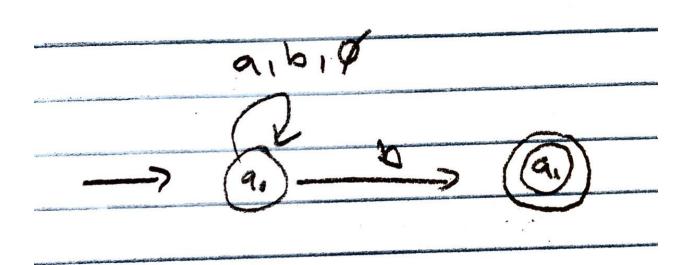
Homework 2: CFG

- 1. N.A.
- 2. . (15) Show the NFA that results from applying the NFA construction method, discussed in the class, to the regular expression ($a \mid b \mid 0$) \land b.



- 3. (40) Context Free Grammar.
 - a. Write a context free grammar for arithmetic expressions which can use numbers, variables and operation + only

i.
$$expr \rightarrow expr + expr$$

ii.
$$expr \rightarrow var \mid num$$

iii.
$$var \rightarrow a \mid b \mid c$$

iv.
$$num \to 1 \mid 2 \mid ... \mid 9$$

b. Write a context free grammar for the arithmetic expressions above that captures right associativity.

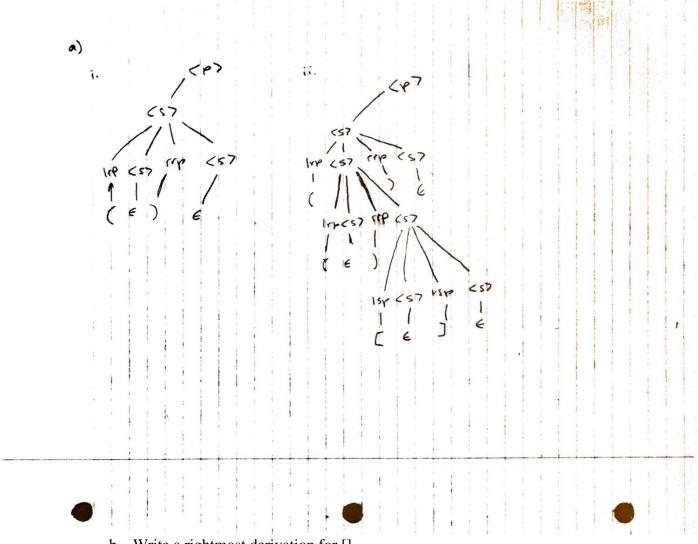
i.
$$expr \rightarrow var + expr \mid num + expr$$

ii.
$$var \rightarrow a \mid b \mid c$$

iii.
$$num \to 1 | 2 | ... | 9$$

- c. Write a context free grammar for arithmetic expressions which can use numbers, variables and binary operations and + only. Your grammar has to capture the precedence that must be computed before +. Your grammar should also capture the left associativity of + and -. For example 5+5-5-5+6-5 should be computed as 5+(5-5-5)+(6-5).
 - i. $expr \rightarrow operand \mid expr + group$
 - ii. $var \rightarrow a \mid b \mid c$
 - iii. $operand \rightarrow num \mid var$
 - iv. $num \to 1 | 2 | ... | 9$
 - v. $group \rightarrow expr expr \mid expr$
- d. Refine your grammar above to allow parenthesis (and). Hint. Recall the techniques on how precedence and associativity were dealt with in the class.
 - i. $expr \rightarrow operand \mid expr + group \mid expr + (group)$
 - ii. $var \rightarrow a \mid b \mid c$
 - iii. $operand \rightarrow num \mid var$
 - iv. $num \to 1 | 2 | ... | 9$
 - v. $group \rightarrow expr expr \mid expr$
- 4. (15) Given a grammar $< P > \rightarrow < S > < S > \rightarrow lrp < S > rrp < S > | lsp < S > rsp < S > | <math>\epsilon$ The terminals are defined as follows $lrp \rightarrow (rrp \rightarrow) lsp \rightarrow [rsp \rightarrow] CS3361$ Concepts of Programming Languages by Y Zhang, TTU, Spring 2021

a. Draw a parse tree for each of the sentences:



b. Write a rightmost derivation for [].

i.
$$\langle P \rangle \rightarrow \langle S \rangle$$

ii.
$$\langle S \rangle \rightarrow lsp \langle S \rangle rsp \langle S \rangle$$

1.
$$\rightarrow lsp < S > rsp \epsilon$$

2.
$$\rightarrow lsp \epsilon rsp \epsilon$$

$$3. \ \to lsp \ rsp$$

5. Consider the CFG for in the slides (of L6) for parsing (Page 16), and an input program sum := A + B

a)

Sum:= A + B write sum (program) CStmt-Yist7 & F estants 25tunt-1197 85 id := (expr) (stml-list) 89 Match! continue ... := (expr) (stmt_list) \$\$:= A + 15 write sum; match!... CXPT) (Stut-11517\$5 A+15 write sum; Ctermo Ctem-tuilo Com Elisto \$1 | A +18 write sumi Gentle 7 (fuct tail) (tern-tail) (start-lut) Atis write sumi id (fact-tail > (tem-tail > (Start-list) 58 | match! CECL-tail > Ctem-tail > Cstimt-list > 11 + 13 write sumi & cterm-tail > (Statent-115+7\$1) + 15 write sum; Ctermity 17 C Start - 15+7 & 5 | + 13 write sum: Cade op Xterm > Cterm_tail Estant list) \$9 + K winte sum + < tem7</tem + color colon + livir &s | martch1 Ctermy Cterm-tuil 7 (strut-list> & 18 write gum Cfactors Cfact tails (terminals cstml 15755 VS Write sum id Cfuct-tally Ctomptally (Stant 16175) Mutel Cfact tail 7 dem tail 7 Cstant 1.547 53 | write sun E Clem-tail7 cstant 1.5+> 54 | write com E cstm + 1'st7 55 | write Sum estimpe stant 1:17 &9 | write sum write cexpro cstm = 1,517 1\$ maleh! Cexo17 65+m+ 1:517 \$1 Sum Cterm > Cterm-tail > cstml 1.5+7 \$\$ | Sum (Suctor) (fact-tuil) (term-tail) (start list) \$8 \ Sum id cfact-tails (form-tails (start 15+7 \$\$ | match! E Cterm-taily estand-1.517 & E <51m+-115+7 \$\$ 8 50 done