Quantifiers and the semantics of movement

1 Review: The denotation of quantifiers

Exercise: Draw a tree and compute the truth conditions for "Every dinosaur (is) red" based on the lexical entries below:

- $[\![\text{every}]\!]_{\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle} = \lambda f_{\langle e,t\rangle}.[\lambda g_{\langle e,t\rangle}.$ for all $x \in D_e$ such that $f(x) = 1$, $g(x) = 1]$
- $[\![\text{dinosaur}]\!]_{\langle e,t\rangle} = \lambda x_e.x$ is a dinosaur
- $[\![\text{red}]\!]_{\langle e,t\rangle} = \lambda x_e.x$ is red

Below are lexical entries for a few other quantifiers we discussed last week:

- $[\![\text{some}]\!]_{\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle} = \lambda f_{\langle e,t\rangle}.[\lambda g_{\langle e,t\rangle}.$ there is some $x \in D_e$ such that $f(x) = 1$ and $g(x) = 1]$
- $[\![\text{two}]\!]_{\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle} = \lambda f_{\langle e,t\rangle}.[\lambda g_{\langle e,t\rangle}.$ there are two $x \in D_e$ such that $f(x) = 1$ and $g(x) = 1]$
- $[\![\text{no}]\!]_{\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle} = \lambda f_{\langle e,t\rangle}.[\lambda g_{\langle e,t\rangle}.$ there is no $x \in D_e$ such that $f(x) = 1$ and $g(x) = 1]$

Call the first argument of a quantifier its *restrictor* and the second its *nuclear scope*.

2 Quantifier scope and movement

'Raising' predicates are predicates whose syntactic subjects are the semantic arguments of an embedded clause.

(1) a. It is required that no student come to class.
    b. No student is required to come to class.

(2) a. It is 3% likely that every coin will land heads.
    b. Every coin is 3% likely to land heads.

(3) a. It is expected that a Kenyan will win the race.
    b. A Kenyan is expected to win the race.

We will assume here that the (a) and (b) sentences in these examples are derivationally related by *movement* of the subject NP.[1]

Q: How do the (a) and (b) sentences differ in meaning?

[1] If the subject doesn't raise out of the embedded clause (a), English requires an expletive 'it' (or 'there') to occupy the subject position.

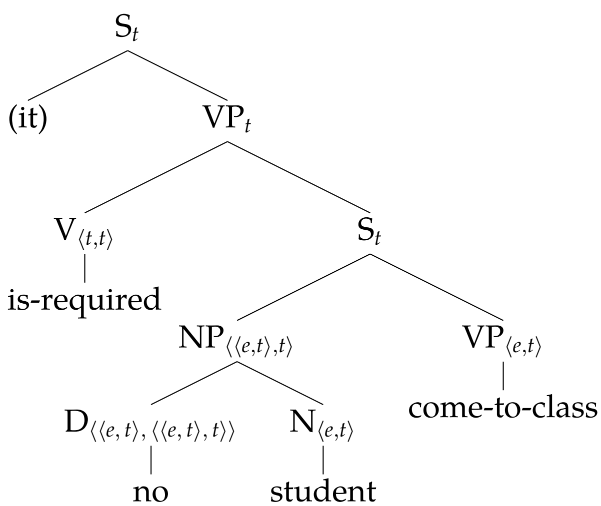
2.1 Computing "It is required that..."

We will assume a fairly simplistic $\langle t, t\rangle$ lexical entry for "is required," where "is required" takes a proposition (type t) and requires that it be true:

(4) $[\![\text{is-required}]\!]_{\langle t,t\rangle} = \lambda p_t.p$ must be true

We will furthermore assume the following simplified syntactic structure:

(5)



Exercise: Compute the truth conditions for (5).

2.2 Computing "NP is required to..."

As stated above, we assume that the (b) sentences are derived from a structure like (5), but with *movement* of the embedded subject to the matrix subject position, which leaves a *trace*.

(6) $[_S$ [No student] is-required $[_S$ *t* to come to class]]

The question is *how movement is interpreted.*

Intuition: "No student is required to come to class" can be rephrased using an "it is required that..." clause, as follows:

(7) No student *is such that* [it is required that [*she/he* come to class]]

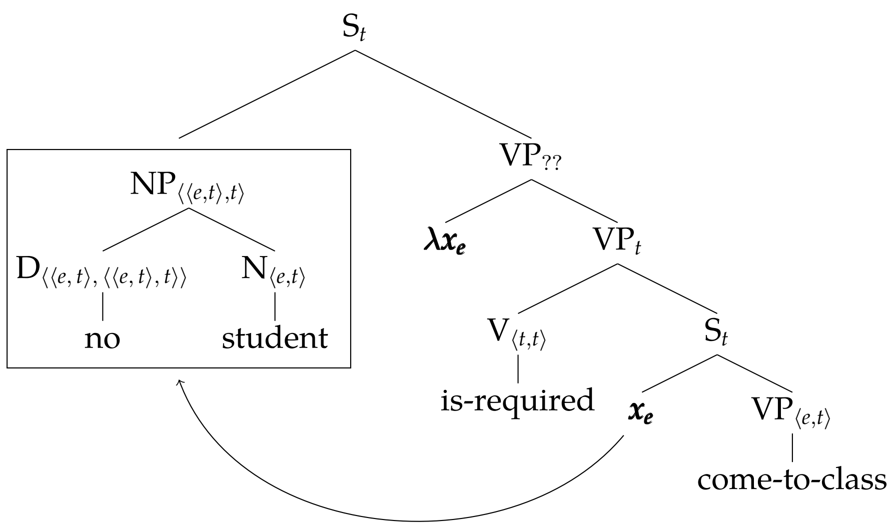
Two parts to this: (a) the trace is interpreted as a variable (here the pronoun *she/he*) and (b) we have to understand "is such that."

More formally:

(8) $[\text{No } \underbrace{[\text{student}]}_{\text{restrictor of } no}] \ \underbrace{[\lambda x.\text{ it is required that } [x \text{ come to class}]]}_{\text{nuclear scope of } no}$

The trace is interpreted as a matching variable (here, x of type e), and we *abstract* over this variable to form a nuclear scope of type $\langle e,t\rangle$. We will do this by inserting a matching λ in the syntax, below the position that we move to. This is called **Predicate Abstraction**:

(9)



Predicate Abstraction takes the type t "is-required [t come to class]" and interpret it as the type $\langle e,t\rangle$ *predicate* "$\lambda x.$ is-required [x come to class]."

Exercise: What type is the VP that dominates λx? Compute the truth conditions for (9).

To summarize:

(10) **Movement of $\boldsymbol{\alpha}$ from position A to position B:**

a. α is moved to position B;

b. α in A is replaced with a trace, interpreted as x;

c. A matching lambda node, λx, is adjoined below B (normally on the sister of B).

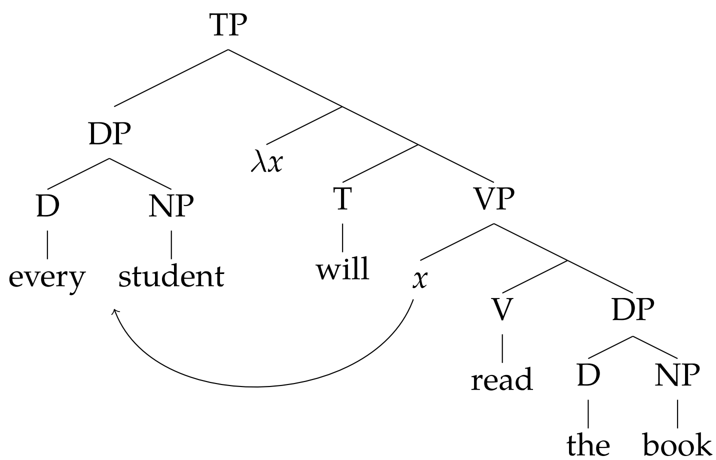
for some choice of variable x—don't reuse variables

Note: This is slightly different in Heim & Kratzer. There, movement leaves a λ with an index (a number) which matches an index on the trace, and then there's a special composition rule to interpret the λ node, called Predicate Abstraction. The more pedantic Heim & Kratzer rule is useful for issues that we will not get into in this course, such as the interpretation of pronouns.

3 Modern syntactic assumptions

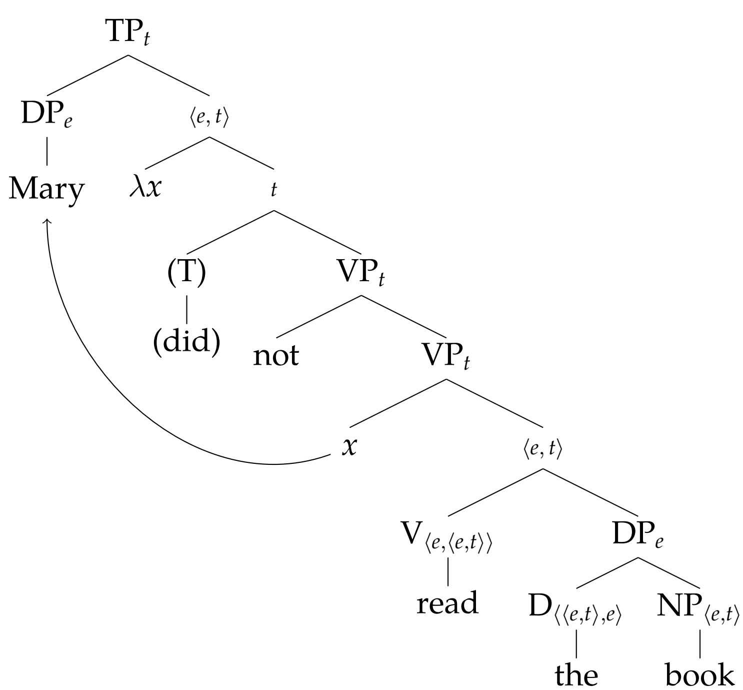
For the purposes of this course, we will assume that clauses have the projections (C(omplementizer)P >) T(ense)P > VP, with the subject generated in the specifier of VP.[2] Nominals will be D(eterminer)P(hrase)s. So "Every student will read the book" looks like this:

(11)



One consequence: we no longer have to cheat when talking about negation. Last week we said that, for example, we should model "Mary did not read the book" as "not [Mary read the book]." This is no longer a problem:

(12)



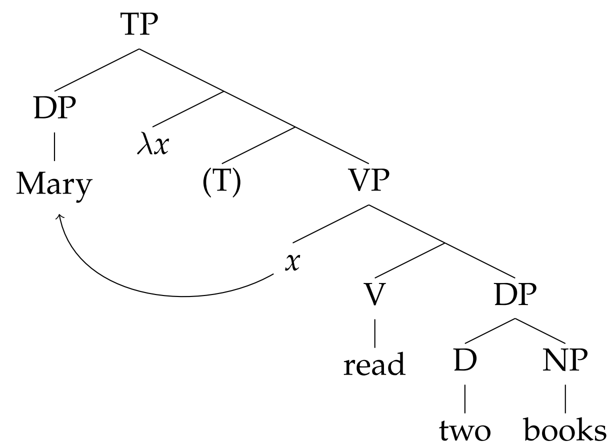
Exercise: Compute the truth conditions for (12). (Why did cheating by pretending the negation was higher work before?)

[2] You might think this is further split up into *v*P > VP. And we might slip up sometimes too.

4 Quantifiers in non-subject position

And now for something completely different... consider “Mary read two books.”

(13)

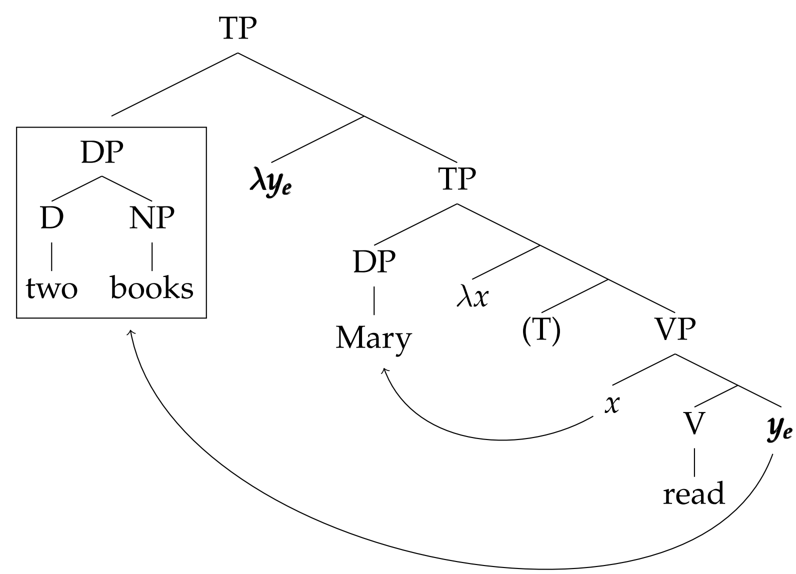


Exercise: Compute the type for each node in (13).

There is a **type mismatch**.

This can be resolved by **Quantifier Raising** (QR) the object (covert movement) to a position of type *t*, resulting in a Logical Form (LF) which differs from the pronounced word order:

(14)



Exercise: Compute the truth condition for (14). (Start by figuring out all the types.)

5 Movement and scope ambiguity

Notice that QR can target VP or TP, which are both type *t* in this system. This can account for some scope ambiguities:

- Moving to VP will make the quantifier takes scope under the subject and under adjuncts above VP;
- Moving to TP will make the quantifier take scope over the subject and adjuncts.

(15) **Making the subject quantificational:**
Every student read two books.
  a. There are two particular books that every student read.
     (ok if we can find two books that were read by everyone)
  b. Each student read two (possibly different) books.
     (ok if for every student, we can find two (possibly different) books that s/he read)

(16) **Adding negation:**
Mary did not read two books.
(Suppose there were ten books assigned.)
  a. There are two books that Mary did not read.
     (ok if Mary read eight books)
  b. The books that Mary read are not (even) two.
     (ok if Mary read less than two books)

Exercise: Take (15) or (16) and QR the object to VP or TP. Did you get the two readings of the sentence?

Next time

Here ends our review/crash-course in compositional semantics. You made it!

Next time we will discuss *only*, in particular the proposal and arguments in *Rooth (1985) pp 27–32, 88–94*.