Assignment 5
Implementing CKY

In this assignment, you will have to implement the CKY parser by writing code into the get_best_parse method of the class PCFGParser. You will be using the Manually Annotated Sub-Corpus (MASC) from the American National Corpus (ANC): http://www.anc.org/data/masc/. The data is provided in the code.

At the beginning of the main method in PCFGParserTester.py some training and test trees are read in. The parser is then used to predict trees for the sentences in the test set. For each test sentence, the parse given by your parser is evaluated by comparing the constituents it generates with the constituents in the hand-parsed version. From this, precision, recall, and the F1 score are calculated.

You should familiarize yourself with these basic classes:

- ling.Tree: CFG tree structures
- Lexicon: Pre-terminal productions and probabilities
- Grammar, UnaryRule, BinaryRule: CFG rules and methods to access them

Tree is a linguistic tree class. Lexicon is a minimal lexicon, but it handles rare and unknown words adequately for the present purposes. Grammar is a class you use to learn a PCFG from the training trees.
UnaryRule and BinaryRule are simply the classes the grammar uses to store these learned productions. They each bear the frequency estimated probabilities from the training set.

Although it is not required, it is strongly recommended that you get your parser working on the miniTest dataset before you attempt the treebank datasets. The miniTest data set consists of 3 training sentences and 1 test sentence from a toy grammar. There are just enough productions in the training set for the test sentence to have an ambiguity due to PP-attachment. There are unary, binary and ternary grammar rules in training sentences. You can binarize and unbinarize trees using the TreeBinarization class.

Make sure you can run the main method of the PCFGParserTester class.

Running:
python assignment5.py --data miniTest –-parser PCFGParser
will train and test your parser on a few sentences from a toy grammar.

Running:
python assignment5.py --data masc –-parser PCFGParser
will train and test your parser on the MASC dataset.

If the option –-parser is omitted, the default parser used is a BaselineParser, which performs quite poorly on the MASC dataset. Setting max_length to a certain value determines the maximum length of sentences to test on (it does not affect the training set). You can lower this value for preliminary experiments, but your final parser should work on sentences of at least length 20 in a reasonable time.

For the miniTest dataset your parser should match the given parse of the test sentence exactly. Once you've got this working you can move on to the MASC dataset.