Austin Brooks
Red ID: 819274879
10/12/2019
CS 320

Homework assignment 2

Problem 1: terminal set {a,b,c,(,)} non-terminal set {E,F,id}

E -> E + F | F
F -> F * F | (id) | (E)
id -> a | b | c

example:

E -> E+F	E-> F
E -> E+(F*F | (id) | (E))	E->F*F | (id) | (E)
E -> E+(F*F | **(a | b | c)** | (E))	E->F*F | **(a | b | c)** | (E)

the function allows for multiple outputs, meaning the grammar is ambiguous
for instance, a,b,c could be output in a variety of ways.

ANSWER: ambiguous

Problem 2:

$E \rightarrow E \cap F \mid E \cup F \mid E \wedge F \mid E \vee F \mid F$ takes in F

$F \rightarrow id \mid (E)$ goes into E

1 is higher precedence, most important, 2 is lower precedence

left to right, ignore |

∩ , U , Λ , V goes left to right (,)?

Operator	Precedence	Associativity
∩	2	left
U	2	left
Λ	2	left
V	2	left
(	1	doesn't have effect
)	1	doesn't have effect

Problem 3:

Terminal: {a, b, c, ¬,Λ,V, (,)} non-terminal: {p, id}

$P \rightarrow id \mid P \vee P \mid P \wedge P \mid \neg P \mid (P)$

$id \rightarrow a \mid b \mid$ c

$(a \wedge b) \vee \neg(b \wedge c)$(leftmost)

ANSWER

leftmost -

$(a \wedge b) \vee \neg(b \wedge c)$

rightmost -

$\neg(b \wedge c) \vee (a \wedge b)$

Problem 4: terminal set is $\{a, b, c, (,)\}$ non-terminal set is $\{E, F, id\}$
eliminate left recursion

$E \rightarrow E + F \mid F$

A' = E B = F

$F \rightarrow F * F \mid id \mid (E)$

A' = F' B = id,E

$id \rightarrow a \mid b \mid c$ will not change?

$A \rightarrow \beta A'$

$A' \rightarrow \alpha A' / \in$

ANSWER

E → FE' using 1st rule

E' → +FE'/ ∈ place e under uncomplemented to make compliment

F → (id)F' | E F' using 1st rule

F' → ((id)F' | E F')/ ∈

id → a|b|c

Problem 5:

Given the following grammar, where the terminal set is {1,2,3, ε, +,$*$, (,)}, the non-terminal set is $\{S, T, L, U, M, V\}$, and the start symbol is S. Please solve the following questions

$S \rightarrow TL$

$L \rightarrow +S \mid \varepsilon$

$T \rightarrow UM$

$M \rightarrow * T \mid \varepsilon$

$U \rightarrow (S) \mid V$

$V \rightarrow 1 \mid 2 \mid 3$

a. Compute the FIRST and FOLLOW for each non-terminal (i.e., complete the following FIRST&FOLLOW table)

Non-terminal	First	Follow
S	{(,1,2,3}	{),$}
L	{+,ε}	{),$}
T	{(,1,2,3}	{+,), $}
M	{$*$, ε}	{+,), $}

U	{(,1,2,3}	{*, +,), $}
V	{1,2,3}	{*, +,), $}

b. Construct a parsing table for the top-down parser (i.e., complete the following parsing table).

Non term	1	2	3	+	*	(	)	$
S	S -> TL	S -> TL	S -> TL			S -> TL		
L				L->+S			L -> e	L -> e
T	T-> UM	T-> UM	T->UM			T->UM		
M				M->e	M->*T		M->e	M->e
U	U -> V	U-> V	U-> V			U->S		
V	V -> 1	V->2	V->3					

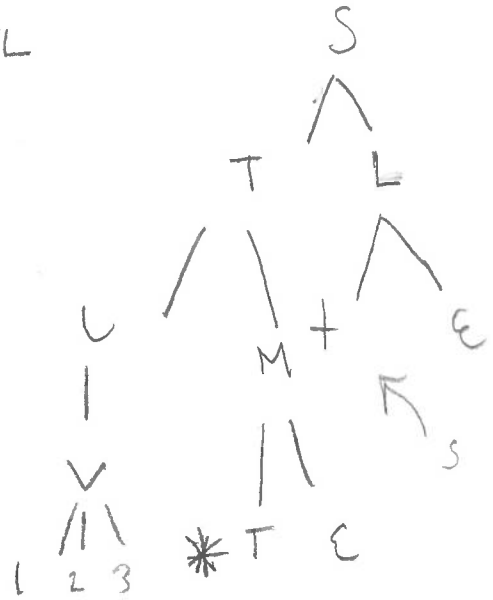
c. Parse trees

5. $S \rightarrow TL$

$L \rightarrow +S \mid \varepsilon$

$T \rightarrow \cup M$

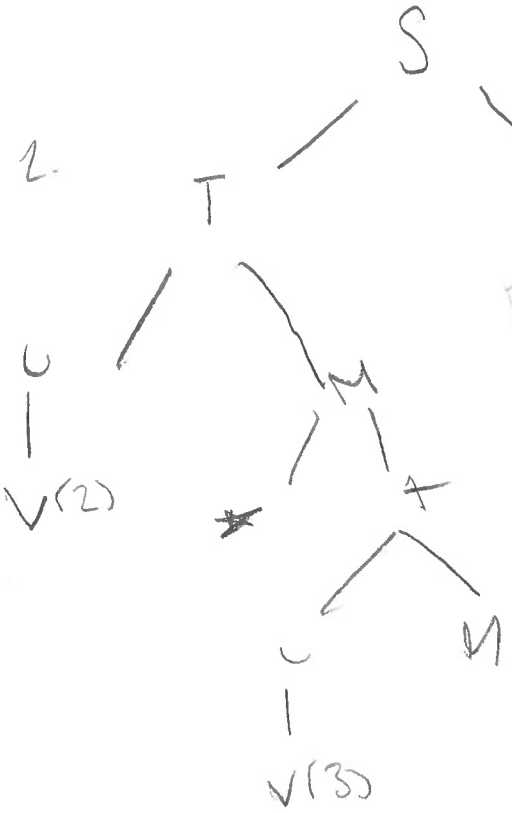
$M \rightarrow *T \mid \varepsilon$



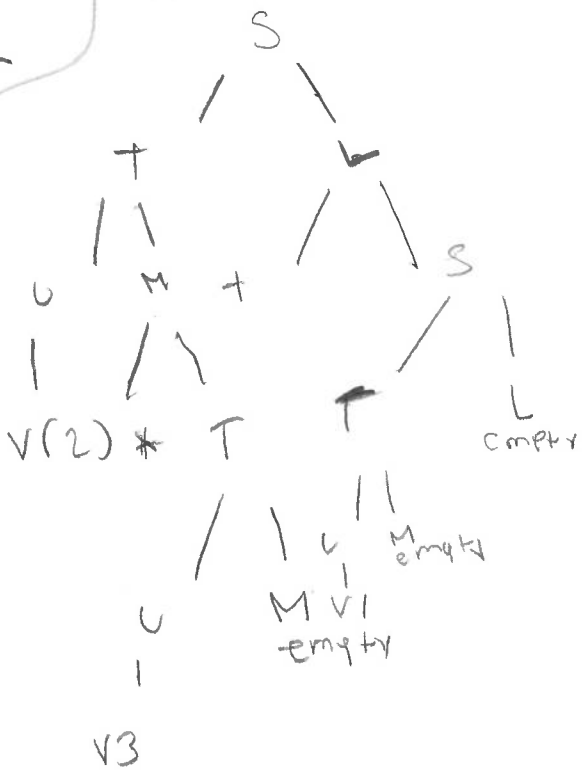
1.

* operator is need to be alone

2.



Answer:



6. T: {1, 2, 3, +, *, (,)} NT: {S, T, U, V}

2 * 3 + 1 $

a.

Stack	Input	Action
$	2 * 3 + 1 $	Shift
$ 2	* 3 + 1 $	Reduce
$ V	* 3 + 1 $	Reduce
$ U	* 3 + 1 $	Reduce
$ T	* 3 + 1 $	Shift
$ T *	3 + 1 $	Shift
$ T * 3		
$ T * V		
$ T * U		
$ T		
$ S		

V → 2? nothing happens

T → U

no *

* For a, b, c, d see next paper (not back)

a

Stack	Input	Action
$	2 * 3 + 1 $	Shift
...	...	combo of shift + reduce
$ S	$	✓ Accept

start

end

b. Fail

Stack	2 * 3 + 1 $	Reduce by ~
$	Wrong	Wrong
Wrong	$	Error

c. 2 * 3 + 1 shift reduce bottom up