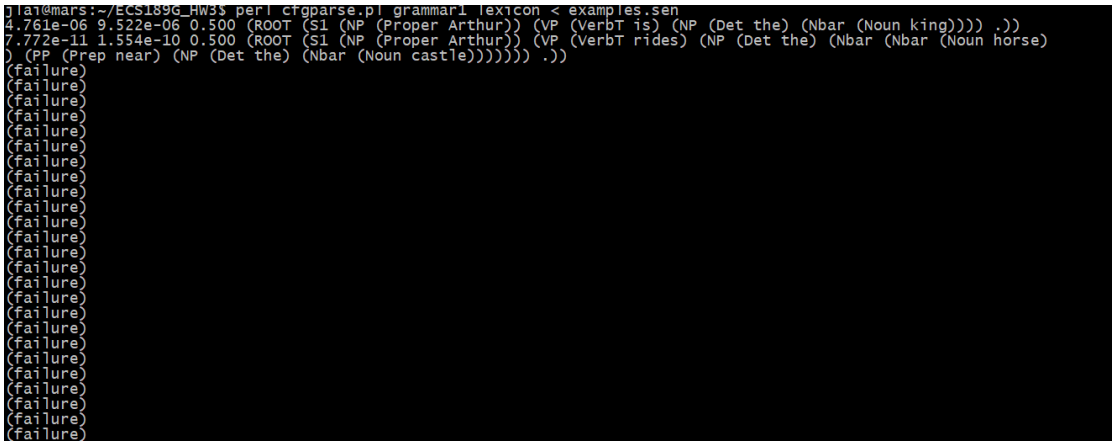
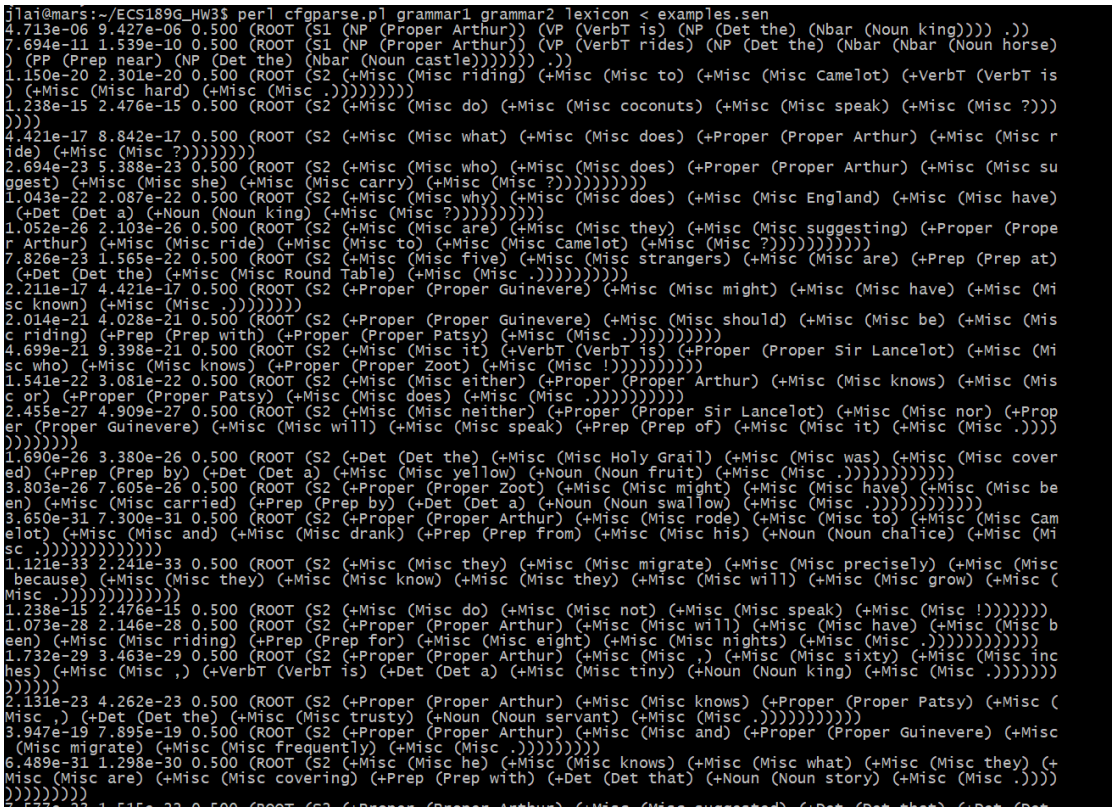
HW3 Report

Task 1

For grammar2 and lexicon, we can see that both of them have one probability for each word and gramma. So this is a UniformModel.

Task 2

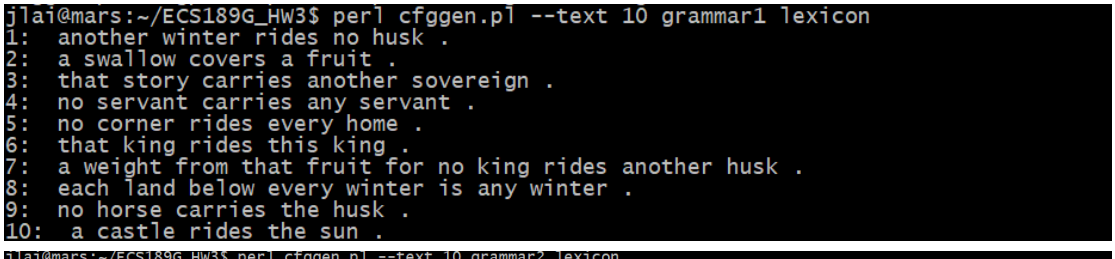


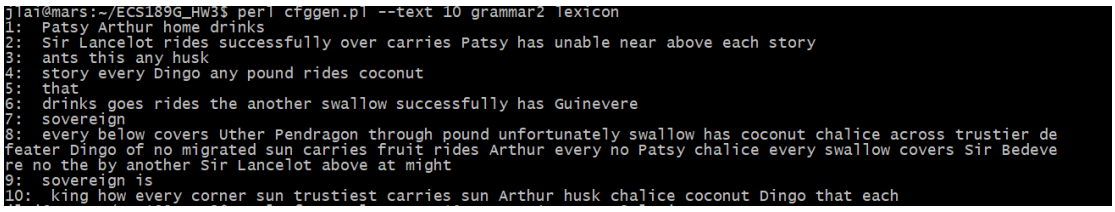


When running the code with only gramma1, it can only parse the first two sentences successfully but the rest cannot parse. When running the code with also gramma2, all sentences could be parsed successfully.

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 few rules for these while the grammer2 has Misc tag. So there are fewer lexicon rules and recursive grammar rules in the garmmar1 file. Hence merging with grammr2, all of the sentences could be parsed successfully.

Task 3





```
jlai@mars:~/ECS189G_HW3$ perl cfggen.pl --text 10 grammar1 grammar2 lexicon
1:  every quest has this master .
2:  each quest across every chalice has another quest .
3:  each land is another sun .
4:  Sir Lancelot is a land .
5:  each husk carries Zoot .
6:  a chalice has Guinevere .
7:  each chalice carries that sovereign .
8:  the winter has each swallow .
9:  no home at the sovereign carries each fruit .
10:  this king is each husk .
```

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. The first step is to create a new set of rules and probabilities for the non-terminal transitions. I used the table example in the book while the probabilities were adjusted to try and make more coherent sentences given the word list. The grammar2 provided was incorporated to ensure every possible sentence could be parsed even without any logical sense.
The second step is to create a lexicon from the word list. I used the lexicon given as a base line and add non-terminals to allow more options for combinations of words. I also added the adjectives and pronouns.

Task 5

There are 11 sentences are grammatically correct in the 20 sentences.