Computational Linguistics 1 - Fall 2011
CMSC/LING723, LBSC 744
Homework 5 - Due 15 November 2011

Submission Guidelines

1. Print out any written portion of the assignment and submit in class on November 15, *as well as* emailing your writeup, in .pdf format, to `compling723.fall2011@gmail.com`.
2. Any code you write for this homework should also be submitted to `compling723.fall2011@gmail.com`.
3. Be sure include your full name and e-mail in your submission, and clearly indicate each problem number in your solution set.

Data

For this assignment, we will be parsing our test corpus. However, we will be ignoring lexical items, and only parsing "down to" the POS tags. You may either build your parser to take full parse trees on input, then ignore all but the POS tags for initializing your CYK chart, or you may build your parser to take just a sequence of POS tags on input. My CYK parser takes just a sequence of POS tags on input, and I created that sequence from the parse files as follows:

```
cat f2-21.test.parse | \
    awk '{for (j=1;j<NF;j++) if (substr($(j+1),1,1)!="(") \
    printf("%s ",substr($j,2,length($j))); print "";}' > \
    f2-21.test.parse-POSonly
```

(you may also write your own script to extract POS sequences, I only provide mine as a suggestion and example.)

Problem 1 (65 points)

(a) (40 points) Write a CYK parser. Using the Chomsky Normal Form grammar that you induced for Homework 4, parse just the first 2000 sentences of our test corpus, f2-21.test.parse.noLEX. (Remember that you can input just POS sequences rather than the full parse, but you will have to add the "dummy" words (Wpos) back into your output.)

(b) (10 points) Were there any parses that failed? Why? How did you handle that in your parser?

(c) (15 points) In order to evaluate the quality of your parses, we need to put them back into the same format of the true parses—the output of your parser differs from the true format in that it is in Chomsky Normal Form. Post-process your trees to transform from CNF to the original grammar rules. Make sure that your final output includes the "dummy" words (Wpos) in the noLEX files.

As a sanity check, my CYK parser implementation returned the following parse for the first sentence in the test set:

```
(TOP (S (NP (NN Wnn) (NN Wnn) (CC Wcc) (NN Wnn) (NNS Wnns))
        (VP (VBD Wvbd) (ADVP (RBR Wrbr) (IN Win) (RB Wrb))) (. W.)))
```

(and this is what it looks like in CNF form):

```
(TOP (S (NP (NN Wnn)
            (NP~NN (NN Wnn)
                       (NP~NN~NN (CC Wcc)
                                     (NP~NN~NN~CC (NN Wnn) (NNS Wnns)))))
        (S~NP (VP (VBD Wvbd)
                      (ADVP (RBR Wrbr)
                                (ADVP~RBR (IN Win) (RB Wrb))))
              (. W.))))
```

Turn in your parser output, with one parse per line, and I will evaluate the F-score accuracy of your parser using the `evalb` evaluation program (available online, if you wish you evaluate your own parses).

As a sanity check, my CYK parser has an F-score of 76.07.

Problem 2 (15 points)

Let's analyze your parser and its output.

(a) (5 points) How long did it take your parser to parse the first 100 sentences? How long did it take your parser to parse the entire test corpus (2000 sentences)?

As a sanity check, my CYK parser took 15 minutes to parse all 2000 sentences.

(b) (5 points) Were there a few sentences that took longer to parse than average? Show one of them, and discuss why it required more computation.

(c) (5 points) Show your parser's prediction for a parse that it got wrong; show the true parse as well. What went wrong, and does it seem like a reasonable error to you (either linguistically or computationally)?