Combinatory Categorial Grammar
CCG is mildly context sensitive

- CCG categories to LIG symbols:
  
  CCG categories can be viewed as their result category plus a stack-valued feature identifying their arguments and the order of their combination.

  e.g. $S/\text{NP}/\text{NP}$ is $S[/\text{NP}, \backslash\text{NP}\ldots]$ in LIG

- Under the principle of inheritance and consistency, combinatory rules map one-to-one to LIG rules.

- Thus CCG is provably equivalent to LIGs (Vijay-Shanker & Weir, 1993), which characterise the class of mildly context sensitive languages.
Example: Dutch crossing dependencies ($B^n_\times$):

$$X/Y \quad (Y \backslash Z) \Rightarrow_{B^n} X \backslash Z$$

..omdat ik Cecilia de nijlpaarden zag voeren
..because I Cecilia the hippopotamuses saw feed
‘..because I saw Cecilia feed the hippopotamuses.’

\[
\begin{aligned}
\text{omdat } \text{ik} \quad \text{Cecilia} \quad \text{de nijlpaarden} \quad \text{zag} \quad \text{voeren}\\
\begin{array}{cccc}
\text{NP}_1 & \text{NP}_2 & \text{NP}_3 & ((S \backslash \text{NP}_1) \backslash \text{NP}_2) / \text{VP} \\
\end{array} \\
\begin{array}{c}
\text{VP} \backslash \text{NP}_3 \quad >_{B^n_\times} \\
((S \backslash \text{NP}_1) \backslash \text{NP}_2) \backslash \text{NP}_3 \\
(S \backslash \text{NP}_1) \backslash \text{NP}_2 \\
S \backslash \text{NP}_1 \\
S
\end{array}
\end{aligned}
\]

\text{note: indices in NPs are just for ease of exposition.}
..omdat ik Cecilia Henk de nijlpaarden zag helpen voeren

‘..because I saw Cecilia help Henk feed the hippopotamuses.’
Conclusion

- CCG is radically lexicalist in the sense that everything projects from the lexicon, and there is no intermediary between UG and the lexicalised (competence) grammar.

- Lexical types do the work of mapping (no multiple levels).

- Competence Grammar (i.e. lexicon) is shaped by UG (its principles).

- The theory attempts to stick to the strong hypothesis that formal restrictiveness (i.e. low automata-theoretic power) might just be the explanation behind all these constraints, which are violable if stated over structures or functions, but seem to follow limited kinds of combinatory dependencies...

  .. which are, semantic in nature. So good syntax is faithful reflection of semantic dependencies.