Introduction to Semantics

Lecture 3: Generalized Quantifier Theory I

Yasutada Sudo
UCL
y.sudo@ucl.ac.uk

OVA 2026 at Debreceni Egyetem

1 Semantic types

So far, we've encountered four types of model-theoretic objects that reside in our model. For an arbitrary model $\mathcal{M}$,

- Truth-values = denotations of (declarative sentences)

  (1) $[\![\texttt{Mike loves Chomsky}]\!]_{\mathcal{M}} \in \{0, 1\}$

- Individuals = denotations of proper names

  (2) a. $[\![\texttt{Mike}]\!]_{\mathcal{M}} \in D$
  b. $[\![\texttt{Chomsky}]\!]_{\mathcal{M}} \in D$

  We denote the set of individuals in $\mathcal{M}$ by D (called the domain of $\mathcal{M}$).

- Functions from individuals to truth-values = denotations of intransitive verbs and verb phrases

  (3) a. $[\![\texttt{yawned}]\!]_{\mathcal{M}} = [\lambda x \in D.\ 1 \text{ iff } x \text{ yawned in } \mathcal{M}]$
  b. $[\![\texttt{loves Chomsky}]\!]_{\mathcal{M}} = [\lambda x \in D.\ 1 \text{ iff } x \text{ loves } [\![\texttt{Chomsky}]\!]_{\mathcal{M}} \text{ in } \mathcal{M}]$

- Functions from individuals to functions from individuals to truth-values = denotations of transitive verbs

  (4) a. $[\![\texttt{loves}]\!]_{\mathcal{M}} = [\lambda x \in D.[\lambda y \in D.1 \text{ iff } y \text{ loves } x \text{ in } \mathcal{M}]]$
  b. $[\![\texttt{met}]\!]_{\mathcal{M}} = [\lambda x \in D.[\lambda y \in D.1 \text{ iff } y \text{ met } x \text{ in } \mathcal{M}]]$

We will give labels to these different types of model-theoretical objects. These labels are called **semantic types**.

- The semantic type of truth-values is $\texttt{t}$.
- The semantic type of individuals is $\texttt{e}$.
- The semantic type of functions from individuals to truth-values is $\langle \texttt{e}, \texttt{t} \rangle$.

- The semantic type of functions from individuals to functions from individuals to truth-values is $\langle \mathtt{e}, \langle \mathtt{e}, \mathtt{t} \rangle \rangle$.

More generally, every semantic type is either $\mathtt{t}$, $\mathtt{e}$, or $\langle \sigma, \tau \rangle$ (alt.: (σ, τ), $(\sigma \to \tau)$) for some semantic types σ and τ.

Here are some examples of semantics types you might encounter:

(5) $\mathtt{t}$ $\mathtt{e}$ $\langle \mathtt{e}, \mathtt{t} \rangle$ $\langle \mathtt{e}, \langle \mathtt{e}, \mathtt{t} \rangle \rangle$ $\langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{t} \rangle$ $\langle \langle \mathtt{e}, \mathtt{t} \rangle, \langle \mathtt{e}, \mathtt{t} \rangle \rangle$ $\langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{e} \rangle$ $\langle \langle \mathtt{e}, \mathtt{t} \rangle, \langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{t} \rangle \rangle$

Here are things that are *not* semantic types:

(6) $\langle \mathtt{e}, \mathtt{e}, \mathtt{t} \rangle$ $\langle \mathtt{e} \rangle$ $\langle \mathtt{et} \rangle$ $\mathtt{et}$

However, Heim & Kratzer 1998 define $\mathtt{et}$ as a shorthand for $\langle \mathtt{e}, \mathtt{t} \rangle$, and you might see it in published papers too. It makes complex types like $\langle \langle \mathtt{e}, \mathtt{t} \rangle, \langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{t} \rangle \rangle$ easier to grasp, $\langle \mathtt{et}, \langle \mathtt{et}, \mathtt{t} \rangle \rangle$.

Formally we define semantic types recursively as follows:

(7) a. $\mathtt{t}$ and $\mathtt{e}$ are semantic types.
b. If σ and τ are semantic types, then $\langle \sigma, \tau \rangle$ is a semantic type.
c. Nothing else is a semantic type.

We give them meaning by defining a 'domain' D_τ for each semantic type τ.

- $D_{\mathtt{t}} = \{ 0, 1 \}$
- $D_{\mathtt{e}} = D$
- $D_{\langle \sigma, \tau \rangle} = \{ f \mid f : D_\sigma \to D_\tau \}$

To indicate the semantic type, you can write something like:

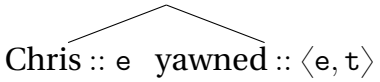
(8) a. $[\![\mathtt{yawned}]\!]_{\mathcal{M}} \in D_{\langle \mathtt{e}, \mathtt{t} \rangle}$
b. $[\![\mathtt{Socrates}]\!]_{\mathcal{M}} \in D_{\mathtt{e}}$
c. $[\![\mathtt{Plato\ smiled}]\!]_{\mathcal{M}} \in D_{\mathtt{t}}$

Sometimes :: is used:

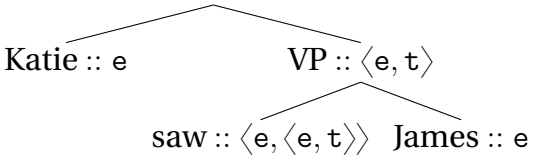
(9) a. $[\![\mathtt{yawned}]\!]_{\mathcal{M}} :: \langle \mathtt{e}, \mathtt{t} \rangle$
b. $[\![\mathtt{Socrates}]\!]_{\mathcal{M}} :: \mathtt{e}$
c. $[\![\mathtt{Plato\ smiled}]\!]_{\mathcal{M}} :: \mathtt{t}$

It's a good exercise to annotate a tree with the semantic type of each constituent.

(10) a.



b.



You will not use most of the functional semantic types (e.g., you will probably never use $\langle\langle \mathtt{t}, \mathtt{e}\rangle, \langle \mathtt{e}, \langle \mathtt{e}, \langle \mathtt{t}, \mathtt{e}\rangle\rangle\rangle\rangle$). But all these functions simply 'exist' in some sense, so we might as well have labels for them.

If your model contains more primitive objects (e.g., time intervals, degrees, events, possible worlds) then you will have more 'atomic' (non-functional) semantic types.

2 Nouns

Let us enrich our fragment with singular count nouns. Plural count nouns and mass nouns are put aside in this course.

For reasons of time, we will just spell out an analysis: Nouns are predicates on a par, and there are one-place nouns, which denote functions of type $\langle \mathtt{e}, \mathtt{t}\rangle$ and two-place nouns, which denote functions of type $\langle \mathtt{e}, \langle \mathtt{e}, \mathtt{t}\rangle\rangle$.

(11) For any model $\mathcal{M}$,

a. $[\![\mathtt{student}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x is a student in $\mathcal{M}$
b. $[\![\mathtt{cat}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x is a cat in $\mathcal{M}$
c. $[\![\mathtt{apple}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x is an apple in $\mathcal{M}$

(12) For any model $\mathcal{M}$,

a. $[\![\mathtt{picture}]\!]_{\mathcal{M}} = \lambda x \in D.\lambda y \in D.\ 1$ iff y is a picture of x in $\mathcal{M}$
b. $[\![\mathtt{sequel}]\!]_{\mathcal{M}} = \lambda x \in D.\lambda y \in D.\ 1$ iff y is a sequel to x in $\mathcal{M}$
c. $[\![\mathtt{part}]\!]_{\mathcal{M}} = \lambda x \in D.\lambda y \in D.\ 1$ iff y is a part of x in $\mathcal{M}$

Generally, noun phrases (NPs) denote functions of type $\langle \mathtt{e}, \mathtt{t}\rangle$. Although we will not give analyses of adjectives (to save time), it'll be handy in the following discussion to have some denotations of NPs containing them, e.g.

(13) For any model $\mathcal{M}$,

a. $[\![$ NP / British student $]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x is a British student in $\mathcal{M}$
b. $[\![$ NP / black cat $]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$iff x is a black cat in $\mathcal{M}$
c. $[\![$ NP / green apple $]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$iff x is a green apple in $\mathcal{M}$

In the following discussion, we will talk about such type-$\langle \mathtt{e}, \mathtt{t}\rangle$ functions and the sets of individuals they characterize interchangeably.

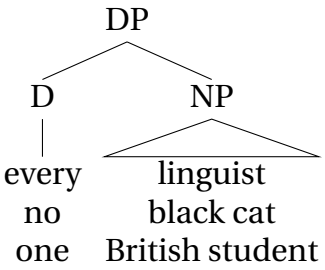
3 Generalized Quantifier Theory

We call DPs like (14) 'quantificational DPs':

(14) a. every linguist
b. no black cat
c. one British student

Let's assume the following structure for them.

(15)



You could work with a different syntactic structure, but in that case you would have to adjust your semantic analysis.

3.1 Quantificational DPs do not refer to individuals

Proper names denote individuals, but quantificational DPs do not.

Let us tentatively assume the contrary, namely, that quantificational DPs do refer to individuals. Then the following two sentences can be given essentially parallel analyses.

(16) a. Alice smiled.
b. Every linguist smiled.

3.1.1 Which individual?

If we assume that quantificational DPs refer to individuals, just like proper names, then a question immediately arises: Which individuals do they refer to?

Take, for example, 'every linguist'. Clearly, this phrase is not about a particular person, but about all linguists at the same time. If it were to refer to an individual, then that individual has to somehow represent the totality of all linguists. You might think that such a theory could be potentially developed, and that might be so.

However, other quantificational DPs like 'no linguist' are harder to analyze this way. If 'no linguist' were to refer to some specific individual, that individual should somehow represent no linguist. Also, we would need to make sure that the denotations of 'no linguist' and 'no student' could be different individuals, because the following sentences have different truth-conditions.

(17) a. No linguist smiled.

b. No student smiled.

Similarly, what would be an adequate individual for the extension of 'few linguists'? It might be any one linguist, or even no linguist, given that 'Few linguists are rich' is true when no linguists are rich (modulo the scalar implicature; see below), as well as when only one or two linguists are rich.

Relatedly, the following passages from *Through the Looking-Glass, and What Alice Found There* by Lewis Carroll, which you might have read, illustrate the inadequacy of the hypothesis we are after. Pay attention to how the White King interprets 'nobody', which I've put in bold font:

> 'Just look along the road, and tell me if you can see either of them.'
>
> 'I see **nobody** on the road,' said Alice.
>
> 'I only wish I had such eyes,' the King remarked in a fretful tone. 'To be able to see **Nobody**! And at the distance too! Why, it's as much as I can do to see real people, by this light!'
>
> .....
>
> 'Who did you pass on the road?' the King went on, holding out his hand to the Messenger for some more hay.
>
> '**Nobody**,' said the Messenger.
>
> 'Quite right,' said the King: 'this young lady saw him too. So of course **Nobody** walks slower than you.'
>
> 'I do my best,' the Messenger said in a sullen tone. 'I'm sure **nobody** walks much faster than I do!'
>
> 'He can't do that,' said the King, 'or else he'd have been here first.'

Here's another quote with similar double-meaning.

> This is a story about four people named Everybody, Somebody, Anybody and Nobody.
>
> There was an important job to be done and Everybody was sure that Somebody would do it.
>
> Anybody could have done it, but Nobody did.
>
> Somebody got angry about this, because it was Everybody's job. Everybody thought Anybody could do it, but Nobody realised that Everybody wouldn't do it.
>
> It ended up with that Everybody blamed Somebody when Nobody did what Anybody could have done.

3.1.2 Wrong inferential patterns

In addition to the question of which individuals they should refer to, quantificational DPs give rise to inferential patterns that are different from proper names, which would be unexpected, if these two types of DPs both referred to individuals.

Take the example in (18). Assuming that everyone speaks at least one language, the following statement is a tautology, meaning it's guaranteed to be true.

(18) Alex is mono-lingual or Alex is multi-lingual.

Now replace the proper name *Alex* with a quantificational DP 'every linguist'.

(19) Every linguist is mono-lingual or every linguist is multi-lingual.

This is not a tautology, unlike (18), and can be false. In fact, it is false with respect to the actual state of affairs, given that there actually are both mono-lingual and multi-lingual linguists.

Similarly, consider the following pair of sentences.

(20) a. Alex lives in London or in Paris.
b. Alex lives in London or Alex lives in Paris.

These two sentences have the same truth-conditions. Now replace the proper name with 'every linguist'.

(21) a. Every linguist lives in London or in Paris.
b. Every linguist lives in London or every linguist lives in Paris.

These two sentences are not synonymous, although one of them entails the other (which entails which?).

Likewise, the sentences in (22) are synonymous, but the sentences in (23) are not.

(22) a. Alex went to Paris and Amsterdam.
b. Alex went to Paris and Alex went to Amsterdam.

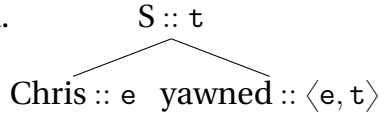
(23) a. No linguist went to Paris and Amsterdam.
b. No linguist went to Paris and no linguist went to Amsterdam.

3.2 Generalized Quantifiers

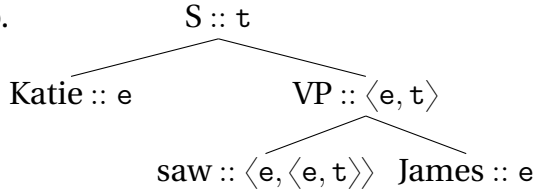
The above considerations point to the conclusion that quantificational DPs do not refer to individuals. Then what are their denotations?

We can actually deduce their semantic types, based on the assumption that quantificational DPs combine with VPs via Functional Application. Recall:

(24) a.

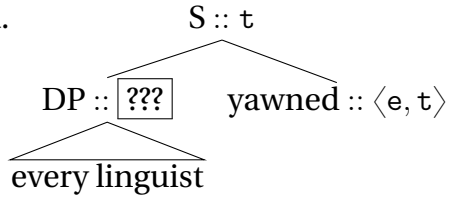


b.

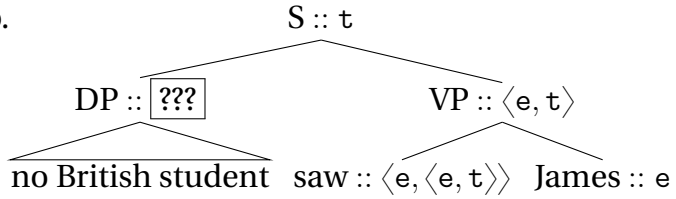


With a quantificational subject:

(25) a.



b.



If these quantificational DPs are not referring expressions, we are left with only one possibility: They are functions that take a type-$\langle \mathtt{e}, \mathtt{t} \rangle$ function and return a type-t object, a truth-value. For any model $\mathcal{M}$,

(26) a. $[\![\texttt{every linguist}]\!]_{\mathcal{M}} :: \langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{t} \rangle$
b. $[\![\texttt{no British student}]\!]_{\mathcal{M}} :: \langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{t} \rangle$

We could furthermore figure out the semantic types of the quantificational determiners, based on the assumption that the NPs are of type $\langle \mathtt{e}, \mathtt{t} \rangle$, but let us do that in the next lecture.

Having figured out the semantic types, let us figure out the denotations. Given the semantic types, the denotations have to look like:

(27) a. $[\![\texttt{every linguist}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \mathtt{e},\mathtt{t} \rangle}.\ 1$ iff ???
b. $[\![\texttt{no British linguist}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \mathtt{e},\mathtt{t} \rangle}.\ 1$ iff ???

The input type-$\langle \mathtt{e}, \mathtt{t} \rangle$ function f is whatever the VP denotes, for example $[\![\texttt{yawned}]\!]_{\mathcal{M}}$. We know the truth-conditions of 'Every linguist yawned' (based on our truth-conditional intuitions), as well as $[\![\texttt{yawned}]\!]_{\mathcal{M}}$ (as we discussed in the previous lecture):

(28) a. $[\![\texttt{yawned}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x yawned in $\mathcal{M}$
b. $[\![\texttt{every linguist yawned}]\!]_{\mathcal{M}} = 1$ iff every linguist in $\mathcal{M}$ yawned in $\mathcal{M}$

Now, crucially, we can rewrite the truth-conditions (28b) in terms of (28a) as follows (OK, this step might not be easy at the beginning, but for now if you understand the equivalence, that's good enough).

$$[\![\texttt{every linguist yawned}]\!]_{\mathcal{M}} = 1 \text{ iff every linguist in } \mathcal{M} \text{ yawned in } \mathcal{M}$$
$$\text{iff for every linguist } x \text{ in } \mathcal{M},\ [\![\texttt{yawned}]\!]_{\mathcal{M}}(x) = 1$$

This representation makes it clear which part of the truth-conditions is coming from 'yawned' and which part is coming from 'every linguist'. That is, by abstracting over this particular VP, we will arrive at the following denotation for 'every linguist':

(29) $[\![\texttt{every linguist}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \mathtt{e},\mathtt{t} \rangle}.\ 1$ iff for every linguist x in $\mathcal{M}$, $f(x) = 1$

By the same reasoning the denotation of 'no British student' should be:

(30) $[\![\texttt{no British student}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \mathtt{e},\mathtt{t} \rangle}.\ 1$ iff for no British student x in $\mathcal{M}$, $f(x) = 1$

The body part of the function could be written in different equivalent ways, e.g.

(31) $[\![\texttt{no British student}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \mathtt{e},\mathtt{t}\rangle}.\ 1$ iff for every British student x in $\mathcal{M}$, $f(x) = 0$

And likewise:

(32) $[\![\texttt{every linguist}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \mathtt{e},\mathtt{t}\rangle}.\ 1$ iff for no linguist x in $\mathcal{M}$, $f(x) = 0$

These functions of type $\langle\langle \mathtt{e}, \mathtt{t}\rangle, \mathtt{t}\rangle$ are called **generalized quantifiers**.[1]

4 Sets and characteristic functions

According to our analysis so far, the denotations of intransitive verbs and those of VPs containing a transitive verb and a proper name object as functions of type $\langle \mathtt{e}, \mathtt{t}\rangle$.

(33) For any model $\mathcal{M}$,
- a. $[\![\texttt{yawned}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x yawned in $\mathcal{M}$
- b. $[\![\texttt{loves Chomsky}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x loves $[\![\texttt{Chomsky}]\!]_{\mathcal{M}}$ in $\mathcal{M}$

Each of these type-$\langle \mathtt{e}, \mathtt{t}\rangle$ functions can be seen as dividing the set D of individuals in $\mathcal{M}$ into two sets: the subset of D that it maps to truth-value 1 and the subset of D that it maps to truth-value 0. Concretely, for (33a), the two sets are (34), and for (33b), they are (35).

(34)
- a. $\{\, x \mid x \in D \text{ and } x \text{ yawned in } \mathcal{M} \,\}$
- b. $\{\, x \mid x \in D \text{ and } x \text{ didn't yawn in } \mathcal{M} \,\}$

(35)
- a. $\{\, x \mid x \in D \text{ and } x \text{ loves } [\![\texttt{Chomsky}]\!]_{\mathcal{M}} \text{ in } \mathcal{M} \,\}$
- b. $\{\, x \mid x \in D \text{ and } x \text{ doesn't love } [\![\texttt{Chomsky}]\!]_{\mathcal{M}} \text{ in } \mathcal{M} \,\}$

We say that the function in (33a) **characterizes** (34a), or equivalently, the function in (33a) is the **characteristic function** of (34a). Similarly, (33b) characterizes (35a).

Generally, any function f that maps individuals to truth-values (in symbols, $f : D \to \{0, 1\}$, or $f : D_{\mathtt{e}} \to D_{\mathtt{t}}$) characterizes the set of individuals $\{\, x \mid f(x) = 1 \,\}$.[2]

It is important to keep in mind that not all functions characterize sets. Only functions that return truth-values do. For instance, a function that takes an individual and returns another function does not characterize a set. So $[\![\texttt{loves}]\!]_{\mathcal{M}}$ does not characterize a set.[3]

When a function f that returns truth-values is given, we can uniquely determine the set it characterizes, namely $\{\, x \mid f(x) = 1 \,\}$. Let's write this as follows. '$\coloneqq$' means 'is defined to be equivalent to'.

$$\mathsf{set}(f) \coloneqq \{\, x \mid f(x) = 1 \,\}$$

[1] More specifically, they are generalized quantifiers of type $\langle 1\rangle$. Type-$\langle n\rangle$ generalized quantifiers bind n-many variables, and type-$\langle 1, 1\rangle$ generalized quantifiers bind a variable in two different formulas/predicates. Type-$\langle 2, 2\rangle$ generalized quantifiers like two-place predicates/relations, for example.

[2] One could generalize this further: Any function f from D_τ to $D_{\mathtt{t}}$ for any semantic type τ characterizes $\{\, x \mid f(x) = 1 \,\}$.

[3] Although functions of type $\langle \mathtt{e}, \langle \mathtt{e}, \mathtt{t}\rangle\rangle$ are 'isomorphic' (see below) to functions from pairs of individuals to truth-values, which characterize sets of pairs of individuals.

Conversely, furthermore, when a set S of individuals is given, we can uniquely define its characteristic function, namely, the function that maps every member of S to truth-value 1 and everything else to truth-value 0.[4] We can write this as:

$$\mathsf{function}(S) = \lambda x \in D.\ 1 \text{ iff } x \in S$$

It is important to notice that if f characterizes $\{\, x \mid f(x) = 1 \,\}$, the characteristic function for $\{\, x \mid f(x) = 1 \,\}$ is f. That is, for any $f \in D_{\langle \mathtt{e},\mathtt{t}\rangle}$:

$$\mathsf{function}(\mathsf{set}(f)) = f$$

Similarly, the characteristic function of any set S characterizes nothing but S itself. That is, for any $S \subseteq D$,

$$\mathsf{set}(\mathsf{function}(S)) = S$$

This means that if a set and its characteristic function 'carry the same amount of information' in the sense that one can be uniquely recovered from the other. This equivalence allows us to use one as a proxy for the other, although they are formally distinct. We say there is a one-to-one correspondence, or isomorphism between functions of type $\langle \mathtt{e}, \mathtt{t}\rangle$ and sets of individuals.

4.1 Generalized quantifiers as sets of sets of individuals

Generalized quantifiers, by virtue of being of type $\langle\langle \mathtt{e}, \mathtt{t}\rangle, \mathtt{t}\rangle$, can be seen as sets of functions of type $\langle \mathtt{e}, \mathtt{t}\rangle$.

(36) For any model $\mathcal{M}$

a. $[\![\texttt{every linguist}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \mathtt{e},\mathtt{t}\rangle}.\ 1$ iff for every linguist x in $\mathcal{M}$, $f(x) = 1$

b. $\mathsf{set}([\![\texttt{every linguist}]\!]_{\mathcal{M}}) = \left\{ f \;\middle|\; \begin{array}{l} f \in D_{\langle \mathtt{e},\mathtt{t}\rangle} \text{ and} \\ \text{for every linguist } x \text{ in } \mathcal{M},\ f(x) = 1 \end{array} \right\}$

And because type-$\langle \mathtt{e}, \mathtt{t}\rangle$ functions can be seen as sets of individuals, generalized quantifiers can be seen as sets of sets of individuals. Let's use SET to denote those.

(37) $$\mathsf{SET}([\![\texttt{every linguist}]\!]_{\mathcal{M}}) = \left\{ S \;\middle|\; \begin{array}{l} S \subseteq D \text{ and} \\ \text{for every linguist } x \text{ in } \mathcal{M},\ \mathsf{function}(S)(x) = 1 \end{array} \right\}$$

It's customary to move things like $S \subseteq D$ to the left of '|', to save space. Let's adopt this convention.

(38) $$\mathsf{SET}([\![\texttt{every linguist}]\!]_{\mathcal{M}}) = \{\, S \subseteq D \mid \text{for every linguist } x \text{ in } \mathcal{M},\ \mathsf{function}(S)(x) = 1 \,\}$$

Note that this is the same thing as:

(39) $$\mathsf{SET}([\![\texttt{every linguist}]\!]_{\mathcal{M}}) = \{\, S \subseteq D \mid \text{for every linguist } x \text{ in } \mathcal{M},\ x \in S \,\}$$

Or if you want to state it more naturally?

[4] If we want to generalize this to all types of sets, we will have to keep track of the semantic type, but we will only talk about sets of individuals here.

(40) $\mathsf{SET}(\llbracket \texttt{every linguist} \rrbracket_{\mathcal{M}}) = \{\, S \subseteq D \mid S \text{ contains every linguist } x \text{ in } \mathcal{M} \,\}$

We can analyze other generalized quantifiers in terms of sets of individuals as well.

(41) a. $\llbracket \texttt{no British student} \rrbracket_{\mathcal{M}} = \lambda f \in D_{\langle \mathtt{e},\mathtt{t} \rangle}.\, 1$ iff for no British student x in $\mathcal{M}$, $f(x) = 1$
b. $\mathsf{SET}(\llbracket \texttt{no British student} \rrbracket_{\mathcal{M}}) = \{\, S \subseteq D \mid \text{for no British student } x \text{ in } \mathcal{M},\, x \in S \,\}$

Again, there are other ways of stating the same thing:

(42) $\begin{aligned} \mathsf{SET}(\llbracket \texttt{no British student} \rrbracket_{\mathcal{M}}) &= \{\, S \subseteq D \mid \text{for every British student } x \text{ in } \mathcal{M},\, x \notin S \,\} \\ &= \{\, S \subseteq D \mid S \text{ contains no British student } x \text{ in } \mathcal{M} \,\} \end{aligned}$

4.2 Predicates of predicates of individuals

According to our fragment, VPs denote type-$\langle \mathtt{e}, \mathtt{t} \rangle$ functions. They are **predicates of individuals**, specifying which individuals have the property in question and which ones do not (in a given model $\approx$ state of affairs). For example, the denotation of the VP 'likes Chomsky' is the predicate of individual, telling you who likes Chomsky and who doesn't (in the model). its set counterpart contains those and only those individuals that like Chomsky.

Generalized quantifiers, which are functions of type $\langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{t} \rangle$, are **predicates of predicates of individuals**, specifying which predicates of individuals have the property in question (in a given model). For example, the denotation of 'every linguist' is the predicate of predicates of individuals, telling you which predicates of individuals are true of every linguist, and which predicates are not true of every linguist. Equivalently, we can regard those predicates of individuals as sets of individuals, in which case the denotation of 'every linguist' tells you which sets of individuals contain every linguist, and which sets of individuals don't. The set counterpart of the denotation of 'every linguist' is the set of all predicates that have this property, or in set talk, the set of sets of individuals that contain eery linguist that you can find in the model.

5 Proper names as generalized quantifiers

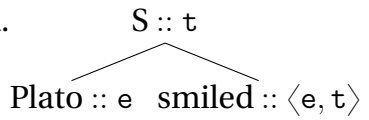
In our analysis so far, we have two types of DPs:

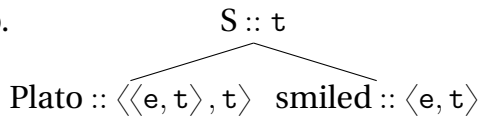
- Proper names, which are referring expressions, and denote individuals (type-$\mathtt{e}$ elements).
- Quantificational DPs, which are non-referring expressions, and denote generalized quantifiers, functions of type $\langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{t} \rangle$.

In the theoretical framework we are adopting, it is not necessary that all DPs have the same semantic type. More generally, it is not necessary for syntactic categories to have a unique semantic type corresponding to them.

Nonetheless, it is interesting to notice that proper names (and other referring expressions) can be analyzed as denoting generalized quantifiers/functions of type $\langle \langle \mathtt{e}, \mathtt{t} \rangle, \mathtt{t} \rangle$, too (whereas quantificational DPs cannot be analyzed as having type-$\mathtt{e}$ denotations; see above).

That is, instead of (43a), we can have (43b).

(43) a. 

b. 

As we will see, the re-analysis of proper name denotations in terms of generalized quantifiers will not change anything substantial, thanks to an isomorphism between individuals and a certain sub-class of generalized quantifiers.

Just to be explicit, let us distinguish these two analyses via subscripts, as in (44).

(44) For any model $\mathcal{M}$,
a. $[\![\texttt{Plato}_{\texttt{ind}}]\!]_{\mathcal{M}}$ is the individual in $\mathcal{M}$ that the name 'Plato' refers to in $\mathcal{M}$
b. $[\![\texttt{Plato}_{\texttt{GQ}}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \texttt{e},\texttt{t}\rangle}.\ 1 \text{ iff } f([\![\texttt{Plato}_{\texttt{ind}}]\!]_{\mathcal{M}}) = 1$

In terms of sets, the generazlied quantifier denotation can be rewritten as (45).

(45)
$$\begin{aligned}\mathsf{SET}([\![\texttt{Plato}_{\texttt{GQ}}]\!]_{\mathcal{M}}) &= \{\, S \subseteq D \mid 1 \text{ iff } \mathsf{function}(S)([\![\texttt{Plato}_{\texttt{ind}}]\!]_{\mathcal{M}}) = 1 \,\} \\ &= \{\, S \subseteq D \mid 1 \text{ iff } [\![\texttt{Plato}_{\texttt{ind}}]\!]_{\mathcal{M}} \in S \,\}\end{aligned}$$

In words, this is the set of all predicates that are true of Plato, or equivalently, the set of all sets that contain Plato.

Importantly, whenever an individual is given, it's easy to construct the corresponding generalized quantifier. You simply collect all the predicates that are true of that individual, and form a set.

(46) For any individual $x \in D$, $\mathsf{GQ}(x) \coloneqq [\lambda f \in D_{\langle \texttt{e},\texttt{t}\rangle}.\ f(x) = 1]$

Similarly, for any generalized quantifier that is constructed this way, it's possible to recover the individual that it is constructed from. This is because for each such generalized quantifier constructed from an individual x, the set version of it will contain a unique singleton set $\{\, x \,\}$, and the sole member of that singleton set is the individual it is constructed from. Formally,

(47) For any generalized quantifier $Q \in D_{\langle\langle \texttt{e},\texttt{t}\rangle,\texttt{t}\rangle}$ for which there is a unique individual $x \in D$ such that $Q(\lambda y \in D_e.\ y = x) = 1$, $\mathsf{indiv}(Q) \coloneqq$ the unique individual x such that $Q(\lambda y \in D_e.\ y = x) = 1$.

Or equivalently,

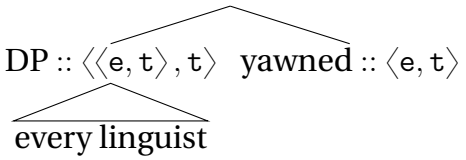
(48) For any generalized quantifier $Q \in D_{\langle\langle \texttt{e},\texttt{t}\rangle,\texttt{t}\rangle}$ for which there is a unique individual $x \in D$ $\{\, x \,\} \in \mathsf{SET}(Q)$, $\mathsf{indiv}(Q) \coloneqq$ the unique individual x such that $\{\, x \,\} \in \mathsf{SET}(Q)$.

And it can be shown that for any individual x, $\mathsf{indiv}(\mathsf{GQ}(x)) = x$, and any generalized quantifier Q that that is in the domain of $\mathsf{indiv}(\cdot)$, $\mathsf{GQ}(\mathsf{indiv}(Q)) = Q$. This means that there is an isomorphism between D and those generalized quantifiers that are in the domain of $\mathsf{indiv}(\cdot)$, and we can regard them as 'the same thing'.

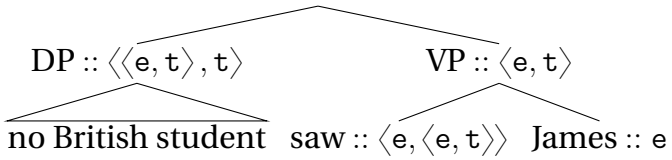
6 Fragment so far

In this lecture, we added singular count nouns and quantificational DPs containing singular count nouns such as the following.

(49) a.



b.



Our fragment therefore have:

- Sentences containing an intransitive verb and a proper name or quantificational DP as subject.
- Sentences containing an transitive verb, a proper name or quantificational DP as subject, and a proper name as object.

We haven't talked about quantificational DPs in object position. We'll discuss them in Lecture 5. For now, all objects are proper names.

Singular count nouns have denotations like the following. There are one-place and two-place ones (just like verbs).

(50) For any model $\mathcal{M}$,

a. $[\![\texttt{student}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x is a student in $\mathcal{M}$
b. $[\![\texttt{cat}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x is a cat in $\mathcal{M}$
c. $[\![\texttt{apple}]\!]_{\mathcal{M}} = \lambda x \in D.\ 1$ iff x is an apple in $\mathcal{M}$

(51) For any model $\mathcal{M}$,

a. $[\![\texttt{picture}]\!]_{\mathcal{M}} = \lambda x \in D.\lambda y \in D.\ 1$ iff y is a picture of x in $\mathcal{M}$
b. $[\![\texttt{sequel}]\!]_{\mathcal{M}} = \lambda x \in D.\lambda y \in D.\ 1$ iff y is a sequel to x in $\mathcal{M}$
c. $[\![\texttt{part}]\!]_{\mathcal{M}} = \lambda x \in D.\lambda y \in D.\ 1$ iff y is a part of x in $\mathcal{M}$

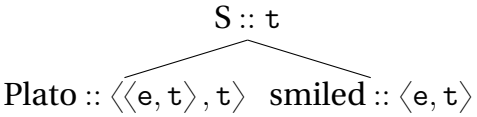
Quantificational DPs denote functions of type $\langle\langle \texttt{e}, \texttt{t}\rangle, \texttt{t}\rangle$, which are called **generalized quantifiers**.

(52)

a. $[\![\texttt{every linguist}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \texttt{e},\texttt{t}\rangle}.\ 1$ iff for every linguist x in $\mathcal{M}$, $f(x) = 1$
b. $[\![\texttt{no British student}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \texttt{e},\texttt{t}\rangle}.\ 1$ iff for no British student x in $\mathcal{M}$, $f(x) = 1$
c. $[\![\texttt{some black cat}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \texttt{e},\texttt{t}\rangle}.\ 1$ iff for some black cat x in $\mathcal{M}$, $f(x) = 1$
d. $[\![\texttt{one book}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \texttt{e},\texttt{t}\rangle}.\ 1$ iff for one book x in $\mathcal{M}$, $f(x) = 1$

We have also discussed an alternative analysis of proper names (in subject position) as quantificational DPs, where they are given generalized quantifiers as their denotations, but importantly, there is no substantial difference from the original analysis of them as referring expressions.

(53)



(54) For any model $\mathcal{M}$,

a. $[\![\texttt{Plato}_{\texttt{ind}}]\!]_{\mathcal{M}}$ is the individual in $\mathcal{M}$ that the name 'Plato' refers to in $\mathcal{M}$
b. $[\![\texttt{Plato}_{\texttt{GQ}}]\!]_{\mathcal{M}} = \lambda f \in D_{\langle \texttt{e},\texttt{t}\rangle}.\ 1 \text{ iff } f([\![\texttt{Plato}_{\texttt{ind}}]\!]_{\mathcal{M}}) = 1$

For now, it doesn't really matter which analysis we take.

The rest of the theory stays the same as before.

- Intransitive verbs denote type-$\langle \texttt{e}, \texttt{t}\rangle$ functions.

(55) For any model $\mathcal{M}$,

a. $[\![\texttt{yawned}]\!]_{\mathcal{M}} = [\lambda x \in D.\ 1 \text{ iff } x \text{ yawned in } \mathcal{M}]$
b. $[\![\texttt{smokes}]\!]_{\mathcal{M}} = [\lambda x \in D.\ 1 \text{ iff } x \text{ smokes in } \mathcal{M}]$
c. $[\![\texttt{smiled}]\!]_{\mathcal{M}} = [\lambda x \in D.\ 1 \text{ iff } x \text{ smiled in } \mathcal{M}]$
d. $[\![\texttt{danced}]\!]_{\mathcal{M}} = [\lambda x \in D.\ 1 \text{ iff } x \text{ danced in } \mathcal{M}]$

- Transitive verbs denote type-$\langle \texttt{e}, \langle \texttt{e}, \texttt{t}\rangle\rangle$ functions.

a. $[\![\texttt{saw}]\!]_{\mathcal{M}} = [\lambda x \in D.\ [\lambda y \in D.\ 1 \text{ iff } y \text{ saw } x \text{ in } \mathcal{M}]]$
b. $[\![\texttt{likes}]\!]_{\mathcal{M}} = [\lambda x \in D.\ [\lambda y \in D.\ 1 \text{ iff } y \text{ likes } x \text{ in } \mathcal{M}]]$
c. $[\![\texttt{invited}]\!]_{\mathcal{M}} = [\lambda x \in D.\ [\lambda y \in D.\ 1 \text{ iff } y \text{ likes } x \text{ in } \mathcal{M}]]$
d. $[\![\texttt{hates}]\!]_{\mathcal{M}} = [\lambda x \in D.\ [\lambda y \in D.\ 1 \text{ iff } y \text{ hates } x \text{ in } \mathcal{M}]]$

- The sole compositional rule is Functional Application.

(56) **Functional Application**

For any model $\mathcal{M}$, if α is made up of β and γ (that is, either  or  , whichever is grammatical) and $[\![\beta]\!]_{\mathcal{M}}$ is a function such that $[\![\gamma]\!]_{\mathcal{M}} \in \text{dom}\left([\![\beta]\!]_{\mathcal{M}}\right)$, then

$$\left[\!\!\left[\begin{array}{c} \alpha \\ \diagup\!\diagdown \\ \beta \quad \gamma \end{array} \right]\!\!\right]_{\mathcal{M}} = [\![\beta]\!]_{\mathcal{M}}\left([\![\gamma]\!]_{\mathcal{M}}\right)$$

Just to be a bit more concrete, consider an example model $\mathcal{L}$, where:

- There are five individuals in total in $\mathcal{L}$.

(57) $D_{\mathcal{L}} = \{a, b, c, d, e\}$

- There are exactly three linguists in $\mathcal{L}$, namely, a, b, and c.

(58) $[\![\texttt{linguist}]\!]_{\mathcal{L}} = \{a, b, c\}$

- Let's also assume that a, b, c and e smiled, but d didn't in $\mathcal{L}$.

(59) $[\![\texttt{smiled}]\!]_{\mathcal{L}} = \{\, x \in D_{\mathcal{L}} \mid x \text{ smiled in } \mathcal{L} \,\}$
$= \{\, a, b, c, e \,\}$

- Now, let's analyze 'every linguist'. According to the fragment, its denotation with respect to $\mathcal{L}$ is:

(60) $[\![\texttt{every linguist}]\!]_{\mathcal{L}} = \lambda f \in D_{\langle \mathsf{e},\mathsf{t}\rangle}.\ 1 \text{ iff for every linguist } x \text{ in } \mathcal{L},\ f(x) = 1$

Consider the set version of this:

(61) $\mathsf{SET}([\![\texttt{every linguist}]\!]_{\mathcal{L}}) = \{\, S \subseteq D_{\mathcal{L}} \mid S \text{ contains every linguist in } \mathcal{L} \,\}$
$= \left\{ \begin{array}{c} \{\, a, b, c \,\}, \\ \{\, a, b, c, d \,\}, \\ \{\, a, b, c, e \,\}, \\ \{\, a, b, c, d, e \,\} \end{array} \right\}$

- $[\![\texttt{smiled}]\!]_{\mathcal{L}}$ is a member of (61), so 'Every linguist smiled' is true with respect to $\mathcal{L}$.
- Similarly, we can represent the meaning of 'no linguist'.

(62) $[\![\texttt{no linguist}]\!]_{\mathcal{L}} = \lambda f \in D_{\langle \mathsf{e},\mathsf{t}\rangle}.\ 1 \text{ iff for no linguist } x \text{ in } \mathcal{L},\ f(x) = 1$

(63) $\mathsf{SET}([\![\texttt{no linguist}]\!]_{\mathcal{L}}) = \{\, S \subseteq D_{\mathcal{L}} \mid S \text{ contains no linguist in } \mathcal{L} \,\}$
$= \left\{ \begin{array}{c} \{\, d \,\}, \\ \{\, e \,\}, \\ \{\, d, e \,\} \end{array} \right\}$

- $[\![\texttt{smiled}]\!]_{\mathcal{L}}$ is not a member of (63), so 'No linguist smiled' is false with respect to $\mathcal{L}$.

References

Heim, Irene & Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Oxford: Blackwell.