

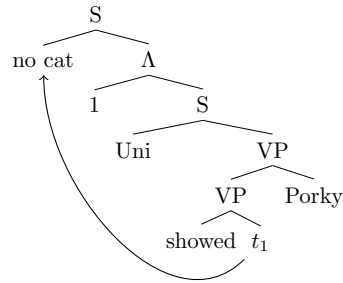
# October 14 & 17: More on LF

## 1 What LF buys us

Last class: using QR to interpret quantificational objects of transitives in situ and derive cases of scope ambiguity and pronominal binding by quantifiers.

Some other upshots of the system:

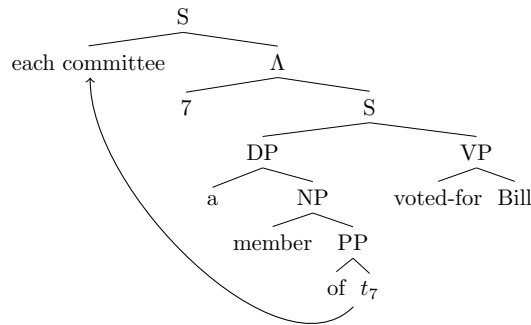
1. In situ quantification with ditransitives:



$$\llbracket \text{no cat} \rrbracket^g (\lambda x. \text{showed}'(x)(p)(u))$$

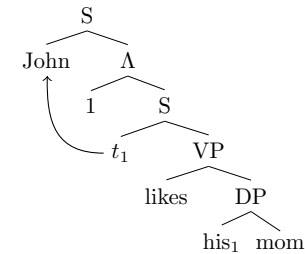
- ▷ In general, QR predicts that a quantificational DP can appear *anywhere* an expression with type  $e$  appears. The reason: the quantificational DP can leave behind a trace of type  $e$  and acquire scope over a node of the right type via covert movement.

2. So-called “inverse linking” constructions, i.e. quantification into DP:



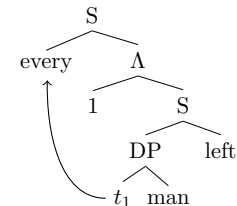
$$\llbracket \text{each committee} \rrbracket^g (\lambda x. \llbracket \text{a member of } t_7 \rrbracket^{g[x/7]} (\text{voted-for}'(b)))$$

3. Certain things are predicted able to QR, and certain things not. For one, we predict that there is a binding derivation of a simple sentence like  $John_i$  likes  $his_i$  mom.

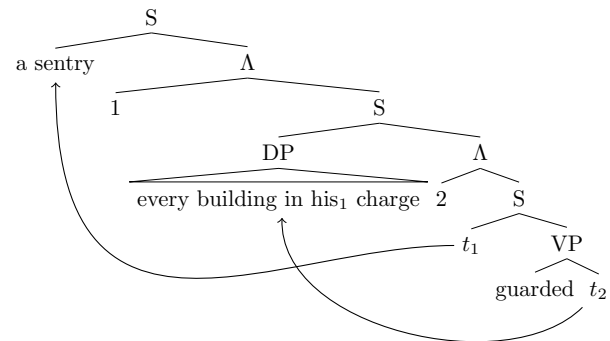


$$(\lambda x. \text{likes}'(\text{mom}'(x))(x))(j) = \text{likes}'(\text{mom}'(j))(j)$$

- ▷ As noted, this interpretation is indistinguishable from the interpretation sans binding. Difficult to imagine any evidence that could bear on the existence or not of such LFs, but it exists (and we'll see it next week).
- ▷ On the other hand, we predict that e.g. determiners can't QR. The following tree isn't interpretable as it stands. Thus, quantificational scope-taking must amount to an **entire DP** taking scope.

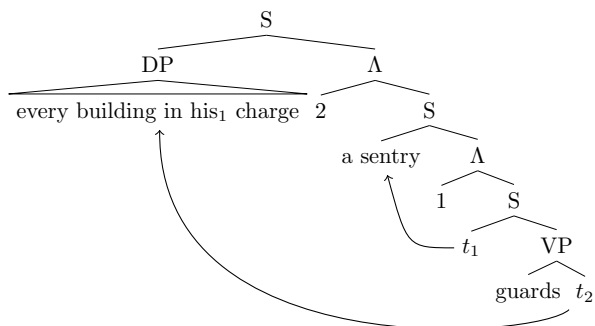


4. “Roofing”: a quantifier  $Q_1$  can only bind into a quantifier  $Q_2$  if  $Q_1$  has scope over  $Q_2$ , where  $Q_1$  has scope over  $Q_2$  iff  $Q_1$  c-commands  $Q_2$  at LF:



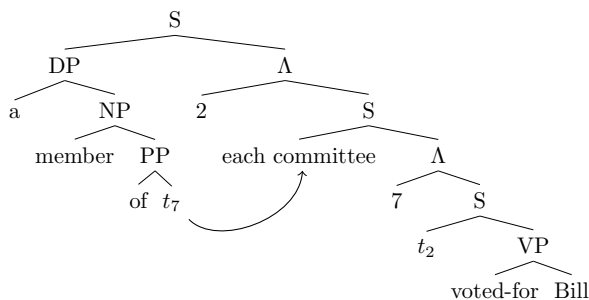
$$\llbracket \text{a sentry} \rrbracket^g (\lambda x. \llbracket \text{every building in his}_1 \text{ charge} \rrbracket^{g[x/1]} (\lambda y. \text{guarded}'(y)(x)))$$

- ▷ There's a lot happening in this tree. Useful to go over in detail to make sure you understand each piece of the derivation.
- ▷ If we try to assign the object quantifier scope over the subject, we inevitably end up unbinding the pronoun his<sub>1</sub>:



$\llbracket \text{every building in his}_1 \text{ charge} \rrbracket^g (\lambda y. \llbracket \text{a sentry} \rrbracket^g (\lambda x. \text{guards}'(y)(x)))$

- ▷ This matches the empirical situation **perfectly**. The inverse scope reading is only available when the pronoun isn't understood as bound. When it's bound, the inverse scope reading disappears.
- 5. Over-generation concern: unlike pronouns, QR should never be able to un-bind traces. That is, the following LF shouldn't be allowed:



$\llbracket \text{a member of } t_7 \rrbracket^g (\lambda x. \llbracket \text{each committee} \rrbracket^g (\lambda y. \text{voted-for}'(b)(x)))$

- ▷ The truth conditions derived in this case are bizarre. Given an assignment  $g$ , they require there to be a member of  $g(7)$  such that, for each committee  $x$ , s/he voted for Bill. Clearly, this isn't a possible reading of the sentence, and so the rule that relates S-structure with LF's will have to be carefully formulated so avoid this sort of outcome.
- 6. Does everything have to QR? That is, is the below tree . This seems to predict, counterintuitively, that the LF's that are closest to surface ones are inverse-scoping LF's:

- 7. Constraints on QR seem, at least in certain cases, to parallel constraints on overt movement. Just as you cannot wh-move out of a relative clause, it seems you cannot QR out of a relative clause. That is, (2)

- (1) \*Who did you meet a man [who likes  $t$ ]?
- (2) Most accidents everybody reported were serious. (\* $\forall > \exists$ )

- ▷ However, the situation is a bit more complicated. Quantifiers don't readily take scope out of tensed clauses, but wh-movement out of tensed clauses is impeccable:

- (3) Who did Bill say John likes  $t$ ?
- (4) Someone said John likes everybody. (\* $\forall > \exists$ )

## 2 Quantifying into XP

### 2.1 VP-internal subjects

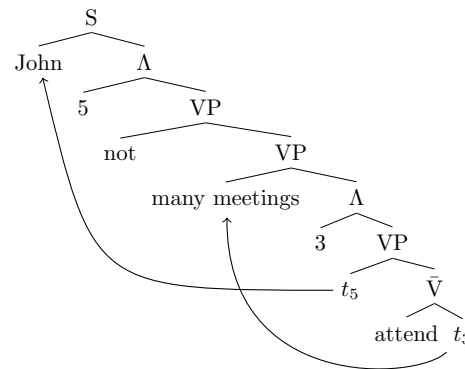
Data point: VP-level negation:

- (5) Kyle doesn't like natto.
- (6) John didn't attend many meetings.

Before, we'd just assumed negation had sentential scope. So why not have a QR-able sentential negation? For one, it would badly over-generate:

- (7) I'm certain not to teach  $\neq$  I'm not certain to teach

A standard solution: VP-internal subjects allow us to quantify into VP. In other words, there is a subject position internal to VP which moves out to subject position, and VPs are thus actually of type  $t$ .



$\llbracket \text{not} \rrbracket^g (\llbracket \text{many meetings} \rrbracket^g (\lambda x. \text{attend}'(x)(j)))$

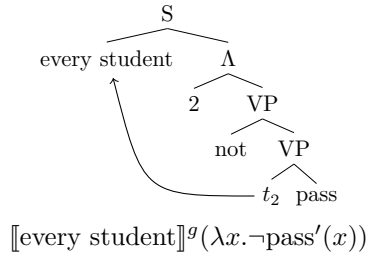
Similar data suggests we may need to posit covert subjects inside PPs, APs, and NPs:

- (8) No student from a foreign country knew the national anthem.
- (9) Everyone interested in more than one country came to the session.
- (10) No owner of an espresso machine drinks tea.

An empirical wrinkle. The following is ambiguous between a reading on which every student  $x$  was such that  $x$  didn't pass, and a reading on which it's false that every student passed:

- (11) Every student didn't pass.

Yet our semantics only derives one reading:



Here is another option: we assume that the trace can be of type  $\langle\langle e, t \rangle, t\rangle$ , and that in such cases **PA** can introduce functions *from generalized quantifiers* into values. Then a tree that looks very much like the previous tree receives the following interpretation:

$$(\lambda \mathcal{Q}. \neg \mathcal{Q}(\text{pass}')) ([[ \text{every student} ]]^g)$$

Independent motivation for generalizing **PA** and assignment functions in this way—that is, allowing higher-order traces and abstraction over things other than individuals—comes from cross-categorial topicalization:

- (12) On the porch, she isn't  $t$
- (13) Hard-working, he is  $t$ .
- (14) I said I was gonna write to him, and write to him I did  $t$ .

Related: cases of so-called A-movement, as in (15). We lack the resources to address cases like this at this point, but the eventual analysis (and choice points) would look a great deal like the case with negation.

- (15) A unicorn seems to be approaching.

### 3 Flexible types

There exist other ways to interpret object quantifiers in situ. For example, we might imagine that there is a silent morpheme that either applies to transitive verbs or quantifiers and allows them to compose directly:

$$[[\text{SAT}_\emptyset]]^g = \lambda R_{\langle e, \langle e, t \rangle \rangle} \cdot \lambda \mathcal{Q}_{\langle \langle e, t \rangle, t \rangle} \cdot \lambda x_e \cdot \mathcal{Q}(\lambda y. R(y)(x))$$

$$[[\text{SAT}_\emptyset]]^g = \lambda \mathcal{Q}_{\langle \langle e, t \rangle, t \rangle} \cdot \lambda R_{\langle e, \langle e, t \rangle \rangle} \cdot \lambda x_e \cdot \mathcal{Q}(\lambda y. R(y)(x))$$

Alternatively, it could be that everything is type  $\langle\langle e, t \rangle, t\rangle$  and transitive verbs are born with a higher type:

$$[[\text{John}]]^g = \lambda P_{\langle e, t \rangle} \cdot P(j) \quad [[\text{met}]]^g = \lambda \mathcal{Q}_{\langle \langle e, t \rangle, t \rangle} \cdot \lambda x \cdot \mathcal{Q}(\lambda y. \text{met}'(y)(x))$$

Yet another possibility is that some expressions are born as type  $e$  but shift into type- $\langle\langle e, t \rangle, t\rangle$  expressions via the following silent morpheme:

$$[[\text{LIFT}_\emptyset]]^g = \lambda x_e \cdot \lambda P_{\langle e, t \rangle} \cdot P(x)$$

And we haven't even come close to exploring the entire logical space of possibilities. For example, it might be the case that functional application is not the sole saturative mode of combination(!).

### 4 Arguments for LF?

#### 4.1 Generalizing flexible types?

These flexible types solutions all allow a transitive verb to combine with a quantificational object in situ. But they struggle to account for other phenomena that QR seems to handle with aplomb:

1. Ditransitives: it seems that, given the flexible types strategy, we'll need separate type-shifters to handle cases like *Uni showed no cat Porky*. The ones mooted above only work for combining quantificational DPs with mono-transitive verbs.
2. Scopal ambiguity: the type-shifters mooted so far only allow for *surface-scope* readings. Thus, it seems we'll need separate type-shifters that allow for the possibility of scopal ambiguity, and that these, as well, will have to allow for both transitive and ditransitive verbs.
3. Inverse linking: we only generate one reading for constructions like *a member of every committee* (do you see why?), though this does allow us to avoid positing covert subjects in "surface-linking" constructions like *nobody from a foreign country*.

4. Binding: it isn't obvious on the flexible types approach how a quantifier can bind a pronoun. Our current treatment relies on a QR'd quantifier's trace being co-indexed with the pronoun to be bound.

We could imagine adding other type-shifters/silent lexical items to fix some of these issues, but it isn't obvious (given what we've seen so far, anyway) how to go about finding a *general* solution to these issues. Ideally, we would like to find a small set of type-shifters that allow us to generate all the scopes we need and do binding.

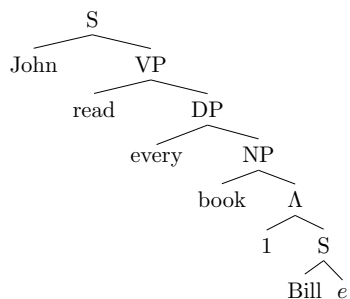
However, for now, the flexible types approach appears too flat-footed to handle the data in a satisfying way.

## 4.2 ACD

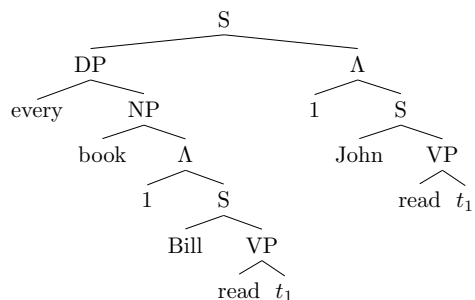
Constructions like the following are known as ACD (short for "antecedent-contained deletion")

- (16) John read everything Bill did.

Ellipsis. Usually taken to involve some sort of identity between antecedent and unpronounced VPs. But where is the antecedent:



ACD (solving the regress, de re, presupposes ACD is VPE)



An argument for this:

- (17) My mom told me to read everything my teacher did.

We haven't given an analysis of *intensional* verbs like wanted yet. But for now, we can rely on the intuitive idea that the ambiguity between the two readings of Mom told me to read every book