SYNCHRONIZING STRUCTURE AND MEANING IN TREE ADJOINING GRAMMARS

Chung-hye Han

Department of Linguistics
Simon Fraser University

TAG+12, Düsseldorf
June 29, 2016
MAIN RESEARCH GOALS
MAIN RESEARCH GOALS

‣ Syntax: structural analysis of natural language sentences
MAIN RESEARCH GOALS

- Syntax: structural analysis of natural language sentences
- Compositional semantics: mapping sentence structures to meaning
PRINCIPLE OF COMPOSITIONALITY
PRINCIPLE OF COMPOSITIONALITY

- The Fregean program
The Fregean program

The meaning of a sentence is determined by the meaning of its parts and how they are put together.
PRINCIPLE OF COMPOSITIONALITY

‣ The Fregean program

The meaning of a sentence is determined by the meaning of its parts and how they are put together.

‣ Importance of syntax in the computation of meaning
ILLUSTRATION OF COMPOSITIONALITY

John saw a woman with binoculars.
John saw a woman with binoculars.
John saw a woman with binoculars.
John saw a woman with binoculars.
John saw a woman with binoculars.
Some boy is dating every girl.
There is a boy who is dating every girl.
(some > every)
SCOPE AMBIGUITY

**Some boy is dating every girl.**

There is a boy who is dating every girl.  
*(some > every)*
Some boy is dating every girl.

There is a boy who is dating every girl.  
(some>every)

For each girl, there is a boy who is dating her.
(every>some)
Some boy is dating every girl.

There is a boy who is dating every girl.  
(some $>$ every)

For each girl, there is a boy who is dating her.  
(every $>$ some)

SCOPE AMBIGUITY
Some boy is dating every girl.
COVERT MOVEMENT OF THE QUANTIFIED PHRASE: QUANTIFIER RAISING

some $\Rightarrow$ every

every $\Rightarrow$ some
COVERT MOVEMENT OF THE QUANTIFIED PHRASE: QUANTIFIER RAISING

some > every

every > some
COVERT MOVEMENT OF THE QUANTIFIED PHRASE: QUANTIFIER RAISING

some>every  every>some
COVERT MOVEMENT OF THE QUANTIFIED PHRASE: QUANTIFIER RAISING

some > every

every > some

\[
\text{TP} \quad \text{TP} \\
\text{DP} \quad \text{DP} \\
\text{Some boy} \quad \text{every girl} \\
\text{DP} \\
\text{T'} \\
\text{T} \\
\text{is} \\
\text{V} \\
\text{dating} \\
\text{DP}
\]
COVERT MOVEMENT OF THE QUANTIFIED PHRASE: QUANTIFIER RAISING

some > every

every > some

Diagram showing the syntactic structure of sentences with quantifiers and the movement of quantifiers.
COVERT MOVEMENT OF THE QUANTIFIED PHRASE: QUANTIFIER RAISING

some $\Rightarrow$ every

$\text{DP}$
$\text{Some boy}$

$\text{TP}$
$\text{DP}$
$\text{every girl}$

$\text{VP}$
$\text{T}$
$\text{is}$
$
\text{V}
\text{dating}$

$\text{DP}$
$\text{Some boy}$

$\text{TP}$
$\text{DP}$
$\text{every girl}$

$\text{VP}$
$\text{T}$
$\text{is}$
$
\text{V}
\text{dating}$

every $\Rightarrow$ some

$\text{DP}$
$\text{some}$

$\text{TP}$
$\text{DP}$
$\text{every}$

$\text{VP}$
$\text{T}$
$\text{is}$
$
\text{V}
\text{dating}$

$\text{DP}$
$\text{some}$

$\text{TP}$
$\text{DP}$
$\text{every}$

$\text{VP}$
$\text{T}$
$\text{is}$
$
\text{V}
\text{dating}$
COVERT MOVEMENT OF THE QUANTIFIED PHRASE: QUANTIFIER RAISING

some > every

every > some

[Diagram showing the movement of quantifiers in a sentence structure]
OUTLINE

- Tree Adjoining Grammar (TAG) Syntax  
  (Joshi, Levy & Takahashi 1975)

- Compositional semantics in Synchronous TAG  
  (Shieber & Schabes 1990, Shieber 1994)

- It-clefts: resolving the tension between surface constituency and semantics  
  (Han & Hedberg 2008)

- Pied-piping in relative clauses: adjoining in semantics  
  (Han 2007)

- Bound variables: using delayed locality  
  (Storoshenko & Han 2015)

- Interleaving of scopal elements from multiple clauses: maintaining tree-locality  
  (Frank & Storoshenko 2012)

- Scope in underspecified feature unification TAG semantics: a comparison  
  (Kallmeyer & Joshi 2003, Kallmeyer & Romero)
OUTLINE

- Tree Adjoining Grammar (TAG) Syntax (Joshi, Levy & Takahashi 1975)
- Compositional semantics in Synchronous TAG (Shieber & Schabes 1990, Shieber 1994)
- *It*-clefts: resolving the tension between surface constituency and semantics (Han & Hedberg 2008)
- Pied-piping in relative clauses: adjoining in semantics (Han 2007)
- Bound variables: using delayed locality (Storoshenko & Han 2015)
- Interleaving of scopal elements from multiple clauses: maintaining tree-locality (Frank & Storoshenko 2012)
- Scope in underspecified feature unification TAG semantics: a comparison (Kallmeyer & Joshi 2003, Kallmeyer & Romero)
TAG
TAG

- A tree-rewriting formal grammar (Joshi, Levy & Takahashi 1975)
TAG

- A tree-rewriting formal grammar (Joshi, Levy & Takahashi 1975)
- Consists of a finite set of lexicalized elementary trees (Joshi & Schabes 1997, Frank 2002)
TAG

- A tree-rewriting formal grammar (Joshi, Levy & Takahashi 1975)

- Consists of a finite set of lexicalized elementary trees (Joshi & Schabes 1997, Frank 2002)

- Elementary trees are composed via two operations: substitution and adjoining.
Some boy is happily dating every girl.

- Syntactic heads and their projections are extended projections of a lexical head.
- All argument slots are included.
- Every syntactic dependency is expressed within an elementary tree.

(Frank 2002)
Some boy is happily dating every girl.

- Syntactic heads and their projections are extended projections of a lexical head.
- All argument slots are included.
- Every syntactic dependency is expressed within an elementary tree.

(Frank 2002)
ELEMENTARY TREES

Some boy is happily dating every girl.

- Syntactic heads and their projections are extended projections of a lexical head.
- All argument slots are included.
- Every syntactic dependency is expressed within an elementary tree.

(Frank 2002)
Some boy is happily dating every girl.

- Syntactic heads and their projections are extended projections of a lexical head.
- All argument slots are included.
- Every syntactic dependency is expressed within an elementary tree.

(Frank 2002)
Some boy is happily dating every girl.
Some boy is happily dating every girl.
Some boy is happily dating every girl.
Some boy is happily dating every girl.
DERIVED TREE AND DERIVATION STRUCTURE

Some boy is happily dating every girl.
Some boy is happily dating every girl.
Some boy is happily dating every girl.
OUTLINE

‣ Tree Adjoining Grammar (TAG) Syntax (Joshi, Levy & Takahashi 1975)

‣ Compositional semantics in Synchronous TAG (Shieber & Schabes 1990, Shieber 1994)

‣ *It*-clefts: resolving the tension between surface constituency and semantics (Han & Hedberg 2008)

‣ Pied-piping in relative clauses: adjoining in semantics (Han 2007)

‣ Bound variables: using delayed locality (Storoshenko & Han 2015)

‣ Interleaving of scopal elements from multiple clauses: maintaining tree-locality (Frank & Storoshenko 2012)

‣ Scope in underspecified feature unification TAG semantics: a comparison (Kallmeyer & Joshi 2003, Kallmeyer & Romero 2008)
SYNCHRONIZING SYNTAX AND SEMANTICS: STAG
SYNCHRONIZING SYNTAX AND SEMANTICS: STAG

- A pairing of TAGs, a TAG for syntax and a TAG for semantics (Shieber & Schabes 1990, Shieber 1994)
SYNCHRONIZING SYNTAX AND SEMANTICS: STAG

- A pairing of TAGs, a TAG for syntax and a TAG for semantics (Shieber & Schabes 1990, Shieber 1994)

- Each syntactic elementary tree is paired with one or more semantic trees.
SYNCHRONIZING SYNTAX AND SEMANTICS: STAG

- A pairing of TAGs, a TAG for syntax and a TAG for semantics (Shieber & Schabes 1990, Shieber 1994)

- Each syntactic elementary tree is paired with one or more semantic trees.

- As syntactic elementary trees compose, corresponding semantic elementary trees compose in parallel.
SYNCHRONIZING SYNTAX AND SEMANTICS: STAG

- A pairing of TAGs, a TAG for syntax and a TAG for semantics (Shieber & Schabes 1990, Shieber 1994)

- Each syntactic elementary tree is paired with one or more semantic trees.

- As syntactic elementary trees compose, corresponding semantic elementary trees compose in parallel.

- Syntactic derivation is isomorphic to semantic derivation.
SYNCHRONIZING SYNTAX AND SEMANTICS: STAG

- A pairing of TAGs, a TAG for syntax and a TAG for semantics (Shieber & Schabes 1990, Shieber 1994)

- Each syntactic elementary tree is paired with one or more semantic trees.

- As syntactic elementary trees compose, corresponding semantic elementary trees compose in parallel.

- Syntactic derivation is isomorphic to semantic derivation.

- Compositional semantics is defined on derivation structure, not derived tree.
Some boy is dating every girl.

(STAG)

F: formulas  T: terms
R: predicates

Each node is typed.
Predicates are unreduced \( \lambda \)-expressions.
Links synchronize the derivation.
Multi-component sets

(Shieber and Nesson 2006)
Some boy is dating every girl.

(Shieber and Nesson 2006)
Some boy is dating every girl.

(STAG)

- F: formulas
- T: terms
- R: predicates
- Each node is typed.
- Predicates are unreduced \( \lambda \)-expressions.
- Links synchronize the derivation.
- Multi-component sets

(Shieber and Nesson 2006)
SEMANTIC COMPOSITION FOR SOME > EVERY

\[
\begin{align*}
\alpha &\text{is}\_\text{dating} \\
\alpha &\text{some}\_\text{boy} \quad \alpha &\text{every}\_\text{girl} \\
&\{\alpha'\text{some}\_\text{boy, } \beta'\text{some}\_\text{boy}\} \quad \{\alpha'\text{every}\_\text{girl, } \beta'\text{every}\_\text{girl}\}
\end{align*}
\]

\[
\begin{align*}
T &\quad F \\
\text{is} &\quad \text{boy}'(x) \quad F \\
\text{some}' &\quad T \\
x &
\end{align*}
\]

\[
\begin{align*}
\alpha &\text{is}\_\text{dating} \\
\alpha &\text{some}\_\text{boy} \quad \alpha &\text{every}\_\text{girl} \\
&\{\alpha'\text{some}\_\text{boy, } \beta'\text{some}\_\text{boy}\} \quad \{\alpha'\text{every}\_\text{girl, } \beta'\text{every}\_\text{girl}\}
\end{align*}
\]

\[
\begin{align*}
T &\quad F \\
\text{is} &\quad \text{boy}'(x) \quad F \\
\text{some}' &\quad T \\
x &
\end{align*}
\]

\[
\begin{align*}
\lambda &\text{y}\lambda x.\text{is-dating}'(x, y) \\
\text{is-dating}' &\quad \text{every}' \quad \text{girl}' \quad F \\
y &
\end{align*}
\]
SEMANTIC COMPOSITION FOR **SOME > EVERY**
SEMANTIC COMPOSITION FOR $\text{SOME} \geq \text{EVERY}$

$\alpha \text{is_dating}$

$\alpha \text{some\_boy} \quad \alpha \text{every\_girl}$

$\{\alpha' \text{some\_boy, } \beta' \text{some\_boy}\}$

$\{\alpha' \text{every\_girl, } \beta' \text{every\_girl}\}$

$\alpha' \text{is_dating}$

$\text{is\_dating}$

$\lambda y. \lambda x. \text{is\_dating'}(x, y)$

$\text{some}_x \quad \text{boy'}(x)$
SEMANTIC COMPOSITION FOR **SOME > EVERY**

\[
\begin{align*}
\alpha & \text{is_dating} \\
\alpha & \text{some-boy} \quad \alpha & \text{every-girl} \\
\{\alpha' \text{some-boy}, \beta' \text{some-boy}\} & \text{every} \\
\{\alpha' \text{every-girl}, \beta' \text{every-girl}\} \\
\end{align*}
\]
SEMANTIC COMPOSITION FOR SOME > EVERY

\[
\begin{aligned}
\alpha \text{is\_dating} & \quad \alpha' \text{is\_dating} \\
\alpha \text{some\_boy} & \quad \alpha \text{every\_girl} \\
\{\alpha' \text{some\_boy}, \beta' \text{some\_boy}\} & \quad \{\alpha' \text{every\_girl}, \beta' \text{every\_girl}\}
\end{aligned}
\]
SEMANTIC COMPOSITION FOR \( \text{SOME} \supset \text{EVERY} \)

\[
\begin{align*}
\alpha \text{is_dating} & \quad \alpha' \text{is_dating} \\
\alpha \text{some}_x \text{boy}_x & \quad \{\alpha' \text{some}_x \text{boy}_x, \beta' \text{some}_x \text{boy}_x\} \\
\alpha \text{every}_y \text{girl}_y & \quad \{\alpha' \text{every}_y \text{girl}_y, \beta' \text{every}_y \text{girl}_y\}
\end{align*}
\]

\[
\begin{align*}
\text{some'}_x \text{boy'}_x & \quad \text{every'}_y \text{girl'}_y \\
\text{is-dating'}(x,y)
\end{align*}
\]
SEMANTIC COMPOSITION FOR SOME > EVERY

\[
\alpha \text{is\_dating} \\
\text{\{\alpha \text{\_some\_boy, \beta \text{\_some\_boy}\}} \\
\text{\{\alpha \text{\_every\_girl, \beta \text{\_every\_girl}\}}}
\]

\[
\text{TP} \\
\text{DP}_1 \\
\text{some boy} \\
\text{T} \\
\text{is} \\
\text{VP} \\
\text{DP}_2 \\
\text{dating every girl}
\]

\[
\text{F} \\
\text{some}'_x \text{ boy}'(x) \\
\text{F} \\
\text{every}'_y \text{ girl}'(y) \\
\text{F}_1 \text{ 2} \\
\text{is-dating}'(x, y)
\]

\[
\text{T}_1 \\
\text{x} \\
\text{V} \\
\text{DP}_2 \\
\text{dating every girl}
\]

\[
\text{T}_2 \\
\text{y} \\
\text{R} \\
\lambda y \lambda x. \text{is-dating}'(x, y)
\]
SEMANTIC COMPOSITION FOR SOME $\geq$ EVERY

$\alpha \text{is\_dating} \leftarrow \begin{cases} \alpha \text{some\_boy} & \alpha \text{every\_girl} \\ \{\alpha' \text{some\_boy}, \beta' \text{some\_boy}\} & \{\alpha' \text{every\_girl}, \beta' \text{every\_girl}\} \end{cases}$

$\alpha' \text{is\_dating} \rightarrow \text{some}'x [\text{boy}'(x)] [\text{every}'y [\text{girl}'(y)] [\text{is\_dating}'(x,y)]]$

$\text{some}'x \leftarrow \text{boy}'(x)$

$\text{every}'y \leftarrow \text{girl}'(y)$

$\text{is\_dating}'(x,y) \leftarrow \lambda y \lambda x. \text{is\_dating}'(x, y)$

Diagram:

- TP
- DP
- T
- VP
- V
- DP
- is
- dating
- every girl
There is a boy who is dating every girl.
SEMANTIC COMPOSITION FOR $\textit{EVERY} \triangleright \textit{SOME}$

\[
\begin{align*}
\text{α}_\text{is\_dating} & \quad \text{α'}_\text{is\_dating} \\
\{\text{α}_\text{some\_boy}, \text{β}_\text{some\_boy}\} & \quad \{\text{α'}_\text{every\_girl}, \text{β'}_\text{every\_girl}\}
\end{align*}
\]
SEMANTIC COMPOSITION FOR $\textit{EVERY} \gg \textit{SOME}$

$$\begin{array}{c}
\alpha \text{is\_dating} \\
\alpha \text{some\_boy} \quad \alpha \text{every\_girl} \\
\{\alpha' \text{some\_boy}, \beta' \text{some\_boy}\} \\
\{\alpha' \text{every\_girl}, \beta' \text{every\_girl}\}
\end{array}$$

$$\begin{array}{c}
\alpha' \text{is\_dating} \\
\alpha' \text{some\_boy} \quad \alpha' \text{every\_girl} \\
\{\alpha' \text{some\_boy}, \beta' \text{some\_boy}\} \\
\{\alpha' \text{every\_girl}, \beta' \text{every\_girl}\}
\end{array}$$

$$\begin{array}{c}
\text{DP} \\
\text{some boy}
\end{array}$$

$$\begin{array}{c}
\text{TP} \\
\text{DP} 1 \\
\text{T'} \\
\text{T} \\
\text{is} \\
\text{VP} \\
\text{V} \\
\text{DP} 2 \\
\text{dating} \\
\text{every girl}
\end{array}$$

$$\begin{array}{c}
\text{DP} \\
\text{some boy}
\end{array}$$

$$\begin{array}{c}
\text{TP} \\
\text{DP} 1 \\
\text{T'} \\
\text{T} \\
\text{is} \\
\text{VP} \\
\text{V} \\
\text{DP} 2 \\
\text{dating} \\
\text{every girl}
\end{array}$$

$$\begin{array}{c}
\text{F} \\
\text{some'}_x \\
\text{boy'}(x) \\
\text{F}
\end{array}$$

$$\begin{array}{c}
\text{F} \\
\text{every'}_y \\
\text{girl'}(y) \\
\text{F}
\end{array}$$

$$\begin{array}{c}
\text{F} \\
\text{1} \\
\text{2}
\end{array}$$

$$\begin{array}{c}
\text{T} \\
\text{1}
\end{array}$$

$$\begin{array}{c}
\text{R} \\
\text{2}
\end{array}$$

$$\begin{array}{c}
\lambda y \lambda x \text{. is\_dating'}(x, y)
\end{array}$$
SEMANTIC COMPOSITION FOR \textit{EVERY} > \textit{SOME}

\[
\begin{array}{c}
\alpha \text{is_dating} \\
\{\alpha \text{some\_boy}, \alpha \text{every\_girl}\} \\
\alpha' \text{is_dating} \\
\{\alpha' \text{some\_boy}, \beta' \text{some\_boy}\} \\
\{\alpha' \text{every\_girl}, \beta' \text{every\_girl}\}
\end{array}
\]

\[
\begin{array}{c}
T \\
F \\
\text{boy}'(x) \\
\text{some}'_x \\
x
\end{array}
\]

\[
\begin{array}{c}
F \\
F_1 F_2 \\
\text{every}'_y \\
girl'(y) \\
\end{array}
\]

\[
\begin{array}{c}
F_1 \\
R \\
\lambda y \lambda x . \text{is\_dating}'(x, y) \\
y
\end{array}
\]

\[
\begin{array}{c}
\text{some\_boy} \\
\text{every\_girl} \\
\end{array}
\]

\[
\begin{array}{c}
\text{is} \\
\text{VP} \\
\text{DP} \_1 \\
\text{DP} \_2
\end{array}
\]

\[
\begin{array}{c}
\text{is} \\
\text{dating} \\
\text{every\_girl} \\
\text{T} \\
\text{TP}
\end{array}
\]
SEMANTIC COMPOSITION FOR EVERY > SOME

\[ \alpha_{\text{is\_dating}} \]
\[ (\alpha_{\text{some\_boy}}, \alpha_{\text{every\_girl}}) \]
\[ \{\alpha'_{\text{some\_boy}}, \beta'_{\text{some\_boy}}\} \quad \{\alpha'_{\text{every\_girl}}, \beta'_{\text{every\_girl}}\} \]

\[ F \]
\[ \text{every}'_y \quad \text{girl}'(y) \]
\[ F \]
\[ \text{some}'_x \quad \text{boy}'(x) \]
\[ F_{12} \]
\[ \text{is\_dating} \]
\[ (x, y) \]
\[ \text{is\_dating} \{ \text{every\_girl}, \text{every\_girl} \} \]
\[ \{ \text{some\_boy}, \text{some\_boy} \} \]
\[ x \quad \lambda y \lambda x. \text{is\_dating}'(x, y) \]

\[ T' \]
\[ \text{is} \]
\[ V \]
\[ \text{dating}_y \quad \text{every\_girl} \]
\[ T \]
\[ \text{some\_boy} \]

\[ \text{DP}_1 \]

\[ \text{TP} \]

\[ \text{DP}_2 \]
SEMANTIC COMPOSITION FOR \textit{EVERY}\textgreater\textit{SOME}
SEMANTIC COMPOSITION FOR \textit{EVERY} > \textit{SOME}

\begin{align*}
\alpha \text{is\_dating} & \quad \alpha' \text{is\_dating} \\
\alpha \text{some\_boy} \quad \alpha \text{every\_girl} & \quad \alpha' \text{some\_boy}, \beta' \text{some\_boy} \quad \alpha' \text{every\_girl}, \beta' \text{every\_girl}
\end{align*}

\begin{align*}
\text{is\_dating} & \quad \text{is\_dating}' \\
\text{some\_boy} & \quad \text{every\_girl} \\
\text{is} & \quad \text{is\_dating}' \\
\text{dating\_every\_girl} & \quad \text{is\_dating'}(x, y)
\end{align*}
SEMANTIC COMPOSITION FOR EVERY > SOME

\[
\begin{align*}
\text{some boy} & \quad \text{every girl} \\
\alpha & \quad \alpha' & \{\alpha'\text{some boy, } \beta'\text{some boy}\} & \{\alpha'\text{every girl, } \beta'\text{every girl}\}
\end{align*}
\]

\[
\begin{align*}
\text{every' } y & \quad [\text{girl'(}y\text{)]} & \text{some' } x & \quad [\text{boy'(}x\text{)]} & \text{is-dating'(}x, y\text{)]} \\
\text{some' } x & \quad [\text{boy'(}x\text{)]} & \text{is-dating'(}x, y\text{)]} \\
\end{align*}
\]
For each girl, there is a boy who is dating her.
STAG AS A THEORY OF SYNTAX AND SEMANTICS INTERFACE
STAG AS A THEORY OF SYNTAX AND SEMANTICS INTERFACE

- No intermediate level between syntax and semantics where covert syntactic operations take place
STAG AS A THEORY OF SYNTAX AND SEMANTICS INTERFACE

- No intermediate level between syntax and semantics where covert syntactic operations take place
- Syntax directly interfaces with semantics.
STAG AS A THEORY OF SYNTAX AND SEMANTICS INTERFACE

- No intermediate level between syntax and semantics where covert syntactic operations take place
- Syntax directly interfaces with semantics.
- Makes compositional semantics computationally feasible and tractable
STAG AS A THEORY OF SYNTAX AND SEMANTICS INTERFACE

- No intermediate level between syntax and semantics where covert syntactic operations take place
- Syntax directly interfaces with semantics.
- Makes compositional semantics computationally feasible and tractable
- Task: to expand the empirical coverage of STAG
OUTLINE

- Tree Adjoining Grammar (TAG) Syntax (Joshi, Levy & Takahashi 1975)
- Compositional semantics in Synchronous TAG (Shieber & Schabes 1990, Shieber 1994)
- *It*-clefts: resolving the tension between surface constituency and semantics (Han & Hedberg 2008)
- Pied-piping in relative clauses: adjoining in semantics (Han 2007)
- Bound variables: using delayed locality (Storoshenko & Han 2015)
- Interleaving of scopal elements from multiple clauses: maintaining tree-locality (Frank & Storoshenko 2012)
- Scope in underspecified feature unification TAG semantics: a comparison (Kallmeyer & Joshi 2003, Kallmeyer & Romero 2008)
IT-CLEFTS: EXPLETIVE APPROACH

It was Ohno [who won]
cleft pronoun + copula + clefted constituent + cleft clause

IT-CLEFTS: EXPLETIVE APPROACH

- Cleft pronoun is an expletive.

IT-CLEFTS: EXPLETIVE APPROACH

- Cleft pronoun is an expletive.
- Cleft clause bears a direct syntactic and semantic relation to clefted constituent, e.g. via predication.

IT-CLEFTS: DISCONTINUOUS CONSTITUENT APPROACH

<table>
<thead>
<tr>
<th>It</th>
<th>was</th>
<th>Ohno</th>
<th>[who won]</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleft pronoun + copula + clefted constituent + cleft clause</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IT-CLEFTS: DISCONTINUOUS CONSTITUENT APPROACH

- Cleft pronoun is not an expletive, but has a semantic function of a definite determiner:


<table>
<thead>
<tr>
<th>It</th>
<th>was</th>
<th>Ohno</th>
<th>[who won]</th>
</tr>
</thead>
<tbody>
<tr>
<td>cleft pronoun + copula + clefted constituent + cleft clause</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Cleft pronoun is not an expletive, but has a semantic function of a definite determiner.

Cleft clause bears a direct syntactic and semantic relation to the cleft pronoun, e.g. via extraposition.

AGAINST EXPLETIVE APPROACH
 AGAINST EXPLETIVE APPROACH

- Cleft pronoun can be replaced with *this* or *that*, subject to pragmatic constraints (Hedberg 2000).

  a. This is not Iowa we're talking about.
  b. That's the French flag you see flying over there.
AGAINST EXPLETIVE APPROACH

- Cleft pronoun can be replaced with *this* or *that*, subject to pragmatic constraints (Hedberg 2000).

  a. This is not Iowa we’re talking about.
  b. That’s the French flag you see flying over there.

- Cleft clause has the internal structure of a restrictive relative clause, but does not relate to clefted constituent like a restrictive relative clause to its head.

  c. It was Ohno who/that won.
  d. It was Ohno Ahn beat.
  e. It was Ohno whose Dad cheered.
  f. * Ohno that won is an American.
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH

- *It*-clefts have existential and exhaustiveness presuppositions just as copular sentences with definite description subjects.

(Hedberg 1990, 2000, Percus 1997)
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH

- *It*-clefts have existential and exhaustiveness presuppositions just as copular sentences with definite description subjects.

(Hedberg 1990, 2000, Percus 1997)

a. The king of France is bald.
b. The king of France is not bald.
c. There is one and only one king of France.
d. It was Ohno who/that won.
e. It was not Ohno who won.
f. Someone won and only one person won.
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)

‣ *It*-clefts can have equative and predicational interpretations, as in simple copular sentences.

(Ball 1977, DeClerck 1988, Hedberg 1990, 2000)
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)

- *It*-clefts can have equative and predicational interpretations, as in simple copular sentences.

(Ball 1977, DeClerck 1988, Hedberg 1990, 2000)

a. The teacher is Sue Johnson.
b. The teacher is a woman.
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)

- \( lt \)-clefts can have equative and predicational interpretations, as in simple copular sentences.

(Ball 1977, DeClerck 1988, Hedberg 1990, 2000)

a. The teacher is Sue Johnson.
b. The teacher is a woman.
c. It was Ohno who won.
d. The one who won was Ohno.
e. THE\( z \) [won(\( z \))] [\( z \) = Ohno]
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)

- *lt*-clefts can have equative and predicational interpretations, as in simple copular sentences.

(Ball 1977, DeClerck 1988, Hedberg 1990, 2000)

a. The teacher is Sue Johnson.
b. The teacher is a woman.
c. It was Ohno who won.
d. The one who won was Ohno.
e. \text{THE}z \ [\text{won}(z)] \ [z = \text{Ohno}]
f. It was a kid who beat John.
g. The one who beat John was a kid.
h. \text{THE}z \ [\text{beat}(z, \text{John})] \ [\text{kid}(z)]
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)

- _lt_-clefts pattern with copular sentences containing definite description subjects regarding anaphor binding. (Percus 1997)
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)

- *lt*-clefts pattern with copular sentences containing definite description subjects regarding anaphor binding.  
  (Percus 1997)

  a. The one that John nominated was himself.
  b. It was himself who John nominated.

  c. * The one that John nominated was him.
  d. * It was him who John nominated.
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)

- *It*-clefts pattern with copular sentences containing definite description subjects regarding anaphor binding. (Percus 1997)
  
  a. The one that John nominated was himself.
  b. It was himself who John nominated.
  
  c. *The one that John nominated was him.
  d. *It was him who John nominated.
  
- Parallelism between copular sentences and *it*-clefts can be accounted for if the cleft pronoun and the cleft clause form a semantic unit as a definite description.
IN FAVOR OF DISCONTINUOUS CONSTITUENT APPROACH (CONT.)

- *it*-clefts pattern with copular sentences containing definite description subjects regarding anaphor binding. (Percus 1997)
  a. The one that John nominated was himself.
  b. It was himself who John nominated.
  
  c. * The one that John nominated was him.
  d. * It was him who John nominated.

- Parallelism between copular sentences and *it*-clefts can be accounted for if the cleft pronoun and the cleft clause form a semantic unit as a definite description.

- Syntactically, cleft clause is a restrictive relative clause and forms a discontinuous constituent with the cleft pronoun.
IN FAVOUR OF EXPLETIVE APPROACH

- But the clefted constituent and the cleft clause do form a surface syntactic constituent. (Delahunty 1982)
IN FAVOUR OF EXPLETIVE APPROACH

But the clefted constituent and the cleft clause do form a surface syntactic constituent.  

( Delahunty 1982)

\[ 
\begin{align*}
\text{a. I said it should have been [Bill who negotiated the new contract], and it should have been.} \\
\text{b. It must have been [Fred that kissed Mary] but [Bill that left with her].}
\end{align*}
\]
TENSION BETWEEN EXPLETIVE AND DISCONTINUOUS CONSTITUENT APPROACHES
TENSION BETWEEN EXPLETIVE AND DISCONTINUOUS CONSTITUENT APPROACHES

- Surface constituency  →  Expletive approach
TENSION BETWEEN EXPLETIVE AND DISCONTINUOUS CONSTITUENT APPROACHES

- Surface constituency → Expletive approach

Interpretive properties → Discontinuous constituent approach
TENSION BETWEEN EXPLETIVE AND DISCONTINUOUS CONSTITUENT APPROACHES

- Surface constituency → Expletive approach

- Interpretive properties → Discontinuous constituent approach

- The distinction TAG makes between derivation structure and derived tree can resolve this tension.
TENSION BETWEEN EXPLETIVE AND DISCONTINUOUS CONSTITUENT APPROACHES

- Surface constituency → Expletive approach

Interpretive properties → Discontinuous constituent approach

- The distinction TAG makes between derivation structure and derived tree can resolve this tension.

Derivation structure: syntactic dependencies between elementary objects and compositional semantics are defined
TENSION BETWEEN EXPLETIVE AND DISCONTINUOUS CONSTITUENT APPROACHES

- Surface constituency → Expletive approach

  Interpretive properties → Discontinuous constituent approach

- The distinction TAG makes between derivation structure and derived tree can resolve this tension.

  Derivation structure: syntactic dependencies between elementary objects and compositional semantics are defined

  Derived tree: surface constituencies are defined
ELEMENTARY TREES FOR CLEFT PRONOUN AND CLEFT CLAUSE
ELEMENTARY TREES FOR CLEFT PRONOUN AND CLEFT CLAUSE

- Multi-Component (MC) set, inspired by analysis of extraposition by Kroch and Joshi (1987), and Abeiile (1994).
ELEMENTARY TREES FOR CLEFT PRONOUN AND CLEFT CLAUSE

- Multi-Component (MC) set, inspired by analysis of extraposition by Kroch and Joshi (1987), and Abeiile (1994).
- Cleft clause has the internal syntax of relative clause.
EQUATIVE COPULA ELEMENTARY TREE

\[(\alpha \text{was})\]

```
( \alpha \text{was} )
   / \  \
  TP   T'
 /     \
DP0_i T
     /  \  \
    T   CopP
     /    \
   was_k Cop FP
     /  \
   t_k DP0 F'
     /  \
   t_i DP0 F DP1
     /  \
   \epsilon
```
FP is a small clause of the copula from which the two DPs being equated originate.
FP is a small clause of the copula from which the two DPs being equated originate.

Similar to Frank's (2002) copular trees
DERIVATION AND DERIVED TREE FOR AN EQUATIVE IT-CLEFT

It was Ohno who won.

\[
\alpha\text{was} \\
\alpha\text{Ohno} \quad \{\alpha\text{it}, \beta\text{who\_won}\}
\]
It was Ohno who won.
It was Ohno who won.

DERIVATION AND DERIVED TREE FOR AN EQUATIVE IT-CLEFT
ELEMENTARY TREE PAIRS: PROPER NOUN AND EQUATIVE COPULA

\[
\begin{array}{c}
(\alpha \text{Ohno}) \\
\text{DP}
\end{array}
\begin{array}{c}
(\alpha' \text{Ohno}) \\
\text{T}
\end{array}
\begin{array}{c}
\text{Ohno'}
\end{array}
\]

\[
\begin{array}{c}
(\alpha \text{was}) \\
\text{TP}
\end{array}
\begin{array}{c}
\text{DP}_0 \quad \text{T'}
\end{array}
\begin{array}{c}
\lambda y x . x = y
\end{array}
\]

\[
\begin{array}{c}
\text{was}_k \\
\text{Cop}
\end{array}
\begin{array}{c}
\text{FP}_1
\end{array}
\begin{array}{c}
\lambda y x . x = y
\end{array}
\]

\[
\begin{array}{c}
\text{DP}_0 \\
\text{F}'
\end{array}
\begin{array}{c}
\text{DP}_1
\end{array}
\begin{array}{c}
\epsilon
\end{array}
\]
ELEMENTARY TREE PAIRS: CLEFT PRONOUN AND CLEFT CLAUSE

- cleft pronoun + cleft clause = definite description = definite quantified phrase

- MC set in the semantics defines the semantics of quantification
SYNCHRONOUS DERIVATION FOR AN EQUATIVE IT-CLEFT

It was Ohno who won.

\[
\begin{array}{c}
\alpha \text{was} \\
\alpha \text{Ohno} \{\alpha \text{it}, \beta \text{who\_won}\} \\
\alpha' \text{was} \\
\alpha' \text{Ohno} \{\alpha' \text{it}, \beta' \text{who\_won}\}
\end{array}
\]

\[
\begin{array}{c}
\lambda x. \text{won}(x) \\
\lambda y \lambda x. x = y
\end{array}
\]

\[
\begin{array}{c}
\text{THE}_z \\
\lambda x. \text{won}(x) \\
\lambda y \lambda x. x = y
\end{array}
\]

\[
\begin{array}{c}
\text{THE}_z \\
\lambda x. \text{won}(x) \\
\lambda y \lambda x. x = y
\end{array}
\]

\[
\begin{array}{c}
\text{THE}_z \\
\lambda x. \text{won}(x) \\
\lambda y \lambda x. x = y
\end{array}
\]
SYNCHRONOUS DERIVATION FOR AN EQUATIVE IT-CLEFT

It was Ohno who won.

\[ \lambda x. \text{won}(x) = z \]

\[ \lambda y \lambda x. x = y\text{Ohno}' \]
It was Ohno who won.
SYNCHRONOUS DERIVATION FOR AN EQUATIVE IT-CLEFT

It was Ohno who won.

\[
\text{THE}_z \lambda x. \text{won}(x) = \gamma \Rightarrow \text{THE}_z [\text{won}(z)] [z = \text{Ohno}']
\]

\[
\begin{array}{c}
\alpha \text{was} \\
\alpha \text{Ohno} \{\alpha \text{it, } \beta \text{who}_\text{won}\}
\end{array} \quad \begin{array}{c}
\alpha' \text{was} \\
\alpha' \text{Ohno} \{\alpha' \text{it, } \beta' \text{who}_\text{won}\}
\end{array}
\]
SYNCHRONOUS DERIVED TREES FOR AN EQUATIVE IT-CLEFT

It was Ohno who won.

\[
\begin{align*}
\lambda x. \text{won}(x) & \quad \lambda y \lambda x. x = y \quad \text{Ohno'}
\end{align*}
\]
It was Ohno who won.
SYNCHRONOUS DERIVED TREES FOR AN EQUATIVE IT-CLEFT

It was Ohno who won.

\[
\lambda x. \text{won}(x) \quad z
\]

\[
\lambda y \lambda x. x = y \quad \text{Ohno}'
\]

THE\(z\) [won(z)] [\(z = \text{Ohno}'\)]
The elementary tree is anchored by a lexical predicative noun.
It was a kid who beat John.

\[
\begin{align*}
\alpha & \text{was_kid} \\
\{\alpha & \text{it, } \beta \text{who_beat}\} \\
\alpha & \text{John}
\end{align*}
\]
It was a kid who beat John.
It was a kid who beat John.
The semantic tree represents a formula with a one-place predicate and a term node.
It was a kid who beat John.

SYNCHRONOUS DERIVATION FOR A PREDICATIONAL IT-CLEFT
SYNCHRONOUS DERIVATION FOR A PREDICATIONAL IT-CLEFT

It was a kid who beat John.
SYNCHRONOUS DERIVATION FOR A PREDICATIONAL IT-CLEFT

It was a kid who beat John.

\[
\begin{array}{c}
\text{THE}z \\
F \\
R \\
\lambda y \lambda x. \text{beat}(x, y) \\
\text{John'}
\end{array} \quad \begin{array}{c}
F \\
T \\
\lambda x. \text{kid}(x) \\
z
\end{array} \quad \begin{array}{c}
F \\
R \\
\alpha \text{was_kid} \\
\{ \alpha \text{it}, \beta \text{who_beat} \} \\
\alpha \text{John}
\end{array} \quad \begin{array}{c}
F \\
T \\
\alpha' \text{was_kid} \\
\{ \alpha' \text{it}, \beta' \text{who_beat} \} \\
\alpha' \text{John}
\end{array}
\]
SYNCHRONOUS DERIVATION FOR A PREDICATIONAL IT-CLEFT

It was a kid who beat John.
SYNCHRONOUS DERIVED TRERES FOR A PREDICATIONAL IT-CLEFT

It was a kid who beat John.
SYNONYMOUS DERIVED TRERES FOR A PREDICATIONAL IT-CLEFT

It was a kid who beat John.
SYNCHRONOUS DERIVED TRERES FOR A PREDICATIONAL IT-CLEFT

It was a kid who beat John.

\[
\text{THE}_z \left[ \text{beat}(z, \text{John}) \right] \left[ \text{kid}(z) \right]
\]
SYNCHRONOUS DERIVED TRERES FOR A PREDICATIONAL IT-CLEFT

It was a kid who beat John.

Tension resolved!
OUTLINE

‣ Tree Adjoining Grammar (TAG) Syntax (Joshi, Levy & Takahashi 1975)

‣ Compositional semantics in Synchronous TAG (Shieber & Schabes 1990, Shieber 1994)

‣ *it*-clefts: resolving the tension between surface constituency and semantics (Han & Hedberg 2008)

‣ Pied-piping in relative clauses: adjoining in semantics (Han 2007)

‣ Bound variables: using delayed locality (Storoshenko & Han 2015)

‣ Interleaving of scopal elements from multiple clauses: maintaining tree-locality (Frank & Storoshenko 2012)

‣ Scope in underspecified feature unification TAG semantics: a comparison (Kallmeyer & Joshi 2003, Kallmeyer & Romero 2008)
PIED-PIPING IN RELATIVE CLAUSE

a. a boy [[ whose brother ]i Mary hit ti ]
b. a boy [[ whose brother's friend ]i Mary hit ti ]

- Predicate-argument structure within the relative clause
- Relation between the head noun and the wh relative pronoun
PHRASE STRUCTURE BASED APPROACH
PHRASE STRUCTURE BASED APPROACH

- Heim and Kratzer approach (1998): *wh* relative pronoun contributes a $\lambda$-operator that binds the gap in relative clause, turning it into a predicate type $<e,t>$.

```
a boy [ whoi [ Mary hit ti ]] 
\lambda x. hit(Mary,x) 
\lambda x. boy(x) \land hit(Mary,x)
```
PHRASE STRUCTURE BASED APPROACH

- Heim and Kratzer approach (1998): *wh* relative pronoun contributes a \( \lambda \)-operator that binds the gap in relative clause, turning it into a predicate type \(<e,t>\).

\[
\begin{align*}
\text{a boy [ who } & \text{ [ Mary hit t } i \text{ ]] } \\
\lambda x. & \text{ hit(Mary,} x) \\
\lambda x. & \text{ boy(} x \text{) } \land \text{ hit(Mary,} x)
\end{align*}
\]

- Pied-piped material must go back to the position of the gap in the covert component.

\[
\begin{align*}
\text{a boy [ who } & \text{ [ Mary hit [ t } i \text{ se brother ]]} \\
\lambda x. & \text{ hit(Mary, the-brother-of}(x)) \\
\lambda x. & \text{ boy(} x \text{) } \land \text{ hit(Mary, the-brother-of}(x))
\end{align*}
\]
STAG ANALYSIS OF PIED-PIPING

In syntax, pied-piped material adjoins to the wh relative pronoun. (Kallmeyer & Scheffler 2004)

In semantics, the wh pronoun provides a generalized quantifier (GQ).

The meaning of pied-piped material adjoins to the GQ.

a. a boy [ [ whose brother ]i Mary hit ti ]
b. a boy [ [ whose brother’s friend ]i Mary hit ti ]
SYNTAX OF RELATIVE CLAUSE

\[(\beta \text{hit})\]

```
NP  
|   CP  
|   |   |   C'  
|   |   |   |   TP  
|   |   |   |   |   DP_\text{i}  
|   |   |   |   |   |   T'  
|   |   |   |   |   |   |   T  
|   |   |   |   |   |   |   |   DP  
|   |   |   |   |   |   |   |   |   V'  
|   |   |   |   |   |   |   |   |   |   t_i  
    |   |   |   |   |   |   |   |   |   |   |   hit  
    |   |   |   |   |   |   |   |   |   |   |   |   t_j  
```
SYNTAX OF RELATIVE CLAUSE

The relative clause elementary tree is an extended projection of a lexical predicate.
SYNTAX OF RELATIVE CLAUSE

The relative clause elementary tree is an extended projection of a lexical predicate.

Will adjoin to the head noun
SYNTAX OF PIED-PIPING

\[
(\beta's\_\text{brother}) \quad \text{DP} \quad \text{D'} \quad \text{DP} \quad \text{D} \quad \text{NP} \quad 's \quad \text{N} \quad \text{brother}
\]
SYNTAX OF PIED-PIPING

Genitive ’s heads a DP structure, according to the DP Hypothesis.

\[
(\beta’\text{'}s\_\text{brother}) \quad \begin{array}{c}
\text{DP} \\
\text{DP} \quad \text{D'} \\
\text{D} \quad \text{NP} \\
\text{’s} \\
\text{brother}
\end{array}
\]
SYNTAX OF PIED-PIPING

- Genitive ’s heads a DP structure, according to the DP Hypothesis.
- A well-formed elementary tree, as a DP is an extended projection of a noun.

(θ’s_brother)  
\[
\begin{array}{c}
\text{DP} \\
\text{DP} & \text{D'} \\
\text{D} & \text{NP} \\
\text{’s} & \text{N} \\
\text{brother}
\end{array}
\]
DERIVATION OF A RELATIVE CLAUSE WITH PIED-PIPING

a boy [ [whose brother]i Mary hit ti ]
DERIVATION OF A RELATIVE CLAUSE WITH PIED-PIPING

a boy [ [whose brother]i Mary hit t_i ]
DERIVATION OF A RELATIVE CLAUSE WITH PIED-PIPEING

a boy [ [whose brother]i Mary hit t_i ]
DERIVATION OF A RELATIVE CLAUSE WITH PIED-PIPING

a boy [ [whose brother]_i Mary hit t_i ]
The relative clause is defined as a predicate, to be adjoined to another predicate.
The relative clause is defined as a predicate, to be adjoined to another predicate. In syntax, the relative pronoun DP substitutes into the relative clause elementary tree.
The relative clause is defined as a predicate, to be adjoined to another predicate.

In syntax, the relative pronoun DP substitutes into the relative clause elementary tree.

In semantics, the logical form of relative pronoun adjoins onto the relative clause.
SEMANTICS OF RELATIVE PRONOUN

\[
(\alpha \text{who}) \quad \text{DP} \quad (\beta' \text{who}) \quad \text{R} \\
\downarrow \quad \downarrow \\
\text{D} \quad \lambda x \\
\downarrow \quad \text{F} \\
\text{who} \quad \text{GQ} \quad \text{R} \\
\downarrow \quad \downarrow \\
\lambda P.P(x)
\]
The relative pronoun introduces a generalized quantifier (GQ).
The relative pronoun introduces a generalized quantifier (GQ).

\( \lambda x \) binds \( x \) in the GQ.
The relative pronoun introduces a generalized quantifier (GQ).

- $\lambda x$ binds $x$ in the GQ.
- Adjoining $(\beta'\text{who})$ onto $(\beta'\text{hit})$ has the effect of abstracting over the variable coming from the relative pronoun, turning the relative clause into a one-place predicate.
The relative pronoun introduces a generalized quantifier (GQ).

\( \lambda x \) binds \( x \) in the GQ.

Adjoining \((\beta'\text{who})\) onto \((\beta'\text{hit})\) has the effect of abstracting over the variable coming from the relative pronoun, turning the relative clause into a one-place predicate.

Ensures that relative clause and head noun are predicating over the same variable.
SEMANTICS OF PIED-PIPING

\[
\begin{align*}
\lambda Q & \cdot \lambda y \cdot \text{Rel}(y, z) \\
\text{THE}_z & \cdot \text{brother}(z) \\
\lambda y & \cdot \text{GQ} \\
\lambda Q & \cdot \text{GQ} \\
\text{F} & \cdot \text{F} \\
\text{F} & \cdot \text{F} \\
\text{Q}(z) & \cdot \\
\end{align*}
\]
SEMANTICS OF PIED-PIPING

- (β’ ’s brother) adjoins onto the GQ node in (β’who).
SEMANTICS OF PIED-PIPING

- $(\beta^\prime$’s brother) adjoins onto the GQ node in $(\beta^\prime$who).
- $\lambda y$ ensures that the variable coming from who is in some relation with the variable coming from the head of the pied-piped DP (whose brother).
SEMANTICS OF PIED-PIPING

- $(\beta' \text{'s brother})$ adjoins onto the GQ node in $(\beta' \text{who})$.
- $\lambda y$ ensures that the variable coming from $\text{who}$ is in some relation with the variable coming from the head of the pied-piped DP $(\text{whose brother})$.

- $\lambda Q$ turns $\text{whose brother}$ into a GQ. Ensures that the variable coming from the head of the pied-piped DP $(\text{whose brother})$ is the argument of the predicate of the relative clause (hit).
PUTTING THE SEMANTICS TOGETHER

a boy [ [whose brother]i Mary hit ti ]

\[
\begin{aligned}
&\text{GQ} \\
&\lambda Q \\
&\text{F} \\
&\text{GQ} \\
&\lambda y \\
&\text{F} \\
&\text{R} \\
&\text{GQ} \\
&\lambda x \\
&\text{F} \\
&\text{R} \\
&\lambda \text{P}.\text{P}(x) \\
&\lambda \text{x} \lambda \text{y}. \text{hit}(x, y) \\
&\text{R} \\
&\text{R[1]} \\
&\text{T[2]} \\
&\text{Mary'}
\end{aligned}
\]
a boy [ [whose brother]i Mary hit t_i ]
PUTTING THE SEMANTICS TOGETHER

a boy [ [whose brother]i Mary hit ti ]
PUTTING THE SEMANTICS TOGETHER

a boy [ [whose brother]i Mary hit t_i ]
PUTTING THE SEMANTICS TOGETHER (CONT.)

a boy [ [whose brother]i Mary hit t_i ]
PUTTING THE SEMANTICS TOGETHER (CONT.)

a boy [ [whose brother]i Mary hit ti ]

\[
\lambda x \lambda y. \text{hit}(x, y) \quad \text{Mary}'
\]

\[
\lambda x. \lambda y. \text{hit}(x, y)
\]

\[
\lambda P. P(x)
\]

\[
\lambda y
\]

\[
\text{THEz } [\text{brother}(z) \land \text{Rel}(x, z)] [Q(z)]
\]

\[
\text{GQ}
\]

\[
\lambda Q
\]

\[
\text{THEz}
\]

\[
\text{brother}(z) \quad \text{Rel}(y, z)
\]
PUTTING THE SEMANTICS TOGETHER (CONT.)

a boy [ [whose brother]_i Mary hit t_i ]

\[
\lambda x . \text{REL}(x, z) \land \text{brother}(z) \land \text{hit}(\text{Mary'}, z)
\]

\[
\lambda Q . \text{REL}(x, z) \land \text{brother}(z) \land Q(z)
\]
PUTTING THE SEMANTICS TOGETHER (CONT.)

a boy [ [whose brother]i Mary hit t; ]

\[ \lambda x. \text{THE}_z [\text{brother}(z) \land \text{Rel}(x, z)] [\text{hit}(\text{Mary}', z)] \]
OUTLINE

- Tree Adjoining Grammar (TAG) Syntax
  (Joshi, Levy & Takahashi 1975)

- Compositional semantics in Synchronous TAG
  (Shieber & Schabes 1990, Shieber 1994)

- *It*-clefts: resolving the tension between surface constituency and semantics
  (Han & Hedberg 2008)

- Pied-piping in relative clauses: adjoining in semantics
  (Han 2007)

- Bound variables: using delayed locality
  (Storoshenko & Han 2015)

- Interleaving of scopal elements from multiple clauses: maintaining tree-locality
  (Frank & Storoshenko 2012)

- Scope in underspecified feature unification TAG semantics: a comparison
  (Kallmeyer & Joshi 2003, Kallmeyer & Romero 2008)
The dependency between the binder (quantifier) and the bound variable (pronoun) is non-local.
PHRASE-STRUCTURE BASED APPROACH

- Co-indexation between the binder and the bound variable
- Covert quantifier raising

Every girl \( i \) believes that she \( i \) is intelligent.

\[
\text{every girl } \lambda \ [ t_i \text{ believes that she } i \text{ is intelligent } ]
\]

\[
\forall x \ [ \text{girl}(x) \ [ \text{believes}(x, \text{intelligent}(x)) ]]
\]
An MC set can compose with different elementary trees as long as the members eventually compose into the same elementary tree.

In the derivation structure, all the members of an MC set must be dominated, but need not be immediately dominated by a single node.

Shown to be weakly equivalent to standard TAG by Chiang & Scheffler (2008)
SYNTAX OF BOUND VARIABLE
PRONOUN

\[
\{(\alpha \text{she}) \text{ DP}[3\text{sgF}] \mid (\beta \text{she}) \text{ DP}^*[3\text{sgF}]\}
\]

\[
\{\mid \text{D} \mid \text{she}\}
\]
(αshe) DP[3sgF]  (βshe) DP*[3sgF]  

D  

she

• (αshe) substitutes into a substitution site in a predicate elementary tree.
SYNTAX OF BOUND VARIABLE PRONOUN

- (αshe) substitutes into a substitution site in a predicate elementary tree.
- (βshe) adjoins to the binder DP, ensuring agreement. (Ryant & Scheffler 2006, Kallmeyer & Romero 2007)
SYNTAX OF BOUND VARIABLE PRONOUN

- ($\alpha$ she) substitutes into a substitution site in a predicate elementary tree.
- ($\beta$ she) adjoins to the binder DP, ensuring agreement. (Ryant & Scheffler 2006, Kallmeyer & Romero 2007)
- Delayed Tree-Local derivation

\begin{align*}
(\alpha \text{she}) & \quad \text{DP}[^{3}\text{sg}F] \\
\quad \quad D \\
\quad \quad \text{she} \\
(\beta \text{she}) & \quad \text{DP}[^{3}\text{sg}F] \\
\quad \quad D \\
\quad \quad \text{she}
\end{align*}
Every girl$_i$ believes [that she$_i$ is intelligent]
Every girl\textsubscript{i} believes [that she\textsubscript{i} is intelligent]
Every girl$_i$ believes [that she$_i$ is intelligent]
Every girl$_i$ believes [that she$_i$ is intelligent]
Every girl\textsubscript{i} believes [that she\textsubscript{i} is intelligent]
Semantics of quantified NP recast as a GQ

- Node labels according to their semantic types
- \(\beta\) (she) adjoins to 3 in \((\alpha\text{every}_\text{girl})\).
- A corresponding semantic tree adjoins to 3 in \((\beta'\text{every}_\text{girl})\).
SEMANTICS OF BOUND VARIABLE PRONOUN

- Delayed Tree-Local Derivation:
  
  - (α’she) substitutes into a predicate tree as an argument.
  
  - (β’she) adjoins to the binder tree.

- With the function in (β’she), $x_s$ (bound variable) and $x_g$ (binder variable) are $\lambda$-converted with $z$, bound under a single $\lambda z$ operator.
Every girl\(_i\) believes [that she\(_i\) is intelligent]
Every girl \(i\) believes [that she \(i\) is intelligent]
Every girl \( i \) believes [that she \( i \) is intelligent]
Every girl\(_i\) believes [that she\(_i\) is intelligent]
Every girl$_i$ believes [that she$_i$ is intelligent]
Every girl$_i$ believes [that she$_i$ is intelligent]
Every girl$_i$ believes [that she$_i$ is intelligent]
Every girl\(_i\) believes [that she\(_i\) is intelligent]
Every girl believes [that she is intelligent]
OUTLINE

- Tree Adjoining Grammar (TAG) Syntax (Joshi, Levy & Takahashi 1975)
- Compositional semantics in Synchronous TAG (Shieber & Schabes 1990, Shieber 1994)
- *It*-clefts: resolving the tension between surface constituency and semantics (Han & Hedberg 2008)
- Pied-piping in relative clauses: adjoining in semantics (Han 2007)
- Bound variables: using delayed locality (Storoshenko & Han 2015)
- Interleaving of scopal elements from multiple clauses: maintaining tree-locality (Frank & Storoshenko 2012)
- Scope in underspecified feature unification TAG semantics: a comparison (Kallmeyer & Joshi 2003, Kallmeyer & Romero 2008)
SCOPES AGAIN: INTERLEAVING SCOPAL ELEMENTS

Some professor remembered [PRO to review every paper].

- Intended scope: some > every > remembered
- This cannot be derived under Tree-Local, or even Set-Local MC-TAG.

(Frank & Storoshenko 2012)

(Nesson & Shieber 2008)
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
SCOPE AGAIN: INTERLEAVING SCOPAL ELEMENTS

Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].

\[\exists x \text{ professor}(x) \quad e \quad \langle e, t \rangle \quad \lambda p \lambda x. \text{remembered}(x, p)\]

\[\forall y \text{ paper}(y) \quad e \downarrow \quad \langle e, t \rangle \quad \lambda z \lambda x. \text{review}(x, z) \quad y\]
SCOPE AGAIN: INTERLEAVING SCOPAL ELEMENTS

Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
SCOPE AGAIN: INTERLEAVING SCOPAL ELEMENTS

Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].

- Some > every > remember cannot be generated!
ALTERNATIVE PERSPECTIVE ON PREDICATE SEMANTIC TREES

- Semantic elementary trees for predicates as MC sets
  - Variable part: $(\alpha'\text{praised})$
  - Scope part: $(\beta'\text{praised})$

(Frank & Storoshenko 2012)
ALTERNATIVE PERSPECTIVE ON PREDICATE SEMANTIC TREES

Kim Praised Sue.

(Frank & Storoshenko 2012)
ALTERNATIVE PERSPECTIVE ON PREDICATE SEMANTIC TREES

Kim Praised Sue.

(Frank & Storoshenko 2012)
Kim Praised Sue.

(Frank & Storoshenko 2012)
ALTERNATIVE PERSPECTIVE ON PREDICATE SEMANTIC TREES

Kim Praised Sue.

(Frank & Storoshenko 2012)
ALTERNATIVE PERSPECTIVE ON PREDICATE SEMANTIC TREES

```
(λx(λy.praised(y, x))).praised(Kim, Sue)
```

Kim Praised Sue.
Some professor remembered [PRO to review every paper].

(Frank & Storoshenko 2012)
Some professor remembered [PRO to review every paper].

(Frank & Storoshenko 2012)
Some professor remembered [PRO to review every paper].

```
TP
  /\  
DP   T'  
    /\    
some professors T    VP
      /\     
     V   TP*
  remembered
     /\    
    DP   PRO

TP
  /\  
DP   T'  
    /\    
to    VP
     /\     
   V    DP
  review  every paper
```
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
DERIVATION OF INTERLEAVING SCOPE

Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].

\[
\exists x \text{ professor}(x) \\
\forall y \text{ paper}(y) \\
\lambda p \lambda x. \text{remembered}(x, p) \\
\lambda x \lambda y. \text{review}(y, x)
\]
Some professor remembered [PRO to review every paper].

\[
\lambda p \lambda x. \text{remembered}(x, p)
\]

\[
\lambda x \lambda y. \text{review}(y, x)
\]

\[
\exists x \ \text{professor}(x) \rightarrow \forall y \ \text{paper}(y) \rightarrow \text{review}(y, x) \quad \text{PRO to review every paper}
\]
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered \([\text{PRO} \text{ to review every paper}]\).
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
Some professor remembered [PRO to review every paper].
DERIVATION OF INTERLEAVING SCOPE (CONT.)

Some professor remembered [PRO to review every paper].

$$\exists x \ [\text{professor}(x)] \ [\forall y \ [\text{paper}(y)] \ [\text{remembered}(x, \text{review}(x, y))]]$$

$$\forall y \ [\text{paper}(y)] \ [\text{remembered}(w, \text{review}(w, y))]$$

$$\exists x \ [\text{professor}(x)] \ [\forall y \ [\text{paper}(y)] \ [\text{remembered}(x, \text{review}(x, y))]]$$

$$\forall y \ [\text{paper}(y)] \ [\text{remembered}(w, \text{review}(w, y))]$$

$$\exists x \ [\text{professor}(x)] \ [\forall y \ [\text{paper}(y)] \ [\text{remembered}(w, \text{review}(w, y))]]$$
OUTLINE

- Tree Adjoining Grammar (TAG) Syntax (Joshi, Levy & Takahashi 1975)
- Compositional semantics in Synchronous TAG (Shieber & Schabes 1990, Shieber 1994)
- *It*-clefts: resolving the tension between surface constituency and semantics (Han & Hedberg 2008)
- Pied-piping in relative clauses: adjoining in semantics (Han 2007)
- Bound variables: using delayed locality (Storoshenko & Han 2015)
- Interleaving of scopal elements from multiple clauses: maintaining tree-locality (Frank & Storoshenko 2012)
- Scope in underspecified feature unification TAG semantics: a comparison (Kallmeyer & Joshi 2003, Kallmeyer & Romero 2008)
LTAG SEMANTICS WITH SEMANTIC UNIFICATION

- Each elementary tree is linked to a pair of a semantic representation and a semantic feature structure description.
LTAG SEMANTICS WITH SEMANTIC UNIFICATION

- Each elementary tree is linked to a pair of a semantic representation and a semantic feature structure description.

- Flat semantic representations: typed labelled formulas, scope constraints
Each elementary tree is linked to a pair of a semantic representation and a semantic feature structure description.

Flat semantic representations: typed labelled formulas, scope constraints

Semantic feature structures contain features for all node positions in elementary trees: S, NP, VP, T (top), B (bottom), I (individual variables), P (propositional labels)
Each elementary tree is linked to a pair of a semantic representation and a semantic feature structure description.

Flat semantic representations: typed labelled formulas, scope constraints

Semantic feature structures contain features for all node positions in elementary trees: S, NP, VP, T (top), B (bottom), I (individual variables), P (propositional labels)

Global features are linked to the elementary tree as a whole, rather than to single node positions.
LTAG SEMANTICS WITH SEMANTIC UNIFICATION

- Each elementary tree is linked to a pair of a semantic representation and a semantic feature structure description.

- Flat semantic representations: typed labelled formulas, scope constraints

- Semantic feature structures contain features for all node positions in elementary trees: S, NP, VP, T (top), B (bottom), I (individual variables), P (propositional labels)

- Global features are linked to the elementary tree as a whole, rather than to single node positions.

- Semantic composition: conjoining feature structure descriptions, and adding feature value equations (feature unification)
AN EXAMPLE OF SEMANTIC UNIFICATION

Everybody laughs.

(Kallmeyer & Romero 2008)
Everybody laughs. (Kallmeyer & Romero 2008)
Everybody laughs.  

- Underspecified semantic representation

\[ l_1 : \text{laugh}(x), \ l_2 : \text{every}(x, 4, 5), \ l_3 : \text{person}(x) \]

\[ 2 \geq l_1, \ 4 \geq l_3, \ 2 \geq 5, \ 5 \geq l_1 \]
Everybody laughs.  

(Kallmeyer & Romero 2008)

- Underspecified semantic representation

\[
\begin{align*}
l_1 &: \text{laugh}(x), \\
l_2 &: \text{every}(x, \text{[4]}, \text{[5]}), \\
l_3 &: \text{person}(x)
\end{align*}
\]

\[
\begin{align*}
2 &\geq l_1, \\
4 &\geq l_3, \\
2 &\geq 5, \\
5 &\geq l_1
\end{align*}
\]

- Disambiguation: assignments for the remaining meta-variables

\[
\begin{align*}
2 &\rightarrow l_2, \\
4 &\rightarrow l_3, \\
5 &\rightarrow l_1
\end{align*}
\]
Everybody laughs. (Kallmeyer & Romero 2008)

- Underspecified semantic representation

\[ l_1 : \text{laugh}(x) \], \[ l_2 : \text{every}(x, 4, 5) \], \[ l_3 : \text{person}(x) \]

\[ 2 \geq l_1, 4 \geq l_3, 2 \geq 5, 5 \geq l_1 \]

- Disambiguation: assignments for the remaining meta-variables

\[ 2 \rightarrow l_2, 4 \rightarrow l_3, 5 \rightarrow l_1 \]

\[ \text{every}(x, \text{person}(x), \text{laugh}(x)) \]
SCOPE AMBIGUITY IN SEMANTIC UNIFICATION

Someone likes everybody.

\(l_1: \text{likes}(1, 12)\)
\(2 \geq 3\)

\(l_2: \text{some}(x, 4, 5)\)
\(4 \geq l_3\)
\(6 \geq 5, 5 \geq 7\)

\(l_3: \text{person}(x)\)

\(l_4: \text{every}(y, 8, 9)\)
\(8 \geq l_5\)
\(10 \geq 9, 9 \geq 11\)

\(l_5: \text{person}(y)\)
Someone likes everybody.

(Kallmeyer & Romero 2008)
Someone likes everybody.

\[ l_1: \text{likes}(1, 12) \]
\[ 2 \geq 3 \]

\[ l_2: \text{some}(x, 4, 5) \]
\[ 4 \geq l_3 \]
\[ 6 \geq 5, 5 \geq 7 \]

\[ l_3: \text{person}(x) \]

\[ l_4: \text{every}(y, 8, 9) \]
\[ 8 \geq l_5 \]
\[ 10 \geq 9, 9 \geq 11 \]

\[ l_5: \text{person}(y) \]
Someone likes everybody.
SCOPE AMBIGUITY IN SEMANTIC UNIFICATION

Someone likes everybody.

(Kallmeyer & Romero 2008)
SCOPE AMBIGUITY IN SEMANTIC UNIFICATION

Someone likes everybody.

(Kallmeyer & Romero 2008)
Someone likes everybody.

(Kallmeyer & Romero 2008)

- Underspecified semantic representation

\[
\begin{align*}
l_1 & : \text{likes}(x, y), \quad l_2 : \text{some}(x, 4, 5), \quad l_3 : \text{person}(x) \\
l_4 & : \text{every}(y, 8, 9), \quad l_5 : \text{person}(y) \\
2 & \geq l_1, \quad 4 \geq l_3, \quad 2 \geq 5, \quad 5 \geq l_1, \quad 8 \geq l_5, \quad 2 \geq 9, \quad 9 \geq l_1
\end{align*}
\]
SCOPE AMBIGUITY IN SEMANTIC UNIFICATION

Someone likes everybody.

(Kallmeyer & Romero 2008)

- Underspecified semantic representation

\[
\begin{align*}
  l_1 &: \text{likes}(x, y), \\
  l_2 &: \text{some}(x, \overline{4}, \overline{5}), \\
  l_3 &: \text{person}(x) \\
  l_4 &: \text{every}(y, \overline{8}, \overline{9}), \\
  l_5 &: \text{person}(y)
\end{align*}
\]

\[
\begin{align*}
  2 & \geq l_1, \\
  4 & \geq l_3, \\
  2 & \geq 5, \\
  5 & \geq l_1, \\
  8 & \geq l_5, \\
  2 & \geq 9, \\
  9 & \geq l_1
\end{align*}
\]

- Disambiguation
Someone likes everybody.

(Kallmeyer & Romero 2008)

- Underspecified semantic representation

\[
\begin{align*}
  l_1 &: \text{likes}(x, y),
  l_2 &: \text{some}(x, [4, 5]),
  l_3 &: \text{person}(x) \\
  l_4 &: \text{every}(y, [8, 9]),
  l_5 &: \text{person}(y) \\

  2 &\geq l_1, 4 \geq l_3, 2 \geq 5, 5 \geq l_1, 8 \geq l_5, 2 \geq 9, 9 \geq l_1
\end{align*}
\]

- Disambiguation

\[
\begin{align*}
  2 &\rightarrow l_2, 4 \rightarrow l_3, 5 \rightarrow l_4, 8 \rightarrow l_5, 9 \rightarrow l_1 \\
\text{some}(x, \text{person}(x), \text{every}(y, \text{person}(y), \text{likes}(x, y)))
\end{align*}
\]
SCOPE AMBIGUITY IN SEMANTIC UNIFICATION

Someone likes everybody. (Kallmeyer & Romero 2008)

- Underspecified semantic representation

\[ l_1: \text{likes}(x, y), l_2: \text{some}(x, [4, 5]), l_3: \text{person}(x) \]
\[ l_4: \text{every}(y, [8, 9]), l_5: \text{person}(y) \]
\[ 2 \geq l_1, 4 \geq l_3, 2 \geq 5, 5 \geq l_1, 8 \geq l_5, 2 \geq 9, 9 \geq l_1 \]

- Disambiguation

\text{some}(x, \text{person}(x), \text{every}(y, \text{person}(y), \text{likes}(x, y)))

\text{every}(y, \text{person}(y), \text{some}(x, \text{person}(x), \text{likes}(x, y)))
CONCLUSION
CONCLUSION

- TAG’s extended domain of locality in the form of MC set captures discontinuous constituency in syntax.
CONCLUSION

- TAG’s extended domain of locality in the form of MC set
  Captures discontinuous constituency in syntax

- The separation between derived tree and derivation tree
  Resolves tension between surface constituency and semantics
CONCLUSION

- TAG’s extended domain of locality in the form of MC set
  Captures discontinuous constituency in syntax

- The separation between derived tree and derivation tree
  Resolves tension between surface constituency and semantics

- Adjoining operation in semantics as well as syntax
  Reduces apparent non-local relation to a local one
CONCLUSION

› TAG’s extended domain of locality in the form of MC set
  - Captures discontinuous constituency in syntax

› The separation between derived tree and derivation tree
  - Resolves tension between surface constituency and semantics

› Adjoining operation in semantics as well as syntax
  - Reduces apparent non-local relation to a local one

› Use of Delayed Tree Locality
  - Handles non-local binding dependencies
CONCLUSION

› TAG’s extended domain of locality in the form of MC set
  Captures discontinuous constituency in syntax

› The separation between derived tree and derivation tree
  Resolves tension between surface constituency and semantics

› Adjoining operation in semantics as well as syntax
  Reduces apparent non-local relation to a local one

› Use of Delayed Tree Locality
  Handles non-local binding dependencies

› Alternative conception of predicate semantic tree
  Can stay in the realm of Tree-Local MC-TAG
CONCLUSION

› TAG’s extended domain of locality in the form of MC set
  Captures discontinuous constituency in syntax

› The separation between derived tree and derivation tree
  Resolves tension between surface constituency and semantics

› Adjoining operation in semantics as well as syntax
  Reduces apparent non-local relation to a local one

› Use of Delayed Tree Locality
  Handles non-local binding dependencies

› Alternative conception of predicate semantic tree
  Can stay in the realm of Tree-Local MC-TAG
ACKNOWLEDGMENTS

- NSERC Discovery Grants: RGPIN/341442