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

\[ A = \{a^i b^j : i \neq j \text{ and } i, j \geq 0 \}. \]

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

\[ \{a^i b^j : i = j \text{ and } i, j \geq 0 \} = \{a^n b^n : n \geq 0 \}. \]

Then modify this automaton in two ways for accepting the languages \( \{a^i b^j : i > j \geq 0 \} \) and \( \{a^i b^j : 0 \leq i < j \} \), respectively. Now that is enough hints! Don’t forget the acceptance criterion!
2. Give a complete proof that the following language is not context free:

\[ L = \{a^n b a^n : n \geq 0\} \]

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 \( uvwyz \) for which

i. \( |wyz| \leq N \),
ii. \( |wy| > 0 \), and
iii. \( uv^m wz^m \in L \) for all \( m \geq 0 \).

Assume \( L \) is a CFL.
Then \( PL \) holds.
Let \( N \) be constant of \( PL \)
Let \( u = a^N b a^N \in L \).

Since \( |u| \geq N \) and \( u \in L \),
\( u = vwyxz \) s.t. i) then iii) hold.

A) - Case \( wy \) contains a "L". Then \( vxz \) contains \( \geq 3 "L" \) and can't be in \( L \).

B) - Case \( wy \) is \( a^t \): Since i) holds \( wy \) can contain \( \leq 2 \) blocks of \( a's \). Thus \( vxz \) begins with \( a^N b a^N \) ends with \( a^N \). \# of \( a's \) in
\( vxz \) is \( 3N - |wy| \). Thus \( vxz \notin L \).

Since \( |wy| \geq 0 \) either A) or B) must hold.
In both cases we have arrived at a contradiction.

\[ \therefore L \text{ is not CFL} \]
3. Give a concise description of the language generated by the following grammar.

\[ S \rightarrow ZN \]
\[ Z \rightarrow 00ZM \mid \Lambda \]
\[ N \rightarrow MN1 \mid \Lambda \]
\[ M \rightarrow a \mid b \]

(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 \( \{a^n b^n c^{2n} : n \geq 0\} \).

\[ L(A) = \{ x \in \{0,1,a,b\}^* : A \xrightarrow{\varepsilon} x \} \]

\[ L(M) = a + b \]

\[ L(N) = \{ (a+b)^n 1^n : n > 0 \} \]

\[ L(Z) = \{ 0^{2n} (a+b)^n : n \geq 0 \} \]

\[ L(S) = L(Z) \cdot L(N) \]

\[ \{ 0^{2^k} (a+b)^i (a+b)^j : i, j \geq 0 \} \]
4. Convert the following context-free grammar into Chomsky Normal Form. Show your steps!

\[ S \rightarrow A BC \]
\[ A \rightarrow aA \mid C \mid \Lambda \]
\[ B \rightarrow Bb \mid b \]
\[ C \rightarrow bCc \mid \Lambda \]

1. find nullable symbols: \{A, C\}
2. remove \( \Lambda \) productions:
   \[ S \rightarrow ABC \mid BC \mid AB \mid B \]
   \[ A \rightarrow aA \mid a \mid C \]
   \[ B \rightarrow bB \mid b \]
   \[ C \rightarrow bCc \mid bc \]
3. eliminate unit productions:
   \[ S \rightarrow ABC \mid BC \mid AB \mid bB \]
   \[ A \rightarrow aA \mid a \mid bCc \mid bc \]
   \[ B \rightarrow bB \mid b \]
   \[ C \rightarrow bCc \mid bc \]
4. restrict rules to CNF form, adding new productions as necessary.
   \[ S \rightarrow X C \mid BC \mid AB \mid bE \mid b \]
   \[ A \rightarrow DA \mid a \mid EY \mid EF \]
   \[ B \rightarrow bE \mid b \]
   \[ C \rightarrow EY \mid EF \]
   \[ D \rightarrow a \]
   \[ E \rightarrow b \]
   \[ F \rightarrow c \]
   \[ X \rightarrow AB \]
   \[ Y \rightarrow CF \]
5. Can the following grammar generate the string $bbbab$? 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?

$$S \rightarrow AB \mid BC$$
$$A \rightarrow BA \mid a$$
$$B \rightarrow CC \mid b$$
$$C \rightarrow AB \mid a$$

$w_1 \ldots w_n$

```
  b  b  a  b
  B  B  A  C  B
  Ø  A  S  S  C
  A  S  C
  S  C
  S
```

contains all non-terminals deriving $w_1w_2 \ldots w_n$

$S$ does the job and hence $L(G)$. 

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

\[ S \rightarrow aSb | aaSb | \lambda . \]

new ambiguous: \[ S \rightarrow aSb | T \]
\[ T \rightarrow aTb | \lambda \]
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 nu trap state are reachable from start state
   and can reach a final state.

II) For all states s ≠ trap-state
    Check whether there is cycle from s to s (by using a graph search alg).

III) L(M) infinite iff ∃ s 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 \( \rightarrow \). Call resulting grammar \( G' \).

Check whether \( S \in \operatorname{L}(G') \), i.e., \( S \) nullable.

\[ \operatorname{L}(G) \neq \{ \} \] iff \( S \) nullable in \( G' \).

The above alg is poly time.

Reasoning similar to reasoning of the following alg.

B) Let \( N_0 = " \text{all non-terminals}" \)

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's a production whose r.l.s. lies in \( N_{i-1} \).

Until \( N_i = N_{i-1} \).

\[ \operatorname{L}(G) \neq \{ \} \] iff \( S \in N_i \)

Loop poly time only

\( \text{loop executed only # of non-term. many times} \)

\( \text{poly time only} \)