Spring'23 CPSC 323.02 Compilers & Languages
HW #2 [100 points]

Submission deadline: April 9th, 11:59PM, submit it on Canvas.

*Any HW content shall **NOT** be made publicly accessible without the written consent of the instructor.*

Part I: [20 points] Derivations and Parse Trees.
Four problems in our required Textbook Page 114-116

Problem 3.1
Problem 3.2
Problem 3.3
Problem 3.7

Part II: [5 points] Parsing and Ambiguity

1) Let G be the context-free grammar with productions
   S -> aSaa | B,   B -> bbBcc | C,   C -> bc

   (a) Draw a parse tree for $a^3b^3c^3a^6$.
   (b) Is G ambiguous? If yes, justify and remove the ambiguity. If not, give the reason.

Part III [60 points]: Design CFG: Find context-free grammar that generate the following languages. *(Note: there is NO unique answer for each question)*

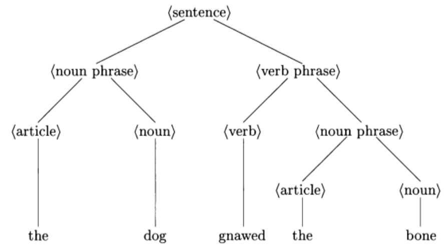
1) L = {$x \in \{0, 1\}^*$ | $x = x^R$ and |x| is even}, x^R is the reverse of the string x.
2) L = {$x \in \{0, 1\}^*$ | the length of x is odd and the middle symbol is 0}
3) L = {binary strings containing the same number of 0s as 1s}
4) L = {binary strings that are palindromes}
5) $\emptyset$
6) L = {$a^nb^ma^{2n}$ | n, m >= 0}

Part IV: [15 points] Construct nondeterministic pushdown automata (PDA) to accept the following languages. *(Note: there is NO unique answer for each question. PDA accept a string if, after processing the string, the stack is empty, and it is in a final state. PDA can be deterministic. In this course, we only require non-deterministic PDA, so the sink state is not necessary in construction.)*

1) L = {ww^R | $w = (a+b)^+$}, w^R is the reverse of w.
2) L = {$w\#w^R$ | w is any binary string}, w^R is the reverse of w.
3) L = {binary strings in the form $0^n1^m0^n$, where n >= 1, and m >= 1}.

Kinsey Vo CPSC 323-02

3.1. Write a leftmost derivation for the dog-bone example of Figure 3.1.



Answer: using leftmost derivation

<sentence> => <noun phrase><verb phrase>
=> <article><noun><verb phase>
=> the<noun><verb phase>
=> the dog<verb phrase>
=> the dog<verb><noun phrase>
=> the dog gnawed<noun phrase>
=> the dog gnawed <article><noun>
=> the dog gnawed the <noun>
=> the dog gnawed the bone

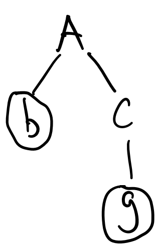
3.2. Given the grammar $G =$

$$A \to Ba \mid bC$$
$$B \to d \mid eBf$$
$$C \to gC \mid g$$

(a) Determine which of the following strings are in $L(G)$. Construct parse trees for those that are.

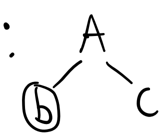
bg *bffd* *bggg* *edfa* *eedffa* *faae* *defa*

√ bg:



(accepted)

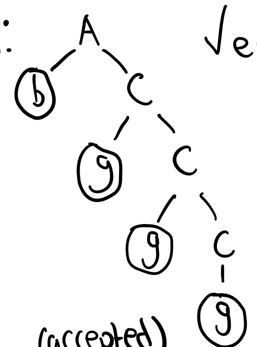
X bffd:



(rejected)

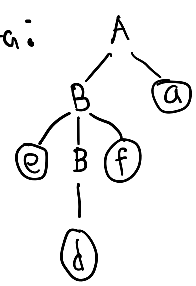
– No terminal symbol 'f' in the rule for C

√ bggg:



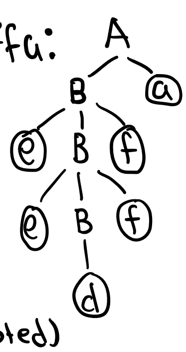
(accepted)

√ edfa:



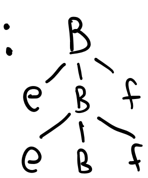
(accepted)

√ eedffa:



(accepted)

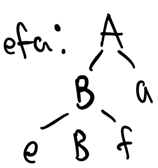
X faae:



(rejected)

– No nonterminal variable starting w/ symbol 'f'

X defa:



(rejected)

– Can't generate concatenation of 'd' and 'e'

(b) Write derivations for those of the given strings that are in $L(G)$.

bg: A => bC
=> bg

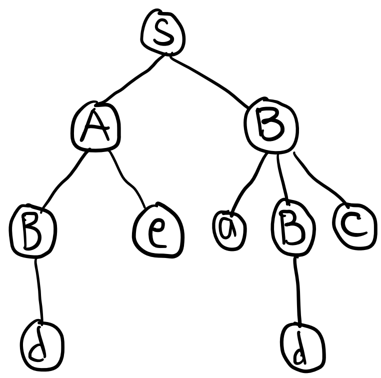
bggg: A => bC
=> bgC
=> bggC
=> bggg

eedffa: A => Ba
=> eBfa
=> eeBffa
=> eedffa

edfa: A => Ba
=> eBfa
=> edfa

3.3. Given the following derivation, construct the corresponding parse tree:

$$S \Rightarrow AB \Rightarrow BeB \Rightarrow deB \Rightarrow deaBc \Rightarrow deadc$$



capital = nonterminals
lowercase = terminals

3.7. Show how the grammar for L_R

$$S \rightarrow aSa \mid bSb \mid c$$

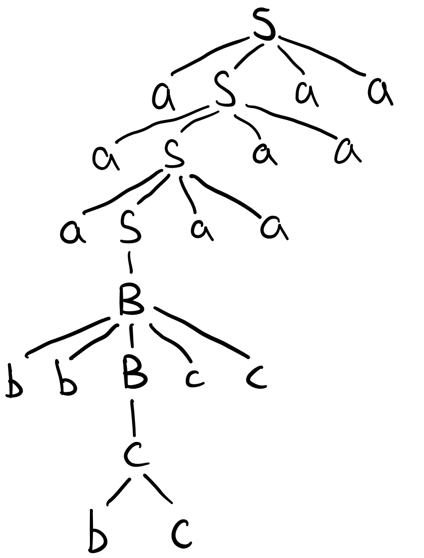
generates the string *aababcbabaa*.

S => aSa
=> aaSaa
=> aabSbaa
=> aabaSabaa
=> aababSbabaa
$\boxed{\text{=> aababcbabaa}}$

1) Let G be the context-free grammar with productions
S -> aSaa | B,   B -> bbBcc | C,   C -> bc

(a) Draw a parse tree for $a^3b^3c^3a^6$.

$a^3b^3c^3a^6$ = aaabbbcccaaaaaa



(b) Is G ambiguous? If yes, justify and remove the ambiguity. If not, give the reason.

No, G is not ambiguous.
We need C -> bc in order to get a string w/ an odd number of b and c.

(b) There might be various ways to show the grammar is unambiguous. One possible way required in this course is to show the structure of parse tree is unique.
Solution: this grammar is unambiguous because of following reasons:
S → aSaa | B, S is replaced by aSaa or B. As shown above tree, the structure of this tree is unique. The left child must have a, and the right child must have aa.
B → bbBcc | C, B is replaced by bbBcc or C. As shown in the above tree, the structure of this tree is unique and expands to the middle side. The left child must have bb, and the right child must have cc.
C → bc, C is replaced by bc, which does not make this grammar ambiguous.
Therefore, there is a unique tree structure. This grammar is unambiguous.

1) $L = \{x \in \{0, 1\}^* \mid x = x^R$ and $|x|$ is even$\}$, x^R is the reverse of the string x.

L contains all strings that are equivalent to their reversed versions

Approach:
1) $\varepsilon \in L$, $S \to \varepsilon$ (empty string character)
2) shortest strings: 00, 11 and production rules $S \to 00 \mid 11$
3) 0000, 1111, 0110, 1001, 011110, 100001, ...
matching relation: $S \to 0S0 \mid 1S1$

$\boxed{\text{Combine all 3 steps: } S \to 0S0 \mid 1S1 \mid 00 \mid 11 \mid \varepsilon}$ or $G = \{\{0, 1\}, \{S\}, S, \{S \to 0S0 \mid 1S1 \mid \epsilon\}\}$

2) $L = \{x \in \{0, 1\}^* \mid$ the length of x is odd and the middle symbol is $0\}$

$\varepsilon \in L$, $S \to \varepsilon$

Shortest strings: 0 and production rules $S \to 0$ (guarantees middle symbol 0)

Possible strings: 000, 001, 100, 101, 01010, 10001, ...

matching relation: $S \to 0S0 \mid 0S1 \mid 1S0 \mid 1S1$

$\boxed{\text{Combine all 3 steps: } S \to 0 \mid 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \mid \varepsilon}$ $G = \{0, 1\}, \{S\}, S, \{S \to 0S0 \mid 1S1 \mid 0S1 \mid 1S0 \mid 0\}\}$

3) $L = \{$binary strings containing the same number of 0s as 1s$\}$

binary strings: $\Sigma = \{0, 1\}$

1) $\varepsilon \in L$, $S \to \varepsilon$

2) 2 cases: Starting w/ 0 or starting w/ 1

$S \to 0S1S$ $\quad$ $S \to 1S0S$

$\boxed{\text{Final: } S \to 0S1S \mid 1S0S \mid \varepsilon}$ $G = \{\{0, 1\}, \{S\}, S, \{S \to 0S1S \mid 1S0S \mid \epsilon\}\}$

4) $L = \{$binary strings that are palindromes$\}$

binary strings: $\Sigma = \{0, 1\}$

Approach:

1) $\varepsilon \in L$, $S \to \varepsilon$

2) Production rules: $S \to 0 \mid 1$

3) 010, 101, ...

matching relation: $S \to 0S0 \mid 1S1$

$\boxed{\text{Final: } S \to 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon}$ $G = \{\{0, 1\}, \{S\}, S, \{S \to 0S0 \mid 1S1 \mid 0 \mid 1 \mid \epsilon\}\}$

5) $\emptyset$

Empty set: $\boxed{S \to S}$ $G = \{\{\}, \{S\}, S, \{S \to S\}\}$

6) $L = \{a^n b^m a^{2n} \mid n, m >= 0\}$

n, m	$a^n b^m a^{2n}$	string
0	$a^0 b^0 a^{2(0)}$	ε
1	$a^1 b^1 a^{2(1)}$	abaa
2	$a^2 b^2 a^{2(2)}$	aabbaaaa
3	$a^3 b^3 a^{2(3)}$	aaabbbaaaaaa

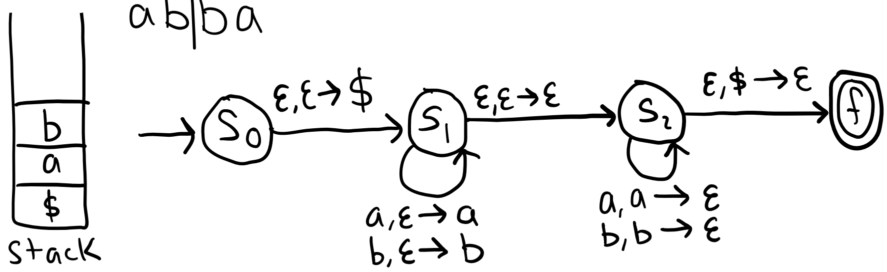
$\varepsilon \in L$, $S \to \varepsilon$

abaa, aabbaaaa, aaabbbaaaaaa

relation: $\boxed{\begin{array}{l} S \to aSaa \mid A \\ A \to bA \mid \varepsilon \end{array}}$ $G = \{\{a, b\}, \{S, A\}, S, \{S \to aSaa \mid A \mid \epsilon, A \to bA \mid \epsilon\}\}$

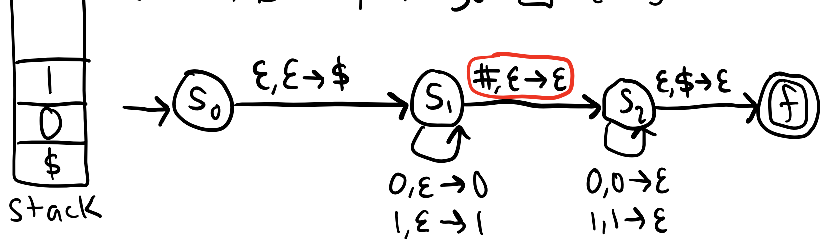
1) $L = \{ww^R \mid w = (a+b)^+\}$, w^R is the reverse of w.

ab|ba

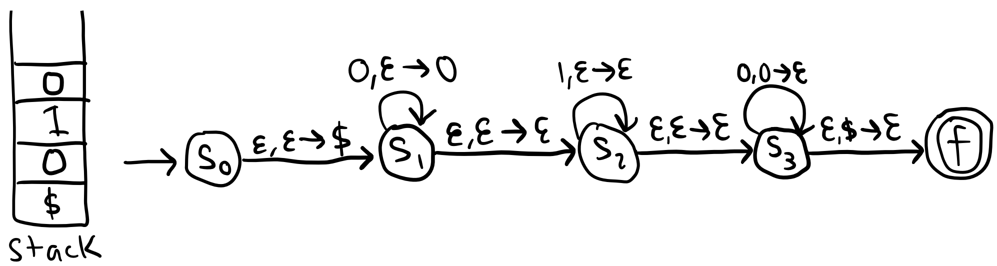


2) $L = \{w\#w^R \mid$ w is any binary string$\}$, w^R is the reverse of w.

w#W, Binary strings: $\Sigma = \{0,1\}$ # = middle position



3) L = {binary strings in the form $0^n1^m0^n$, where $n >= 1$, and $m >= 1$}.



– PDA accepts a string if the stack is empty and it is in the final state after processing the string