

Dragon

4.2.1

$$S \rightarrow SS + | SS \cdot | a$$

a) $aa + a^*$

LMD

$$S \rightarrow \text{start}$$

1) SS^*

1) $SS + S^*$

3) $aS + S^*$

3) $aa + S^*$

3) $aa + a^*$

$$\beta S := SS^*$$

2) SS^*

3) a

b)

RMD

$$S \rightarrow \text{start}$$

2) SS^*

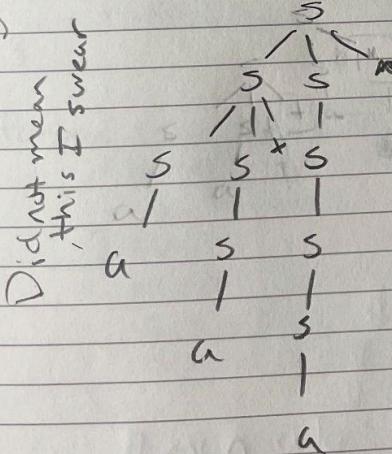
3) Sa^*

1) $SS + a^*$

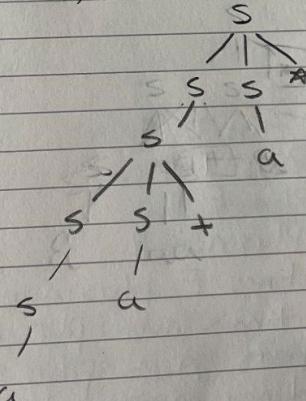
3) $Sa + a^*$

3) $aa + a^*$

c)



RMD



Creating a Compiler 4.7 5.8c

4.7 num plus num times num plus num \$
 a) LMD
 S Start
 T E \$
 T T plus E \$
 S F plus E \$
 T Num plus E \$

2

2 Num plus T plus E \$
 Y num plus T time E plus E \$

3

5 num plus F time E plus E \$
 T Num plus num time E plus E \$

4

3 num plus num time T plus E \$
 T num plus num times F plus E \$

5

2 num plus num times num plus E \$
 T num plus num times num plus T \$

6

3 num plus num times num plus num plus E \$
 T num plus num times num plus num plus E \$

7

b) RMD

S start

T E \$

T T plus E \$

S T plus F \$

T T plus num \$

Y T times F plus num \$

T T times num plus num \$

S F times num plus num \$

G E times num plus num \$

G T plus F times num plus num \$

S T plus F times num plus num \$

T T plus num times num plus num \$

S F plus num times num plus num \$

G num plus num times num plus num \$

- c) This defines a ambiguous grammar for one I had to take a completely different path when it came to the RMD (including a extra rule b). I also believe that this will "break math" because the times is in the middle so the next will never be correct without any specification in the grammar.

5.2c

1) Start \rightarrow Value \$

2) Value \rightarrow num

3) \rightarrow lvalue expr (paren increment)

4) Expr \rightarrow plus value value

5) \rightarrow prod values

6) Values \rightarrow value values

7) \rightarrow n

function Parse_Stack

function Parse_Expr (E)

function Parse_Value (E)

function Parse_Plus (E)

function Parse_Minus (E)

function Parse_Times (E)

function Parse_Divide (E)

function Parse_LValue (E)

function Parse_Increment (E)

function Parse_Error (E)

function Parse_Paren (E)

function Parse_Prod (E)

function Parse_Value (E)

function Parse_Plus (E)

function Parse_Minus (E)

function Parse_Times (E)

function Parse_Divide (E)

function Parse_LValue (E)

function Parse_Increment (E)

function Parse_Error (E)

function Parse_Paren (E)

function Parse_Prod (E)

function Parse_Value (E)

function Parse_Plus (E)

function Parse_Minus (E)

function Parse_Times (E)

function Parse_Divide (E)

function Parse_LValue (E)

function Parse_Increment (E)

function Parse_Error (E)

function Parse_Paren (E)

function Parse_Prod (E)

function Parse_Value (E)

function Parse_Plus (E)

function Parse_Minus (E)

function Parse_Times (E)

function Parse_Divide (E)

function Parse_LValue (E)

function Parse_Increment (E)

function Parse_Error (E)

function Parse_Paren (E)

function Parse_Prod (E)

function Parse_Value (E)

function Parse_Plus (E)

function Parse_Minus (E)

function Parse_Times (E)

function Parse_Divide (E)

function Parse_LValue (E)

function Parse_Increment (E)

function Parse_Error (E)

function Parse_Paren (E)

function Parse_Prod (E)

function Parse_Value (E)

function Parse_Plus (E)

function Parse_Minus (E)

function Parse_Times (E)

function Parse_Divide (E)

function Parse_LValue (E)

function Parse_Increment (E)

function Parse_Error (E)

function Parse_Paren (E)

function Parse_Prod (E)

function Parse_Value (E)

Crafting A Compiler 4.9

First & Follow set for $S \rightarrow aSg$

1) $S \rightarrow aSg$ $\text{Follow } S = \{\$$

$\text{S follows } B \text{ so, turn}$

2) $\rightarrow B$ $\text{Follow } B = \text{Follow } S$

$\text{B follows } B \text{ so, turn}$

3) $B \rightarrow bB(e)$ $\text{Follow } B = \{\$$

$\text{B follows } C \text{ so, turn}$

4) $\rightarrow C$ $\text{Follow } C = \{\$$

$\text{C follows } C \text{ so, turn}$

5) $C \rightarrow cC(e)$ $\text{Follow } C = \{\$$

$\text{C follows } D \text{ so, turn}$

6) $\rightarrow d$ $\text{Follow } D = \{\$$

$\text{D follows } D \text{ so, turn}$

$\text{C since it ends in } \underline{\text{temp}} \text{ so, turn}$

Non-Terminals	Null?	First	Follow
S	no	$\{a, b, c, d\}$	$\{\$, e\}$
B	no	$\{b, c, d\}$	$\{\$, e\}$
C	no	$\{c, d\}$	$\{\$, e\}$

Dragon Exercise 4.4.3

First & Follow set for Grammer in 4.2.1

- 1) $S \rightarrow SS^+$
- 2) $\quad \quad \quad \rightarrow SS^\diamond$
- 3) $\quad \quad \quad \rightarrow a$

Non-terminal	null	First	Follow
S	No	$\{\epsilon, a\}$	$\{S^+, S^\diamond\}$

Crafting A Compiler 9.2

Sign test (Exp) {

neg: stmts

zero: stmts

pos: stmts

3

AST

