Assignment 5

201014560 & 057931354

Question 1

(a)

NP -> NP PP

"every delicious fine floor under every sandwich on the chief of staff with a sandwich **pickled** every delicious sandwich with president with the delicious pickle ."

The long sequence has the form of multiple nouns joined by preposions. More specifically, NP is expanded to Noun and PP is expanded to Prep (under/on) and again to NP. We see these long sequence often since the probability to this rule (NP -> NP PP) is equal to the probability of NP -> Det Noun.

(b)

Noun -> Adj Noun

Noun -> president | sandwitch | pickle | chief of staff | floor

Probability of 1 / 6 * 1 / 6 two adjctive followed by a Noun ("the the fine perplexed pickle")

(c)

For the long sentences problem – change the proportions of the weights between NP -> NP PP and NP -> Det Noun by giving more weight to last rule.

Specifically we gave 0.8 to NP -> Det Noun and 0.2 to NP -> NP PP

Here is an example of produced sentence:

"the pickle on the president ate a sandwich . "

For the proble of hardly producing multiple adjectives – here we changed the proportions of the weights between the different Noun rules. We gave 0.5 to Noun -> Adj Noun and 0.1 to each terminal rule.

Here is an example of produced sentence:

"a perplexed sandwich under a delicious fine chief of staff **kissed** every fine delicious pickled sandwich . "

Question 3

We implemented cfg to cnf conversion (PCFG.to_cnf()).

To run it:

```
python generate.py grammar2.txt –cnf grammar2-CNF.txt
```

and then we feed it to CKY:

```
python cky.py grammar2-CNF.txt sents.txt
```

Since even the original cfg is not CNF *("is it true that* the president ate a sandwich?"), we ensured that the resulted CNF supports the original form as well as the extended form.

In our conversion to CNF we handle **unit rules** as well. This obviously allows us to define in our cfg grammar unit rules.

Question 3 first Part (a)

a – *Sally ate a sandwich .*
We added NNP -> Sally and NP -> Sally rules. Note that Sally cannot be a Noun since Noun is preceded by a Det

b - *Sally and the president wanted and ate a sandwich .*
We added NP -> NP CC NP and VP -> VP CC VP and CC -> and to support coordination.

c – *the president sighed .*
Here we had to distinguish between intransitive verb (sighed) and transitive verbs (in fact all verbs in the original grammar are transitive, that is, they accept NP). We added VP -> Vi , VP -> Vt NP Vi -> sighed , Vt -> ate ... Note, that verb transitive can behave sometimes as verb intransitive (we can simply say "he thought" or "the president wanted and ate the sandwich" in which "wanted" does not request NP). Therefore we add additional rules: VP -> ViVt CC VP , ViVt -> Vi , ViVt -> Vt and even VP -> Vi

d – "the president thought that a sandwich sighed ."

Here we added the rules SBAR -> COMP S and COMP -> that , VP -> Vt SBAR
The SBAR is expanded to that and a sentence through the sentence is subordinated to main sentence (the president thought).

We ensured that the sentence "the president thought that a sandwich sighed a pickle ." since sighed is intransitive verb.

e - it perplexed the president that a sandwich ate Sally .
Here "that" follows NP (the president). We added NP -> NP SBAR. In addition we added PRP -> it and NP -> PRP (personal pronoun)
The "perplexed" is treated as a (transitive) verb.

f - the very very very perplexed president ate a sandwich .
Note that "very" is an adverb though it precedes aan adjective (perplexed). In fact advervs can "modify" both verbs and adjectives. We had simply to add Adv -> very and Noun -> Adv Noun

g - the president worked on every proposal on the desk .

Here we add the rule VP -> ViVt PP to be able to parse "on every proposal on the desck". Note that "on every proposal on desc" is by PP -> Prep NP and NP -> NP PP and again PP -> Prep NP. Note also that the verb can be either transitive or intransitive. In our sentence worked is defined as verb transitive but it does not accept a Noun

h – Sally is lazy .

Here we treated "is" as Copula, that is, it "makes" a relation between a subject and a predicate (Noun and an Adjective)
We added a rule S -> NP Copula Adj

i - Sally is eating a sandwich .

here the "is" is treated as a verb, third person singular person and "eating" is deined as VBG Verb, present participle. In short we added these rules:
VBZ -> is , VBG_i -> eating and VP -> VBZ VBG_t NP and VP -> VBZ VBG_i
As can be seen, we tried to continue distinguishing between transitive and intransitive verbs.

j - the president thought that Sally is a sandwich .

we have seen already the SBAR expression. The difference here is that the subordinated sentence is of the form "something is something". We treated "is" as Copula and added another rule: S -> NP Copula NP

Question 3 second Part (b)

Relative clauses:

the pickle kissed the president that ate the sandwich .

Here the subordinated sentence (clause) is not really a sentence since it starts with a verb (ate). The missing noun of course is "the president". We added the rule SBAR -> COMP VP

the pickle kissed the sandwich that the president ate

this is similar to e in the previou section however here the transitive verbe (ate) is not followed by NP. As was remarked earlier, we added VP -> Vt and gave it a small probability (though honestly, we are not sure whether it is more likely to see transitive verbs with nouns than without).

the pickle kissed the sandwich that the president thought that Sally ate .

Here we have multiple subordinated clauses (or relative clauses). The rules: NP -> NP SBAR and SBAR -> COMP S handle such sentences.

Singular vs. plural agreement

the president chooses the chief of staff

we added the following rules:

S -> Agree
Agree -> NP_singl VP_singl
VP_singl -> VBZ NP
VP_singl -> VBZ PP
NP _singl -> NP_singl PP
NP -> _singl Det Noun
NP ->_singl NNP

the citizens choose the president

We added the following rules:

S -> Agree

Agree -> NP_plur VP_plur

VP_plur -> VBP NP

VP_plur -> VBP PP

NP_plur -> NP_plur PP

NP_plur -> Det Noun_plur

NP_plur -> NP_singl CC NP_singl

Noun_plur -> NNS

Nouns_plur -> Adj Noun_plur

NNS -> citizens

VBP -> choose

the president and the chief of staff choose the sandwich

this sentence is covered by the rule above NP_plur -> NP_singl CC NP_singl