Homework 2: Formal Languages, Parsing, and Semantics

Toni Kazic

Fall, 2024

1 Introduction

In this homework, the syntactic and semantic rubber hits the road. This homework introduces the deeper structures of language, especially when phrased formally; looks at syntax and parsing; and extends the notion of parsing to semantics.

2 Who's Who and Solution Patterns

2.1 Lead Person: yellow

2.2 Group Members

first name last name	color
Eric Chin	yellow yellow
Ying-Chen Lin	green green
James Wiliams	purple violet

2.3 Three Member Solution Patterns

i is the question number.

color	draft solution	revise solution
greengreen	$i \mod 3 = 1$	$i \mod 3 = 0$
yellowyellow	$i \mod 3 = 2$	$i \mod 3 = 1$
purpleviolet	$i \mod 3 = 0$	$i \mod 3 = 2$

2.4 Two Member Solution Patterns

color	draft solution	revise solution
green green[redacted]	odds	evens
yellow yellow[redacted]	evens	odds

3 General Instructions

- *Fill out the group members table and follow the solution patterns* in Section 2.
- *If the question is unclear, tell me your interpretation of it as part of your answer.* Feel free to ask about the questions in class or on the Slack channel (use `@channel` as others will probably be puzzled too).
- *For questions using corpora, use the corpus of the lead person.*
- *Put your draft answers right after each question using a* ***complete, functional*** `org` *mode code or example block.* Make sure your code block is complete and functional by testing it in your copy of this homework file.
- *Each group member reviews the others' draft solutions and you revise them together.*
- *Discuss each other's draft and reviews, finalizing the answers.*
- *Show all your work: code, results, and analysis.* Does your code work in this file and produce **exactly** the results you show?
- *Post the completed file to Canvas no later than noon on the Tuesday indicated* in the schedule in the syllabus, naming your file with each person's first name (no spaces in the file name, and don't forget the `.org` extension!). Only one person should submit the final file.

4 Hints

4.1 Make sure the structure of the grammar can be parsed by the parser.

For example, a recursive descent parser cannot terminate the parse of a left-recursive grammar.

4.2 Use re-entrancy if you need it.

`NLTK` has some notation for this.

5 Questions

1. Remember those silly tags from hw1.org? Let

$N = \{\text{FOO,BAR,EGO,NEED,ADS,DUCK,MANSE}\}$ and
$T = \{dog, black, racing, was, squirrel, tree, burrow, groundhog, bushes, towards, hunting, back, wee\}$

For each of the following production rules, state from which class of language they come and *show why* by derivation from the language definitions (that is, write the proof and describe it).

rule 1 FOO $\longrightarrow$ EGO NEED DUCK

ANSWER This is a production of a Context Free Language.

- This is because the rule can be rewritten as $A \to BCD$ where $B, C, D \in N$
- $BCD \in w$ given that $w \in \{N, T\}^*$

rule 2 FOO DUCK $\longrightarrow$ EGO NEED DUCK

ANSWER This is a Context Sensitive Language production!

- This is because this production is of the form $u_1 A u_2 \to u_1 w u_2$
- $\epsilon = u_1, FOO = A, DUCK = u_2, EGO\ NEED = w$
- The following holds true: $u_1, u_2, w \in (N \cup T)^*, A \in N, w \neq \epsilon$

rule 3 FOO $\longrightarrow$ EGO dog DUCK

ANSWER This is a production of a Context Free Language.

- This is because the rule can be rewritten as $A \to BCD$ where $B, D \in N$ and $C \in T$
- $BCD \in w$ given that $w \in \{N, T\}^*$

rule 4 FOO groundhog $\longrightarrow$ EGO dog DUCK squirrel

ANSWER

rule 5 FOO $\longrightarrow$ black dog DUCK

ANSWER This is a Regular Language!

- This is because the rule can be redescribed as $A \rightarrow XY$ where $X \in T^*$ and $Y \in N$

Each of the rules is abstracted from a different grammar!

2. Consider the following grammar.

```
N = {A,B,C,D}
T = {foo,bar}
S = {C}
P = {C -> A B
     B -> A D
     B -> A
     D -> A A
     A -> T
     }
```

Is the grammar left-, right-, neither-, or both-recursive? Why?

3. By hand, generate a construct for each unique length of output using the grammar of question 2 and show them and their derivation as an `org` table:

sentence	rule sequence and comments
$foobar$	$S \rightarrow C \rightarrow AB \rightarrow AA \rightarrow AT \rightarrow TT \rightarrow "foo"T \rightarrow "foo"\ "bar"$
$foofoofoofoo$	$S \rightarrow C \rightarrow AB \rightarrow TB \rightarrow "foo"B \rightarrow "foo"AD \rightarrow "foo"TD \rightarrow$ $"foo""foo"D \rightarrow "foo""foo"AA \rightarrow "foo""foo"TA \rightarrow$ $"foo""foo""foo"A \rightarrow "foo""foo""foo"T \rightarrow "foo""foo""foo""foo"$

4. For the terminals in question 1, write the minimum number of Python **regular expressions** (*not production rules from context-free or greater grammars!*) to distinguish among them *when the entire group of words is presented* (not one-by-one!). Do not use trivial regexes that match one and only one word. Use conditionals and order the regexs into a tree until all terminals are recognized without ambiguities. Carry your tree out until each terminal has a regular expression that places it in the leaves. Include a sketch of your tree if you think it will help!

5. Now implement your regular expression tree in question 4 and show the code and results.

6. Write a grammar that captures the following sentences:

sentence 1 “The cheerful black dog slept quietly by the chair.”

sentence 2 “A sleepy yellow dog stretched his back.”

sentence 3 “Somebody downstairs made the coffee.”

Put the phrases generated by each rule from these sentences alongside the rules, again as an `org` table (this example is **JUST to illustrate format, it is not correct!**):

rule	phrase
S $\rightarrow$ A B	(foo bar barbar)

7. Now implement the grammar of question 6 as a recursive descent parser. Parse each sentence, showing the results as a prettily printed tree, and compare them. What do you observe?

8. Following on, implement the grammar of question 6 as a chart parser. Parse each sentence, showing the results, and compare these chart parsing results to your results in question 7. What do you observe?

9. Extend your grammar for the sentences in question 7 so that it can parse sentences 4–6 below. Time the implementation's performance for each sentence, doing this 1000 times for each sentence for better estimates, and put the results in an `org` table.

   sentence 4 “We had a long walk to the park and Vinny played with three other dogs.”

   sentence 5 “It was sunny today but might not be tomorrow.”

   sentence 6 “There are 49 angels dancing on the head of this pin.”

10. Consider this attribute-value matrix:

$$
\begin{bmatrix}
\text{CAT} & s \\
\text{HEAD} & \begin{bmatrix}
\text{AGR} & \boxed{1} \begin{bmatrix} \text{NUM} & sg \\ \text{PER} & 3 \end{bmatrix} \\
\text{SUBJ} & \begin{bmatrix} \text{AGR} \ \boxed{1} \end{bmatrix}
\end{bmatrix}
\end{bmatrix}.
$$

Draw the corresponding directed acyclic graph, ideally in Python. (A hand-drawn figure is fine: just photograph it and include the image below, as is done in notes.org.)

11. Now extend your grammar from question 9 to include features relevant to subject-verb agreement, using `nltk.FeatStruct()` from chapter nine, so that you can parse sentences 1–9. Using `cp.parse()`, print and study the parse trees for each sentence. Do you agree with them? Why or why not?

    sentence 7 "The black dogs are playing with the elf toy."

    sentence 8 "The yellow dog slept in my pajamas."

    sentence 9 "We will take two long rides in the country next week."

12. Chapter 10 of BLK and the semantics howto march one through the basics of applying the FOPC and the λ calculus to reifying the semantics of context-free sentences. One of the practical difficulties in this approach is ensuring that the implementation of the universe of discourse (they call it the *domain of discourse*, same thing) actually covers the intended universe.

    To see this, let's use their `sem2.fcfg` grammar to parse the following sentences syntactically and semantically, and output the reification of the sentences into the FOPC and the λ calculus.

    (HINT: be sure to

    ```
    from nltk.sem import *
    ```

    so you get all the parts and save yourself frustration!)

    For each of the following sentences, parse them and print the sentence, its parse, and its semantics; and then explain the results you get and exactly how you would fix the problems encountered.

    - Suzie sees Noosa.
    - Fido barks.
    - Tess barks.

6 Grading Scale

This homework is worth 15 points. The grading scale is:

fraction correctly reviewed and answered	points awarded
≥ 0.95	15
0.90 – 0.94	14
0.85 – 0.89	13
0.80 – 0.94	12
0.75 – 0.79	11
0.70 – 0.74	10
0.65 – 0.69	9
0.60 – 0.64	8
0.55 – 0.59	7
0.50 – 0.54	6
0.45 – 0.49	5
0.40 – 0.44	4
0.35 – 0.39	3
0.30 – 0.34	2
0.25 – 0.29	1
< 0.25	0

7 Scoring

question	max pts	answer ok?
1	1	
2	1	
3	1	
4	1	
5	1	
6	2	
7	1	
8	1	
9	1	
10	1	
11	2	
12	2	
total score	15	0
percentage		0
total points		