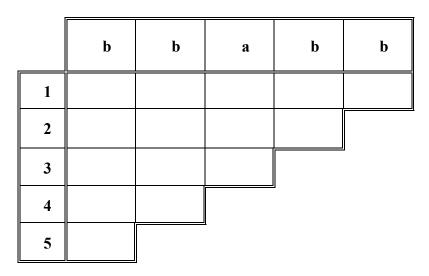
- 1. Write a Context Free Grammar for the language  $L = \{ a^k b^m c^n | k = n + m, or m = k + n, or n = k + m, k > 0, m > 0, n > 0 \}.$
- **2.** Consider the language

 $L = \{ a^n b^{n!} | n > 0 \}.$ Use the Pumping Lemma for Context-Free Languages to show that L is not context-free.

- 3. Present the CKY recognition matrix for the string **bbabb** assuming the Chomsky Normal Form grammar, G = ({S,A,B,C,D}, {a,b}, R, S), specified by the rules R:

  - $C \rightarrow a$
  - $D \rightarrow b$



4. Choosing from among (Y) yes, (N) No, categorize each of the following closure properties. No proofs are required.

Problem / Language Class (C)	Regular	Context Free
Closed under union with Context Free languages?		
Closed under quotient with languages of its own class (C), i.e., L1/L2		
Closed under difference with languages of its own class (C), i.e., (difference (L1, L2) = L1 – L2 )?		
Closed under intersection with Closed with languages of its own class?		

5. Prove that any class of languages, C, closed under union, concatenation, intersection with regular languages, homomorphism and substitution (e.g., the Context-Free Languages) is closed under Erase Middle with Regular Sets (em), where L ∈ C, R is Regular, L and R are over the alphabet Σ, and L em R = { xz | x,z ∈ Σ<sup>+</sup> and ∃y ∈ R, such that xyz ∈ L }. You may assume substitution f(a) = {a, a'}, and homomorphisms g(a) = a' and h(a) = a, h(a') = λ. Here a∈Σ and a' is a distinct new character associated with each a∈Σ.

You must be very explicit, describing what is produced by each transformation you apply.

- 6. Consider the CFG G = ( { S, T }, { a, b }, R, S ) where R is:  $S \rightarrow a T T | T S | a$  $T \rightarrow b S T | b$
- **a.)** Present a pushdown automaton that accepts the language generated by this grammar. You may (and are encouraged) to use a transition diagram where transitions have arcs with labels of form  $\mathbf{a}, \alpha \rightarrow \beta$  where  $\mathbf{a} \in \Sigma \cup \{\lambda\}, \alpha, \beta \in \Gamma^*$ . Note: I am encouraging you to use extended stack operations.

What parsing technique are you using?	
(Circle one) <b>top-down</b> or <b>bottom-up</b>	
How does your PDA accept? (Circle	
one) final state or empty stack or final	
state and empty stack	
What is the <b>initial state</b> ?	
What is the <b>initial stack content</b> ?	
What are your <b>final states</b> (if any)?	

What parsing technique are you using? (Circle one) <u>top-down</u> or bottom-up How does your PDA accept? (Circle one) final state or <u>empty stack</u> or final state and empty stack What is the initial state? What is the initial stack content? What are your final states (if any)? b.) Now, using the notation of IDs (Instantaneous Descriptions, [q, x, z]), describe how your PDA accepts strings generated by G. 7. Consider the context-free grammar  $G = ( \{ S, A, B \}, \{ a, b \}, R, S )$ , where R is:

 $S \rightarrow SAB \mid BA$  $A \rightarrow AB \mid a$  $B \rightarrow bS \mid b \mid \lambda$ 

a.) Remove all λ-rules from G, creating an equivalent grammar G'. Show all rules.
 *Nullable* = { }
 G':

b.) Remove all unit rules from G', creating an equivalent grammar G''. Show all rules.
Unit(S)=Chain(S)={ }; Unit(A)={ }; Unit(B)={ }
G'':

c.) Convert grammar G'' to its Chomsky Normal Form equivalent, G'''. Show all rules.G''':

In exam I may have some Unproductive non-terminals and some Unreachable ones.