1. Give a pushdown automaton for accepting the following language. State your acceptance criterion.

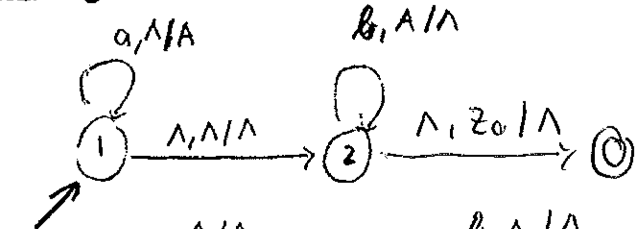
A $\{a^ib^j : i \neq j \text{ and } i, j \geq 0\}.$

Hint: First construct a 2-state (non-deterministic) pushdown automaton for accepting

$$\{a^ib^j : i = j \text{ and } i, j \geq 0\} \quad = \quad \{a^nb^n : n \geq 0\} . \qquad B$$

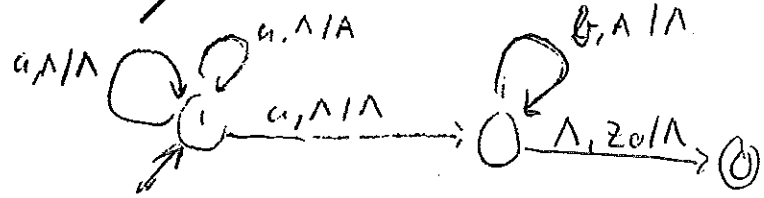
Then modify this automaton in two ways for accepting the languages $\{a^ib^j : i > j \geq 0\}$ and $\{a^ib^j : 0 \leq i < j\}$, respectively. Now that is enough hints! Don't forget the acceptance criterion! C

A



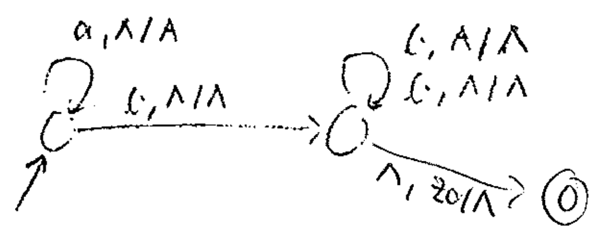
either criterion

B



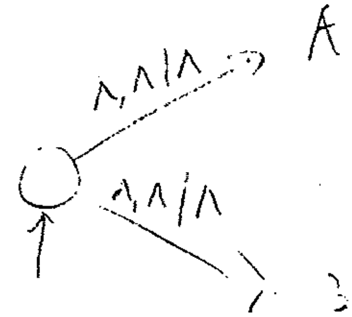
i>j.

C



i<j.

C :



2. Give a complete proof that the following language is not context free:

$$L = \{a^n b a^n b a^n : n \geq 0\}.$$

Hint: Use the following version of the pumping lemma for context free languages:

If L is context free, then there exists N such that: If $u \in L$ and $|u| \geq N$, then u can be written as $vwxyz$ for which

i • $|wxy| \leq N$,
ii • $|wy| > 0$, and
iii • $vw^m xy^m z \in L$ for all $m \geq 0$.

Assume L CFL.
Then PL holds.
Let N be constant of PL
Let $u = a^N b a^N b a^N$.
Since $|u| \geq N$ and $u \in L$
$u = vwxyz$ s.t. i) thru iii) hold.

A) – Case wy contains a "b". Then vxz contains < 3 "b" and can't be in L.

B) – Case $wy \in a^+$: Since i) holds wy can touch ≤ 2 blocks of a's. Thus vxz begins with $a^N b$ or ends with $b a^N$. The # of a's in vxz is $3N - |wy|$. Thus $vxz \notin L$.

Since $|wy| > 0$ either A) or B) must hold!
In both cases we arrived at a contradiction.

⇒ L is not CFL

3. Give a concise description of the language generated by the following grammar.

$$
\begin{aligned}
S &\to ZN \\
Z &\to 00ZM \mid \Lambda \\
N &\to MN1 \mid \Lambda \\
M &\to a \mid b
\end{aligned}
$$

(The non-terminals are upper case and $\{0, 1, a, b\}$ are terminal.)

Hint: Give a concise description of the language generated by M, N, Z and S, respectively.

The kind of descriptions we want are regular expressions, or mathematical formulas such as $\{a^n b^n c^{2n} : n \geq 0\}$.

Let $L(A) = \{x \in \{0,1,a,b\}^* : A \overset{*}{\underset{G}{\Rightarrow}} x\}$

$L(M) = a + b$

$L(N) = \{(a+b)^n 1^n : n \geqslant 0\}$

$L(Z) = \{0^{2n} (a+b)^n : n \geqslant 0\}$

$L(S) = L(Z) \cdot L(N)$

$\{0^{2i} (a+b)^i \cdot (a+b)^j 1^j, \; i, j \geqslant 0\}$

4. Convert the following context-free grammar into Chomsky Normal Form. Show your steps!

$$S \rightarrow ABC$$
$$A \rightarrow aA \mid C \mid \Lambda$$
$$B \rightarrow Bb \mid b$$
$$C \rightarrow bCc \mid \Lambda$$

1. find nullable symbols: {A,C}
2. remove Λ productions:
   S → ABC | BC | AB | B
   A → aA | a | C
   B → Bb | b
   C → bCc | bc
3. eliminate unit productions:
   S → ABC | BC | AB | Bb | b
   A → aA | a | bCc | bc
   B → Bb | b
   C → bCc | bc
4. restrict rules to CNF form, adding new productions as necessary.
   S → XC | BC | AB | BE | b
   A → DA | a | EY | EF
   B → BE | b
   C → EY | EF
   D → a
   E → b
   F → c
   X → AB
   Y → CF

5. Can the following grammar

$$\begin{aligned} S &\to AB \mid BC \\ A &\to BA \mid a \\ B &\to CC \mid b \\ C &\to AB \mid a \end{aligned}$$

generate the string *bbab*? Use the CYK algorithm. Show your work by giving the table. How can the table be used to answer the question whether the string can be generated or not!

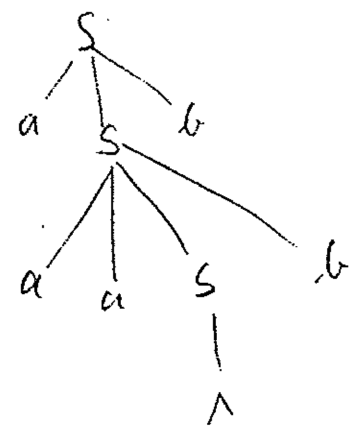
w_1	...		w_4
b	b	a	b
B	B	A C	B
∅	A, S	S, C	
A	S, C		
S, C			

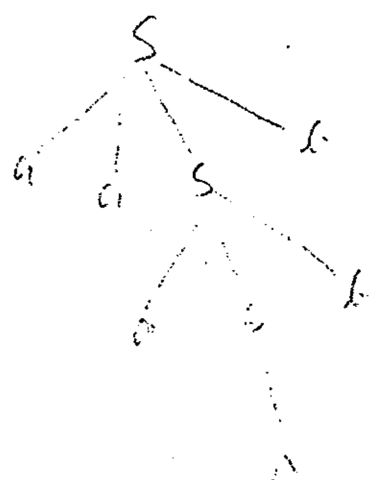
contains all non-terminals deriving $w_1 w_2 \ldots w_4$

S does the job and thus $bbab \in L(G)$

6. Show that the following CFG grammar is ambiguous and find an equivalent unambiguous CFG.

$$S \to aSb \mid aaSb \mid \lambda \ .$$





non-ambiguous: $S \to aSb \mid T$

$T \to aaTb \mid \Lambda$

7. Give an algorithm that decides whether a given Finite State Automaton accepts a finite language. Does your algorithm have polynomial running time?

I) Run the minimization alg. for the FA. Now all states except possible the trap state are reachable from start state and can reach a final state.

II) For all states $s \neq$ trap-state

Check whether there is cycle from s to s (by using a graph search alg).

III) $L(M)$ infinite iff $\exists$ such a cycle.

Minimization alg in poly time.
Searches in poly time.
Thus the above alg is poly time.

8. Give an algorithm that decides whether a given Context Free Grammar generates the empty language. Does your algorithm have polynomial running time?

A) Change all non-terminals to λ. Call resulting grammar G'.
Check whether $\lambda \in L(G')$, i.e.
S nullable

$L(G) \neq \{\}$ iff S nullable in G'.
↑
original

The above alg is poly time
Reasoning similar to reasoning of the following alg.

B) Let N_0 = "all non-terminals"

For $i = 1$ step 1 do

$N_i := N_{i-1}$
For all terminals not in N_{i-1}
add them to N_i if there is a production whose r.h.s. lies in N_{i-1}^*

loop poly in size of grammar

until $N_i = N_{i-1}$.

$L(G) = \{\}$ iff $S \notin N_i$

Loop executed only # of non-term. many times

poly time alg