Combinatory Categorial Grammar

- widely used in computational linguistics
- mildly context-sensitive grammar formalism
- efficiently parsable
- constituency-based structures

Application Example

Mary likes musicals

V NP

NP VP

S

How does a CCG work?

- **lexicon** associates each input symbol with a set of categories
- **rule system** describes how adjoining categories can be combined
- derivation is successful if the derivation tree is rooted in an initial category
- accepted tree languages of a CCG are the relabeled sets of derivation trees
- derivation trees are always binary trees

Example for $a^n b^n$

- input alphabet $\Sigma = \{a, b\}$
- atomic categories $A = \{C, D\}$
- set of initial categories $I = \{C\}$

- lexicon $L$ with $L(a) = \{C/D, C/D/C\}$ and $L(b) = \{D\}$

Rule System

**Forward Rule**

- $D/C/E/D \rightarrow D/E/C$
- $D/C/E/E \rightarrow C/D/E/C$

**Backward Rule**

- $C/D/E \rightarrow D/C/E/D$
- $C/D/E/C \rightarrow D/C/E/D$

What classes of tree languages can CCG accept?

- 0-CCG $\subset$ regular tree languages
- 1-CCG = regular tree languages
- $k$-CCG = tree adjoining languages

Why is this question important?

For parsing natural languages, you want as much power as necessary to model linguistic structures but not more to limit parsing complexity.

Future Work

- tree language expressivity of
  - CCG with substitution rules
  - weighted CCG