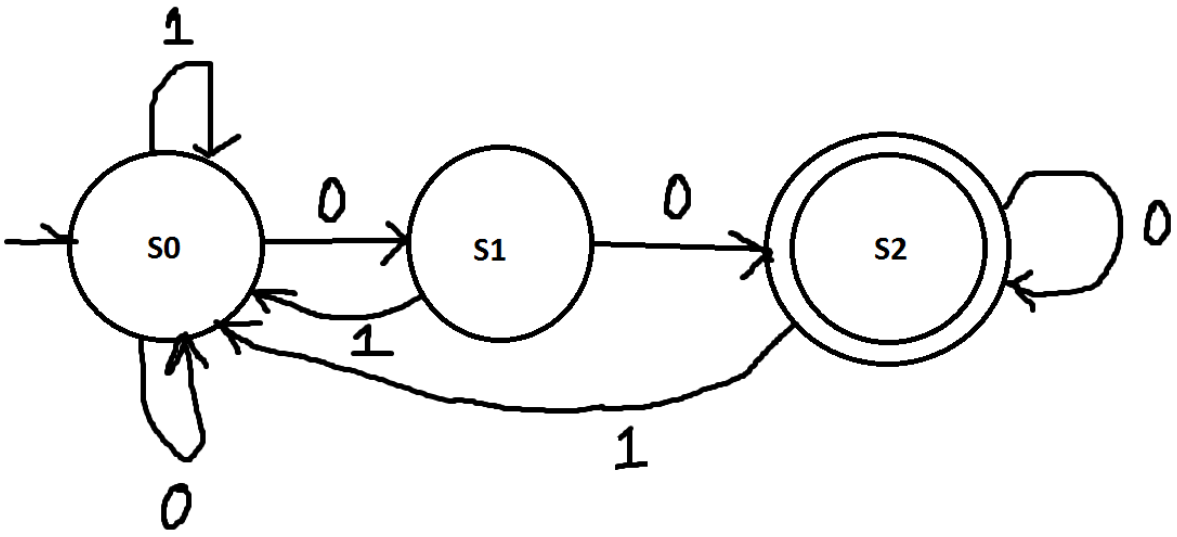
Jessie George
Section 1
HW2
2/13/16

Problem 1
Assuming that input is at least 2 binary numbers long.
NFA 20/20 pts

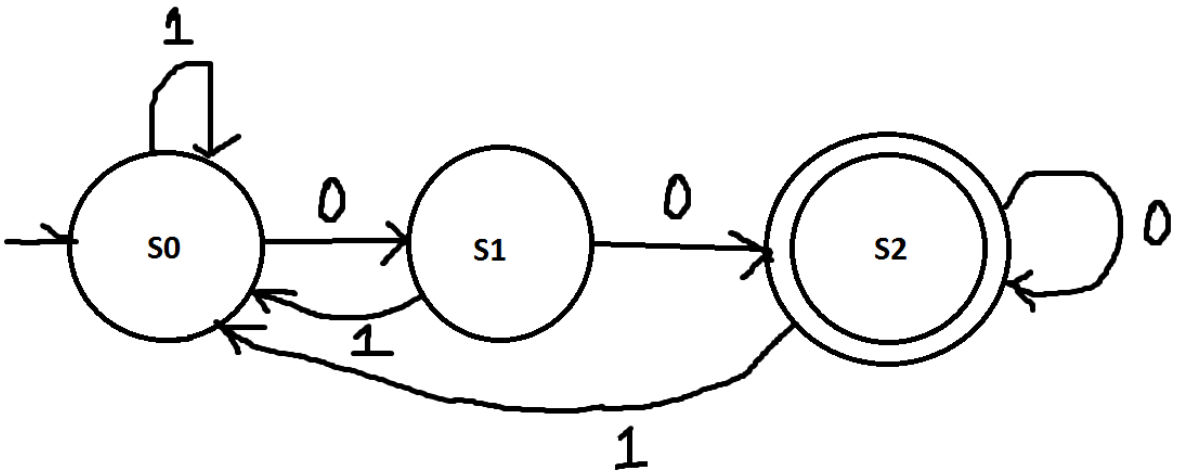


<S, s, F, T>
S = {S0, S1, S2}
s = S0
F = {S2}
T =

State		0	1	Input
	S0	S0, S1	S0	
	S1	S2	S0	
	S2	S2	S0	

See next page for DFA

DFA



<S, s, F, T>
S = {S0, S1, S2}
s = S0
F = {S2}
T =

State		0	1	Input
	S0	S1	S0	
	S1	S2	S0	
	S2	S2	S0	

Problem 2
Sub-part 1 12/40
Non-CFL
Pumping Lemma
Let s = aaabbbccc
m=n=o=3
p=9
u=aa, v=ab, x=b, y=bc, z=cc
|vy| = 2 + 2 = 4 >0
|vxy| = 2+1+2 <= 9

Let i=2
uv^2xy^2z = aaababbbcbccc which does not belong to the language. Contradiction.
Therefore, the language is a Non-CFL.
Regex = $a^*b^+c^+$

Sub-part 2
Non-CFL
Pumping Lemma
Let s = aaaabbbccc
m=4, n=o=3
p=10
u=aaa, v=ab, x=b, y=bc, z=cc
|vy| = 2 + 2 = 4 >0
|vxy| = 2+1+2 <= 10
Let i=2
uv^2xy^2z = aaaababbbcbccc which does not belong to the language. Contradiction.
Therefore, the language is a Non-CFL.
No regular expression.

Sub-part 3
Non-CFL
Pumping Lemma +4
Let s = aaabbbccc
m=n=o=3
p=9
u=aa, v=ab, x=b, y=bc, z=cc
|vy| = 2 + 2 = 4 >0
|vxy| = 2+1+2 <= 9
Let i=2
uv^2xy^2z = aaababbbcbccc which does not belong to the language. Contradiction.
Therefore, the language is a Non-CFL.
No regular expression.

Sub-part 4
Non-CFL
Pumping Lemma
Let s = aaaabbbbbb

n=2
p=10
u=aaa, v=ab, x=b, y=bb, z=bb
|vy| = 2 + 2 = 4 >0
|vxy| = 2+1+2 <= 10
Let i=2
uv^2xy^2z = aaaababbbbbbbb which does not belong to the language. Contradiction.
Therefore, the language is a Non-CFL.
No regular expression.

Sub-part 5
Non-CFL
Pumping Lemma
Let s = aabbaa
w=aab, w^R=baa
p=6
u=a, v=ab, x=b, y=a, z=a
|vy| = 2 + 1 = 3 >0
|vxy| = 2+1+1 <= 6
Let i=2
uv^2xy^2z = aababbaaa which does not belong to the language. Contradiction.
Therefore, the language is a Non-CFL.
No regular expression.

Sub-part 6
Non-CFL
Pumping Lemma
Let s = bbbbcccc
n=0, m=4
p=8
u=bbb, v=bc, x=c, y=c, z=c
|vy| = 2 + 1 = 3 >0
|vxy| = 2+1+1 <= 8
Let i=2
uv^2xy^2z = bbbbcbccccc which does not belong to the language. Contradiction.
Therefore, the language is a Non-CFL.
No regular expression.

Sub-part 7 +4

Non-CFL

Pumping Lemma

Let s = bbbbdddd

n=0, m=4

p=8

u=bbb, v=bd, x=d, y=d, z=d

|vy| = 2 + 1 = 3 >0

|vxy| = 2+1+1 <= 8

Let i=2

uv^2xy^2z = bbbbdbdddddd which does not belong to the language. Contradiction.

Therefore, the language is a Non-CFL.

No regular expression.

Sub-part 8

Non-CFL

Pumping Lemma

Let s = ccccdddd

n=0, m=4

p=8

u=ccc, v=cd, x=d, y=d, z=d

|vy| = 2 + 1 = 3 >0

|vxy| = 2+1+1 <= 8

Let i=2

uv^2xy^2z = ccccdcdddddd which does not belong to the language. Contradiction.

Therefore, the language is a Non-CFL.

No regular expression.

Sub-part 9

Non-CFL

Pumping Lemma

Let s = aaaabbbb

n=2

p=8

u=aaa, v=ab, x=b, y=b, z=b

|vy| = 2 + 1 = 3 >0

|vxy| = 2+1+1 <= 8
Let i=2
uv^2xy^2z = aaaababbbbb which does not belong to the language. Contradiction.
Therefore, the language is a Non-CFL.
No regular expression.

Sub-part 10

+4

CFL
The CFG in BNF notation is <T,N,P,S> where:
Terminal T = a,b
Non-Terminal N = <w> , <alph>
Production Rules P

1. <w> → <alph><alph><alph><alph><alph><alph>
2. <alph> → a|b|<alph><alph>

Start S = <w>
Regex = (a|b) (a|b) (a|b) (a|b) (a|b) $(a|b)^+$

Problem 3 on next page

Problem 3

Sub-part 1

Leftmost Derivation

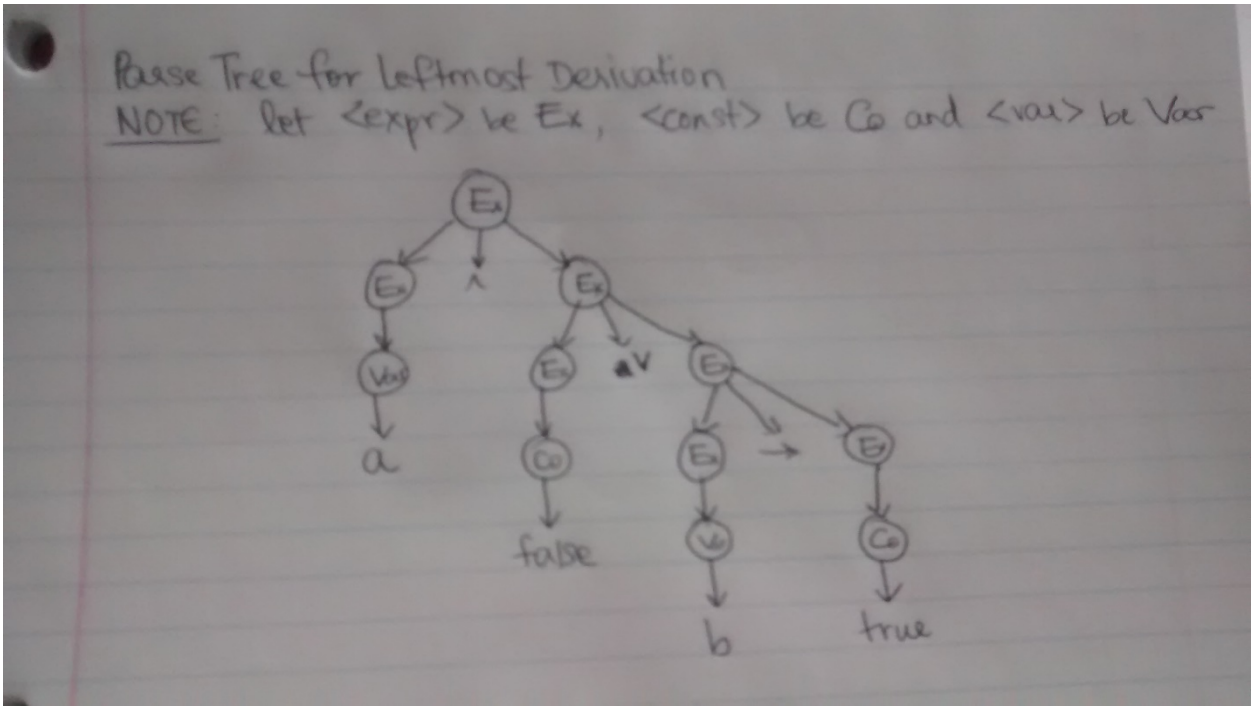
Rule	Sentential Form
1	<expr>
2	<expr> ^ <expr>
2	<var> ^ <expr>
4	a ^ <expr>
2	a ^ <expr> ˅ <expr>
2	a ^ <const> ˅ <expr>
3	a ^ false ˅ <expr>
2	a ^ false ˅ <expr> → <expr>
2	a ^ false ˅ <var> → <expr>
4	a ^ false ˅ b → <expr>
2	a ^ false ˅ b → <const>
3	a ^ false ˅ b → true

Rightmost Derivation

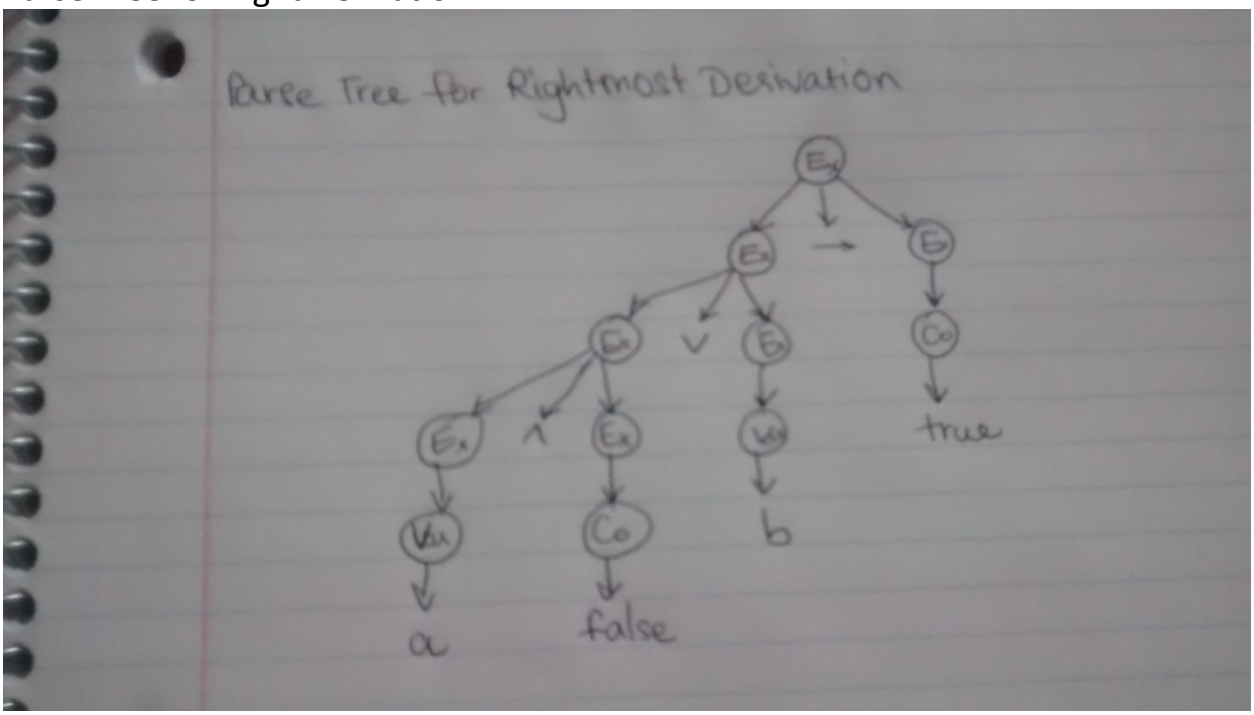
Rule	Sentential Form
1	<expr>
2	<expr> → <expr>
2	<expr> → <const>
3	<expr> → true
2	<expr> ˅ <expr> → true
2	<expr> ˅ <var> → true
4	<expr> ˅ b → true
2	<expr> ^ <expr> ˅ b → true
2	<expr> ^ <const> ˅ b → true
3	<expr> ^ false ˅ b → true
2	<var> ^ false ˅ b → true
4	a ^ false ˅ b → true

Sub-part 2 10/10 pts

Parse Tree for Left Derivation

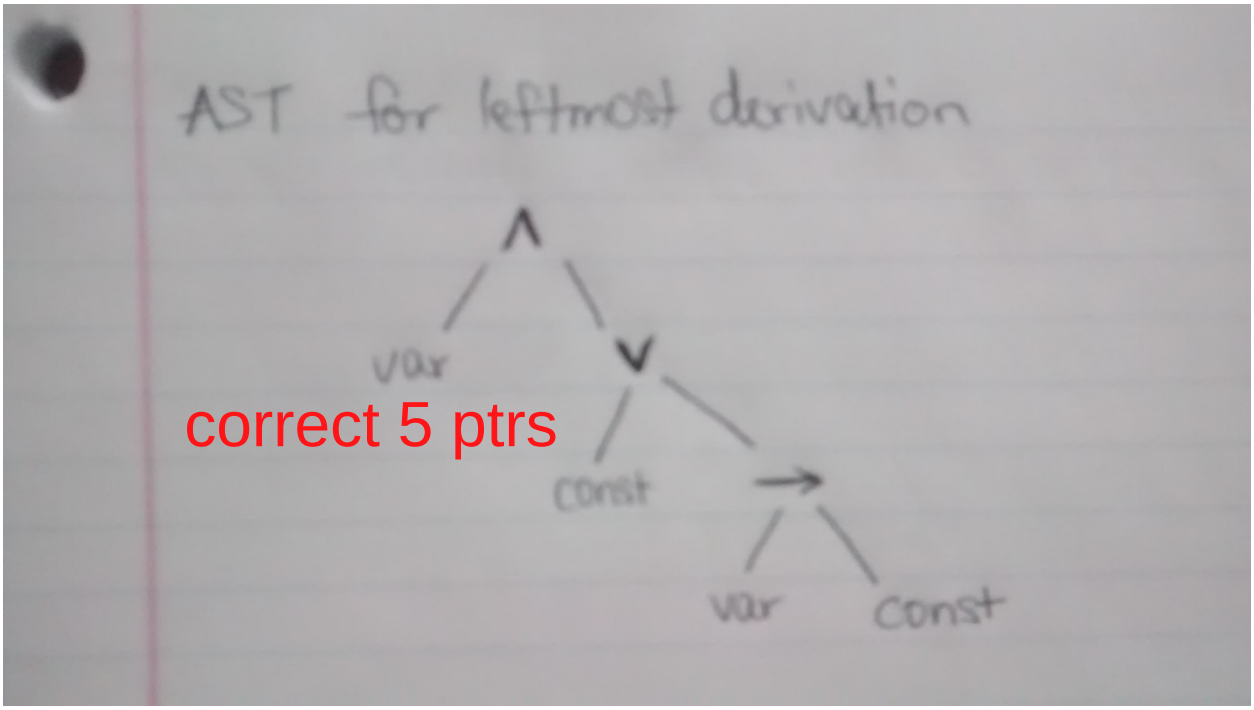


Parse Tree for Right Derivation

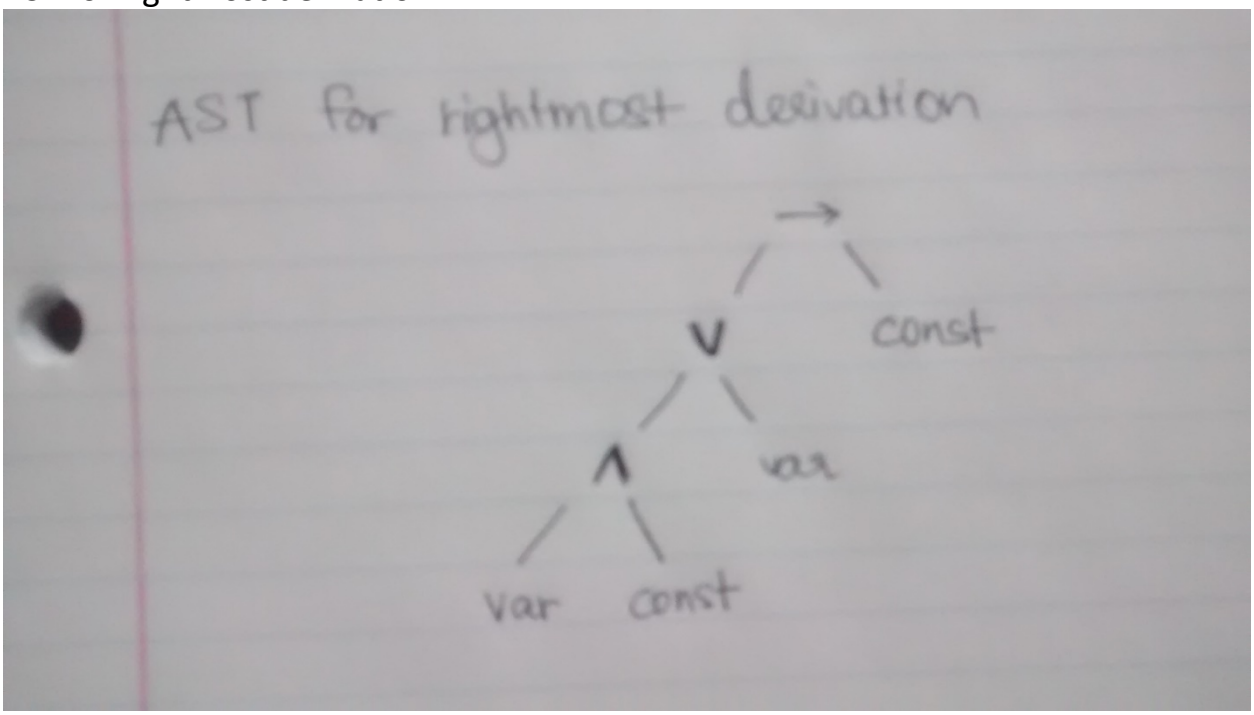


Sub-part 3

AST for leftmost derivation



AST for rightmost derivation



Sub-part 4

It is ambiguous because in sub-part 2, we should 2 distinct parse trees that have the same root <expr> and yield the same output a ^ false ᵛ b → true.

5 ptrs

Sub-part 5

cannot guarantee right associative

no such expression

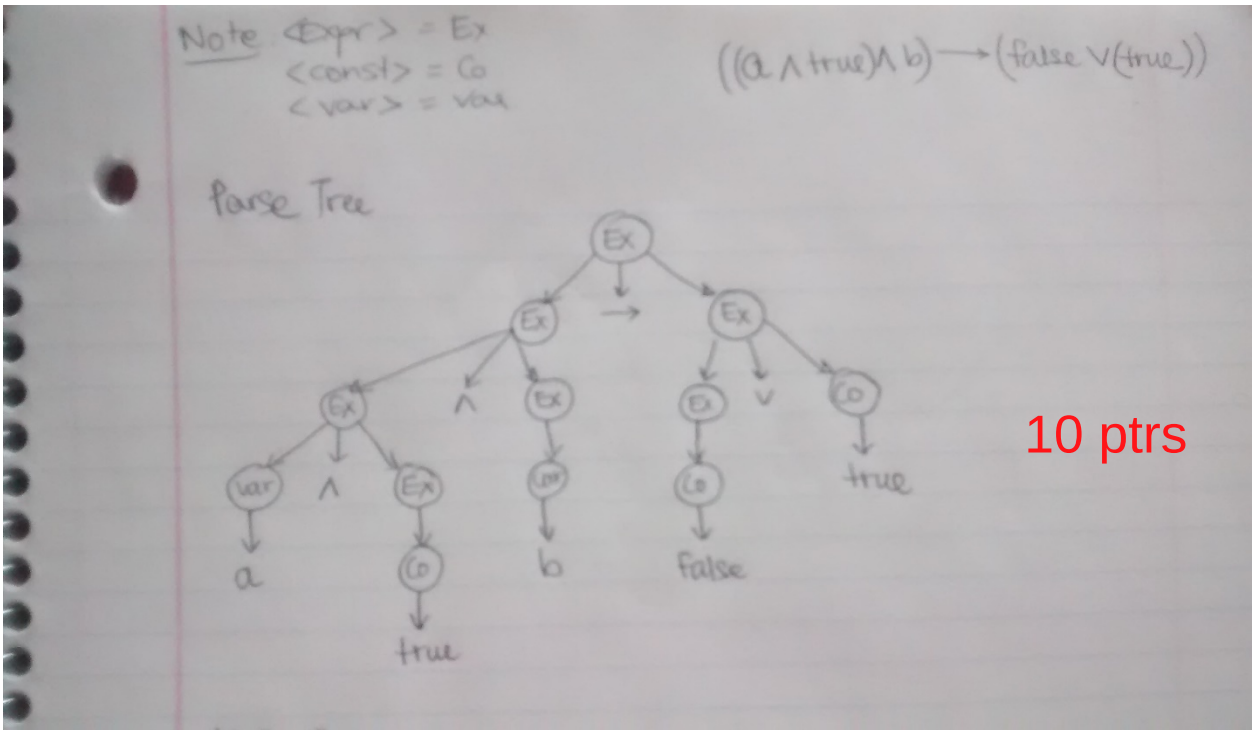
5 ptrs

1. <start> ::= <expr>
2. <expr> ::= <expr> → < expr|const|var > | <const|var>
3. <expr> ::= < expr|const|var > ^ <expr> | <const|var>
4. <expr> ::= <expr > ᵛ < expr|const|var > | <const|var>
5. <const> ::= true|false
6. <var> ::= a|b|c|…|z

has similar problems to the above

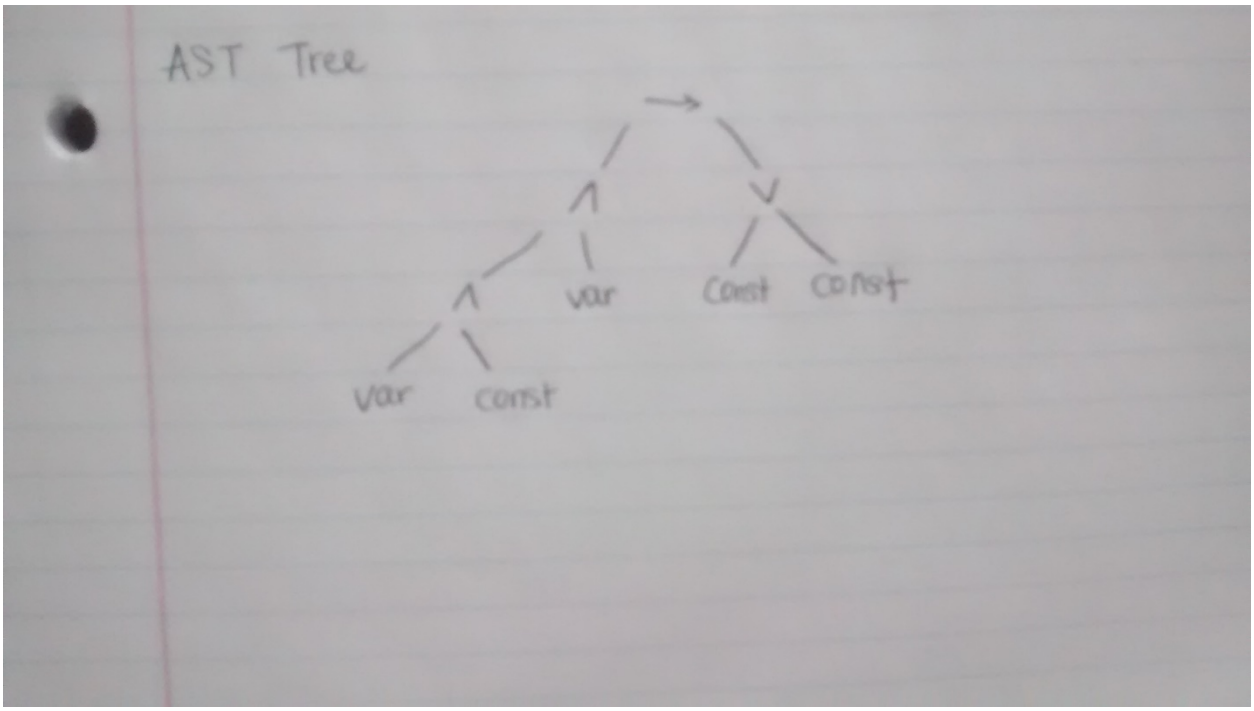
Sub-part 6

Parse Tree



10 ptrs

AST (see next page)


AST Tree
var
const
var
const
const