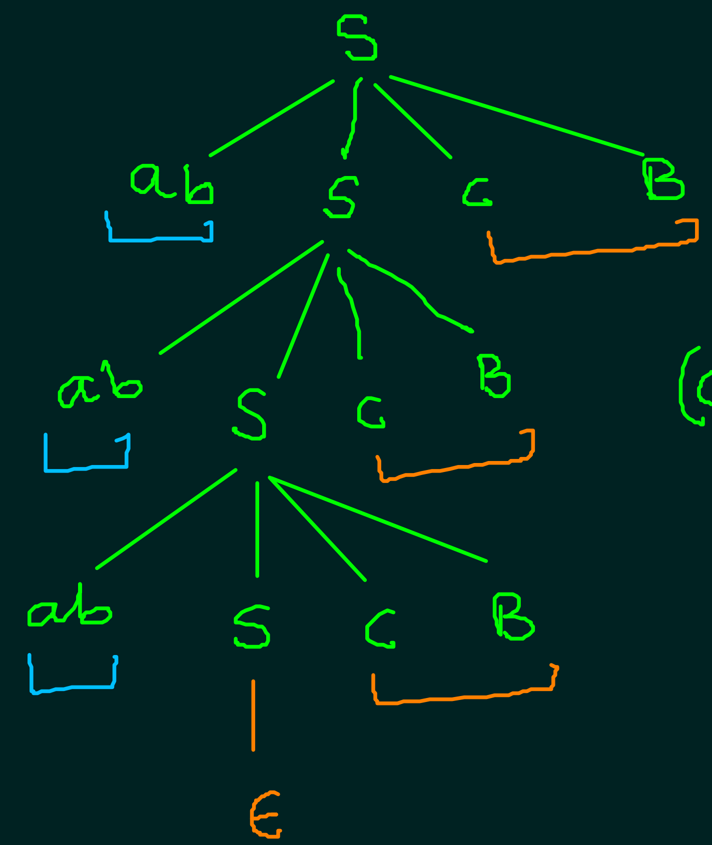
What context-free languages will be generated by the following two (separate) CFGs, $G = \langle \{S, A, B\}, \{a, b, c\}, P, S \rangle$, where P consists of the following production rules?

(a) $S \rightarrow ASB \mid \varepsilon \quad A \rightarrow a \quad B \rightarrow bb \mid b$

(b) $S \rightarrow abScB \mid \varepsilon \quad B \rightarrow bB \mid b$

(a) $L_a = \{ a^m b^n \mid m, n \geq 0,\ m \leq n \leq 2m \}$

(b) $L_b = \{ (ab)^n (cb^+)^n \mid n \geq 0 \}$



$(cB)^n \rightarrow b^+$

Define the context-free grammars for the following context-free languages?
Are your defined CFGs ambiguous / non-ambiguous?

- $L = \{ a^i b^j c^k \mid i,j,k \geq 0, \text{ and } i=j \text{ or } i=k \}$
- $L' = \{ a^i b^j c^k \mid i,j,k \geq 0, \text{ and } i+j = k \}$

0 1 1

① $S \rightarrow S_1 \mid S_2$

$a^n b^n c^m$
$$\begin{cases} S_1 \rightarrow T_1 C \\ T_1 \rightarrow a T_1 b \mid \epsilon \\ C \rightarrow cC \mid \epsilon \end{cases}$$

ambiguous

$a^n b^m c^n$
$$\begin{cases} S_2 \rightarrow a S_2 c \mid A \\ A \rightarrow bA \mid \epsilon \end{cases}$$

② $S \rightarrow aSc \mid A$

$A \rightarrow bAc \mid \epsilon$

non ambiguous

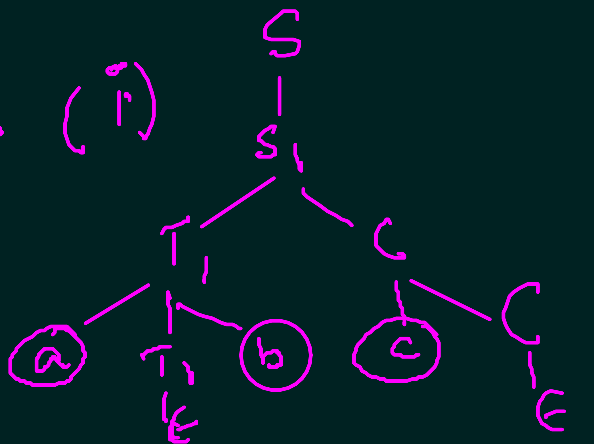
n_1 n_2 $(n_2 + n_1)$

a a a bbb ccc c c c

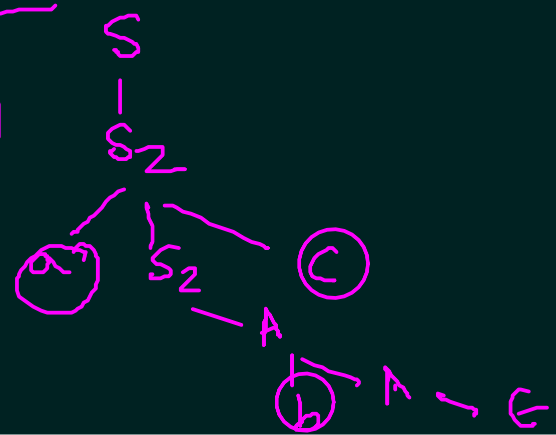
[Give one string which has 2 derivations (left/right)]

abc

(i)



(ii)



Which of the following language(s) is/are context-free? Give justifications.

- $L1 = \{ a^m b^n \mid m,n \geq 0, m = 2n \}$
- $L2 = \{ a^m b^n \mid m,n \geq 0, m \neq 2n \}$
- $L3 = \{ a^m b^m c^{m+n} \mid m,n \geq 1 \}$
- $L4 = \{ a^m b^n c^{m+n} \mid m,n \geq 1 \}$
- $L5 = \{ a^l b^m c^n \mid l \geq 0, l < m \text{ and } l < n \}$ ✓

① CF $a^{2n} b^n \equiv (a^2)^n b^n \equiv (aa)^n b^n \Rightarrow S \rightarrow aaSb \mid \epsilon$ ✓ +PDA

② CF $T \rightarrow AS \mid SB$

$A \rightarrow aA \mid a$ $B \rightarrow bB \mid b$

$(m \neq 2n)$ ✗ $\left.\begin{matrix} m=0 \\ n=0 \end{matrix}\right\}$ $m = 2n$

$(m,n \geq 1)$ $\left\{\begin{matrix} m=0, & n=1 \checkmark \\ m=1, & n=0 \checkmark \end{matrix}\right.$ ✗

③ NOT CFL. k

$a^k b^k c^{k+1} \in \alpha$

$= u\underbrace{vxy}_{\leq k}z$

(ab)

$uv^ixy^iz \notin \alpha$

$i = 3, 2, 0$

contr.

④ CFL

$a^m b^n c^n c^m$ ✓

n, m

$a^m b^n c^m d^n$ ✗

⑤ NOT CFL

k

$a^k b^{k+1} c^{k+1} \in \alpha$

$\boxed{u\underbrace{vxy}_{\leq k}z}$ ✓

$\leq k$ ✗

$i = 2$ $uv^2xy^2z \notin \alpha$ ✓

① $v = a^b$

$y = a^q$

$a^{k+b+q} b^{k+1} c^{k+1} \notin \alpha$

① ② ③ ④ ⑤

$\underbrace{aa \ldots a}_{k} \underbrace{bbb \ldots b}_{k+1} \underbrace{cc \ldots c}_{k+1}$

Consider the two CFGs G and G' with the start symbols S and S' and with the only productions:

Productions of G : $S \rightarrow aS \mid B,\ B \rightarrow bB \mid b$

Productions of G′ : $S' \rightarrow aA' \mid bB',\ A' \rightarrow aA' \mid B',\ B' \rightarrow bB' \mid \varepsilon$

Prove that, $L(G) \subset L(G')$, i.e., L(G) is strictly contained in L(G').

$L(G) = \mathcal{L}(\underline{a^* b^+})$ ×

$\rightarrow a$ ✓ $\in \mathcal{L}(G')$

$\notin \mathcal{L}(G)$

$L(G') = \mathcal{L}\left[\underline{a^+ b^*} + \underline{b^+}\right]$ ✓

$\mathcal{L}(G) \subsetneq \mathcal{L}(G')$

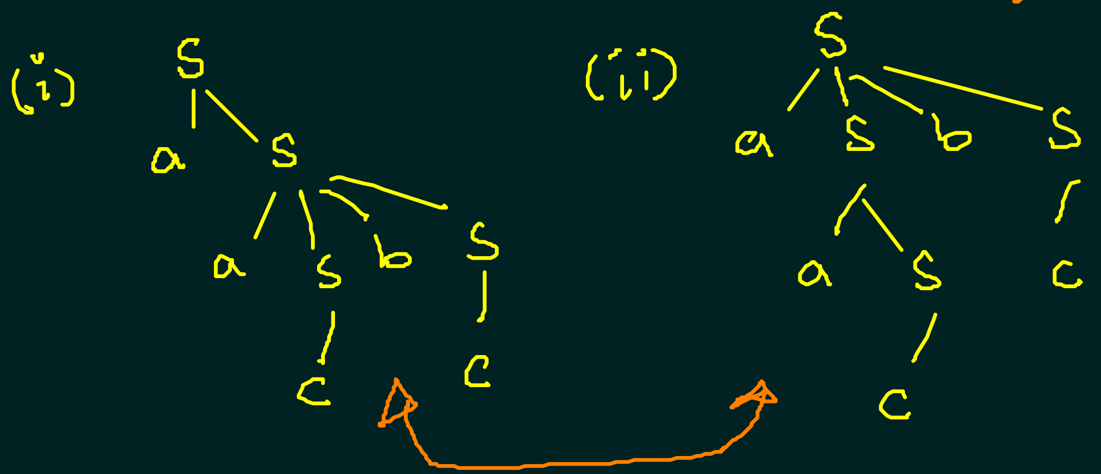
To prove, $\mathcal{L}(G) \subseteq \mathcal{L}(G')$

Induction over length of derivation

Prove that, the following context-free grammar, G = ({S}, {a, b, c}, P, S), is ambiguous.
Here, the production rules (P) are given as: $S \rightarrow aS \mid aSbS \mid c$.
Construct a non-ambiguous grammar, G', that derives the same language. Also prove L(G) = L(G').

String: a a c b c

ambiguity



Unambiguous Grammar:

$$\left\{ \begin{array}{l} S \rightarrow aST \mid c \\ T \rightarrow bS \mid \epsilon \end{array} \right\}$$

$L(G') = L(G)$

$$\left\{ \begin{array}{l} S \rightarrow T \mid U \\ T \rightarrow aTbT \mid c \\ U \rightarrow aS \mid aTbU \end{array} \right\}$$

Given the following languages over the alphabet {a, b}, design one-state PDAs that accepts by empty stack (separate PDA for each one).

- L =(a+b)*b
- L' = all palindromes over {a,b}

Since L is a regular languages, can you directly present the left linear and the right linear grammar for L and then formally derive the NFAs from these grammars?