } y \text{ in } M]](x)(q)]) \quad \text{(Lexicon)}$$

$$= \left[\!\!\left[\begin{array}{cc} \multicolumn{2}{c}{\diagup\!\diagdown} \\ \text{no} & \text{linguist} \end{array} \right]\!\!\right]^{a,M} ([\lambda x \in D_e.\ [\lambda z \in D_e.\ 1 \text{ iff } z \text{ likes } x \text{ in } M](q)]) \quad (\lambda\text{-conv.})$$

$$= \left[\!\!\left[\begin{array}{cc} \multicolumn{2}{c}{\diagup\!\diagdown} \\ \text{no} & \text{linguist} \end{array} \right]\!\!\right]^{a,M} ([\lambda x \in D_e.\ 1 \text{ iff } q \text{ likes } x \text{ in } M]) \quad (\lambda\text{-conv.})$$

This computation shows that the sister to 'no linguist' denotes the type $\langle e,t\rangle$-function $[\lambda x \in D_e.\ 1 \text{ iff } q \text{ likes } x \text{ in } M]$, which maps any individual that q (=Quine) likes in the model to 1 and everyone else to 0. Recall that the quantificational DP $[\![\text{no linguist}]\!]^{a,M}$ takes any function f and says for no linguist x, $f(x) = 1$, i.e. f maps no linguist to 1. Since the argument of this quantificational DP here is the type-$\langle e,t\rangle$ function that maps any individual that q likes to 1, the sentence will mean that this function maps no linguist to 1, i.e. q likes no linguist. The rest of the computation proves this:

$$\left[\!\!\left[\begin{array}{cc} \multicolumn{2}{c}{\diagup\!\diagdown} \\ \text{no} & \text{linguist} \end{array} \right]\!\!\right]^{a,M} ([\lambda x \in D_e.\ 1 \text{ iff } q \text{ likes } x \text{ in } M])$$

$$= [\![\text{no}]\!]^{a,M}([\![\text{linguist}]\!]^{a,M})([\lambda x \in D_e.\ 1 \text{ iff } q \text{ likes } x \text{ in } M]) \quad \text{(FA)}$$

$$= [\![\text{no}]\!]^{a,M}([\lambda y \in D_e.\ 1 \text{ iff } y \text{ is a linguist in } M])([\lambda x \in D_e.\ 1 \text{ iff } q \text{ likes } x \text{ in } M]) \quad \text{(Lexicon)}$$

$$= \left[\lambda f_{\langle e,t\rangle}.\ \left[\lambda g_{\langle e,t\rangle}.\ \begin{array}{c} 1 \text{ iff for no } z \\ \text{s.t. } f(z)=1, \\ g(z)=1 \end{array}\right]\right] \left(\left[\lambda y_e.\ \begin{array}{c} 1 \text{ iff } y \text{ is a} \\ \text{linguist in } M \end{array}\right]\right) \left(\left[\lambda x_e.\ \begin{array}{c} 1 \text{ iff } q \text{ likes} \\ x \text{ in } M \end{array}\right]\right) \quad \text{(Lexicon)}$$

$$= \left[\lambda g \in D_{\langle e,t\rangle}.\ \begin{array}{l} 1 \text{ iff for no individual } z \text{ such that} \\ \left[\lambda y_e.\ \begin{array}{c} 1 \text{ iff } y \text{ is a} \\ \text{linguist in } M \end{array}\right](z) = 1, \\ g(z) = 1 \end{array}\right] \left(\left[\lambda x \in D_e.\ \begin{array}{c} 1 \text{ iff } q \text{ likes} \\ x \text{ in } M \end{array}\right]\right) \quad (\lambda\text{-conv.})$$

$$= \left[\lambda g \in D_{\langle e,t\rangle}.\ \begin{array}{l} 1 \text{ iff for no individual } z \text{ such that} \\ z \text{ is a linguist in } M,\ g(z) = 1 \end{array}\right] ([\lambda x \in D_e.\ 1 \text{ iff } q \text{ likes } x \text{ in } M]) \quad (\lambda\text{-conv.})$$

$$= 1 \text{ iff } \begin{array}{l} \text{for no individual } z \text{ such that } z \text{ is a linguist in } M, \\ [\lambda x \in D_e.\ 1 \text{ iff } q \text{ likes } x \text{ in } M](z) = 1 \end{array} \quad (\lambda\text{-conv.})$$

$$\text{iff for no individual } z \text{ such that } z \text{ is a linguist in } M,\ q \text{ likes } z \text{ in } M \quad (\lambda\text{-conv.})$$

(iff q likes no linguist in M)

3 Scope Ambiguity

We can see Quantifier Raising (QR) as the mechanism behind quantifier scope. There are several important properties of quantifier scope in natural language, one of which is scope ambiguity. To illustrate, consider the following sentence.

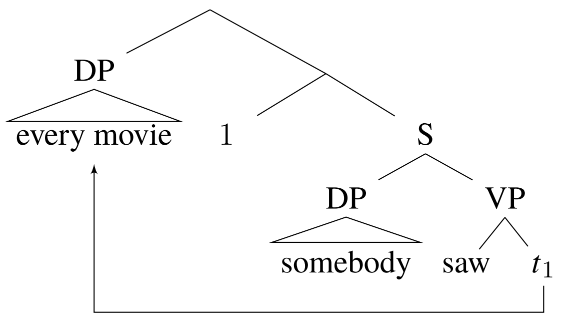
(10) Somebody saw every movie.

Suppose that there are three movies, Movie 1, Movie 2, and Movie 3, and four people, Alice, Bob, Chris, and Daniel. Suppose further that Alice and Bob saw Movie 1, Chris saw Movie 2 and Daniel saw Movie 3. This situation can be represented as (11).

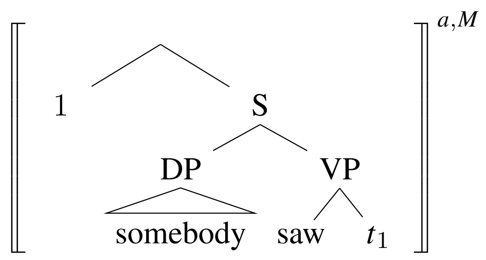
(11)
$$\begin{tikzcd}
\text{Alice} \arrow[r] & \text{Movie 1} \\
\text{Bob} \arrow[ur] & \\
\text{Chris} \arrow[r] & \text{Movie 2} \\
\text{Daniel} \arrow[r] & \text{Movie 3}
\end{tikzcd}$$

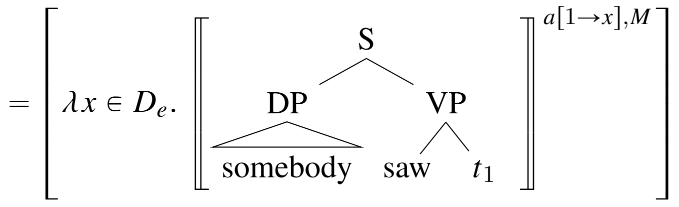
In this situation, (11) is judged to be true.

This judgment is in fact predicted by our theory. After QR, the structure of (11) looks like (12).

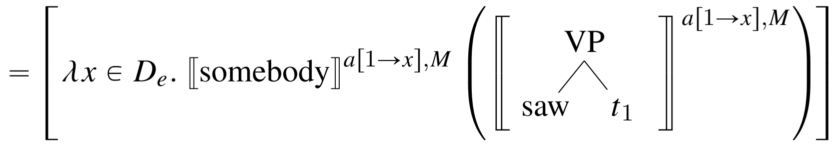
(12)


Let's compute the denotation of this sentence. The sister to the moved quantificational DP will denote the following type-$\langle e,t\rangle$ function.





(PA)


$= \lambda x \in D_e.\ [\![\text{somebody}]\!]^{a[1\to x],M}$
VP
saw
t_1
$a[1\to x],M$

(FA)

$$= \left[\lambda x \in D_e.\ [\![\text{somebody}]\!]^{a[1\to x],M}([\![\text{saw}]\!]^{a[1\to x],M}([\![t_1]\!]^{a[1\to x],M}))\right] \qquad \text{(FA)}$$

$$= \left[\lambda x \in D_e.\ [\![\text{somebody}]\!]^{a[1\to x],M}([\![\text{saw}]\!]^{a[1\to x],M}(a[1\to x](1)))\right] \qquad \text{(TR)}$$

$$= \left[\lambda x \in D_e.\ [\![\text{somebody}]\!]^{a[1\to x],M}([\![\text{saw}]\!]^{a[1\to x],M}(x))\right]$$

$$= \left[\lambda x \in D_e.\ [\![\text{somebody}]\!]^{a[1\to x],M}([\lambda y \in D_e.[\lambda z \in D_e.\ 1 \text{ iff } z \text{ saw } y \text{ in } M]](x))\right] \qquad \text{(Lexicon)}$$

$$= \left[\lambda x \in D_e.\ [\![\text{somebody}]\!]^{a[1\to x],M}([\lambda z \in D_e.\ 1 \text{ iff } z \text{ saw } x \text{ in } M])\right] \qquad (\lambda\text{-conv.})$$

$$= \left[\lambda x \in D_e.\ \left[\lambda f \in D_{\langle e,t\rangle}.\ \begin{array}{c}1 \text{ iff for some person}\\ y \text{ in } M,\ f(y) = 1\end{array}\right]([\lambda z \in D_e.\ 1 \text{ iff } z \text{ saw } x \text{ in } M])\right] \quad \text{(Lexicon)}$$

$$= \left[\lambda x \in D_e.\ \begin{array}{l}1 \text{ iff for some person } y \text{ in } M,\\ [\lambda z \in D_e.\ 1 \text{ iff } z \text{ saw } x \text{ in } M](y) = 1\end{array}\right] \quad (\lambda\text{-conv.})$$

$$=[\lambda x \in D_e.\ 1 \text{ iff for some person } y \text{ in } M,\ y \text{ saw } x \text{ in } M] \quad (\lambda\text{-conv.})$$

Recall that $[\![\text{every movie}]\!]^{a,M}$ maps any function f of type $\langle e, t\rangle$ to 1 iff for every movie x in M, $f(x) = 1$. In our example, the argument of $[\![\text{every movie}]\!]^{a,M}$ is the function $[\lambda x \in D_e.\ 1$ iff for some person $y \in D_e$, y saw x in $M]$, so the entire sentence denotes 1 iff for every movie x in M, for some person $y \in D_e$, y saw x in M. In the above example situation, each movie indeed has at least one person who saw it, so the sentence is correctly predicted to be true.

As you might have learned in other modules, the sentence in (18) is said to have a different reading with different truth-conditions. Specifically, it is often assumed that (18) has a reading that denotes 1 iff there is one particular person who saw every movie. This reading is false in the above scenario (11), because no one saw every movie.

But we should be skeptical here: Is there really a separate reading that is false in the above scenario, in addition to a reading that is true? It is not immediately clear whether this is the case, because the second reading, which is false in the scenario in (11), entails the first reading, which is true in (11), but not vice versa. To facilitate the following discussion, let's give names to these two readings. We call the first reading the *inverse scope reading*, because the order of quantifiers in the semantic representation is the inverse of the order of quantifiers as phonologically realized, and the second reading the *surface scope reading*, as the order of quantifiers matches between the two representations.

(13) a. *Inverse Scope Reading*:
For every movie, there is somebody who saw it.
b. *Surface Scope Reading*:
There is somebody who saw every movie.

As can be seen easily, whenever the surface scope reading (13b) is true, the inverse scope reading (13a) is also true. If there is a single person who saw every movie, then every movie has at least one person who saw it. But the entailment is only one way, as demonstrated by the above scenario (11). In (11), the inverse scope reading (13a) is true, but the surface scope reading (13b) is false.

This asymmetric entailment relation has confused a lot of linguists, and led them to claim that one of these two readings is actually not there as a separate reading. The idea is that one of the readings subsumes the other reading, so there is not real empirical evidence for the subsumed reading and hence is dispensable. Interestingly, some claimed that only the inverse scope reading was necessary, while others claimed that only the surface scope reading was necessary.

Those who thought that only the surface scope reading was necessary are plainly wrong. Their reasoning seems to be the following. The surface scope reading entails the inverse scope reading, which means that whenever the surface reading is true, the inverse scope reading is also true, so the inverse scope reading is superfluous. This reasoning is simply fallacious. It is indeed true that whenever the surface scope reading is true, the inverse scope reading is also true, but because the entailment here is only one-way, there are situations where the inverse scope reading is true but the surface scope reading is false. One such case is given above in

(11). There, only the inverse scope reading is true. So this shows that we at least need to say that the sentence as the inverse scope reading, because the sentence is judged as true there.

The other view is more sensible, according to which the sentence only has the inverse scope reading. The idea is as follows. The inverse scope reading is true in a superset of situations where the surface scope reading is true, so whether or the surface scope reading exists, the set of situations where the sentence is true will be the same, namely the situations where the inverse scope reading is true.

We will give five reasons to think that both the inverse scope and surface scope readings exist. Some of the arguments are indirect and weak, but jointly they make it plausible that the surface scope reading does exist.

3.1 Double Judgments

Consider the sentence (18) again.

(18) Somebody saw every movie.

In the above situation in (11), repeated here, we said that the sentence is judged as true.

(11)

$$\begin{tikzcd}
\text{Alice} \arrow[r] & \text{Movie 1} \\
\text{Bob} \arrow[ur] & \\
\text{Chris} \arrow[r] & \text{Movie 2} \\
\text{Daniel} \arrow[r] & \text{Movie 3}
\end{tikzcd}$$

The inverse scope reading is true here, while the surface scope reading is false. If the sentence really has these two readings, it should be judged as true and false at the same time. If you (or your native speaker informants) think that the sentence is both true and false at the same time, this is good evidence that both readings exist.

Notice that similar double judgments arise with other ambiguous sentences, e.g.

(14) a. I went to the bank on Wednesday.
b. John saw the man with binoculars.

3.2 Embedding

When the sentence is embedded under certain linguistic contexts (more precisely, non-upward monotonic contexts), the entailment relation changes. To see this, consider the following sentence.

(15) It's false that somebody saw every movie.

The two readings of this sentence are now:

(16) a. *Inverse Scope Reading*:
It's false that for every movie there is somebody who saw it.
b. *Surface Scope Reading*:
It's false that there is somebody who saw every movie.

Now the entailment goes from (16a) to (16b). If the inverse scope reading is true, i.e. it's false that for every movie, there is somebody who saw it, then there is a movie that nobody saw. This also makes the surface reading true, because if there is a movie that nobody saw, then it's guaranteed to be false that there is somebody who saw every movie. The entailment from (16b) to (16a) does not go through. You can easily imagine a context where (16b) is true but (16a) is false. And notice that such a context will constitute evidence that the surface scope reading exists, because if the sentence is judged true against such a context, that must be because of the surface reading. In fact, the context in is a case in point. There, the surface scope reading is true, while the inverse scope reading is false.

3.3 Switching the Quantifiers

Another way to change the entailment patter is by switching the quantifiers, as in the following example.

(17) Everybody saw some movie.

Now the universal quantifier 'everybody' is the subject. The two readings of the sentence are as in (18).

(18) a. *Inverse Scope Reading*:
There is some movie that everybody saw.
b. *Surface Scope Reading*:
For everybody, there is some movie that he or she saw.

Now, the inverse scope reading (18a) entails the surface scope reading (18b), but not vice versa. Then, we can create a context where only the weaker reading, namely the surface scope reading, is true. The scenario in again serves as a test case. If the sentence in (17) is judged as true, that must be because the sentence has a surface scope reading. And (17) indeed seems to be judged as true.

3.4 Other Quantifiers

There are quantifiers that don't give rise to asymmetric entailment patterns. Consider, for example, (19).

(19) Somebody saw exactly two movies.

The two readings of (19) are paraphrased as (20).

(20) a. *Inverse Scope Reading*:
There is somebody who saw exactly two movies.
b. *Surface Scope Reading*:
For exactly two movies, there is somebody who saw them.

These two readings do not stand in any entailment relation. To see this concretely, consider the following two situations.

(21)
$$\begin{tikzcd}
\text{Alice} \arrow[r] \arrow[dr] & \text{Movie 1} \\
\text{Bob} \arrow[ur] & \\
\text{Chris} \arrow[r] & \text{Movie 2} \\
\text{Daniel} \arrow[r] & \text{Movie 3}
\end{tikzcd}$$

(22)
$$\begin{tikzcd}
\text{Alice} \arrow[r] & \text{Movie 1} \\
\text{Bob} \arrow[ur] & \\
\text{Chris} \arrow[r] & \text{Movie 2} \\
\text{Daniel} \arrow[ur] & \text{Movie 3}
\end{tikzcd}$$

In (21), only the surface scope reading is true, while in (22), only the inverse scope reading is true. Whether (and how easily) the inverse scope reading is available is not very clear, but everyone would agree that the surface scope reading exists, i.e. the sentence is judged true in (21).

3.5 Other Languages

In certain languages (e.g. Korean, Mandarin Chinese, Japanese, German, Russian), a translation of (18) doesn't have an inverse scope reading, i.e. the sentence is judged false in the scenario depicted in (11). Here's a Japanese translation of (18).

(23)

dareka-ga	subeteno	eiga-o	mita.
someone-NOM	all	movie-ACC	saw

'Someone saw all the movies.'

Side note: Interestingly, the judgments change if a different word order is used. In Japanese, the object can optionally precede the subject (a phenomenon known as 'scrambling'), in which case, both readings seem to be available. Generally, the word order matters for scope judgments in many other languages.

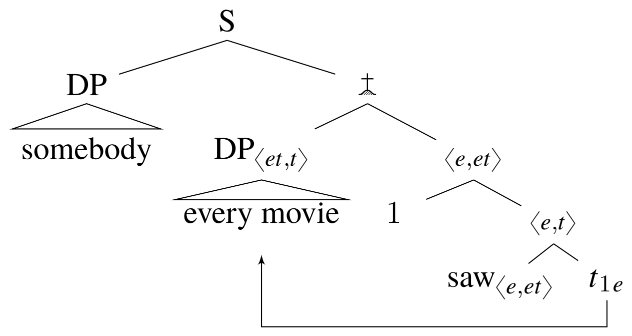
The Japanese sentence in (23) is judged true precisely when the surface scope reading is predicted to be true. Thus, the surface scope reading exists at least in languages like Japanese. Then there must be a mechanism that generates the surface scope reading. Of course, this does not entail that the English sentence in (18) has the surface scope reading, but it at least shows that it would not be so outlandish to assume that it has the surface scope reading as well.

3.6 How to Derive the Surface Scope Reading

Some of the above arguments are weak. For instance, the last one one is not about English, so you can't actually directly conclude that the English sentence in (18) has a surface scope reading. Also, the second, third, and fourth arguments are about different sentences than (18), which leaves open the possibility that there are sentences that have surface scope readings but (18) does not have one. Yet, generally speaking, our semantic theory needs to be able to generate the surface scope reading, in addition to the inverse scope reading.

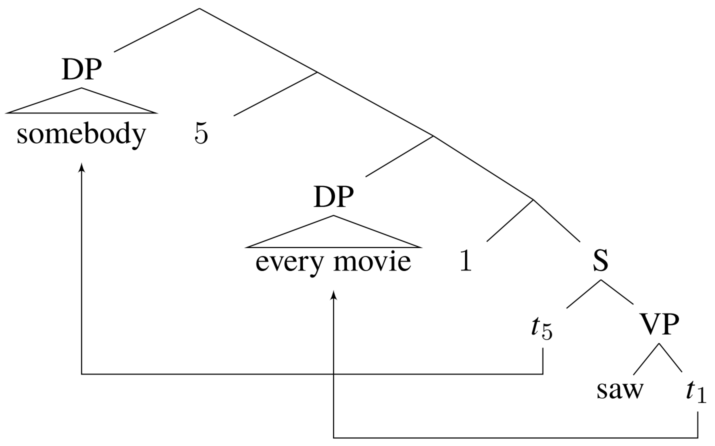
As we have seen, the inverse scope reading is easily captured by QRing the object quantifier above the subject, as demonstrated above. How do we derive the surface scope reading? Notice that moving the object quantifier below the subject quantifier won't work because of a type-mismatch.

(24)



If we assume that QR can optionally apply to the subject quantifier as well, we can have the representation in (25).

(25)



This will derive the surface scope reading. The computation is left for the homework assignment.

To sum up, QR can be seen as the mechanism that decides the scopes of quantificational DPs. In particular it accounts for scope ambiguity by assigning different interpreted structures for the same surface string. In the final week, we will discuss various constraints on quantifier scope, which under the present analysis can be seen as constraints on QR.