

TRA301/COS401: Homework 3

Due Date: March 26, 2009

1. **Parsing:** You now know the limitations of finite-state techniques and you are eager to use context-free grammars (CFG) for modeling natural language. To do this, you will need a CFG parser.
 - (a) (30 points) Implement a CKY parser in your favorite programming language. The program should read in a file containing CFG rules, read in input sentences and produce parse trees for the input sentences according to the grammar. If there are multiple parses according to the grammar, then the parser should be capable of producing all of them.
 - (b) (10 points) Convert the grammar in file “3-1b.txt” to Chomsky Normal Form. The grammar rules have the format *left hand side symbol* followed by a *list of right hand side symbols*. “S” is the start symbol for the grammar. Words that are in lowercase are terminal symbols of the grammar.
 - (c) (20 points) Using the converted grammar from 1(b) parse the sentences in “3-1c.txt” using your CKY parser. Show all parse trees for each sentence. You can output parse trees as indented lines, you don’t have to draw pretty trees.
 - (d) (20 points) Some sentences in “3-1c.txt” cannot be parsed by the grammar. Extend your parser to recover longest spanning partial parses. Partial parses are parses for substrings of the input. A partial parse that spans the most number of tokens is a longest spanning partial parse. A set of partial parses is needed to span the tokens of a sentence. You can use a greedy approach to selecting the partial parses.
 - (e) (20 points) How will you “fix” the grammar so that the ungrammatical sentences in “3-1e.txt” do not parse? Show all the rules of the “fixed” grammar.