Assignment2 – Answers

Dorin Keshales – 313298424

Part 1 | Weights

1. The program generates so many long sentences, since it has some recursive grammar rules, which are :

   - `1 NP → NP PP`
   - `1 PP → Prep NP`

   For example, these rules can generate the following:
   `NP → NP PP → NP PP PP → NP PP PP PP → NP PP PP PP PP → …`
   which can lead to Maximum recursion depth runtime error.

   Specifically, the grammar rule which is responsible for that phenomena is '`NP → NP PP`', since every derivation start from the '`S → NP VP`' rule and then in a 50% chance the generator chooses the '`NP → NP PP`' rule. If so, at this point a recursion between this rule and the '`PP → Prep NP`' rule most likely to occur , making the '`PP → Prep NP`' the only source from which the recursion can occur.

2. Sentences with multiple adjectives are generated so rarely, because the rule '`Noun → Adj Noun`', which is a recursive rule, is one of 6 optional rules for `Noun`. All of the rules for `Noun` has the same weight (equals 1) making the probability for the generator to pick the recursive rule is only 1/6. That is, when generating `Noun`, the generator will pick the recursive rule 1/6 of the times.

3. In order to make the sentences shorter, I gave more weight to the '`NP → Det Noun`' rule i.e. greater probability for the non-recursive rule to be picked by the generator when generating an NP. The rule '`NP → NP PP`' still has a weight of 1, but now the '`NP → Det Noun`' has a weight of 5, making the relative ratio of the NP rules be 5:1, so, when generating an NP, the generator will pick the non-recursive rule, i.e. '`NP → Det Noun`', 5/6 of the times and the recursive rule, i.e. '`NP → NP PP`', 1/6 of the times.

   Similarly, in order to make the adjectives more frequent, I gave more weight to the '`Noun → Adj Noun`' rule - I raised its weight to 5. Making the relative ratio of the Noun rules be 5:1:1:1:1:1, so, when generating a Noun, the generator will pick the recursive rule, i.e. '`Noun → Adj Noun`', 5/10 (=1/2) of the times and each of the other lexical non-recursive rules 1/10 of the times. That is, now the odds for getting more adjectives are 50%.

4. In order to favor a set of more natural sentences I gave more weight to nouns, verbs and adjectives that in my opinion are more likely to appear in sentences of natural language. For example, I gave more weight to nouns such as 'president', 'sandwich' and 'chief of staff', verbs such as 'ate', 'wanted' and 'understood', adjectives like 'fine' and 'delicious'. Also, I gave more weight to sentences ending with a dot than sentences ending with a question mark or exclamation point. All of these changes are in addition to the changes I made in section 3 in order to solve the two problems from sections 1 and 2.

Part 2 | Extending the Grammar

In order to make the grammar more organized tense wise I added the following rules in order to distinguish between using Past Simple tense and Present Progressive tense:

- `S → S_PAST`
- `S → S_PRESENT`

- `S_PAST → NP VP_PAST`
- `S_PRESENT → NP VP_PRES`

Where `S_PAST` is a non-terminal from which we generate a sentence in Past Simple tense and `S_PRESENT` is a non-terminal from which we generate a sentence in Present Progressive tense.

The modifications I've done in the grammar in order to support the types of phenomena illustrated in the given sentences:

a. For sentence a, I added the rule '`NP → Nnp`' In order to derive Proper-nouns names of people and places, such as Sally, as a noun-phrase.

b. For deriving sentence b, I had to support conjunction words in noun-phrases and in verb-phrases. Therefore, I added the rules:

   - `NP → NP Conj NP`
   - `Vpast_trans → Vpast_trans Conj Vpast_trans`

   where `Conj` stands for the conjunction words 'and' and `Vpast_trans` stands for transitive verbs in Past Simple tense such as 'kissed', 'hugged' and etc. In addition, the conjunction word 'and' can also connect sentences, so for supporting this behavior I added the rule '`S → S Conj S`'.

c. For deriving sentence c, I had to support verb-phrases that end with a verb, but not just a regular verb – intransitive verb, which is a verb that is not used with a direct object. Therefore, I added the following rule – '`VP_PAST → Vpast_intrans`', where `Vpast_intrans` stands for intransitive verbs in Past Simple tense such as 'sighed', 'laughed' and etc.

d. For deriving sentence b, I had to support subordinating conjunctions words which in this case it's the word that. These words are used in complex sentences - sentences that have two clauses, one main (or independent) and one subordinate (or dependent). To support that I added the rule '`VP_PAST → Vpast_that SBAR`', where '`Vpast_that`' stands for Past Simple tense verbs that can be followed by the word 'that' (a subordinating conjunction). `SBAR` is a non-terminal I added for deriving the subordinate clause that comes after the word 'that', by adding the following rule '`SBAR → SubC S_PAST`', where `SubC` is an abbreviation for subordinating conjunction words (in this case it represents the word 'that') and `S_PAST` is for deriving a new sentence as the second clause of the main sentence.

e. For sentence e, I added the rule '`S_PAST → Prp Vpast_np_that NP SBAR`' in order to be able to start a sentence with personal pronoun (`Prp`), in this case the word 'it', and then a verb which takes two arguments – a noun-phrase and the verb's cause. This kind of verb is represented by the pre-terminal `Vpast_np_that`. As mentioned, this verb is then followed by an `NP` and the verb's cause, where the verb's cause is basically taking a subordinating conjunction word and a subordinate clause, which can be derived using the `SBAR` rule I presented in section d.

f. For sentence f, I added the following rules in order to support adverbial-phrases (`ADVP`) and adjectival-phrases (`ADJP`):

   - `ADVP → Adv`
   - `ADVP → Adv ADVP`
   - `ADJP → ADVP Adj`
   - `ADJP → Adj`

   Then, I added the following rules in order to derive a sentence like sentence f and even with a prepositional phrase followed the noun:

   - `NP → Det ADJP Noun`
   - `NP → Det ADJP Noun PP`

g. For deriving sentence g, I had to support verb-phrases of verb followed by a prepositional-phase. Therefore, I added the following rule '`VP_PAST → Vpast_prep PP`' , where `Vpast_prep` stands for Past Simple tense verbs that prepositional phrase as an argument such as 'worked'.

h. For deriving sentences like sentence h, I had to support simple Present Progressive tense sentences with adjectival phrase (for deriving sentence h only one adjective is enough, but I extended it to support multiple adjectives and even with an option to include adverbial phrase). I added the following rules for supporting this kind of sentences:

   - `VP_PRES → Vbz ADJP`
   - `VP_PRES → Vbz ADJP Conj ADJP`

Where `Vbz` is a pre-terminal represents verb 3rd person singular present form words like the verb 'is'.

i. For deriving sentence i, I added the rule '`VP_PRES → Vbz Vpresent NP`', where `Vpresent` is a pre-terminal which represents verbs in Present Progressive tense.

j. For deriving sentence j, I added the rule '`VP_PAST → Vbz NP`', but also the rule '`VP_PRES → Vbz NP`' in order to support the second clause of the sentence (the one that starts after the word 'that') as a sentence in itself.

Question: Furthermore, note that handling sentences (b) and (h)/(i) can interact in a bad way, to create ungrammatical sentences. You do not need to solve this issue in this part of the assignment, but you do need to discuss it and explain what the problem is, using an example and a short explanation.

Answer: Handling sentences (b) and (h)/(i) can interact in a bad way, to create ungrammatical sentences, since that in sentence (b) I added rules which allows me to recursively concatenate noun-phrases using the conjunction word 'and' for deriving a new noun-phrase, which might cause interact in a bad way when we'll try to generate sentences like sentence (h) or sentence (i). For example, the generator will generate sentence like:

```
'Sally and the president is fine'

'Sally and the president is eating a sandwich'
```

Instead of:

```
'Sally and the president are fine'

'Sally and the president are eating a sandwich'
```

** For my convenience and for variety on the generated sentences I have added to the vocabulary more proper nouns, nouns, adjectives and transitive and intransitive Past Simple tense verbs.

Part 4 | Additional Linguistic Structures

I chose the additional linguistic phenomena of Yes-No questions (b) and WH-word questions (d).

(b) Yes-No questions

The auxiliary verbs I supported : `did`, `will` and `is`.
I'll briefly discuss how I extended the grammar from part 2 to handle Yes-No questions.

First of all, in Past Simple tense when making questions the verb returns to its base form. Therefore, I had to add to the dictionary the base form of all the Past Simple verbs I had in grammar2. In Future Simple tense, we already use the base form of the verb, so there was no need to add particular verbs for this matter.

I used the following derivation rules for deriving both questions in Past Simple tense and questions in Future Simple tense:

- `ROOT → YNQ ?`
- `YNQ → YNQ_DID_WILL`
- `YNQ_DID_WILL → Aux NP VPQ`
- `VPQ → Vbase_trans NP` i.e. "did Sally ___ (and ___) the sandwich ?"
- `VPQ → Vbase_prep PP` i.e. "did Sally ___ on the desk ?"
- `VPQ → Vbase_that SBAR` i.e. "did Sally ___ that ... ?"
- `VPQ → Vbase_intrans` i.e. "did Sally ___ ?"
- `Vbase_trans → Vbase_trans Conj Vbase_trans` i.e. "eat and understand and..."
- `YNQ_DID_WILL → Aux Prp Vbase_np_that NP SBAR` i.e. "did it perplex the president that ... ?"

where:

- `YNQ_DID_WILL` – is a non-terminal for deriving Yes-No questions in Past Simple and in Future simple.
- `Aux` – is a pre-terminal represents the auxiliary verbs 'did' and 'will'.
- `Vbase_trans` – is a pre-terminal that represents base form verbs which are transitive.
- `Vbase_prep` – is a pre-terminal that represents base form verbs followed by a prepositional phrase.
- `Vbase_that` – is a pre-terminal which represents base form verbs that are followed by subordinating conjunction word such as 'that'.
- `Vbase_intrans` – is a pre-terminal which represents base form verbs which are intransitive.
- `Vbase_np_that` – is a pre-terminal which represents base form verbs that take 2 arguments - a noun-phrase and the verb's cause.

For deriving questions in Present Progressive tense, I had to make other rules, because in Present-Progressive questions the verb is not in its base form, but with an 'ing' extension. So, I used the following derivation rules for deriving questions in Present Progressive tense:

- `YNQ → YNQ_IS`

- `YNQ_IS → Vbz NP NP` i.e. "is Sally a sandwich ?"
- `YNQ_IS → Vbz NP ADJP` i.e. "is Sally (very) funny ?"
- `YNQ_IS → Vbz NP ADJP Conj ADJP` i.e. "is Sally (very) funny and (very) cynical ?
- `YNQ_IS → Vbz NP Vpresent NP` i.e. "is Sally eating the sandwich ?"

where:

- `YNQ_IS` – is a non-terminal for deriving Yes-No questions in Present Progressive.
- `Vbz` – is a pre-terminal which represents verb 3rd person singular present form such as 'is'.
- `Vpresent` – is a pre-terminal which represents Present Progressive tense verbs.

Examples of generated Yes-no questions:

- "did chandler work with rachel ?"
- "is rachel eating ross ?"
- "did monica eat ?"
- "will monica play with chandler ?"
- "will chandler sigh ?"
- "is chandler lazy ?"
- "did chandler buy a pizza ?"
- "is rachel funny ?"
- "is phoebe eating the loyal cynical president ?"
- "is a cynical president hugging ross ?"
- "will the sandwich eat with the delicious president ?"
- "will the pickle think that rachel understood a president ?"
- "is monica a delicious funny pickle ?"
- "will it perplex the very loyal sandwich that joey ate ?"
- "will it perplex chandler that monica hugged a floor ?"
- "did chandler laugh ?"
- "is a doctor kissing chandler ?"
- "will joey understand that it perplexed a cynical pizza that ross laughed ?"
- "did it perplex joey that joey kissed rachel ?"
- "did ross understand that it perplexed joey that joey laughed ?"
- "did the president understand that phoebe laughed ?"

(d) WH-word questions

The Wh-words I supported in are `where`, `what` and `who`, using the auxiliary verbs I supported in the Yes-no questions section. I'll briefly discuss how I extended the grammar to handle WH-word questions.

Supporting the Yes-no questions of section (b) helped me here – I reused some of the rules, but I also had to add new ones since for each Wh-word different kind of questions can be asked. As in the Yes-no questions section, I was able to use the same rules to derivate both Past Simple tense questions and Future Simple tense questions. As given from the Yes-no questions section, for the auxiliary verb 'is' I had to derivate other rules in order to support WH questions.

The rules for the WH-word **`where`**:

- `ROOT → WHQ?`

- `WHQ → where YNQ_DID_WILL` `# For Past Simple and Future Simple #`

`# For Present Progressive #`
- `WHQ → where Vbz NP` i.e. "where is Sally ?"
- `WHQ → where Vbz NP Vpresent` i.e. "where is Sally eating ?"
- `WHQ → where Vbz NP Vpresent NP` i.e. "where is Sally eating the sandwich ?"

The rules for the WH-word **`what`**:

- `WHQ → what Aux NP VPQ1` `# For Past Simple and Future Simple #`

- `VPQ1 → Vbase_trans NP with` i.e. "what did Sally ___ the sandwich with ?"
- `VPQ1 → Vbase_trans` i.e. "what did Sally ___ ?"
- `VPQ1 → Vbase_that` i.e. "what did the president think/understand ?"
- `VPQ1 → Vbase_prep Prep` i.e. "what did the president ___ on ?"
- `VPQ1 → Vbase_that SubC NP Vpast_tran` i.e. "what did the president ___ that Sally ___ ?"

`# For Present Progressive #`
- `WHQ → what Vbz NP Vpresent` i.e. "what is Sally ___ eating ?"
- `WHQ → what Vbz NP` i.e. "what is the sandwich ?" (Answer: a pickle)*

where:

- `SubC` – is a pre-terminal that represents conjunction words.
- `Vpast_tran` – is a pre-terminal that represents transitive verbs in Past Simple tense.

* Grammatically, when asking questions on the SUBJECT we use `who` for people and `what` for things.

The rules for the WH-word **who**:

```
# For Past Simple and Future Simple #
```

- `WHQ → who Aux NP Vbase_prep with` i.e. "who did the president ___ with ?"
- `WHQ → who Aux NP Vbase_trans` i.e. "who did the president ___ ?"
- `WHQ → who VPQ2`

- `VPQ2 → Vpast_trans NP` i.e. "who ___ (and ___) the sandwich ?"
- `VPQ2 → Vpast_prep PP` i.e. "who ___ on the desk ?"
- `VPQ2 → Vpast_that SBAR` i.e. "who thought/understood that ... ?"
- `VPQ2 → Vpast_intrans` i.e. "who ___ ?"

```
# For Present Progressive #
```

- `WHQ → who Vbz Vpresent` i.e. "who is eating ? "
- `WHQ → who Vbz Vpresent NP` i.e. "who is eating the sandwich ?"
- `WHQ → who Vbz ADJP` i.e. "who is funny ?"
- `WHQ → who Vbz ADJP Conj ADJP` i.e. "who is funny and cynical ?"
- `WHQ → who Vbz NP` i.e. "who is the president ?"

where:

- `Vpast_prep` – is a pre-terminal which represents Past Simple verbs followed by a prepositional phrase.
- `Vpast_that` – is a pre-terminal which represents Past Simple verbs that are followed by subordinating conjunction word such as 'that'.
- `Vpast_intrans` – is a pre-terminal which represents Past Simple verbs which are intransitive.

Overall, For the WH questions I had to add more rules than for the Yes-no questions, but I really tried to keep it to a minimum.

Examples of generated WH questions:

- "what will a proposal think that sally pickled ?"
- "who will rachel eat with ?"
- "who played with joey ?"
- "where is phoebe ?"
- "where is phoebe eating a funny pickle ?"
- "who is cynical ?"
- "who is kissing ross ?"
- "what is the pickle hugging ?"
- "who is very funny and very loyal ?"
- "where is every delicious pizza ?"
- "who did a very delicious president kiss ?"
- "who thought that it perplexed monica that ross is a floor ?"
- "what did a sandwich think ?"
- "what is monica and ross eating ?"

- “who is kissing a president ?”
- “who kissed monica and a very very very pickled pizza ?”
- “who did the pickle hug ?”
- “where is chandler kissing rachel ?”
- “where did sally work ?”
- “who thought that chandler is a pickle ?”
- “where did rachel laugh ?”
- “where did ross pickle and hug the pizza ?”
- “where did the chief of staff eat the cynical pickle ?”
- “what will a loyal sandwich play with ?”
- “what did every delicious sandwich buy the floor with ?”
- “where is the very funny president ?”
- “what did phoebe understand ?”

Part 5 | Extra

In this section I have expanded the grammar in some interesting ways:

a. First thing first, I fixed the problem from part 2 regarding handling sentences (b) and (h)/(i) that can interact in a bad way, to create ungrammatical sentences. Now my grammar fully supports plural form, as the auxiliary verb is generated according to the noun-phrase form that was generated before it. In the process I also added plural nouns and plural determiners to the vocabulary, so now I get richer sentences with more meaning that are much like sentences in natural language. To support this change I had to distinguish between singular noun-phrases to plural noun-phrases which made me to duplicate some rules so I can have full separation between singular noun-phrases to plural noun-phrases and then to also distinguish between verb-phrases that fit to singular noun-phrases to verb-phrases that fit to plural noun-phrases.

   For example:
   - “some lecturers played in monica .”
   - “the loyal teachers played with joey !”
   - “ross and all pickles are very cynical and lazy !”
   - “these very very cynical pickles are cynical !”
   - “it perplexed the lecturers that all pickled doctors are those sandwiches .”

b. Then, I extended my grammar to support Singular vs. Plural agreement in Present Simple tense (the linguistic phenomenon presented in Part 4 option e). Since most of the big changes were already made before, I only had to duplicate few rules just to replace the Past Simple tense verbs with the Present Simple tense verbs.

   For example:
   - “it perplexes a loyal delicious proposal that rachel is every funny sandwich !”
   - “monica pickles some cynical doctors !”
   - “all very funny presidents think that ross is chandler .”

- “is it true that phoebe laughs ?”
- “these cynical citizens eat .”
- “those loyal lecturers sigh .”

c. After that, I took the linguistic phenomena I supported in Part 4 – Yes-no questions and WH-word questions – and extended them to also support Present Simple tense questions using the auxiliary verbs ‘do’ and ‘does’. In addition, I also extended them to support plural Present Progressive tense questions using the auxiliary verb ‘are’. It was pretty easy to do it since for supporting Present Simple tense questions I mostly used exiting derivation rules and added just few ones and for supporting Present Progressive tense questions I already had a ready infrastructure following the support of Present Progressive tense questions with the auxiliary verb 'is'.

   For example:
   - “does joey play under a doctor ?”
   - “who does the cynical proposal work with ?”
   - “who does every president work with ?”
   - “does it perplex those very loyal doctors that phoebe smiles ?”
   - “are some lecturers the citizens ?”
   - “where are all doctors kissing these students ?”
   - “do these citizens eat with sally ?”
   - “where are the very fine students ?”
   - “do all teachers think that ross is joey ?”
   - “who smiles ?”
   - “who eats some pickles ?”

d. After that, I made some QA in an interview format, where Yes-No questions and WH questions are generated with their possible answer, using some few new derivation rules. For the Yes-No questions, I tried to make it a little more interesting by answering in a changing level of certainty.

   For example:
   - “Q: are these fine citizens very pickled ? Ans: yes, absolutely !”
   - “Q: does it perplex joey that ross smiled ? Ans: yes, of course .”
   - “Q: is chandler very very cynical and delicious ? Ans: yes, absolutely !”
   - “Q: did rachel and every president work under a loyal chief of staff ? Ans: yes .”
   - “Q: is ross every sandwich ? Ans: definitely not !”
   - “Q: are all funny presidents some very very cynical teachers ? Ans: no !”
   - “Q: will it perplex all very loyal sandwiches on the floor that joey is every very cynical doctor ? Ans: of course not !”
   - “Q: who is funny and lazy ? Ans: those very very cynical citizens .”
   - “Q: what does joey think ? Ans: that those perplexed pickles and a sandwich and the chief of staff are all cynical presidents under phoebe !”
   - “Q: what does chandler think ? Ans: that those very cynical students under monica are all students !”
   - “Q: who will some very funny pickles play with ? Ans: with monica !”
   - “Q: where is ross hugging all doctors ? Ans: in the playground .”

- “Q: what will some teachers understand ? Ans: that every doctor ate the sandwich!”
- “Q: who will some very delicious lecturers play with ? Ans: with the cynical president.”
- “Q: where did it perplex ross that every sandwich played with rachel ? Ans: in italy.”
- “Q: what will rachel think ? Ans: that these very funny students are all cynical teachers .”

e. In the spirit of the interview format, I also created a format for news reports in which I took some popular TV news reporting sentences and made them use part of my grammar in order to generate some funny and interesting live news.

For example:

- “Stay with us, we'll be right back after this commercial of all citizens eating joey .”
- “Our sources tell us that a very funny president worked , stay with us for more information .”
- “We've got some important news for you: monica laughed .”
- “Stay with us, we'll be right back after this commercial of a pickle hugging every pizza .”
- “This is chandler , reporting from paris that all students and every desk understood and hugged a floor .”
- “We are very sorry to announce that joey and the cynical teachers laughed , stay with us for more information .”
- “We are very glad to announce that it perplexed ross that a proposal is rachel , join us live on ' loyal cynical floor ' channel .”
- “And now , we are about to tell you on monica hugging rachel and phoebe , stay with us for more information .”
- “We are very glad to announce that those perplexed lecturers and phoebe pickled rachel , join us live on ' proposal ' channel .”
- “We've got some important news for you: those very very delicious pickles ate joey.”
- “We have some breaking news reaching us here at ' chief of staff ' News, we're receiving reports that monica played in chandler .”