Flexible Composition in LTAG: Quantifier Scope and Inverse Linking

joint work with Aravind K. Joshi and Maribel Romero

Laura Kallmeyer
Lattice, Paris 7
laura.kallmeyer@linguist.jussieu.fr

Réunion Genl, 29.09.2003
Overview

- The data: Nested Quantifiers
- The framework:
  - LTAG semantics
  - Quantifier scope
- The solution:
  - Flexible composition
  - Quantifier set approach
- Conclusion
Nested Quantifiers (1)

\([Qu_1 [ Qu_2]]\) : both scope orderings are possible: \(Qu_1 > Qu_2\) (surface reading) and \(Qu_2 > Qu_1\) (inverse linking reading).

(1) Every president of an African country came to the meeting.
   \(Qu_1 > Qu_2:\ \forall x[\exists y[y\text{ Afr.}_\text{country} \land x\text{ president}_\text{of} y] \rightarrow x\text{ came to the meeting}]\)

(2) A representative from every African country came to the meeting.
   \(Qu_2 > Qu_1:\ \forall x[x\text{ Afr.}_\text{country} \rightarrow \exists y[y\text{ repres. from} x] \land y\text{ came to the meeting}]\)
Nested Quantifiers (2)

\( \text{Qu}_1 \ldots [\text{Qu}_2 [\text{Qu}_3]] \): the scope readings where \( \text{Qu}_1 \) intervenes between \( \text{Qu}_2 \) and \( \text{Qu}_3 \) are impossible (Hobbs & Shieber 1987; Larson 1987):
**Nested Quantifiers (2)**

$Qu_1 \ldots [Qu_2 [Qu_3]]$: the scope readings where $Qu_1$ intervenes between $Qu_2$ and $Qu_3$ are impossible (Hobbs & Shieber 1987; Larson 1987):

- Possible scope orders:
  - $Qu_1 > Qu_2 > Qu_3$
  - $Qu_1 > Qu_3 > Qu_2$
  - $Qu_2 > Qu_3 > Qu_1$
  - $Qu_3 > Qu_2 > Qu_1$
Nested Quantifiers (2)

\[ \text{Qu}_1 \ldots [\text{Qu}_2 \ [ \text{Qu}_3]] \]: the scope readings where \( \text{Qu}_1 \) intervenes between \( \text{Qu}_2 \) and \( \text{Qu}_3 \) are impossible (Hobbs & Shieber 1987; Larson 1987):

**Possible scope orders:**
- \( \text{Qu}_1 > \text{Qu}_2 > \text{Qu}_3 \)
- \( \text{Qu}_1 > \text{Qu}_3 > \text{Qu}_2 \)
- \( \text{Qu}_2 > \text{Qu}_3 > \text{Qu}_1 \)
- \( \text{Qu}_3 > \text{Qu}_2 > \text{Qu}_1 \)

**Impossible scope orders:**
- * \( \text{Qu}_2 > \text{Qu}_1 > \text{Qu}_3 \)
- * \( \text{Qu}_3 > \text{Qu}_1 > \text{Qu}_2 \)
(3) Two politicians spy on someone from every city. (Larson 1987)
Nested Quantifiers (3)

(4) Two politicians spy on someone from every city.
(Larson 1987)

* Qu₂ Qu₁ Qu₃ = * ∃₂ ∀:

\[ \exists z \left[ \text{person}'(z) \land 2x \left[ \text{politicians}'(x) \land \forall y[\text{city}'(y) \to \text{from}'(z, y)] \land \text{spy}'(x, z) \right] \right] \]

Problem: \text{spy}'(x, z)\text{ in nuclear scope of } \exists z \Rightarrow 2x \text{ also in nuclear scope of } \exists z \Rightarrow \forall y \text{ also in nuclear scope of } \exists z \Rightarrow \text{from}'(z, y) \text{ also in nuclear scope of } \exists z

Reading can therefore be excluded for logical reasons
Nested Quantifiers (3)

(5) Two politicians spy on someone from every city. (Larson 1987)

* Qu₂ Qu₁ Qu₃ = * ∃ 2 ∀:
  ∃z [ person′(z) ∧ 2x [politicians′(x) ∧
    ∀y[city′(y) → from′(z, y)] ∧ spy′(x, z)] ] ]

Problem: spy′(x, z) in nuclear scope of ∃z ⇒ 2x also in nuclear scope of ∃z ⇒ ∀y also in nuclear scope of ∃z ⇒ from′(z, y) also in nuclear scope of ∃z
Reading can therefore be excluded for logical reasons

* Qu₃ Qu₁ Qu₂ = * ∀ 2 ∃: Inverse linking
  ∀y [ city′(y) → 2x [politicians′(x) ∧
    ∃z[[person′(z) ∧ from′(z, y)] ∧ spy′(x, z)] ] ]
LTAG semantics (1)

Kallmeyer & Joshi (2003)

- elementary trees are linked to flat semantic representations
- the derivation tree shows how the semantic representations are combined

Underspecified representations:
- enrich formulas with labels $l_1, l_2, \ldots$ and holes $h_1, h_2, \ldots$ (metavariabes ranging over labels)
- scope constraints $x \geq y$ with $x$ and $y$ being labels or holes or variables
(6) John always laughs.

\[
\begin{align*}
\text{l}_1 & : laugh'(x_1), h_1 \geq l_1 \\
\text{arg: } & \langle x_1, 1 \rangle
\end{align*}
\]
LTAG semantics (3)

Result:

\[
\begin{align*}
l_1 : & \text{laugh}'(x), \ john'(x), \ l_2 : \text{always}'(h_2), \\
& h_1 \geq l_1, \ h_1 \geq l_2, \ h_2 \geq l_1 \\
\hline
\text{arg: } & -
\end{align*}
\]
LTAG semantics (3)

\[
\begin{align*}
    l_1 : & \text{laugh}'(x), \quad john'(x), \quad l_2 : always'(h_2), \\
    h_1 & \geq l_1, \quad h_1 \geq l_2, \quad h_2 \geq l_1
\end{align*}
\]

\arg: -

Disambiguation: Bijection from holes to labels such that

(a) subordination on the disambiguated representation is a partial order

(b) no label is subordinated to two labels that are siblings
LTAG semantics (3)

Result:

\[
\begin{align*}
l_1 & : \text{laugh}'(x), \ john'(x), \ l_2 : \text{always}'(h_2), \\
h_1 & \geq l_1, \ h_1 \geq l_2, \ h_2 \geq l_1 \\
\text{arg: } & -
\end{align*}
\]

Disambiguation: Bijection from holes to labels such that

(a) subordination on the disambiguated representation is a partial order

(b) no label is subordinated to two labels that are siblings

here: \( h_1 \geq l_2 > h_2 \geq l_1 \), therefore just one disambiguation:
\( h_1 \rightarrow l_2, \ h_2 \rightarrow l_1 \sim \ john'(x) \land \text{always}'(\text{laugh}'(x)) \)
Quantifier scope (1)

Idea: separating scope and predicate argument information:

(7) every dog barks
Quantifier scope (2)

\[ l_1 : \text{bark}'(x_1), h_1 \geq l_1 \]
arg: \( \langle x_1, (1) \rangle \)

\[ l_2 : \forall x(h_2, h_3) \]
arg: \( - \)

\[ l_3 : p_1(x'), h_2 \geq l_3, h_3 \geq s_1 \]
arg: \( s_1, \langle p_1, (2) \rangle \)

\[ q_1 : \text{dog}' \]
arg: \( - \)
Quantifier scope (3)

\[
\begin{array}{l}
  l_1 : \text{bark}'(x), \ l_2 : \forall x(h_2, h_3), \ l_3 : \text{dog}'(x), \\
  h_1 \geq l_1, \ h_3 \geq l_1, \ h_2 \geq l_3 \\
  \text{arg: } -
\end{array}
\]

just one disambiguation:

\[ h_1 \to l_2, \ h_2 \to l_3, \ h_3 \to l_1 \]

\[ \sim \forall x(\text{dog}'(x), \text{bark}'(x)) \]
Quantifier scope (4)

Underspecified representations for scope ambiguities:

(8) some student loves every course

\[
\begin{align*}
    l_2 & : \exists x(h_2, h_3),
    l_4 & : \forall y(h_4, h_5), \\
    l_1 & : loves'(x, y),
    l_3 & : student'(x),
    l_5 & : course'(y),
    h_2 & \geq l_3, 
    h_3 & \geq l_1, 
    h_4 & \geq l_5, 
    h_5 & \geq l_1, 
    h_1 & \geq l_1
\end{align*}
\]

arg: –

two disambiguations:

- $h_1 \rightarrow l_2, h_2 \rightarrow l_3, h_3 \rightarrow l_4, h_4 \rightarrow l_5, h_5 \rightarrow l_1$
  (wide scope of $\exists$)

- $h_1 \rightarrow l_4, h_2 \rightarrow l_3, h_3 \rightarrow l_1, h_4 \rightarrow l_5, h_5 \rightarrow l_2$
  (wide scope of $\forall$)
Flexible composition (1)

General idea: consider substitutions and adjunctions as 
*attachments* that can go in either direction. 

*Flexible composition*: attaching a tree \( t \) or a set of trees 
\( \{t_1, \ldots, t_n\} \) to an elementary tree (or tree set) \( u \)

- Allows different orders when traversing the derivation tree.
- Extends the generative capacity of TAG.

For our purpose only restricted use of flexible composition: 
standard TAG derivation trees with a bottom-up traversal. 
(This special case is weakly equivalent to TAG.)
Flexible composition (2)

Flexible composition derivation for (2) two politicians spy on someone from every city

1. tree set for from every city is built and it attaches to the tree set for someone

⇒ identification of scope parts of someone and every
Flexible composition (3)

2. the tree sets for *two politicians* and *someone from every city* attach simultaneously to *spy*:

⇒ identification of scope parts of *two* on the one hand and *someone* and *every* on the other hand
Quantifier set approach (1)

Observation: whenever an identification of scope parts takes place,

- all scope orders are possible between the quantifier groups involved in that identification, and
- no other quantifier can intervene between them.

⇒ quantifiers that are identified are ‘glued together’ such that nothing else can intervene.
Quantifier set approach (2)

Formalization with *quantifier sets*:
Quantifier set approach (2)

Formalization with *quantifier sets*:

- introduce *quantifier sets*: whenever quantifiers scope trees are identified, a new set is built containing the scope parts of these quantifiers. (Eventually, these scope parts are already sets.)
Quantifier set approach (2)

Formalization with *quantifier sets*:

- introduce *quantifier sets*: whenever quantifiers scope trees are identified, a new set is built containing the scope parts of these quantifiers. (Eventually, these scope parts are already sets.)

- additional condition on scope order for disambiguated representations:
  
  (c) if one part of a quantifier set $Q_1$ is subordinated by one part of another quantifier set $Q_2$, then all quantifiers in $Q_1$ must be subordinated by all quantifiers in $Q_2$. 
Quantifier set approach (3)

Semantic representation of (2):

\[
\{l_1 : 2x(h_1, h_2), \{l_3 : \forall y(h_3, h_4), l_6 : \exists z(h_6, h_7)\}\} \\
l_2 : \text{politicians}'(x), l_4 : \text{city}'(y), l_5 : \text{from}'(z, y), \\
l_7 : \text{person}'(z), l_8 : \text{spy}'(x, z) \\
h_1 \geq l_2, h_2 \geq l_8, h_3 \geq l_4, h_4 \geq l_5, h_5 \geq l_5, \\
h_6 \geq h_5, h_5 \geq l_7, h_6 \geq l_7, h_7 \geq l_8, h_8 \geq l_8
\]

arg: –

Inverse linking reading \(\forall 2 \exists = l_3 > l_1 > l_6\) excluded: For \(Q_1 := \{l_3 : \forall \ldots, l_6 : \exists \ldots\}\) and \(Q_2 := l_1 : 2 \ldots\), the scope order condition (c) would not be satisfied because \(l_3 > l_1\) and \(l_6 \neq l_1\).
Conclusion

Data: In Qu₁ ... [Qu₂ [ Qu₃]], the inverse linking reading where Qu₁ intervenes between the host Qu₂ and the nested Qu₃ is impossible: * Qu₃ > Qu₁ > Qu₂.

Account:

- Using scope parts for quantifiers and flexible composition, quantifier sets are constructed that group argumentally related quantifiers.
- Constraints are imposed on quantifier sets: given two quantifier sets Q₁ and Q₂, all the quantifiers in Q₁ must have the same scopal relation to all the quantifiers in Q₂.

The flexible composition approach as used here does not increase the weak generative capacity of TAG.