CCG and Unbounded Dependencies

Mark McConville

ESSLLI: CCG and Linguistic Diversity
8-12 August 2005
What is CCG?

CCG is a grammar formalism

- a theory of a particular class of languages

- a specification of
  - a set of grammars
  - a ‘generates’ relation i.e. Which grammars generate which strings?
  - a compositional semantics
What is a CCG?

A CCG over alphabet $\Sigma$ is an ordered triple $\langle A, S, L \rangle$

- $A$ is an alphabet of saturated category labels
- $S \in A$
- $L$ is a finite mapping from $\Sigma$ to CCG categories over $A$
What are CCG categories?

saturated categories:

• for all $X \in A$, $X$ is a CCG category over $A$

unsaturated categories:

• for all CCG categories $x$ and $y$ over $A$ and all modalities $m \in \{\ast, \diamond, \times\}$:
  
  – $X / m Y$ is a CCG category over $A$
  
  – $X \setminus m Y$ is a CCG category over $A$
CCG generation

CCG $\langle A, S, L \rangle$ over alphabet $\Sigma$ generates string $s \in \Sigma^*$ iff.

- $\langle s, S \rangle$ is in the combinatory projection of $L$

Combinatory projection of CCG lexicon $L$:

- the closure of $L$ under the universal CCG combinatory operations
  - application
  - composition
  - type raising
  - substitution
CCG compositional semantics

A CCG of language $L$ is also a compositional theory of $L$’s semantics.

A recipe:

1. Take CCG $\langle A, S, L \rangle$ over alphabet $\Sigma$

2. Define a function $\delta$ from $A$ to semantic domains e.g.
   - $\delta(S) = \{true, false\}$
   - $\delta(NP) = \{john, mary\}$
CCG compositional semantics (ctd)

3. Semantic domains of unsaturated categories follow:

\[ \delta(X/Y) = \delta(X\backslash Y) = \delta(X)^{\delta(Y)} \]

e.g. \[ \delta(S\backslash NP) = \delta(S)^{\delta(NP)} = \{true, false\}\{john, mary\} = \emptyset(\{john, mary\}) \]

4. For every \( \langle s, X \rangle \in L \): assign \( s \) a denotation from \( \delta(X) \) e.g.

- John ⊩ NP : john
- loves ⊩ (S\backslash NP)/NP : \{⟨john, john⟩, ⟨john, mary⟩, ⟨mary, john⟩\}
CCG compositional semantics (ctd)

5. Denotata of phrases follow automatically, since every CCG combinatory operation is also an operation on denotata

\[ X/Y : f \quad Y : x \quad \Rightarrow \quad X : f(x) \]

\[ X/Y : f \quad Y/Z : g \quad \Rightarrow \quad X/Z : g \circ f \]

John loves Mary

\[
\begin{array}{c}
\text{NP} : j \\
(S \backslash \text{NP})/\text{NP} : \{\langle j, \{\langle j, t \rangle, \langle m, t \rangle \rangle, \langle m, \{\langle j, t \rangle, \{m, f \}\} \rangle \}
\end{array}
\]

\[
\begin{array}{c}
\text{NP} : m \\
S \backslash \text{NP} : \{\langle j, t \rangle, \langle m, f \rangle \}
\end{array}
\]

\[
\begin{array}{c}
S : t
\end{array}
\]
CCG as a formalism

The CCG formalism specifies:

- a set of grammars
- a ‘generates’ relation
- a compositional semantics

[Our theories] allow only a finite number of core grammars anyway (apart from the lexicon) . . . (Chomsky, 1983)
What is CCG?

CCG is a *restrictive* grammar formalism

- a grammar formalism is a theory of a particular class of languages
  i.e. its *generative capacity*

- CCG’s generative capacity is a *proper subset* of:
  - the recursively enumerable languages
  - the context-sensitive languages
The Chomsky hierarchy

\[ \varnothing(\Sigma^*) \]

RE, CS, MCS, CF, R
Implications

CCG is

• a restrictive grammar formalism

• a theory of human linguistic competence

So, CCG predicts

• a fixed upper bound on the complexity of natural language

• some constructions are not found in human languages
Unbounded dependency constructions

Three kinds:

• unbounded *nested* dependencies

• unbounded *cross-serial* dependencies

• unbounded *scrambling*
Nested dependencies

\[ n_n \ldots n_1 v_1 \ldots v_n \]

- \( n_i \) is a dependent of \( v_i \)
- \( v_i \) is a dependent of \( v_{i+1} \)
- examples:
  
  \[
  n_2 n_1 v_1 v_2 \\
  n_3 n_2 n_1 v_1 v_2 v_3 \\
  n_4 n_3 n_2 n_1 v_1 v_2 v_3 v_4
  \]
Nested dependencies in English

The man who said that . . . is arriving today.

The men who said that . . . are arriving today.

Unbounded nested dependencies:

- The man$_2$ who said that the men$_1$ who did it are$_1$ arriving today is$_2$ arriving today.
Nested dependencies in German

Jan sagt, daß wir dem Hans das Haus anstreichen halfen.

Jan says that we helped Hans paint the house.

Jan sagt, daß wir die Kinder dem Hans das Haus anstreichen liessen.

Jan says that we let the children help Hans paint the house.
Generating nested dependency constructions

Chomsky’s *Syntactic Structures*:

- languages with unbounded nested dependencies are *not* regular

- some context-free languages exhibit unbounded nested dependencies

e.g. CFG for $a^nb^n$:

\[
S \rightarrow ab \\
S \rightarrow aSa
\]
Nested dependency constructions in CCG

The generative capacity of CCG includes all the context-free languages.

So: CCG predicts that human linguistic competence may contain unbounded nested dependency constructions

e.g. CCG for $n_n \ldots n_1 v_1 \ldots v_n$

\[
\begin{align*}
v_1 & \vdash S_1 \backslash NP_1 \\
v_{i \geq 2} & \vdash (S_i \backslash NP_i) \backslash S_{i-1} \\
n_i & \vdash NP_i
\end{align*}
\]
Deriving nested dependency constructions

\[
\begin{align*}
  n_3 & \quad n_2 & \quad n_1 & \quad v_1 & \quad v_2 & \quad v_3 \\
  \overline{NP}_3 & & \overline{NP}_2 & & \overline{NP}_1 & & S_1 \setminus \overline{NP}_1 & & (S_2 \setminus \overline{NP}_2) \setminus S_1 & & (S_3 \setminus \overline{NP}_3) \setminus S_2 \\
  & & & & S_1 & & S_2 \setminus \overline{NP}_2 & & S_2 & & S_3 \setminus \overline{NP}_3 & & S_3 \\
  & & & & & & & & & & & \\
\end{align*}
\]
Cross-serial dependencies

\[ n_n \ldots n_1 v_n \ldots v_1 \]

- \( n_i \) is a dependent of \( v_i \)
- \( v_i \) is a dependent of \( v_{i+1} \)
- examples:
  - \( n_2 n_1 v_2 v_1 \)
  - \( n_3 n_2 n_1 v_3 v_2 v_1 \)
  - \( n_4 n_3 n_2 n_1 v_4 v_3 v_2 v_1 \)
Cross-serial dependencies in Swiss German

Jan säit, das mer em Hans$_2$ es huus$_1$ hälfe$_2$ aastriiche$_1$.

Jan says that we the.DAT Hans the.ACC house helped paint

*Jan says that we helped Hans paint the house.*

Jan säit, das mer d’chind$_3$ em Hans$_2$ es huus$_1$ lönd$_3$ hälfe$_2$ aastriiche$_1$.

Jan says that we the.ACC children the.DAT Hans the.ACC house let help paint

*Jan says that we let the children help Hans paint the house.*
Generating cross-serial dependency constructions

Languages with unbounded cross-serial dependencies are *not* context-free.

Possible solution: deny that human linguistic competence exhibits unbounded cross-serial dependency constructions.

*One could argue that the phenomenon of cross-serial clause structure is bounded by, say, five embeddings or, to be more generous, one hundred. In either case, the language with bounded cross-seriality would be context-free, regardless of case-marking properties. Down this path lies tyranny. Acceptance of this argument opens the way to proofs of natural languages as regular, nay, finite. The linguist proposing this counterargument to salvage the context-freeness of natural language may have won the battle, but has certainly lost the war.* (Shieber)
Cross-serial dependency constructions in CCG

Some CCGs generate languages exhibiting unbounded cross-serial dependencies.

e.g. CCG for \( n_n \ldots n_1 v_n \ldots v_1 \)

\[
\begin{align*}
v_1 & \vdash S_1 \setminus \text{NP}_1 \\
v_{i \geq 2} & \vdash (S_i \setminus \text{NP}_i) / S_{i-1} \\
n_i & \vdash \text{NP}_i
\end{align*}
\]
Deriving cross-serial dependency constructions

\[
\begin{array}{cccccccc}
  n_4 & n_3 & n_2 & n_1 & v_4 & v_3 & v_2 & v_1 \\
  \text{NP}_4 & \text{NP}_3 & \text{NP}_2 & \text{NP}_1 & (S_4 \backslash \text{NP}_4)/S_3 & (S_3 \backslash \text{NP}_3)/S_2 & (S_2 \backslash \text{NP}_2)/S_1 & S_1 \backslash \text{NP}_1 \\
  & & & & \xrightarrow{B \times 2} & \xrightarrow{B \times 2} & \xrightarrow{B \times 2} & \\
  & & & (S_4 \backslash \text{NP}_4)/(S_4 \backslash \text{NP}_4) \backslash \text{NP}_3)/S_2 & (S_3 \backslash \text{NP}_3)/(S_3 \backslash \text{NP}_3) \backslash \text{NP}_2)/S_1 & (S_2 \backslash \text{NP}_2)/(S_2 \backslash \text{NP}_2) \backslash \text{NP}_1 \xrightarrow{B \times 1} & \\
  & & & & & (S_4 \backslash \text{NP}_4)/(S_4 \backslash \text{NP}_4) \backslash \text{NP}_3) \backslash \text{NP}_2 & (S_4 \backslash \text{NP}_4)/(S_4 \backslash \text{NP}_4) \backslash \text{NP}_3) \backslash \text{NP}_2 & \\
  & & & & & (S_4 \backslash \text{NP}_4)/(S_4 \backslash \text{NP}_4) \backslash \text{NP}_3) \backslash \text{NP}_2 & (S_4 \backslash \text{NP}_4)/(S_4 \backslash \text{NP}_4) \backslash \text{NP}_3) \backslash \text{NP}_2 & \\
  & & & & & S_4 \backslash \text{NP}_4 & S_4 \backslash \text{NP}_4 & \\
  & & & & & S_4 & S_4 & \\
\end{array}
\]
Unbounded dependency constructions

CCG predicts human linguistic competence may contain

• unbounded nested dependency constructions

• unbounded cross-serial dependency constructions
Local scrambling

Local scrambling in Turkish:

Ayse kitab-i oku-yor.
Ayse.NOM book-ACC read-PROG

Ayse reads the book.

Kitab-i Ayse oku-yor.
book-ACC Ayse.NOM read-PROG

Ayse reads the book.
Local scrambling (ctd)

Within embedded clauses:

Fatma.NOM [Ayse-GEN book-ACC read-GER] know-PROG

Fatma knows that Ayse reads the book.

Fatma.NOM [book-ACC Ayse-GEN read-GER] know-PROG

Fatma knows that Ayse reads the book.
Local scrambling (ctd)

Of embedded clauses:


[Ayse-GEN book-ACC read-GER] Fatma.NOM know-PROG

Fatma knows that Ayse reads the book.
Long-distance scrambling

Turkish also exhibits *non-local* scrambling

book-ACC Fatma.NOM [Ayse-GEN read-GER] know-PROG

Fatma knows that Ayse reads the book.
Doubly unbounded scrambling

No limit on the *distance* an argument can move.
No limit on the *number* of arguments that can move.

\[ \sigma(n_1 \ldots n_n) \ v_1 \ldots v_n \]

- \( n_i \) is a dependent of \( v_i \), and \( v_i \) is a dependent of \( v_{i+1} \)

- examples:
  \[
  n_3n_1n_4n_1v_1v_2v_3v_4 \\
  n_2n_4n_1n_3v_1v_2v_3v_4
  \]
Generating doubly unbounded scrambling

Languages with doubly unbounded scrambling constructions are not mildly context-sensitive.

i.e. they cannot be generated by CCGs.

But: CCGs can derive some long-distance scrambling constructions

e.g.

\[ v_1 \vdash S_1 \backslash NP_1 \]
\[ v_{i \geq 2} \vdash (S_i \backslash NP_i) \backslash S_{i-1} \]
\[ n_i \vdash NP_i \]
Deriving LDS constructions in CCG

i.e. any string of form $\sigma(n_1n_2n_3) \ v_1v_2v_3$ can be derived.
But: strings of form $\sigma(n_1n_2n_3)n_4v_1v_2v_3v_4$ cannot be derived in CCG.

Conclusion: CCG can derive all the LDS constructions for which reliable native speaker judgments can be obtained.
Conclusions

CCG is a restrictive grammar formalism.

CCG predicts human linguistic competence may exhibit:

- unbounded nested dependency constructions
- unbounded cross-serial dependency constructions

But *not* doubly unbounded scrambling constructions.