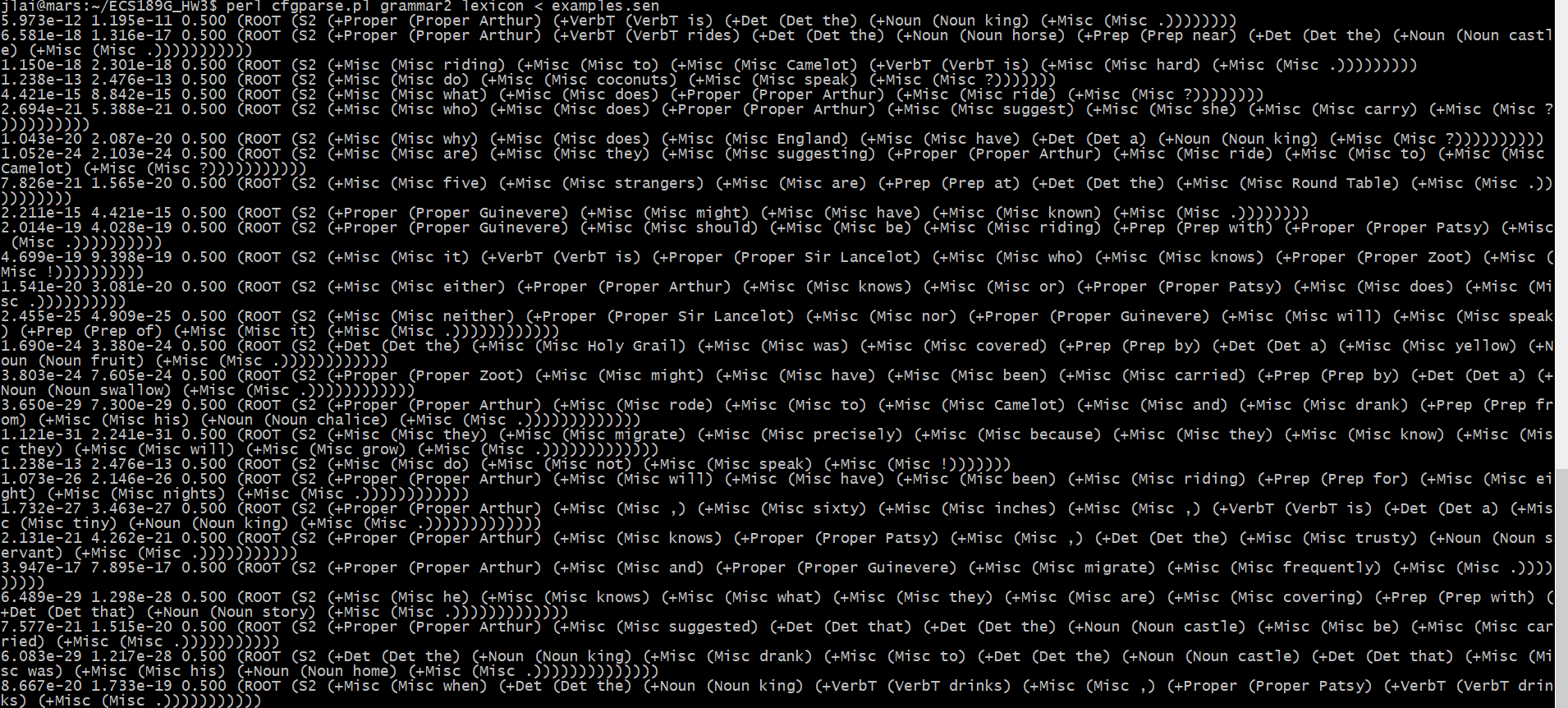
HW3 Report

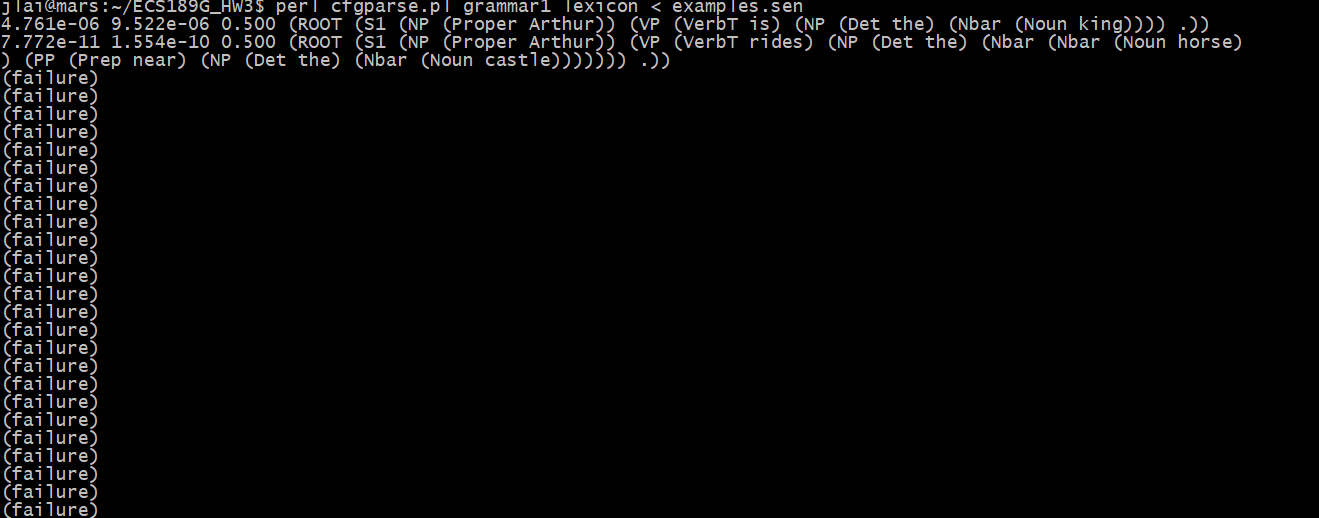
Task 1

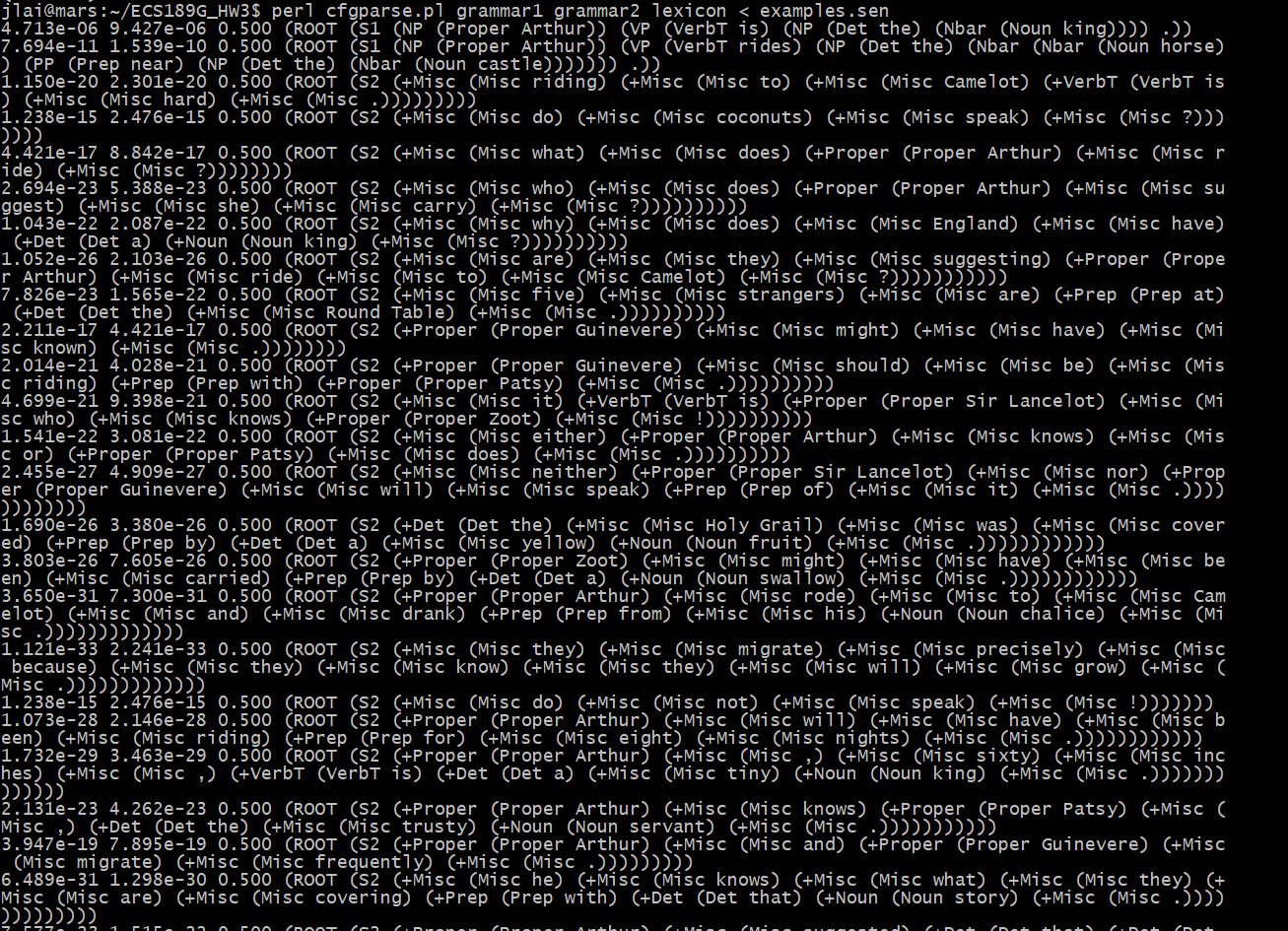


This PCFG implements a quadgram language model because it requires three transitions to reach the terminal after the ROOT. For example, S2->+Proper, +Proper->Proper+ + VerbT, and Proper->Authur. The probability of parsing “Authur” is

, which is exactly a quadgram model.

Task 2

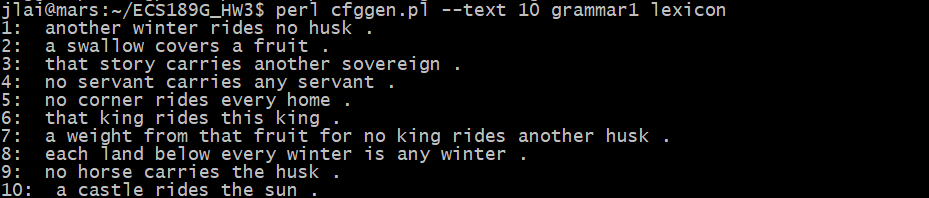


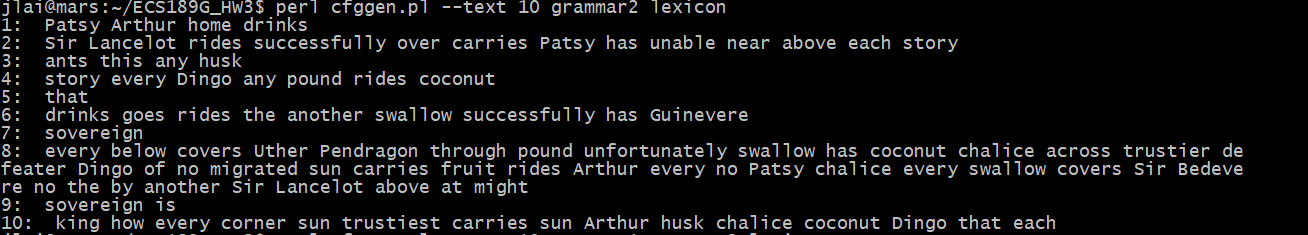
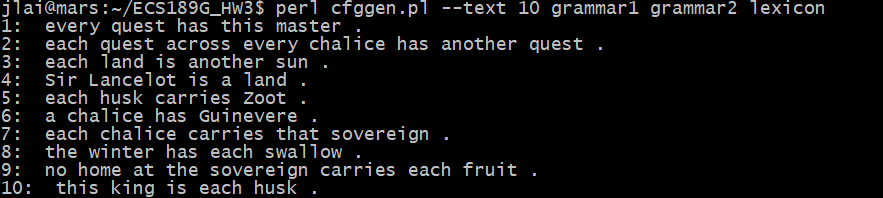


When running the code with only gramma1, it can only parse the first two sentences successfully but the rest cannot parse. When running the code with also gramma2, all sentences could be parsed successfully.

There are several reasons accounting for this difference. Firstly, the grammar1 dose not cover all POS tags and has little rules, like it has no Misc. Secondly, considering examples.sen, there are many sentences ending with question mark and exclamation. The grammar1 has little rule for these while the grammer2 has Misc tag. Hence with grammr2, all of the sentences could be parsed successfully.

Task 3



Using only grammar1, we can see the sentences have almost the same length, which is five words. There are few very short sentences which has less than four words or very long sentences which has more than seven word. And the length cannot be very long because grammar1 has no recursive rules.

Using only grammar2, we can see the sentences have different lengths. There are very short sentences and also very long sentences because there are many recursive rules in grammar2.

Using both grammar1 and grammar2, we can see that there are difference lengths of sentences but mostly are five-word, which indicates that grammar1 is called more frequently because ROOT derives into S1 in 99 times out of 100 times basically. And grammar2 is called less frequently. So there exists some different length sentences.

And the probabilities are also changed as the weights from grammar1 and grammar2 are merged.

Task 4

I use NLTK package of Python to generate my production rules and lexicon rules and combine Penn Treebank and Brown’s production rules. But it seems not ideal. For example, they have many rules with same probabilities. So I amplify some of the probability of the regular rules. Besides, some relation tags are too complex that I make them into groups together, like grouping JJR, JJS into JJ.

Task 5

There are 6 sentences are grammatically correct in the 20 sentences.