HW #3
CKY Parsing

- Goals:
  - Complete implementation of CKY parser
  - Implement dynamic programming approach
  - Incorporate/follow backpointers to recover parse
Implementation

• Build full parser:
  • Can use any language for implementation
    • As long as it works on the cluster

• You may use existing data structures for rules, trees
  • E.g. NLTK has nice tree data structure
  • CKY algorithm must be your own

• Dynamic programming tabulation crucial

• Smaller grammar (similar to HW#1)
  • (Back to ATIS for HW#4)
Notes

• Teams:
  • You may work in teams of two on this assignment
    • Submit the assignment itself to one teammate’s CollectIt
    • Put pointer note in other teammate’s CollectIt
    • Discuss contributions in readme

• Test grammar:
  • Pre-converted to CNF
  • Start symbol: TOP
  • Parse should span input and be rooted at TOP