Clay Ramey

COMP-3220

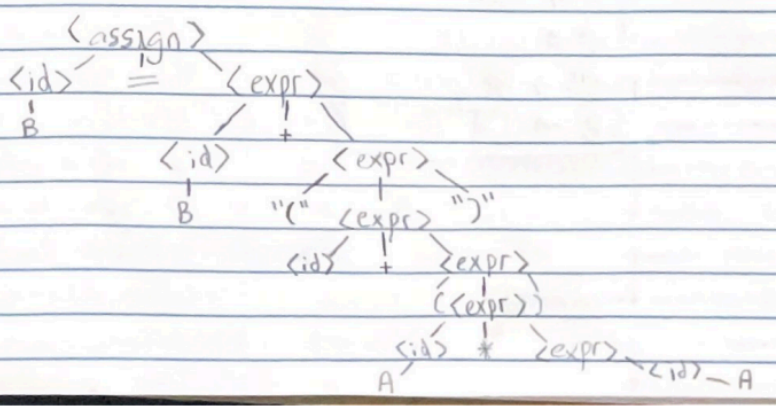
HW 3 2/16/24

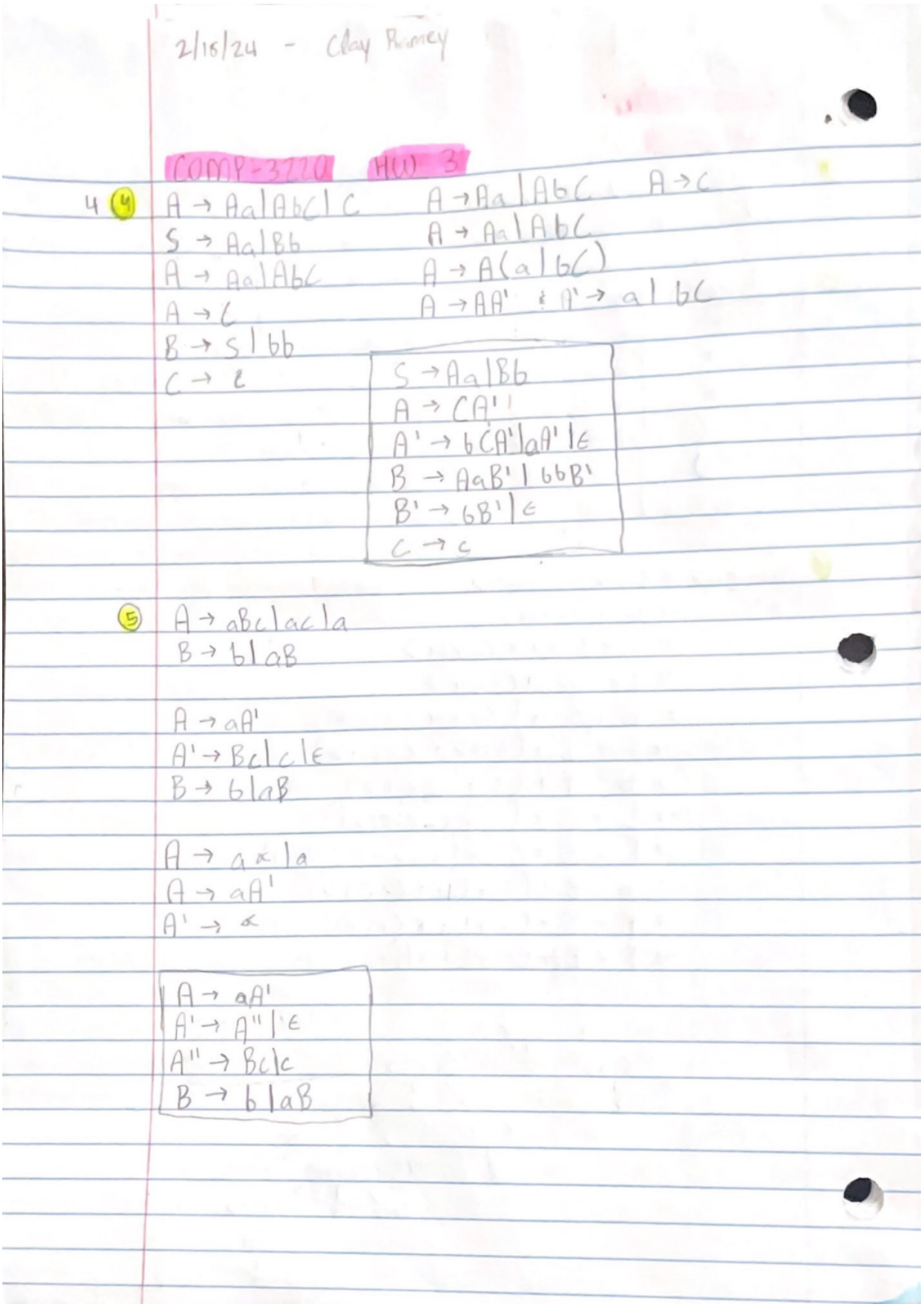
1. A & D are valid sentences
   B & C are incorrect

2.

lexeme	token
=	Assign_op
+	Add_op
*	Mult_op
(	LPAREN
)	RPAREN
A, B, C	Identi

3. <assign> → <id> = <expr>
   → B = <expr>
   → B = <id> + <expr>
   → B = B + <expr>
   → B = B + (<expr>)
   → B = B + (<id> + <expr>)
   → B = B + (C + <expr>)
   → B = B + (C + (<expr>))
   → B = B + (C + (<id> * <expr>))
   → B = B + (C + (A * <expr>))
   → B = B + (C + (A * <id>))
   → B = B + (C + (A * A))





2/16/24 - Clay Ramey

COMP-3220 HW 3

4 (4) $A \to Aa \mid AbC \mid C$

$S \to Aa \mid Bb$

$A \to Aa \mid AbC$

$A \to C$

$B \to S \mid bb$

$C \to \epsilon$

$A \to Aa \mid AbC \quad A \to C$

$A \to Aa \mid AbC$

$A \to A(a \mid bC)$

$A \to AA' \;\&\; A' \to a \mid bC$

$S \to Aa \mid Bb$
$A \to CA''$
$A' \to bCA' \mid aA' \mid \epsilon$
$B \to AaB' \mid bbB'$
$B' \to bB' \mid \epsilon$
$C \to c$

(5) $A \to aBc \mid ac \mid a$

$B \to b \mid aB$

$A \to aA'$

$A' \to Bc \mid c \mid \epsilon$

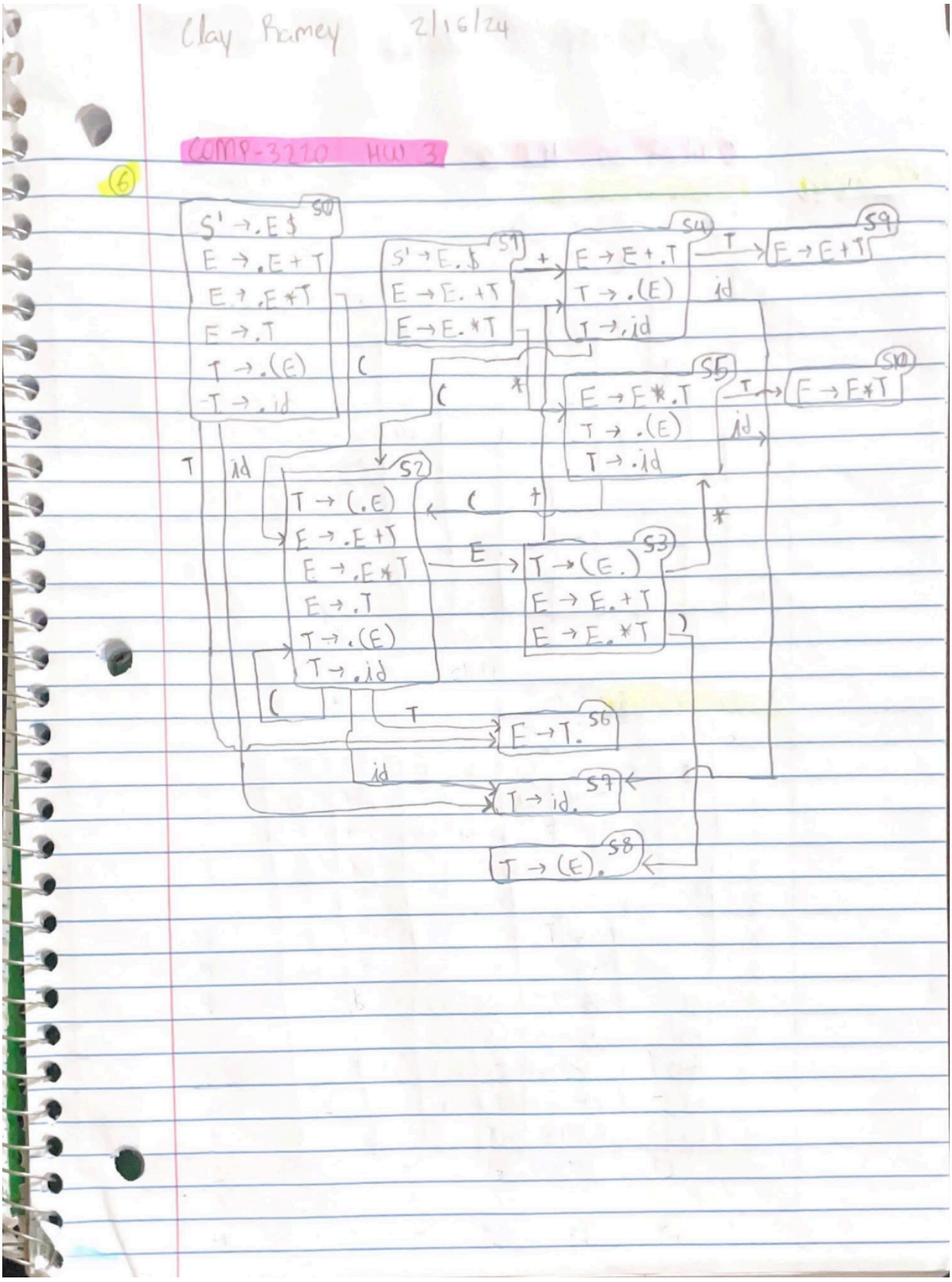
$B \to b \mid aB$

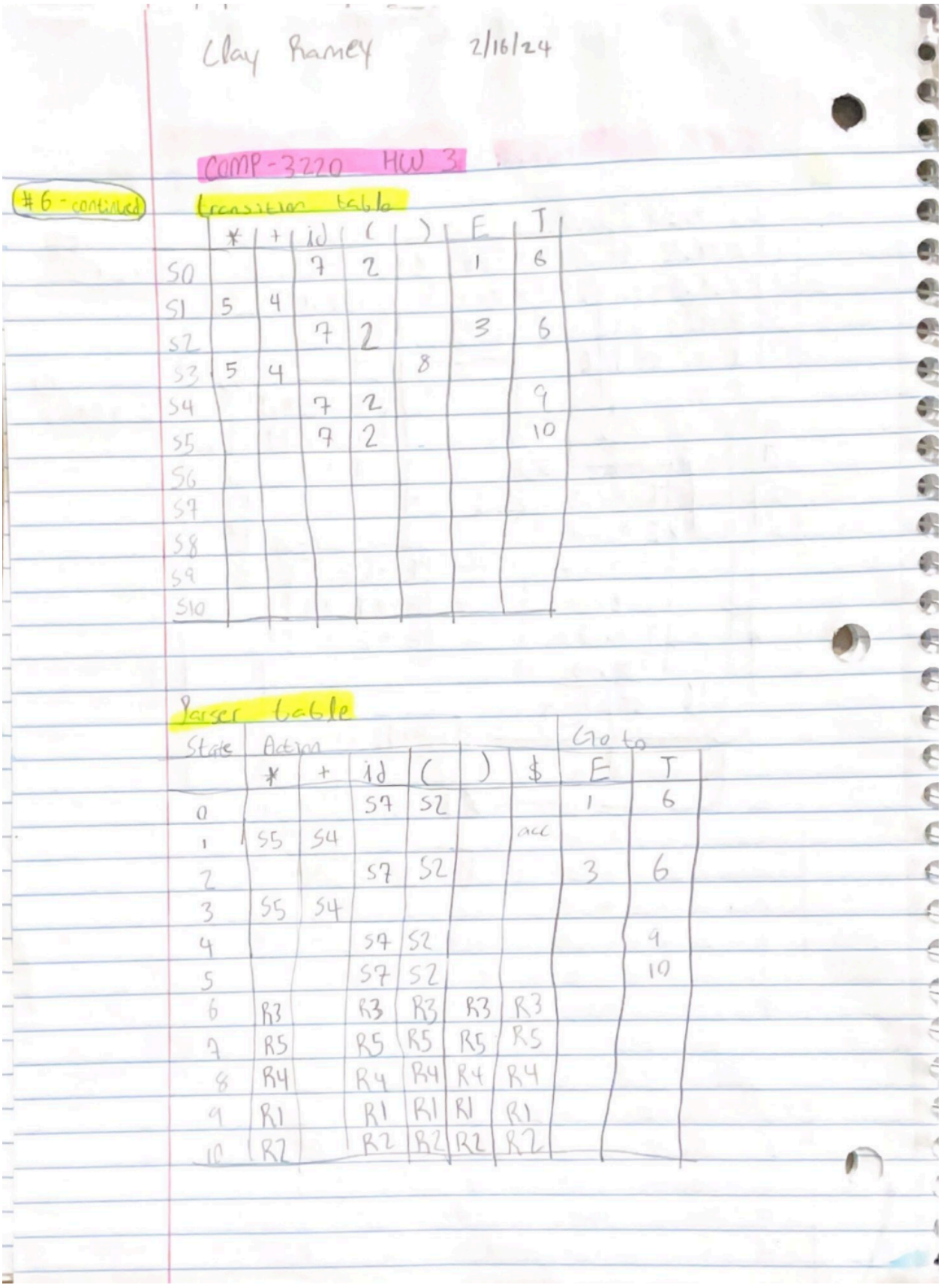
$A \to a\alpha \mid a$

$A \to aA'$

$A' \to \alpha$

$A \to aA'$
$A' \to A'' \mid \epsilon$
$A'' \to Bc \mid c$
$B \to b \mid aB$


Clay Ramey 2/16/24
COMP-3220 HW 3
6
S0
S' → .E$
E → .E+T
E → .E*T
E → .T
T → .(E)
T → .id
S1
S' → E.$
E → E.+T
E → E.*T
S4
E → E+.T
T → .(E)
T → .id
S9
E → E+T
S5
E → E*.T
T → .(E)
T → .id
S10
E → E*T
S2
T → (.E)
E → .E+T
E → .E*T
E → .T
T → .(E)
T → .id
S3
T → (E.)
E → E.+T
E → E.*T
S6
E → T.
S7
T → id.
S8
T → (E).
T
id
(
E
+
*
)



Clay Ramey 2/16/24

COMP-3220 HW 3

#6 - continued

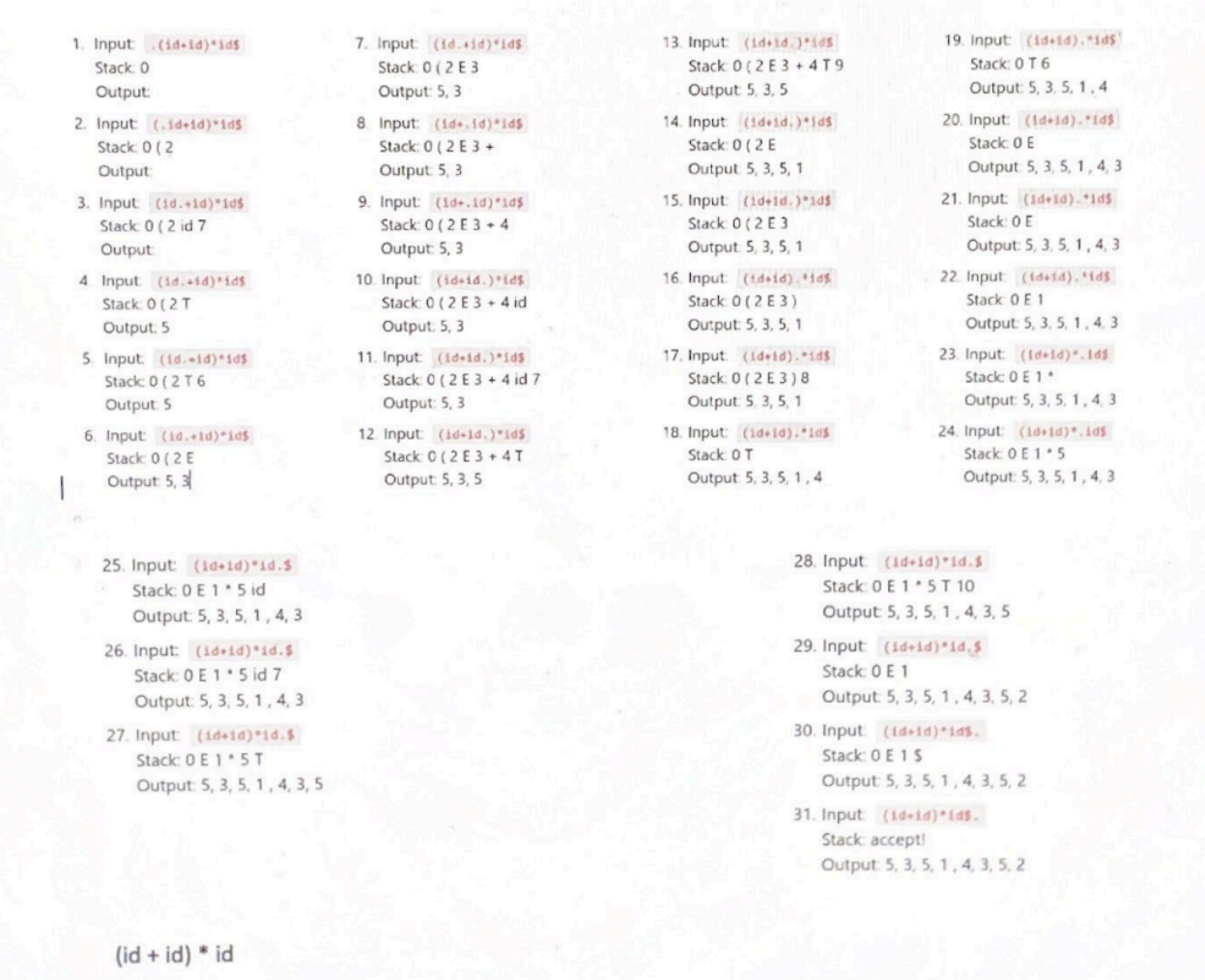
Transition table

	*	+	id	(	)	E	T
S0			7	2		1	6
S1	5	4					
S2			7	2		3	6
S3	5	4			8		
S4			7	2			9
S5			7	2			10
S6							
S7							
S8							
S9							
S10							

Parser table

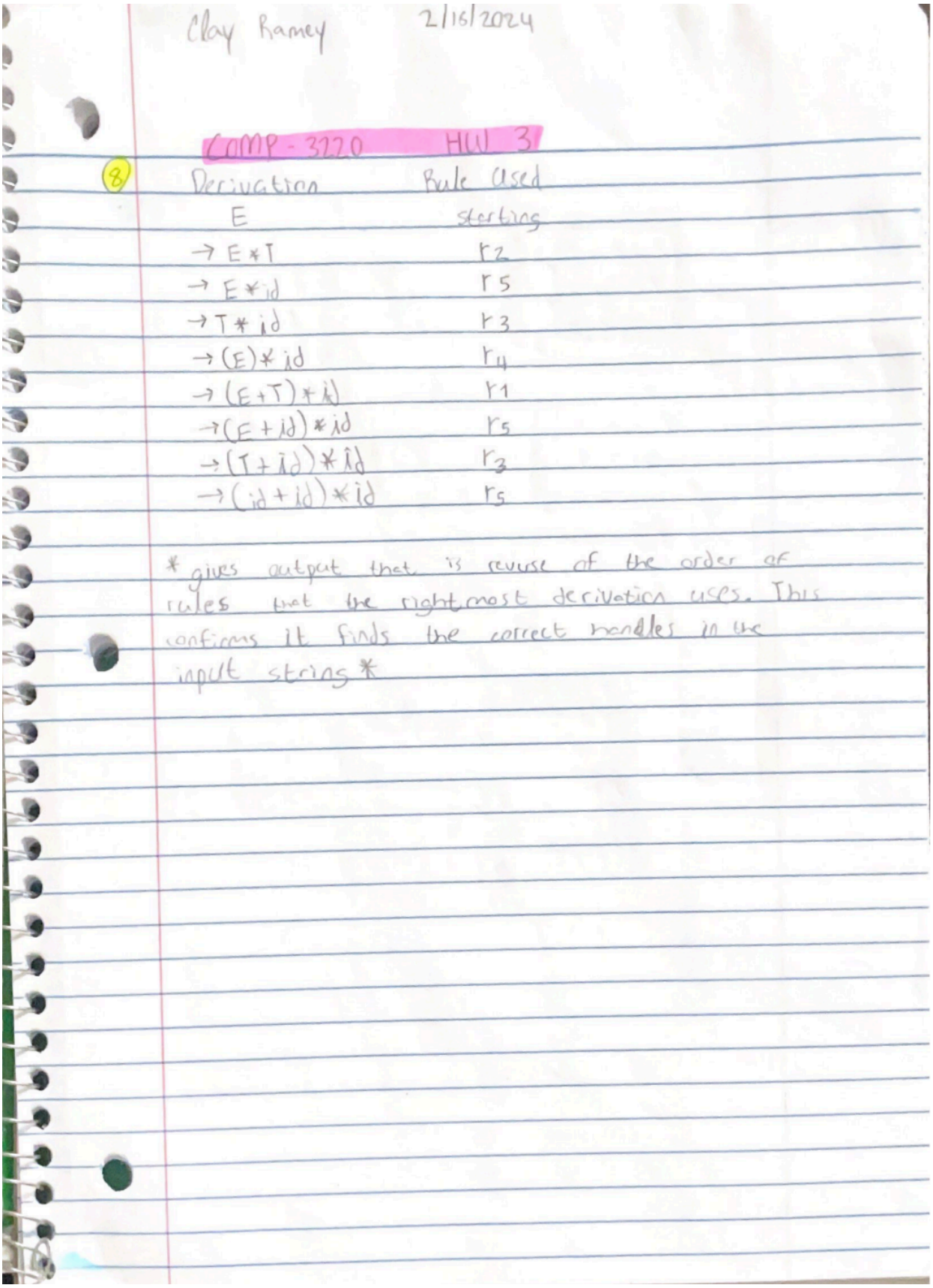
State	Action						Go to	
	*	+	id	(	)	$	E	T
0			S7	S2			1	6
1	S5	S4				acc		
2			S7	S2			3	6
3	S5	S4						
4			S7	S2				9
5			S7	S2				10
6	R3		R3	R3	R3	R3		
7	R5		R5	R5	R5	R5		
8	R4		R4	R4	R4	R4		
9	R1		R1	R1	R1	R1		
10	R2		R2	R2	R2	R2		

COMP 3220 – HW3 – Clay Ramey



I used chatGPT and Chegg for this question. I also used an online PowerPoint for bottom up parsing to help me solve.

Ppt link: https://karkare.github.io/cs335/lectures/07BottomUpParsing.pdf



Clay Ramey 2/15/2024

COMP - 3220 HW 3

(8)

Derivation	Rule Used
E	starting
→ E * T	r2
→ E * id	r5
→ T * id	r3
→ (E) * id	r4
→ (E + T) * id	r1
→ (E + id) * id	r5
→ (T + id) * id	r3
→ (id + id) * id	r5

* gives output that is reverse of the order of rules that the rightmost derivation uses. This confirms it finds the correct handles in the input string *