Q1. What are some advantages of modeling French grammar with a CFG?
The first advantage is that we are able to easily model the basic grammars in French, such as the number and gender matching, and the subject-verb agreement. We could simply list all possible combinations, similar to what I did in my CFG. The second advantage is that the grammar of French is much preciser than that of English since it is artificially defined. In another word, there are a few edge cases that don't fit in the regular grammar of French. As a consequence, we could easily build a CFG for the French. The third advantage is that after modeling French with a CFG, we are able to do lots of NLP works related to French. Considering there are a lot of famous literature written in French, the NLP works related to French are really meaningful.

Q2. What are some disadvantages of modeling French grammar with a CFG?
There are three main disadvantages of modeling French grammar with a CFG. The first one is that it's hard for us to consider slang and foreign words. The reason is that they may not fit in the regular grammar of French, so we have to create separate rules for them in our CFG. The second point is that there are lots of lexical forms in my CFG. For verbs, adjectives, nouns and etc., we all need to consider the number and gender. As a consequence, there are several lexical forms in the CFG in order to distinguish them. The third disadvantage is about the contractions and capitalizations in French. In this assignment, we just simply ignore these two things, but they would be big problems when modeling all French grammars.

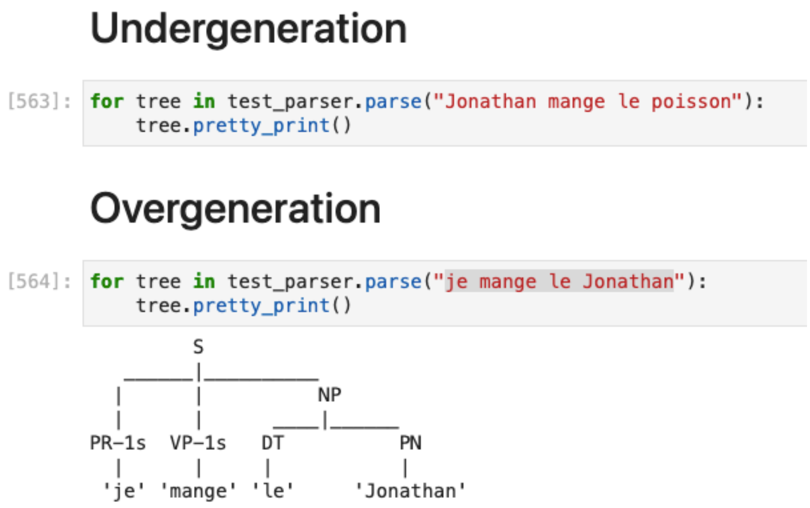
Q3. What are some aspects of French grammar that your CFG does not handle?
One of the most important parts we don't handle is the Subject-Link Verb-Predicative structure, which is an essential part of French. We need to match the noun and the pronoun in number and gender. For example, there might be two versions of a sentence: "Je suis chinois" (the person who is saying this sentence is male) and "Je suis chinoise" (the person who is saying this sentence is female). There is no sample sentence with Subject-Link Verb-Predicative structure in our assignment, so my CFG does not support this part. Moreover, in my CFG, I don't support the adjectival possessive pronouns, like "mon", "ma", "mes", "ton", "ta", "tes" etc. This part is more sophisticated than that of English as well. The adjectival possessive pronouns should be matched with the following noun. For example, we say "mon petit ami" when talking about boyfriends. In contrast, we use "ma petite amie" to express girlfriends. Additionally, we don't consider any complex sentence structures. For example, in "Les boutiques et les cafés que vous avez vus attirent beaucoup de monde.", we are using "que" to construct a complex sentence, as same as using "that" in English. In addition to the three aspects I mentioned above, there are still many other aspects of French grammar that I don't consider when constructing my CFG, like tense in French. Consequently, my CFG could be improved further.

Examples in instruction :
Je regarde la télévision
S -> PR-1s VP-1s NP, NP -> DT-fs N-fs
By using these two rules, we could parse this sentence.

Il la regarde
S -> PR-3s DT VP-t-3s
By using this rule, we could parse this sentence.

Je mange les
S -> PR-1s VP-1s NP, NP -> DT-p NPp
Except the determiner, we also need the noun word as well. This is why my CFG can reject this invalid sentence.

Undergeneration and Overgeneration :



Undergeneration

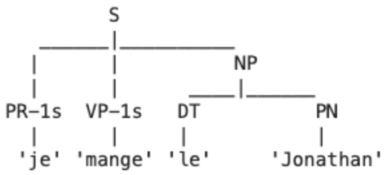
```
[563]: for tree in test_parser.parse("Jonathan mange le poisson"):
           tree.pretty_print()
```

Overgeneration

```
[564]: for tree in test_parser.parse("je mange le Jonathan"):
           tree.pretty_print()
```

```
             S
     ________|___________
    |        |           NP
    |        |       ____|______
  PR-1s    VP-1s   DT          PN
    |        |     |           |
   'je'   'mange' 'le'     'Jonathan'
```

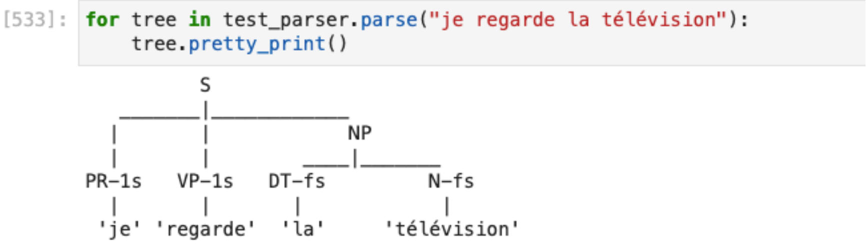
Undergeneration:
Jonathan mange le poisson
In this case, we should consider the proper noun “Jonathan” as the subject. However, in my CFG, I missed the rule to accept this valid sentence, which is a case of undergeneration. A valid sentence is rejected by my CFG.
Overgeneration:
je mange le Jonathan
This could be parsed as the following tree:



```
             S
     ________|___________
    |        |           NP
    |        |       ____|______
  PR-1s    VP-1s   DT          PN
    |        |     |           |
   'je'   'mange' 'le'     'Jonathan'
```

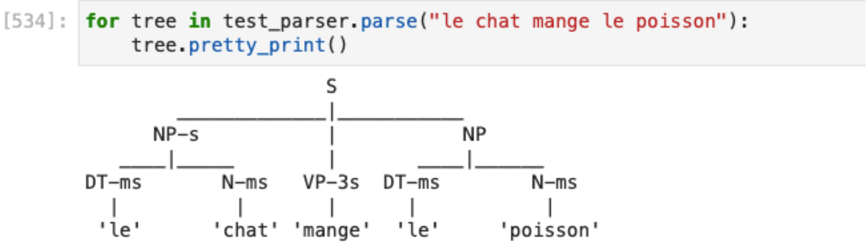
Even though this is an invalid sentence(since we should not have ‘le’ before ‘Jonathan’), my CFG still accepts this sentence, which is a case of overgeneration. An ungrammatical sentence has been generated.

Parse all sample sentences by using self-implemented CYK parser:



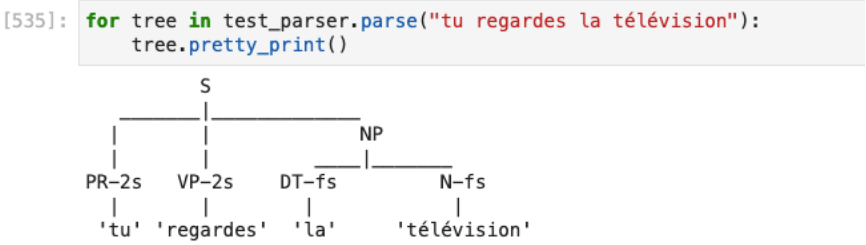
```
[533]: for tree in test_parser.parse("je regarde la télévision"):
           tree.pretty_print()
```

```
          S
   _______|____________
  |       |            NP
  |       |        ____|_______
PR-1s   VP-1s   DT-fs         N-fs
  |       |       |            |
 'je' 'regarde'  'la'     'télévision'
```



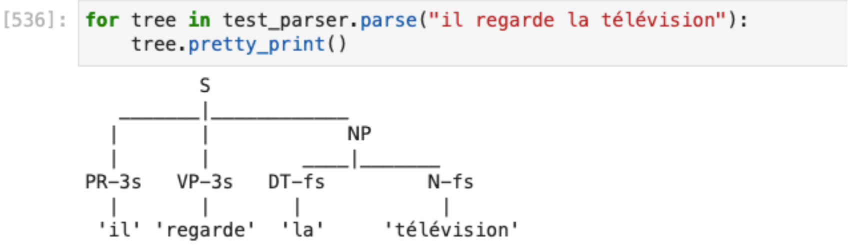
```
[534]: for tree in test_parser.parse("le chat mange le poisson"):
           tree.pretty_print()
```

```
                     S
        _____________|___________
      NP-s           |           NP
   ____|_____        |       ____|______
DT-ms       N-ms   VP-3s  DT-ms        N-ms
  |          |       |      |           |
 'le'      'chat' 'mange'  'le'     'poisson'
```



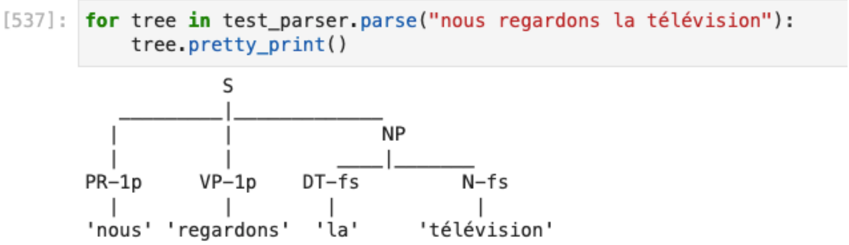
```
[535]: for tree in test_parser.parse("tu regardes la télévision"):
           tree.pretty_print()
```

```
          S
   _______|_____________
  |       |             NP
  |       |         ____|_______
PR-2s   VP-2s    DT-fs         N-fs
  |       |        |            |
 'tu' 'regardes'  'la'     'télévision'
```



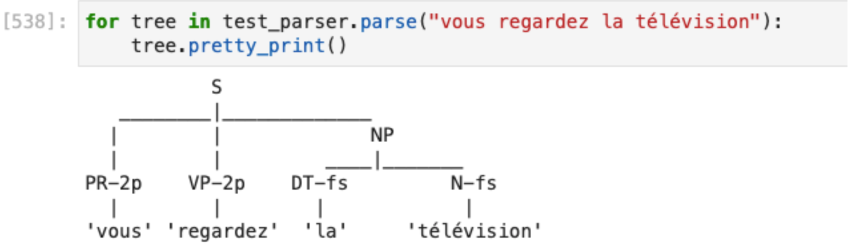
```
[536]: for tree in test_parser.parse("il regarde la télévision"):
           tree.pretty_print()
```

```
          S
   _______|____________
  |       |            NP
  |       |        ____|_______
PR-3s   VP-3s   DT-fs         N-fs
  |       |       |            |
 'il' 'regarde'  'la'     'télévision'
```



```
[537]: for tree in test_parser.parse("nous regardons la télévision"):
           tree.pretty_print()
```

```
            S
   _________|_____________
  |         |             NP
  |         |         ____|_______
PR-1p     VP-1p    DT-fs         N-fs
  |         |        |            |
'nous' 'regardons'  'la'     'télévision'
```



```
[538]: for tree in test_parser.parse("vous regardez la télévision"):
           tree.pretty_print()
```

```
           S
   ________|_____________
  |        |             NP
  |        |         ____|_______
PR-2p    VP-2p    DT-fs         N-fs
  |        |        |            |
'vous' 'regardez'  'la'     'télévision'
```

```
[539]: for tree in test_parser.parse("ils regardent la télévision"):
           tree.pretty_print()
```

```
           S
   ________|______________
  |        |             NP
  |        |         ____|_______
PR-3p    VP-3p    DT-fs         N-fs
  |        |        |            |
'ils' 'regardent'  'la'     'télévision'
```

```
[540]: for tree in test_parser.parse("tu ne regardes pas la télévision"):
           tree.pretty_print()
```

```
               S
   ____________|___________________
  |    |       |        |          NP
  |    |       |        |      ____|_______
PR-2s NEG    VP-2s     NEG  DT-fs         N-fs
  |    |       |        |     |            |
 'tu' 'ne' 'regardes' 'pas'  'la'     'télévision'
```

```
[541]: for tree in test_parser.parse("le chat ne mange pas le poisson"):
           tree.pretty_print()
```

```
                          S
        __________________|_________________
      NP-s         |      |      |          NP
   ____|_____      |      |      |      ____|______
DT-ms      N-ms  NEG   VP-3s   NEG   DT-ms        N-ms
  |          |     |      |      |     |           |
 'le'      'chat' 'ne' 'mange' 'pas'  'le'      'poisson'
```

```
[542]: for tree in test_parser.parse("le chat"):
           tree.pretty_print()
```

```
       S
   ____|____
DT-ms      N-ms
  |         |
 'le'     'chat'
```

[543]:

```
for tree in test_parser.parse("la télévision"):
    tree.pretty_print()
```

```
       S
   ____|_______
DT-fs         N-fs
  |            |
 'la'     'télévision'
```

[544]:

```
for tree in test_parser.parse("les chats"):
    tree.pretty_print()
```

```
       S
   ____|_____
 DT-p       N-mp
  |          |
'les'     'chats'
```

[545]:

```
for tree in test_parser.parse("les télévisions"):
    tree.pretty_print()
```

```
       S
   ____|________
 DT-p          N-fp
  |             |
'les'     'télévisions'
```

[546]:

```
for tree in test_parser.parse("Jonathan"):
    tree.pretty_print()
```

```
    S
    |
'Jonathan'
```

[547]:

```
for tree in test_parser.parse("Montréal"):
    tree.pretty_print()
```

```
    S
    |
'Montréal'
```

[548]:

```
for tree in test_parser.parse("le Canada"):
    tree.pretty_print()
```

```
      S
  ____|_____
 DT         PN
 |          |
'le'     'Canada'
```

[549]:

```
for tree in test_parser.parse("il la regarde"):
    tree.pretty_print()
```

```
       S
   ____|_______
PR-3s  DT   VP-t-3s
  |    |       |
 'il' 'la' 'regarde'
```

[550]:

```
for tree in test_parser.parse("le chat noir"):
    tree.pretty_print()
```

```
        S
   _____|________
DT-ms  N-ms  A-ms-post
  |     |        |
 'le' 'chat'   'noir'
```

[551]:

```
for tree in test_parser.parse("le chat heureux"):
    tree.pretty_print()
```

```
        S
   _____|________
DT-ms  N-ms  A-ms-post
  |     |        |
 'le' 'chat' 'heureux'
```

[552]:

```
for tree in test_parser.parse("le beau chat"):
    tree.pretty_print()
```

```
         S
   ______|_______
DT-ms A-ms-pre  N-ms
  |      |       |
 'le'  'beau'  'chat'
```

[553]:

```
for tree in test_parser.parse("le joli chat"):
    tree.pretty_print()
```

```
         S
   ______|_______
DT-ms A-ms-pre  N-ms
  |      |       |
 'le'  'joli'  'chat'
```

[554]:

```
for tree in test_parser.parse("la dernière semaine"):
    tree.pretty_print()
```

```
          S
   _______|__________
DT-fs  A-fs-pre     N-fs
  |       |          |
 'la' 'dernière' 'semaine'
```

[555]:

```
for tree in test_parser.parse("la semaine dernière"):
    tree.pretty_print()
```

```
          S
   _______|_________
DT-fs    N-fs   A-fs-post
  |       |         |
 'la' 'semaine' 'dernière'
```

[556]:

```
for tree in test_parser.parse("les chats noirs"):
    tree.pretty_print()
```

```
         S
   ______|________
 DT-p   N-mp  A-mp-post
  |      |        |
'les' 'chats'  'noirs'
```

[557]:

```
for tree in test_parser.parse("la télévision noire"):
    tree.pretty_print()
```

```
           S
   ________|___________
DT-fs     N-fs     A-fs-post
  |        |           |
 'la' 'télévision'  'noire'
```

[558]:

```
for tree in test_parser.parse("les télévisions noires"):
    tree.pretty_print()
```

```
            S
   _________|___________
 DT-p      N-fp     A-fp-post
  |         |           |
'les' 'télévisions'  'noires'
```

Cannot be parse

```
[559]: for tree in test_parser.parse("je mangent le poisson"):
           tree.pretty_print()
```

```
[560]: for tree in test_parser.parse("les noirs chats mangent le poisson"):
           tree.pretty_print()
```

```
[561]: for tree in test_parser.parse("la poisson mangent le poisson"):
           tree.pretty_print()
```

```
[562]: for tree in test_parser.parse("je mange les"):
           tree.pretty_print()
```