Test2

Context-Free Grammar

(a) **Convert** the given regular expression into a context-free grammar. (3 points)

$$(\mathtt{ab}^* + \mathtt{\#a})^* + b$$

Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{\#}\}$. Consider the following languages over Σ. Recall that for a string w, $|w|$ denotes the length of w.

$$L_1 = \{\mathtt{b}^{5i} \text{ , where } i \geq 0\}$$

$$L_2 = \{\mathtt{a}^j \text{ , where } j \geq 0\}$$

$$L_3 = \{x\#y \text{ , where } x \in L_1, y \in L_2 \text{ and } |x| = |y|\}$$

Now solve the following problems.

(b) Your Friend writes the following context-free grammars for the language L_1. **Choose** the correct grammar for L_1. There could be more than one correct grammar. (3 points)

(i) $S \to ASA \mid \mathtt{b}$
$A \to bb$

(ii) $S \to AAAAAS \mid \varepsilon$
$A \to bA \mid b$

(iii) $S \to AASB \mid \varepsilon$
$A \to bb$
$B \to b$

(iv) $S \to bP \mid bbbbQ \mid \varepsilon$
$P \to bbR$
$R \to bQ$
$Q \to bS$

(c) **Write** down a string of any length in L_3. (1 point)

(d) **Give** a context-free grammar for the language L_3. (3 points)

Derivations, Parse Tree, Ambiguity

$$S \to xxS \mid xyS \mid yxS \mid yyS \mid B$$
$$B \to xxY \mid yxY$$
$$C \to xC \mid yC \mid \epsilon$$

a. Give left most derivation for the string xyyxyyxx

b. Sketch the parse tree corresponding to the derivation you gave in (a)

c. Demonstrate that the given grammar is ambiguous by showing one more parse tree apart from the one you already found in (b)

d. Find a string of length nine for the given CFG that has only one parse tree

---

Pushdown Automata

Let $\Sigma = \{0, 1\}$. Consider the following language.

$$L_1 = \{w \text{ contains } 01 \text{ as a substring}\}$$
$$L_2 = \{w = (01)^n 0^{3n} \text{ , where } n \geq 0\}$$

(a) **Give** the state diagram of a pushdown automaton that recognizes L_1. (4 points)

(b) **Give** the state diagram of a pushdown automaton that recognizes L_2. (6 points)

Regular Expression

$$L_1 = \{w \ \ \epsilon\{0,1\}^* | \text{ w does not contain } 00\}$$

a. Design a **Regular Expression** that recognizes the language L_1

b. Convert the following Regular Expression into a NFA and then CFG

$$(0 + 11^*) + 0^*111^*$$