Homework 2 - CFGs and LR Parsing

Question 1

(a)

Derivation 1: $S \to aSbS \to abS \to abaSbS \to ababS \to abab$

Derivation 2: $S \to aSbS \to abSaSbS \to abaSbS \to ababS \to abab$

(b)

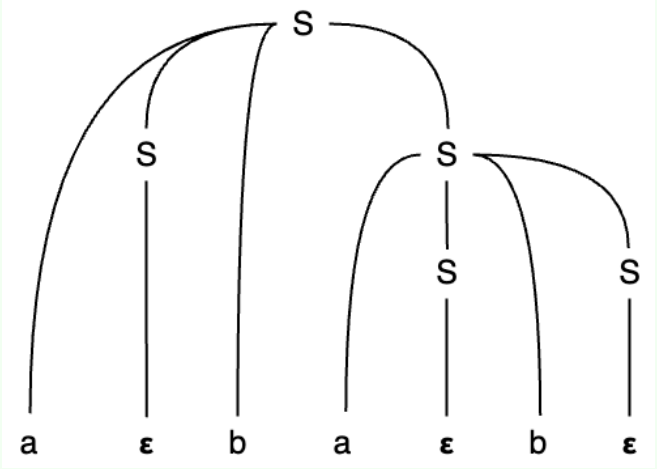
Derivation 1: $S \to aSbS \to aSbaSbS \to aSbaSb \to aSbab \to abab$

Derivation 2: $S \to aSbS \to aSb \to abSaSb \to abSab \to abab$

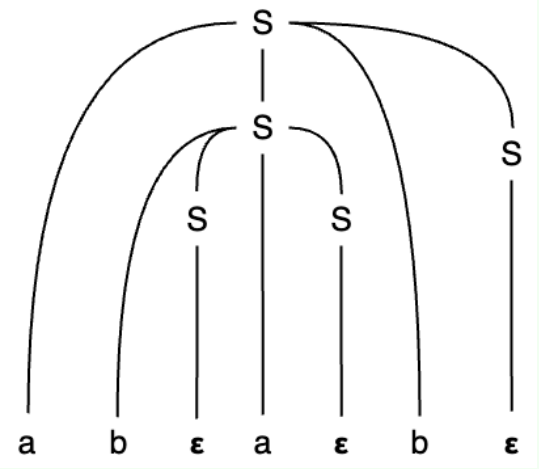
(c)

Note: parse tree for derivation 1 in both leftmost and rightmost are the same (shown on the left), and the parse tree for derivation 2 in both leftmost and rightmost are the same (shown on the right).

Parse tree for Derivation 1s:



Parse tree for Derivation 2s:



Question 2

(a)

Left-most derivation:

$$S \to (L) \to (L, S) \to (S, S) \to (x, S) \to (x, (L)) \to (x, (L, S)) \to (x, (S, S)) \to (x, (x, S)) \\ \to (x, (x, x))$$

(b)

Right-most derivation:

$$S \to (L) \to (L, S) \to (L, (L)) \to (L, (L, S)) \to (L, (L, x)) \to (L, (S, x)) \to (L, (x, x)) \\ \to (S, (x, x)) \\ \to (x, (x, x))$$

(c)

Stack	Input	Action
$	(x, x, x) $	shift
$(	$x, x, x)$ $	shift
$(x	$, x, x)$ $	reduce
$(S	$, x, x)$ $	reduce
$(L	$, x, x)$ $	shift
$(L,	$x, x)$ $	shift
$(L, x	$, x)$ $	reduce
$(L, S	$, x)$ $	reduce
$(L	$, x)$ $	shift
$(L,	$x)$ $	shift
$(L, x	$)$ $	reduce
$(L, S	$)$ $	reduce
$(L	$)$ $	shift
$(L)	$	reduce
$S	$	accept

(d)

Stack	Input	Action
\$	(x, x, x) \$	shift
\$(	$x, x, x)$ \$	shift
\$(x	$, x, x)$ \$	reduce
\$(S	$, x, x)$ \$	shift
\$($S,$	$x, x)$ \$	shift
\$(S, x	$, x)$ \$	reduce
\$(S, S	$, x)$ \$	shift
\$($S, S,$	$x)$ \$	shift
\$(S, S, x	$)$ \$	reduce
\$(S, S, S	$)$ \$	reduce
\$(S, S, L	$)$ \$	reduce
\$(S, L	$)$ \$	reduce
\$(L	$)$ \$	shift
\$(L)	\$	reduce
\$ S	\$	accept

The depth of the stack during self-reduce parse **increases** if we replace the left-recursive production $L ::= L, S$ (max depth: 4) with right-recursive production $L ::= S, L$ (max depth: 6).

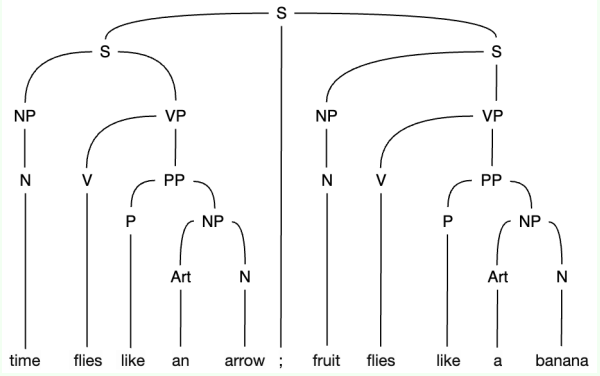
Question 3

(a)

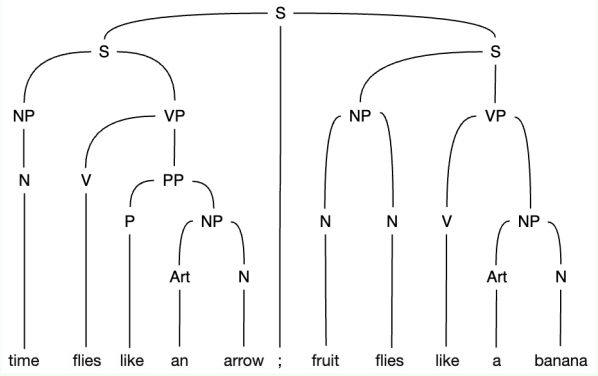
Non-terminals: S (sentence), NP (noun phrase), VP (verb phrase), PP (prepositional phrase), N (noun), V (verb), Art (article), P (preposition)

Grammar Rules:
$S ::= S; S \mid NP \;\; VP$
$NP ::= Art \;\; N \mid N \;\; N \mid N$
$VP ::= V \;\; NP \mid V \;\; PP$
$PP ::= P \;\; NP$
$N ::= \text{time} \mid \text{arrow} \mid \text{banana} \mid \text{fruit} \mid \text{flies}$
$V ::= \text{flies} \mid \text{like}$
$Art ::= \text{a} \mid \text{an} \mid \text{the}$
$P ::= \text{like}$

Parse tree #1:



Parse tree # 2:



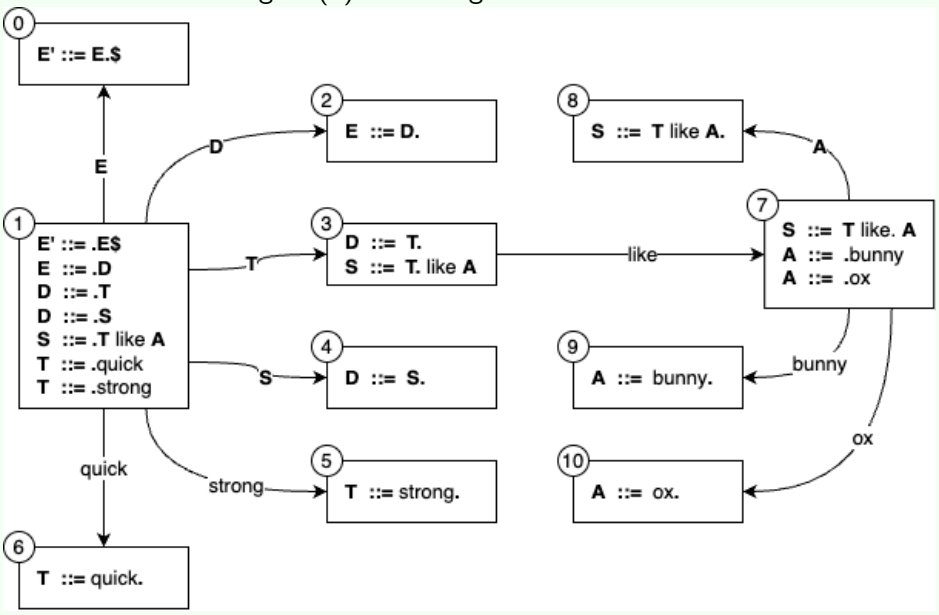
Question 4

(a)

We add a production E' with the epithets symbol E followed by end of file ($), so we have grammar rule:

0. $E' ::= E\ \$$
1. $E ::= D$
2. $D ::= T$
3. $D ::= S$
4. $S ::= T$ like A
5. $T ::=$ quick
6. $T ::=$ strong
7. $A ::=$ bunny
8. $A ::=$ ox

We have the following LR(0) state diagram:



and parse table:

State	Action						Goto				
	like	quick	strong	bunny	ox	$	E	D	T	S	A
0						acc					
1		s6	s5				g0	g2	g3	g4	
2	r1	r1	r1	r1	r1	r1					
3	s7,r2	r2	r2	r2	r2	r2					
4	r3	r3	r3	r3	r3	r3					
5	r6	r6	r6	r6	r6	r6					
6	r5	r5	r5	r5	r5	r5					
7				s9	s10						g8
8	r4	r4	r4	r4	r4	r4					
9	r7	r7	r7	r7	r7	r7					
10	r8	r8	r8	r8	r8	r8					

(b)

FIRST, FOLLOW, nullable for each non-terminal:

Non-terminals	nullable	FIRST	FOLLOW
E	no	quick, strong	$
D	no	quick, strong	$
S	no	quick, strong	$
T	no	quick, strong	like, $
A	no	bunny, ox	$

(c)

SLR parse table:

State	Action						Goto				
	like	quick	strong	bunny	ox	$	**E**	**D**	**T**	**S**	**A**
0						acc					
1		s6	s5				g0	g2	g3	g4	
2						r1					
3	s7					r2					
4						r3					
5	r6					r6					
6	r5					r5					
7				s9	s10						g8
8						r4					
9						r7					
10						r8					

(d)

This grammar is not LR(0) because it has shift-reduce conflicts in state 3 without lookaheads (as shown in the LR(0) parse table). This shift-reduce conflict in state 3 is resolved in the SLR parse table by using the FOLLOW sets to determine when a reduction should occur (as shown in the SLR parse table). Therefore, this grammar is **SLR**.