Quantifier scope

Chris Potts, Ling 230b: Advanced semantics and pragmatics, Spring 2018
April 16

1 Three related issues

• *Composition*: apparent mismatches between the quantifier’s type and its environment.

• *Scope ambiguities*: sentences with multiple quantifiers often have multiple construals based on the quantifiers’ relative scope order.

• *Lexical variation*: differences in how different quantifiers can take scope.

For each of the approaches discussed below, let’s try to determine the extent to which it can tackle these issues.

2 Interpreting multiply-quantified formulae

All the theories we discuss converge on the same meaning representations, in which scope corresponds to linear order in the logical representation and the variables indicate argument/structure associations. As a short-hand, you can also think of a quantifier as “binding” the lambda variable that is outermost in its argument, though it’s not actually a direct binding relationship, but rather just function application.

(1) Some student read every book.

a. \[\text{\texttt{some student}}(\lambda y (\text{every book} (\lambda x ((\text{read} x y)))))\]  
   \[\text{\texttt{student}} \cap \{ y : \text{\texttt{book}} \subseteq \{ x : y \text{ read} x \} \neq \emptyset \]  
   = there is at least one student with the property of reading every book

b. \[\text{\texttt{every book}}(\lambda x (\text{\texttt{some student}}(\lambda y ((\text{read} x y)))))\]  
   \[\text{\texttt{book}} \subseteq \{ x : \text{\texttt{student}} \cap \{ y : y \text{ read} x \} \neq \emptyset \]  
   = every book has the property of being read by at least one student

(2) Let \[\text{\texttt{student}} = \{ s_1, s_2 \} \] and \[\text{\texttt{book}} = \{ b_1, b_2, b_3 \} \]

Model for (1a) and (1b): \[\text{\texttt{read}} = \{ (s_1, b_1), (s_1, b_2), (s_1, b_3) \} \]  
Model only for (1b): \[\text{\texttt{read}} = \{ (s_1, b_1), (s_1, b_2), (s_2, b_3) \} \]
3 Approaches

3.1 A new construction rule

We leave the meanings the same but stipulate that there is an additional mode of semantic composition (Barwise & Cooper 1981):

\[
\lambda x \ (Q \ (\lambda y \ ((R \ y) \ x))) : \langle e, t \rangle
\]

\[
R : \langle e, \langle e, t \rangle \rangle \quad Q : \langle \langle e, t \rangle, t \rangle
\]

3.2 Quantifier raising (QR)

A phrase \(Q\) of type \(\langle \langle e, t \rangle, t \rangle\) can move to adjoin to a node \(N\) that c-commands it and has a denotation of type \(\langle e, t \rangle\), leaving behind a trace of type \(e\). We might impose other conditions, for example, that the node \(N\) involves lambda abstraction over the variable in the original position of \(Q\).

\[
\alpha : t \\
\vdots
\]

\[
R : \langle e, \langle e, t \rangle \rangle \quad Q : \langle \langle e, t \rangle, t \rangle
\]

\[
Q(\lambda x \ \alpha) : t
\]

\[
\lambda x \ \alpha : \langle e, t \rangle \\
\vdots
\]

\[
R : \langle e, \langle e, t \rangle \rangle \quad x : e
\]
3.3 Cooper storage

In Cooper Storage (Cooper 1983), quantifiers are interpreted as variables (as with QR), but they go into a store (a multiset) as pairs of the quantifier and its associated variable. At specified nodes \( \varphi \) (a subset of the type \( t \) nodes), a quantifier–variable pair \( \langle Q, \chi \rangle \) can be removed from the store and interpreted as \( Q(\lambda \chi \varphi) \).

\[(5) \quad \langle \text{everyone} (\lambda x \text{someone} (\lambda y (\text{see } x y))), [] \rangle
\]
\[
\langle \text{someone} (\lambda y (\text{see } x y)), [\langle \text{everyone}, x \rangle] \rangle
\]
\[
\langle (\text{see } x y), [\langle \text{everyone}, x \rangle, \langle \text{someone}, y \rangle] \rangle
\]
\[
\langle y, [\langle \text{someone}, y \rangle] \rangle \quad \langle (\text{see } x), [\langle \text{everyone}, x \rangle] \rangle
\]
\[
\langle \text{see}, [] \rangle \quad \langle x, [\langle \text{everyone}, x \rangle] \rangle
\]

I think the following is a complete bare-bones theory of Cooper Storage. Since stores are multisets, we use multiset union (\( \cup \)) and multiset subtraction (\( \ominus \)).

\[(6) \quad \begin{align*}
a. \quad & \langle \varphi, [] \rangle \quad \text{where } \varphi \text{ is not of type } \langle \langle e, t \rangle, t \rangle \\
b. \quad & \langle \chi, [\langle \varphi, \chi \rangle] \rangle \quad \text{where } \varphi \text{ is of type } \langle \langle e, t \rangle, t \rangle \\
c. \quad & \langle (\varphi \psi), S_i \cup S_j \rangle \quad \text{where } \varphi : \langle \sigma, \tau \rangle \text{ and } \psi : \sigma \text{ (sibling node order irrelevant)} \\
d. \quad & \langle Q (\lambda \chi \varphi), S \ominus [\langle Q, \chi \rangle] \rangle \quad \text{where } Q : \langle e, \langle e, t \rangle \rangle \text{ and } \varphi : t
\end{align*}
\]

The above leaves open whether we actually allow stores to contain multiple occurrences of the same \( \langle Q, \chi \rangle \) pair. Cooper Storage bears a striking formal resemblance to GPSG’s Slash Categories for handling long-distance syntactic dependencies (Gazdar et al. 1985), and it is arguably one way to formalize QR (as opposed to being a competitor to it).
3.4 Argument raising

Hendriks (1993) defines very general type-shifting operations that, among other things, map any two-place relation on entities into two different relations on generalized quantifiers. The rules basically build two local operations of QR directly into the meaning of the verb.

Let \( q = \langle\langle e, t\rangle, t\rangle \):

\[
R: \langle e, \langle e, t\rangle \rangle \xrightarrow{S \circ Q} \lambda P \left( \lambda Q \left( Q \left( \lambda y P \left( \lambda x ((R x) y) \right) \right) \right) \right) : \langle q, \langle q, t\rangle \rangle
\]

(7)

\[
R: \langle e, \langle e, t\rangle \rangle \xrightarrow{Q \circ S} \lambda P \left( \lambda Q \left( P \left( \lambda x Q \left( \lambda y ((R x) y) \right) \right) \right) \right) : \langle q, \langle q, t\rangle \rangle
\]

(8)

See Barker 2015:§2.4 for the general form of the theory.

3.5 Continuations

- The guiding idea behind continuations is that, in giving expressions flexible access to their own contexts, we can account for scope-taking and scope-ambiguities. And by closing off access to contexts at certain points, we can create scope islands.

- Barker 2015 includes an excellent general overview of continuation-based approaches with applications to a wide range of phenomena. There is a bit of overhead: one has to learn some categorial grammar. The pay-off is that the tower notation built on that foundation is intuitive and easier to work with than the raw continuized grammars.

- Barker 2002 uses a formalism more like the one we’ve developed in this class (and in Linguist 130a): each syntactic rule is paired with a continuized interpretation rule. It might be a good starting point. The goal of the continuations question on assignment 2 is to get you over some of the notational hurdles.

\[
\text{run} \ jesse : t \quad \text{continuize} \ jesse \ \text{via} \ lift \implies \ (lift \ jesse)(run) : t
\]

\[
\begin{align*}
\text{jesse} : e & \quad \text{run} : \langle e, t \rangle \\
\text{lift} & \quad \text{run} : \langle e, t \rangle \\
\lambda p \ ((lift \ run)(\lambda f ((lift \ jesse)(\lambda x (p (f x))))) : \langle t, t \rangle \\
\text{lift} & \quad \text{jesse} : \langle e, t \rangle \\
\lambda P ((lift \ see)(\lambda R (\text{everyone} (\lambda x (P (R x))))) : \langle \langle e, t \rangle, t \rangle \\
\text{lift} & \quad \text{see} : \langle e, \langle e, t \rangle \rangle \\
\text{everyone} & \quad \text{run} : \langle e, t \rangle
\end{align*}
\]
4 Scope ambiguities and entailment relations

Examples like (1) are often used to motivate scope ambiguities. However, the wide-scope-object reading is strictly more general (logically weaker) than the narrow-scope-object reading. Thus, one might conclude that we need generate only the more general reading, allowing the other one to emerge via pragmatic enrichment.

For results concerning entailment relations and scope interactions, see van Benthem (1989), Zimmermann (1993), Westerståhl (1996), Ben-Avi & Winter (2004), Altman et al. (2005), and Peters & Westerståhl (2006:§10). (There is still important work to be done in this area.) For present purposes, we can rely on the fact that non-monotonic quantifiers like exactly three unicorns ensure that the two scope orderings are independent assuming that the other quantifier is monotonic.

(9) A man loves exactly three unicorns.

a. \( \left( \text{a man} \right) \left( \lambda x \left( \text{(exactly three unicorn)} \left( \lambda y \left( (\text{love} \ y) \ x) \right) \right) \right) \)

b. \( \left( \text{exactly three unicorn} \right) \left( \lambda y \left( \left( \text{a man} \right) \left( \lambda x \left( (\text{love} \ y) \ x) \right) \right) \right) \)

c. \( \|\text{man}\|^M = \{a, b, c\} \)

d. \( \|\text{unicorn}\|^M = \{u_1, u_2, u_3, u_4\} \)

e. A model for the surface reading (9a) but not the inverse reading (9b):
\[ \|\text{love}\|^M = \{\langle a, u_1 \rangle, \langle a, u_2 \rangle, \langle a, u_3 \rangle, \langle b, u_4 \rangle \} \]

f. A model for the inverse reading (9b) but not the surface reading (9a):
\[ \|\text{love}\|^M = \{\langle a, u_1 \rangle, \langle b, u_2 \rangle, \langle c, u_3 \rangle \} \]

5 QR as a syntactic operation

**Hypothesis** QR involves the same movement operation that creates constituent questions, topicalizations, and other long-distance syntactic dependencies.

(10) a. A man loves every unicorn. \( \xrightarrow{\text{QR}} \) every unicorn (\( \lambda x \) a man loves \( t_x \))

b. Which unicorn does a man love \( t_x \) ?

The present section reviews this hypothesis briefly. For a detailed exploration of the syntactic theories of the 1970s, 1980s, and 1990s, as well as the semantic shortcomings of all of them, see Reinhart 1997:§1, as well as Ruys 1992, Winter 1997, and Szabolcsi 1997 and the references therein.
5.1 Parallels with syntactic islands

(11) a. A man who loves every unicorn saw John. \( \text{QR} \)
    every unicorn (\( \lambda x \) a man who loves \( t_x \) saw John)

b. *Which unicorn does a man who loves \( t_x \) saw John?

(12) a. A man will worry if John dates every woman. \( \not\text{QR} \)
    every unicorn (\( \lambda x \) a man will worry if John dates \( t_x \))

b. *Which woman will a man worry if John dates \( t_x \)?

Other syntactic islands to check out

(13) a. *Which woman did John suggest the possibility that Mary saw \( t_x \)? (complex NLP)
    b. *Which woman did John wish to see Mary and \( t_x \)? (coordination)
    c. *Which woman did John see Mary, who likes \( t_x \)? (appositive)

5.2 The parallel breaks down: universals

Universal quantifiers tend to be much more restricted than movement operations. I think the consensus is that they can scope out of infinitival clauses but not out of tensed ones (Barker 2015:§1.6), whereas Wh-movement is routine out of both environments:

(14) a. Which woman did John want to wish to see \( t_x \)?
    b. A man wanted to wish to see every woman \( \text{QR} \)
       every woman (\( \lambda x \) a man wanted to wish to see \( t_x \))

(15) a. Which woman did John believe that Mary saw \( t_x \)?
    b. A man believed that Mary saw every woman. \( \not\text{QR} \)
       every woman (\( \lambda x \) a man believed that Mary saw \( t_x \))

5.3 The parallel breaks down: existentials

Existentials are much, much less restricted than movement operations. I think the consensus is that unmodified indefinites can scope out of any environment, including most syntactic islands.

(16) a. Exactly three men will worry if John dates a unicorn.
    b. Exactly three men who saw a unicorn dated Mary.
    c. Exactly three men saw Mary and a unicorn.

Reinhart (1997) includes a thorough review of the various responses to this problem. Syntacticians sought to exempt indefinites from certain movement restrictions; Reinhart shows that this will not deliver the right semantics. There had earlier been attempts to treat wide-scope indefinites as quasi-referential expressions (Fodor & Sag 1982), but such approaches tend to struggle with instances where an indefinites takes exceptional scope but co-varies with an even higher quantifier (intermediate scope cases; Farkas 1981; Kratzer 1998; see also Schwarzschild 2002).
6 Resolution

I think the following represents the broad consensus about these issues:

• The arguments for QR as a syntactic operation no longer depend on a tight correspondence with overt movement (Reinhart 1997 was a definitive blow), but rather on other kinds of syntactic processes, especially ellipsis (and especially antecedent contained ellipsis, or ACD).

• In purely semantic arguments, QR might just be an informal descriptive device that is implicitly cashed out with type-shifters, continuations, or mechanisms like Cooper Storage.

• Reinhart’s (1997) proposal is basically that a QR-like mechanism is correct for non-existential quantifiers, but that existential ones are to be modeled as choice functions. A choice function is a function from \(\langle e, t \rangle\) to \(e\) that, for every input function \(f\), returns some entity \(d\) such that \(f(d) = T\):

\[
\|C\|^M\left(\begin{array}{c}
\bigcirc & \rightarrow T \\
\bigcirc & \rightarrow T \\
\bigtriangleup & \rightarrow F
\end{array}\right) = \bigcirc
\]

These variables are then existentially bound at the root level. This idea proved extremely influential. Here’s a sample of earlier references and recent compendiums with extensive bibliographies: Ruys 1992; Winter 1997; Kratzer 1998; Matthewson 1999; von Fintel & Matthewson 2008; Chung & Ladusaw 2004; von Heusinger 2004; Schwarz 2011.
References


