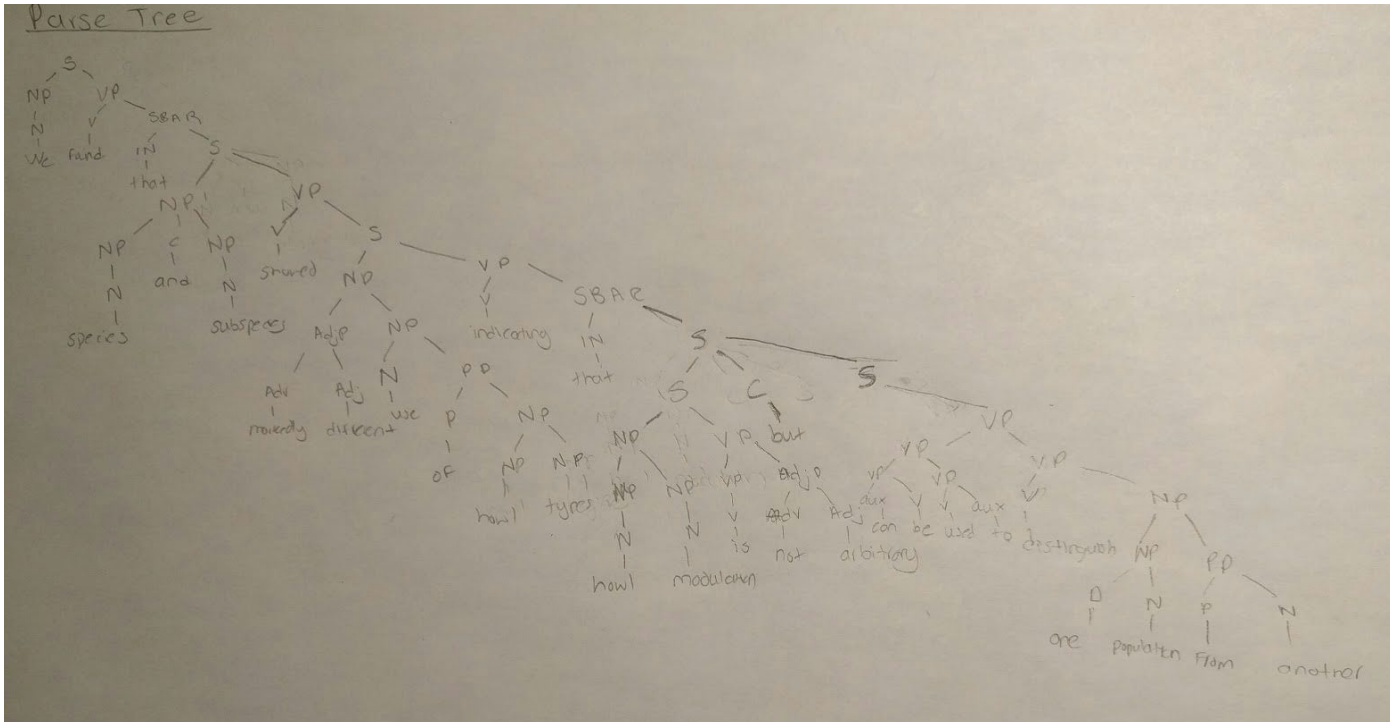
Leslie Manrique
Natural Language Processing

**Manual Parse**




Rules Used:
S -> NP VP | S C S | VP
PP -> P NP
NP -> Det NP | Det N PP | NP C NP | Det Adj N | NP PP | AdjP NP | NP NP | N
AdjP -> Adj | Adv Adj
VP ->|V SBAR| V S| V NP | VP PP | V NP | V AdjP |Aux VP | V VP | V
PP -> P NP
SBAR -> IN S

Difficulties:

There were some difficulties creating this phrase structure tree. This was primarily due to the fact that it was a very long sentence, that could have easily been separated into many sentences and had many conjunctions. I ended up having to create the phrase rule VP -> V S, which allowed me to attach another sentence part. Also, I added conjunctions to account for words like ‘but’ and ‘and’ which create unions between two same parts of speech. ‘That’ wasn’t a regular part of speech, but was used to help join sentences, so I declared it with SBAR - not sure if this was done correctly. Also, I had some trouble deciphering constituents and which part of speech words belonged to. For example, should I treat howl as an adjective or a noun? Turns out, nouns can serve as adjectives to other nouns. Therefore for “howl modulation” I had to write a rule NP -> NP NP, to account for nouns serving as adjectives.

**CFG Parser**

After constant modification, I was able to get my CFG parser to work for the entire sentence. I'm not sure how well i implemented it because after running it, I recieved several different parsings about 1300 with many repeats. Perhaps I could shorten the amount of rules I have, though I'm not sure how to do so at the moment. I pasted two different parsings below. There is a lot of ambiguity, for example In the last two lines of both "one population from another", the parser is not sure if the phrase is attached to a single noun phrase constituent (parse 2) or if it's connected to a verb phrase from before.

1)

```
(S
  (NP We)
  (VP
    (V found)
    (SBAR
      (IN that)
      (S
        (S
          (NP
            (AdjP (Adj different))
            (NP (NP (N species)) (C and) (NP (N subspecies))))
          (VP
            (V showed)
            (S
              (NP
                (AdjP (Adv markedly) (Adj different))
                (NP
                  (NP (NP (N use)) (PP (P of) (NP (N howl))))
                  (NP (N types))))
              (VP
                (V indicating)
                (SBAR
                  (IN that)
                  (S
                    (NP (NP (N howl)) (NP (N modulation)))
                    (VP (V is) (AdjP (Adv not) (Adj arbitrary)))))))))
        (C but)
        (S
          (VP
            (Aux can)
            (VP
              (VP
                (V be)
                (VP
                  (V used)
                  (S
                    (VP
                      (Aux to)
                      (VP
                        (V distinguish)
                        (NP (AdjP (Adj one)) (NP (N population))))))))
              (PP (P from) (NP (N another)))))))))
```

2)

```
(S
 (S
  (NP We)
  (VP
   (V found)
   (SBAR
    (IN that)
    (S
     (NP
      (AdjP (Adj different))
      (NP (NP (N species)) (C and) (NP (N subspecies))))
     (VP
      (V showed)
      (S
       (NP
        (NP
         (AdjP (Adv markedly) (Adj different))
         (NP (N use)))
        (PP (P of) (NP (NP (N howl)) (NP (N types)))))
       (VP
        (V indicating)
        (SBAR
         (IN that)
         (S
          (NP (NP (N howl)) (NP (N modulation)))
          (VP (V is) (AdjP (Adv not) (Adj arbitrary)))))))))))
 (C but)
 (S
  (VP
   (Aux can)
   (VP
    (V be)
    (S
     (VP
      (V used)
      (S
       (VP
        (Aux to)
        (VP
         (V distinguish)
         (NP
          (AdjP (Adj one))
          (NP
           (NP (N population))
           (PP (P from) (NP (N another))))))))))))))
```

**Shift Reduce Parser**

Due to the number of rules I had attached to single parts of speeches, for example NP has 8 ways of parsing, I wasn't able to yield any results due to the many choices available for the parser to make. Once the choice is made, it cannot be undone. The parser made too many wrong choices for it to generate a parse.

**Left Corner Parser**

The left corner parser ran identically to the CFG parser and also had about 1300 results. Instead of processing top down, it first processes the left corner of pre-terminals and finds which rule fits the constituent best. This allowed for faster parsing because it limited the number of matches possible, by first examining pre-terminals.