1 Object Quantifiers

(1) a. \[[\forall_x \text{ Every linguist}] \forall_y \text{ offended Paul}]\]
   b. \[[\forall_x \text{ Paul} \forall_y \text{ offended } \forall_x \text{ every linguist}]\]

- Type Mismatch

- What is the second argument of every in (1b) and how is it derived?

- Ambiguity

(2) Some philosopher offended every linguist.
   a. Reading 1: There is a philosopher who offended every linguist.
   b. Reading 2: Every linguist was offended by some philosopher or the other.

2 Type Raising

(3) Ordinary Quantifiers, type \(\langle n \ell \rangle t\)
   a. \[[\text{somebody} h] = \lambda f \in D_{\text{ref}} \exists x \in D.f(x) = 1\]
   b. \[[\text{everybody} h] = \lambda f \in D_{\text{ref}} \forall x \in D.f(x) = 1\]

(4) Type Raised Quantifiers, type \(\langle n \ell \rangle \ell\)
   a. \[[\text{somebody} h] = \lambda f \in D_{\text{ref}} \exists x \in D.f(y)(x) = 1\]
   b. \[[\text{everybody} h] = \lambda f \in D_{\text{ref}} \forall x \in D.f(y)(x) = 1\]

- Systematic Type-Ambiguity - Type Raising Rules

- Further Type-Raising

(5) Ann introduced everybody to Maria.

- Separating the Restrictor from the Quantifier - the denotations of every, some, a etc.
3 Repairing Type Mismatch by Movement

Before Movement

\[
S \\
\text{Paul} \rightarrow \text{VP} \\
\text{offended} \rightarrow \text{DP} \\
\text{every} \rightarrow \text{linguist}
\]

After Movement of the Quantifier Phrase (i.e. Quantifier Raising):

\[
S \\
\text{DP} \\
\text{every} \rightarrow \text{linguist} \\
\text{S} \\
\text{Paul} \rightarrow \text{VP} \\
\text{offended} \rightarrow \text{t}
\]

But this structure is not interpretable as it stands.

A modification that is interpretable:

\[
S \\
\text{DP} \\
\text{every} \rightarrow \text{linguist} \\
1 \\
\text{S} \\
\text{Paul} \rightarrow \text{VP} \\
\text{offended} \rightarrow \text{t}
\]

(6) Predicate Abstraction Rule (PA):

Let \( \alpha \) be a branching node with daughters \( \beta \) and \( \gamma \), where \( \beta \) dominates only a numerical index \( i \). Then, for any variable assignment function \( \alpha, [\beta^*] = \lambda x \in D. [\gamma^*] \).

4 Arguments for Quantifier Raising

4.1 Ambiguities: Inverse Scope

Sentences with more than one Quantifier Phrase are often ambiguous:

(7) a. Somebody offended everybody.
    b. Alan gave a present to every student.
    c. No student read two books by Tarski.
    d. Exactly two students read every book by Church.

Sometimes one of the readings entails the other - in such cases, we can talk about a weak reading and a strong reading.
4.1.1 The Inadequacy of Type Raising

- The type-raising approach does not generate multiple readings.
- The QR approach generates multiple readings.

4.1.2 ‘Inverse Linking’

(8) One apple in every basket is rotten.

4.1.3 The Role of Syntax

(9) a. Patsy gave a student every book.
   b. Some teacher thinks [that every student likes him].
   c. Some Englishman wrote the poem [that every student memorized].  
      \( \rightarrow \) perceived as unambiguous

Scrambling Languages (Japanese, Korean, Hindi etc.)

(10) Hindi

a. Subject - Object - Verb
   Mona-ne Ravi-ko dāːtːaː
   Mona-Subj Ravi-Obj scolded
b. Object - Subject - Verb
   Ravi-ko Mona-ne dāːtːaː
   Ravi-Obj Mona-Subj scolded
Mona scolded Ravi.

(11) Hindi

a. Subject - Object - Verb
   [kisi larki]-ne [har lark]-ko dāːtːaː
   some girl-Subj every boy-Obj scolded
   ‘Some girl scolded every boy.’  (only: \( \exists > \forall \))
   b. Object - Subject - Verb
   [har lark]-ko [kisi larki]-ne dāːtːaː
   every boy-Obj some girl-Subj scolded
   ‘Some girl scolded every boy.’  (both: \( \exists > \forall, \forall > \exists \))
Mona scolded Ravi.
4.2 Antecedent Contained Deletion

Handling VP-Ellipsis

(12) a. I \[_{\nu \rho \gamma \lambda} \text{read } \text{War and Peace} \] [before you did \[_{\nu \rho \gamma \lambda} \_ \_ \_ \_] \n\text{VP}_2 = \text{\textquoteright read War and Peace\textquoteright} (\equiv \text{VP}_1) \n\n\text{VP}_2 = \text{\textquoteright go to Tanglewood\textquoteright} (\equiv \text{VP}_1) \n\n\text{VP}_2 = \text{\textquoteright leave tomorrow\textquoteright} (\equiv \text{VP}_1) \n\n\text{VP}_2 = \text{\textquoteright leave tomorrow\textquoteright} (\equiv \text{VP}_1) \n\n\text{Ellipsis Resolution = copying/checking for identity} \n\n\text{Antecedent-Contained Deletion} \n\n(13) I \[_{\nu \rho \gamma \lambda} \text{read \ [every \ [_{\nu \rho \gamma \lambda} \text{book \ [_{\nu \rho \gamma \lambda} \text{that you did \ [_{\nu \rho \gamma \lambda} \_ \_ \_ \_ \_] \_ \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_] \_ \_ \_
4.4 Conditions on Binding

(18) $\alpha$ binds $\beta$ iff:

a. C-command:
\[ \alpha \text{ c-commands } \beta . \]

b. Co-indexation:
\[ \alpha \text{ is co-indexed with } \beta . \]

c. Locality:
\[ \text{There is no } \gamma \text{ which satisfies (a) and (b) that is closer to } \beta . \]

(19) Strong Crossover:
A pronoun cannot be co-indexed with a trace that it c-commands.

(20) a. *He$_{\alpha}$ likes [every student].
(\# Every student likes himself.)
b. *the student [who$_{\alpha}$ [he$_{\alpha}$ likes t$_{\alpha}$]]
(\# the student [who$_{\alpha}$ [t$_{\alpha}$ likes himself$_{\alpha}$]])

de. the student [who$_{\alpha}$ [t$_{\alpha}$ thinks [that Mary likes her$_{\alpha}$]]]
(\# the student [who$_{\alpha}$ [t$_{\alpha}$ likes herself$_{\alpha}$]])

(21) Weak Crossover:
A QP cannot bind a pronoun that it does not c-command in surface syntax.

(22) Her$_{\alpha}$, mother loves [every student].
(\# [Every student] is loved by her$_{\alpha}$ mother.)

(23) Condition B:
A pronoun cannot be bound by a c-commanding phrase that is in the same minimal finite clause.

(24) a. *John$_{\alpha}$/[Every student] likes him$_{\alpha}$.
(\# John$_{\alpha}$/[Every student] likes himself$_{\alpha}$.)
b. John$_{\alpha}$/[Every student] thinks that Mary likes him$_{\alpha}$.
c. *the student [who$_{\alpha}$ [t$_{\alpha}$ likes her$_{\alpha}$]]
(\# the student [who$_{\alpha}$ [t$_{\alpha}$ likes herself$_{\alpha}$]])
d. the student [who$_{\alpha}$ [t$_{\alpha}$ thinks [that Mary likes her$_{\alpha}$]]]