Calc-Discrete Assignment 29   Due 6/9/21

This material is quite a bit different than the other concepts we have looked at. This material has applications to the problem of parsing, for example the algebraic expression parser you were working on and for building compliers. Many of these ideas, especially the idea of a formal grammar was developed by the linguist Noam Chomsky. You might be interested to learn more about him particularly since he was born Jewish though he is not a religious Jew.

Pages 794 – 795: Exercises 4, 5, 7. 19, 22,

Pages 814 – 815: Exercises 1, 5, 9, 11, 16, 19, 23

I. See what you can learn about converting an infix expression into a postfix or prefix expression. Then learn about the stack and a queue data structures, and how these can be used to evaluate either a prefix or a postfix expression. This can be used to implement your algebraic parser.

Page 794

Section 12.1

4.

   a. $S \rightarrow 1S \rightarrow 11S \rightarrow 111S \rightarrow 11100A \rightarrow 111000$
   b. None of the productions can place a 1 on the right of a 0
   c. 1^n0^m where $n \geq 0$ and $m > 3$

5.

   a. $S \rightarrow 1A \rightarrow 10B \rightarrow 101A \rightarrow 1010B \rightarrow 10101$
   b. No set of productions can place two 1s next to each other
   c. $0(01)^n$ or $1(01)^n$ where $n > 0$

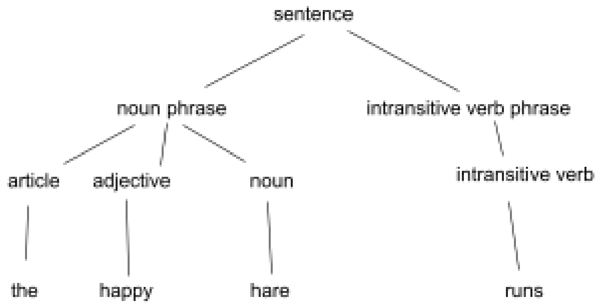
7. $S \rightarrow 0S1 \rightarrow 00S11 \rightarrow 000S111 \rightarrow 000111$

19.

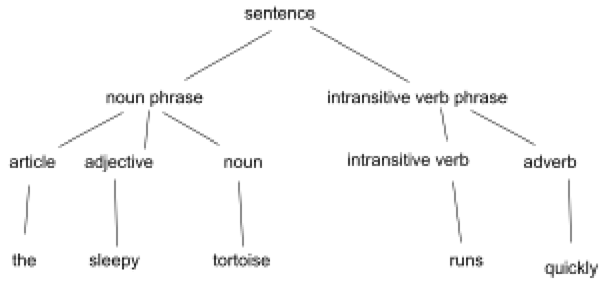
   a. Type 2
   b. Type 3
   c. Type 0
   d. Type 2
   e. Type 2
   f. Type 0
   g. Type 3
   h. Type 0
   i. Type 2
   j. Type 2

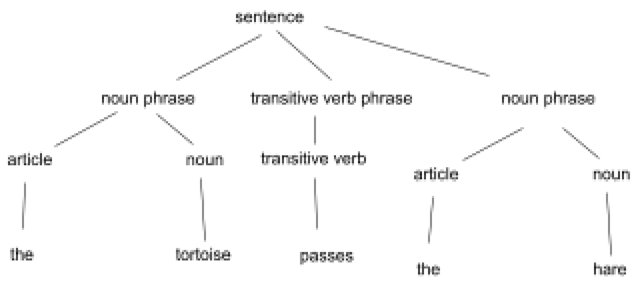
23.

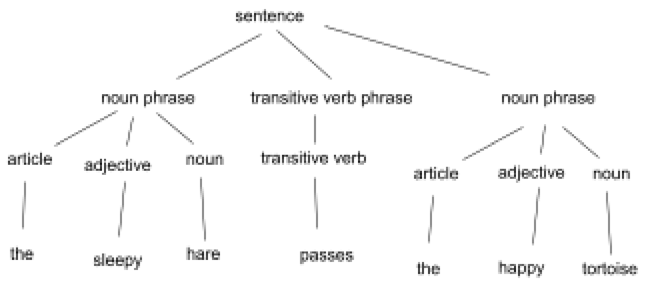
a.



b.



c. 

d. 

Page 814

Section 12.3

1.
   a. $\{000, 001, 1100, 1101\}$
   b. $\{000, 0011, 010, 0111\}$
   c. $\{00, 011, 110, 1111\}$
   d. $\{000000, 010000, 100000, 010100, 000001, 010001, 100001, 010101\}$

5.
   a. Contains all strings of 01 repeating
   b. Contains all strings of 111 repeating (including empty string)
   c. Contains strings that can be assembled by repeatedly taking one or more 0s followed by a 1
   d. Contains strings that can be assembled by repeatedly taking two or more 2s followed by a 0 and adding a 1 at the end

9.
   a. Yes
   b. Yes
   c. No
   d. No
   e. Yes
   f. Yes

11.
   a. Yes
   b. No
   c. Yes

d. No

16. $M = (S, I, f, s_0, F)$ where $S = \{s_0, s_1, s_2\}$ $I = \{0, 1\}$ $F = \{s_0, s_1\}$ $f =$

	0	1
s_0	s_2	s_1
s_1	s_1	s_1
s_2	s_1	s_2

19. $M = (S, I, f, s_0, F)$ where $S = \{s_0, s_1, s_2\}$ $I = \{0, 1\}$ $F = \{s_1\}$ $f =$

	0	1
s_0	s_0	s_1
s_1	s_2	s_1
s_2	s_2	s_2

23. $M = (S, I, f, s_0, F)$ where $S = \{s_0, s_1, s_2, s_3\}$ $I = \{0, 1\}$ $F = \{s_2\}$ $f =$

It's easier for me to make the table than draw it but it's pretty simple anyways.

	0	1
s_0	s_1	s_3
s_1	s_3	s_2
s_2	s_2	s_2
s_3	s_3	s_3

I. I believed we discussed this in class already. If you're interested, I have a github repo. It's still incomplete and might not compile but you can see the existing functionality here https://github.com/BOBONA/UsefulCalculator