cs401, 10 / 27 / 11

- hw/proj questions?
- parsing non-LR CFGs
- (maybe) semantics in CFGs
GLR Parsing

- extension of LR parsing that works for any cfg: allow multiple actions in a table cell
- like LR: bottom-up with tables for actions
- can base on any LR formulation: LR(0), SLR, LR(1), etc
- $O(n^3)$, but faster when low ambiguity
Simple English Grammar

S → NP VP
NP → N | D NP | NP PP
VP → V NP | VP PP
PP → P NP

N → girl | boy | telescope
V → saw
P → with
D → the
English Grammar and LR(0) DFA

S \rightarrow NP \ VP
NP \rightarrow N \mid D \ NP \mid NP \ PP
VP \rightarrow V \ NP \mid VP \ PP
PP \rightarrow P \ NP

N \rightarrow \text{girl} \mid \text{boy} \mid \text{telescope}
V \rightarrow \text{saw}
P \rightarrow \text{with}
D \rightarrow \text{the}
English Grammar and LR(0) dfa
Chart Parsing

• Top-down: Earley
  – $O(n^3)$ but faster when low ambiguity
• Bottom-up: CYK
  – $O(n^3)$
  – Can deal well w/ malformed input
• Both can work left to right
Parsing General CFGs

• Advantages of GLR parsing
  – Once the table is built, parsing is fast
  – Low complexity when low ambiguity

• Advantages of chart parsing
  – Easy to understand, code and parallelize
  – LR parse tables must be rebuilt on grammar change
  – GLR parse tables are humongous

• In practice, chart parsing used for NLP and even for PLs in some research environments