Regular tree grammars as an underspecification formalism

Alexander Koller
University of Edinburgh

ICCS Seminar
Edinburgh, 23 May 2008

joint work with Stefan Thater and Michaela Regneri
Things I won’t talk about

(1) Sentence generation as planning

**Runtimes**

“The Adj₁...Adjₙ rabbit sleeps.”

**TAG vs. CCG**

- TAG and CCG are weakly equivalent ("mildly context-sensitive languages").
- But: Differences in strong expressive power!

Representable with TAG, but not with CCG

Representable with CCG, but not with TAG
Things I won’t talk about

(2) Instruction giving in virtual environments

“Push the left button!”
Things I won’t talk about

(3) Other topics
Today: Scope underspecification

• Scope underspecification

• Regular tree grammars and how to use them for underspecification

• Applications of RTGs for useful underspecification tasks

• Outlook
Scope ambiguities

\[
\begin{align*}
S & \rightarrow \text{NP \ VP} \\
\text{NP} & \rightarrow \text{every student} \\
\text{VP} & \rightarrow \text{V \ NP} \\
\text{V} & \rightarrow \text{reads} \\
\text{NP} & \rightarrow \text{a book}
\end{align*}
\]

\[
\text{semantic construction}
\]

\[
\begin{align*}
\text{every}_x(\text{student'}(x), \text{a}_y(\text{book'}(y), \text{read'}(x,y))) \\
\text{a}_y(\text{book'}(y), \text{every}_x(\text{student'}(x), \text{read'}(x,y)))
\end{align*}
\]
Scope ambiguities

\[
\text{some}_x(\text{student}'(x), \text{every}_z(\text{lang}'(z), \text{two}_y(\text{dialect-of}'(y,z),\text{read}'(x,y)))) \\
\text{some}_x(\text{student}'(x), \text{two}_y(\text{every}_z(\text{lang}'(z), \text{dialect-of}'(y,z)),\text{read}'(x,y)))) \\
\text{every}_z(\text{lang}'(z), \text{some}_x(\text{student}'(x), \text{two}_y(\text{dialect-of}'(y,z),\text{read}'(x,y)))) \\
\text{every}_z(\text{lang}'(z), \text{two}_y(\text{dialect-of}'(y,z), \text{some}_x(\text{student}'(x), \text{read}'(x,y)))) \\
\text{two}_y(\text{every}_z(\text{lang}'(z), \text{dialect-of}'(y,z)), \text{some}_x(\text{student}'(x), \text{read}'(x,y)))
\]
Scope ambiguities in corpora

Computing all scope readings = about 1.5 years

(Rondane corpus, ERG, July 2006)
Scope underspecification

Semantic construction:

every\textsubscript{x}(\text{student'}(x), a\textsubscript{y}(\text{book'}(y), \text{read'}(x,y)))

a\textsubscript{y}(\text{book'}(y), every\textsubscript{x}(\text{student'}(x), \text{read'}(x,y)))

Semantic representations

(Egg et al. 01; Althaus et al. 03)
Scope underspecification

Syntactic structure → Usp. representation (dominance graph) → Semantic representations
Scope underspecification

- Syntactic structure
- Usp. representation (dominance graph)
- Minimal Recursion Semantics
- Hole Semantics

Redundancy elimination, anaphora, etc.

Fast solvers

Semantic representations

(Fuchss et al. 04)
(K. et al. 03)
(K. & Niehren 00; K. & Thater 06)
(Althaus et al. 03; K. & Thater 07)
Regular tree grammars

• Regular tree grammar is a 4-tuple $G = \langle S, N, \Sigma, R \rangle$ consisting of
  ▸ an alphabet $\Sigma$ of terminal symbols with arities
  ▸ an alphabet $N$ of nonterminal symbols
  ▸ a start symbol $S \in N$
  ▸ a finite set $R$ of production rules

• Regular tree grammars describe languages of trees.
RTGs: An example

\[ S \rightarrow f(A,S) \]
\[ S \rightarrow c \]
\[ A \rightarrow a \]
\[ A \rightarrow b \]

RTG, G

L(G): An (infinite) language of finite trees.
RTGs: Theoretical properties

- Languages generated by RTGs = languages accepted by regular tree automata.
- Regular tree languages closed under intersection, complement, etc.
- For any context-free string grammar, the language of parse trees is a regular tree language (but not vice versa).
RTGs in scope underspecification

(K. et al., ACL 08)

- Sets of semantic representations are finite tree languages.
- All finite tree languages are regular.
- So, let’s use RTGs to describe them!
RTGs for underspecification

“A representative of some company saw every sample.”

\[
S \rightarrow a_x(A_1,A_2) \mid a_z(B_1,A_3) \mid every_y(B_3,A_4)
A_1 \rightarrow a_z(B_1,B_2)
A_2 \rightarrow every_y(B_3,B_4)
A_3 \rightarrow a_x(B_2,A_2) \mid every_y(B_3,A_5)
A_4 \rightarrow a_x(A_1,B_4) \mid a_z(B_1,A_5)
A_5 \rightarrow a_x(B_2,B_4)
B_1 \rightarrow comp_z \quad B_2 \rightarrow repr-of_{x,z}
B_3 \rightarrow sample_y \quad B_4 \rightarrow see_{x,y}
\]

\( G \)
RTGs and dominance graphs

- **Syntactic structure**
- **Minimal Recursion Semantics**
- **Hole Semantics**
- **Usp. representation (dominance graph)**
- **Usp. representation (RTG)**
- **Semantic representations**
RTGs for dominance graphs

Compute RTG from dominance graph with existing algorithm!
(K. & Thater 05; Bodirsky et al. 03)

\{1,2,3,4,5,6,7\} \rightarrow a_x(\{2,4,5\}, \{3,6,7\})
\{1,2,3,4,5,6,7\} \rightarrow a_z(\{4\}, \{1,3,5,6,7\})
\{1,2,3,4,5,6,7\} \rightarrow \text{every}_y(\{6\}, \{1,2,4,5,7\})
\{1,3,5,6,7\} \rightarrow a_x(\{5\}, \{3,6,7\})
\{1,3,5,6,7\} \rightarrow \text{every}_y(\{6\}, \{1,5,7\})
\{1,2,4,5,7\} \rightarrow a_x(\{2,4,5\}, \{7\})
\{1,2,4,5,7\} \rightarrow a_z(\{4\}, \{1,5,7\})
\{2,4,5\} \rightarrow a_z(\{4\}, \{5\})
\{3,6,7\} \rightarrow \text{every}_y(\{6\}, \{7\})
\{1,5,7\} \rightarrow a_x(\{5\}, \{7\})
\{4\} \rightarrow \text{comp}_z \quad \{5\} \rightarrow \text{repr-of}_{x,z}
\{6\} \rightarrow \text{sample}_y \quad \{7\} \rightarrow \text{see}_{x,y}
RTGs and dominance graphs
Applying RTGs (1)

- Elimination of logical redundancy, e.g.:
  “A representative of a company saw a sample.”

- Goal: Strengthen USR to eliminate readings that are equivalent to remaining readings.
Redundancy elimination

• Useful in corpora, e.g. Rondane corpus:

  “For travellers going to Finnmark there is a bus service from Oslo to Alta through Sweden.”
  (3960 readings / all equivalent)

  “We quickly put up the tents in the lee of a small hillside and cook for the first time in the open.”
  (480 readings / two classes)
Redundancy elimination

Underspecified description

Semantic representations

\[ a_x(a_z(P, Q), R) \rightarrow a_z(P, a_x(Q, R)) \]

Rewrite system (defines equivalence)
Redundancy elimination with RTGs

Dominance graph

Relabelled RTG, G

Relabelled semantic representations
Filter grammar

- Regular tree grammar $G_F$ that accepts only canonical orderings of adjacent permutable quantifiers:

  $a_x(a_z(P, Q), R) \rightarrow a_z(P, a_x(Q, R))$

  **Rewrite system**

  $S \rightarrow 1(S, S) \mid 2(S, Q_2) \mid 3(S, S)$
  $Q_2 \rightarrow 2(S, Q_2) \mid 3(S, S)$
  $S \rightarrow 4 \mid 5 \mid 6 \mid 7$
  $Q_2 \rightarrow 4 \mid 5 \mid 6 \mid 7$

  **Filter grammar, $G_F$**

  **Some trees in $L(G_F)$**
Redundancy elimination

- Obtain underspecified description for irredundant readings by grammar intersection:

\[
\begin{align*}
\{1,2,3,4,5,6,7\} & \rightarrow 1(\{2,4,5\}, \{3,6,7\}) \\
\{1,2,3,4,5,6,7\} & \rightarrow 2(\{4\}, \{1,3,5,6,7\}) \\
\{1,2,3,4,5,6,7\} & \rightarrow 3(\{6\}, \{1,2,4,5,7\}) \\
\{2,4,5\} & \rightarrow 2(\{4\}, \{5\}) \\
\{3,6,7\} & \rightarrow 3(\{6\}, \{7\}) \\
\{1,3,5,6,7\} & \rightarrow 1(\{5\}, \{3,6,7\}) \\
\{1,3,5,6,7\} & \rightarrow 3(\{6\}, \{1,5,7\}) \\
\{1,2,4,5,7\} & \rightarrow 1(\{2,4,5\}, \{7\}) \\
\{1,2,4,5,7\} & \rightarrow 2(\{4\}, \{1,5,7\}) \\
\{1,5,7\} & \rightarrow 1(\{5\}, \{7\}) \\
\{4\} & \rightarrow 4 \\
\{5\} & \rightarrow 5 \\
\{6\} & \rightarrow 6 \\
\{7\} & \rightarrow 7 \\
\end{align*}
\]

Filter grammar, \(G_F\)

\[
\begin{align*}
S & \rightarrow 1(S, S) \mid 2(S, Q_2) \mid 3(S, S) \\
Q_2 & \rightarrow 2(S, Q_2) \mid 3(S, S) \\
S & \rightarrow 4 \mid 5 \mid 6 \mid 7 \\
Q_2 & \rightarrow 4 \mid 5 \mid 6 \mid 7
\end{align*}
\]

Relabelled RTG, \(G\)
Redundancy elimination

• Obtain underspecified description for irredundant readings by grammar intersection:

\[
\begin{align*}
\{1,2,3,4,5,6,7\}_s & \rightarrow 1(\{2,4,5\}_s, \{3,6,7\}_s) \\
\{1,2,3,4,5,6,7\}_s & \rightarrow 2(\{4\}_s, \{1,3,5,6,7\}_{Q2}) \\
\{1,2,3,4,5,6,7\}_s & \rightarrow 3(\{6\}_s, \{1,2,4,5,7\}_s) \\
\{3,6,7\}_s & \rightarrow 3(\{6\}_s, \{7\}_s) \\
\{1,3,5,6,7\}_{Q2} & \rightarrow 3(\{6\}_s, \{1,5,7\}_s) \\
\{1,2,4,5,7\}_s & \rightarrow 1(\{2,4,5\}_s, \{7\}_s) \\
\{1,2,4,5,7\}_s & \rightarrow 2(\{4\}_s, \{1,5,7\}_{Q2}) \\
\{1,5,7\}_s & \rightarrow 1(\{5\}_s, \{7\}_s) \\
\{4\}_s & \rightarrow 4 & \{5\}_{Q2} & \rightarrow 5 & \{5\}_s & \rightarrow 5 \\
\{6\}_s & \rightarrow 6 & \{7\}_s & \rightarrow 7
\end{align*}
\]

Relabelled intersection grammar, G'

\[L(G') = L(G) \cap L(G_F)\]
Redundancy elimination

- Obtain underspecified description for irredundant readings by grammar intersection:

\[
\begin{align*}
\{1,2,3,4,5,6,7\}_s & \rightarrow a_x(\{2,4,5\}_s, \{3,6,7\}_s) \\
\{1,2,3,4,5,6,7\}_s & \rightarrow a_z(\{4\}_s, \{1,3,5,6,7\}_Q) \\
\{1,2,3,4,5,6,7\}_s & \rightarrow \text{every}_y(\{6\}_s, \{1,2,4,5,7\}_s) \\
\{3,6,7\}_s & \rightarrow \text{every}_y(\{6\}_s, \{7\}_s) \\
\{1,3,5,6,7\}_Q & \rightarrow \text{every}_y(\{6\}_s, \{1,5,7\}_s) \\
\{1,2,4,5,7\}_s & \rightarrow a_x(\{2,4,5\}_s, \{7\}_s) \\
\{1,2,4,5,7\}_s & \rightarrow a_z(\{4\}_s, \{1,5,7\}_Q) \\
\{1,5,7\}_s & \rightarrow a_x(\{5\}_s, \{7\}_s) \\
\{4\}_s & \rightarrow \text{comp}_z \\
\{5\}_Q & \rightarrow \text{repr-of}_{x,z} \\
\{5\}_s & \rightarrow \text{repr-of}_{x,z} \\
\{6\}_s & \rightarrow \text{sample}_y \\
\{7\}_s & \rightarrow \text{see}_{x,y}
\end{align*}
\]

\( G' \) with original node labels \quad \rightarrow \quad \text{3 semantic representations}
Evaluation on corpus

![Graph showing distribution of sentences with different numbers of readings. The x-axis is labeled as \(\log_2(\text{# readings})\), and the y-axis is labeled with counts. There are two bars for each bin: one for the original count and one for the count after redundancy elimination.]

- 7 hours \(\Rightarrow 25\) seconds
- Billions of years \(\Rightarrow 75\) seconds
• RTGs can be extended to weighted RTGs (Graehl & Knight 04): Add numeric weight to each production rule.

\[
\begin{align*}
S & \rightarrow f(A,S) \ [0.7] \\
S & \rightarrow c \ [1] \\
A & \rightarrow a \ [0.5] \\
A & \rightarrow b \ [0.5]
\end{align*}
\]
Applying RTGs (2)

• Use RTG weights to represent preferences on scope ambiguities.

• Compute best reading of hardest sentence in Rondane (with random weights): 1 sec.

• Open problem: Decent statistical model for scope preferences.
Applying RTGs (3)

(Regneri et al., ACL short paper 08)

- Use dominance graphs for underspecified representation of discourse structures.

C1. I try to read a novel
C2. if I feel bored
C3. or I am unhappy.

(Gardent & Webber 98)
Applying RTGs (3)

• Using wRTGs, we can:
  ‣ express fine-grained structural constraints about the discourse structure;
  ‣ express numeric preferences;
  ‣ compute the best reading efficiently.

• Fine-tuned RTG implementation computes best reading of longest discourse in RST Treebank (300 sentences; $10^{178}$ readings) in three minutes.

(Regneri et al., ACL short paper 08)
Summary so far

- **Syntactic structure**
- **Usp. representation** (dominance graph)
- **Minimal Recursion Semantics**
- **Hole Semantics**
- **Usp. representation** (RTG)
- **Semantic representations**

Redundancy elimination:
- **best reading**
Direct semantic construction

- Parse charts of many grammar formalisms can be seen as regular tree grammars!

\[
\begin{align*}
[S,0,3] & \rightarrow s([NP,0,1],[VP,1,3]) \\
[NP,0,1] & \rightarrow np(john) \\
[VP,1,3] & \rightarrow vp([V,1,2],[NP,2,3]) \\
[V,1,2] & \rightarrow v(loves) \\
[NP,2,3] & \rightarrow np(john)
\end{align*}
\]

Parse chart = RTG

Parse tree
Example: Scope in CCG
(with Mark Steedman)

“Everyone loves someone.”

\[
S \to [0,3,S, \forall x.\text{love}(sk^\{x\})(x)] \mid [0,3,S, \forall x.\text{love}(sk^\emptyset)(x)]
\]

\[
[0,3,S, \forall x.\text{love}(sk^\{x\})(x)] \to [0,3,S, \forall x.\text{love}(skolem)(x)]
\]

\[
[0,3,S, \forall x.\text{love}(sk^\emptyset)(x)] \to s([0,1,NP, \lambda P \forall x. P(x)], [1,3,S\setminus NP, \text{love}(sk^\emptyset)])
\]

\[
[0,3,S, \forall x.\text{love}(skolem)(x)] \to s([0,1,NP, \lambda P \forall x. P(x)], [1,3,S\setminus NP, \text{love}(skolem)])
\]

\[
[0,1,NP, \lambda P \forall x. P(x)] \to \text{np(everyone)}
\]

\[
[1,3,S\setminus NP, \text{love}(sk^\emptyset)] \to \text{vp([1,2, S\setminus NP/NP, love])}
\]

\[
[1,3,S\setminus NP, \text{love}(skolem)] \to \text{vp([1,2, S\setminus NP/NP, love])}
\]

\[
[1,2,S\setminus NP/NP, \text{love}] \to v(\text{loves})
\]

\[
[2,3,NP, \text{sk}^\emptyset] \to [2,3,NP, \text{skolem}]
\]

\[
[2,3,NP, \text{skolem}] \to \text{np(someone)}
\]

\[
\{1,2,3\} \to \text{every}_x(\{2,3\})
\]

\[
\{1,2,3\} \to a_y(\{1,3\})
\]

\[
\{2,3\} \to a_y(\{3\})
\]

\[
\{1,3\} \to \text{every}_x(\{3\})
\]

\[
\{3\} \to \text{love}_{x,y}
\]
Minimal Recursion Semantics

Hole Semantics

Syntactic structure

Usp. representation (dominance graph)

Semantic representations

Extract semantic RTG from parse chart (for CCG, synchronous grammars, etc.)

Usp. representation (RTG)

redundancy elimination

best reading

Utool
Outlook

• Explore direct RTG semantic construction by extraction from parse charts.

• Explore further uses of RTGs in underspecification (computation of weakest readings?).

• Explore semantic construction based on tree transducers.