Use only one side of the paper and start each problem on a new page!!

Please show all work to receive ANY credit!!!!

1. (25 points) Consider the grammar $G$ shown below, where $S$ is the start symbol:

   $S \rightarrow AB | \epsilon$

   $A \rightarrow SB | x$

   $B \rightarrow Ay | y$

   a. (6 points) Find all strings in $L(G)$ that can be produced by derivations of 5 or fewer steps. A string is in $L(G)$ if it consists of only terminal symbols.

   b. (4 points) Is $xyxy$ a string in the language? Why or why not?

   c. (15 points) Redesign the grammar given above into an equivalent form that is suitable for top-down parsing by first removing any left recursion and then removing any $\epsilon$ moves.

2. (25 points) The grammar:

   $S \rightarrow SS | (S) | \epsilon$

   has a cycle, since the derivation

   $S \rightarrow SS \rightarrow S$

   can occur. Develop an algorithm that when applied to a grammar will remove cycles. Demonstrate that your algorithm works by applying it to the above grammar. Hint: It may be helpful for you to consider the strings being generated by the grammar to answer this question.
3. (20 points) Consider the grammar, with FIRST and FOLLOW computed for each non-terminal in the grammar:

1. A ---> BA'
2. A' ---> oBA'
3. A' ---> ε
4. B ---> D
5. B ---> ( C )
6. C ---> DC'
7. C' ---> aDC'
8. C' ---> ε
9. D ---> x

FIRST(A) = { x, ( }  FOLLOW(A) = { $ }
FIRST(A') = { o, ε }  FOLLOW(A') = { $ }
FIRST(B) = { x, ( }  FOLLOW(B) = { $, o }  FOLLOW(C) = { ) }
FIRST(C) = { x }  FOLLOW(C') = { ) }  FOLLOW(C') = { ) }
FIRST(D) = { x }  FOLLOW(D) = { $, o, a, ) }

a. (18 points) As part of the design of a top-down parser, construct the LL(1) parsing table for the grammar. Use the rule numbers to identify your table entries.

b. (2 points) Is the grammar LL(1)? Why or why not?

4. (30 points) In LaTeX, it is possible to generate all kinds of mathematical equations. Equation specifications are given delineated by $$$. Some sample equations and their corresponding output are shown below, with ^ for superscript and _ for subscript:

1. $$X^{(2)}$$
2. $$X_{(2)}$$
3. $$X^{(Y^{(2)})}$$
4. $$X_{(Y_{(2)})}$$
5. $$X^{(Y^{(2)})}$$
6. $$X_{(Y_{(2)})}$$
7. $$X^{(Y)_{(2)}}$$
8. $$X_{(2)^{(Y)}}$$

X^2 \ x_2 \ X^{y^2} \ x_{y_2} \ X^y_{x^2} \ x^{y_2} \ x^{y_{x^2}} \ X^{y_{x^2}} \ x^{y_{x^2}}

a. (25 points) Design a context free grammar for the above equations that can specify simple boolean expressions of the form TERM REL_OP TERM, where a TERM can be: a variable that may contain subscripts and superscripts; an integer; or a real number. Possible relational operations include <, <=, >, >=, <>, and <=. Here are the first few grammar rules:

EQUATION_SPEC --> "$" TERM REL_OP TERM "$"
  | "$" TERM "$"

Show that your grammar works by doing derivations for equations 1, 4, and 8.

b. (5 points) Extend your answer in part a. to be general to any level of superscripting and subscripting, e.g., a superscript of a superscript of a superscript, etc.
#1a. Strings in L(G) are: epsilon, xx, xy, and yx

#1b. \[ S \rightarrow AA \rightarrow SBA \rightarrow AABA \rightarrow SBABA \rightarrow BABA \rightarrow yABA \rightarrow yxBA \rightarrow yxyA \rightarrow yxySB \rightarrow yxyB \rightarrow yxyAy \rightarrow yxyxy \]

#1c. \[ S \rightarrow AA \mid \text{epsilon} \]
\[ A \rightarrow BA' \mid xA' \mid B \mid x \]
\[ A' \rightarrow ABA' \mid AB \]
\[ B \rightarrow xA'yB' \mid yB' \mid xyB' \mid xA'y \mid xy \mid y \]
\[ B' \rightarrow A'yB' \mid yB' \mid A'y \mid y \]

#2. TRY IT AGAIN!!! (SEE NEXT PAGE)

#3. \[
\begin{array}{cccc}
| a | o | x | ( | ) | S | \\
\hline
A | | | | | | \\
A' | | | | | | \\
B | | | | | | \\
C | | | | | | \\
C' | | | | | | \\
D | | | | | | \\
\end{array}
\]

WHERE WOULD THE SYNC ACTIONS GO?

#4a. TERM \rightarrow SINGLE \mid DOUBLE \mid NESTED

SINGLE \rightarrow VARIABLE SUBSUP CURLYB

DOUBLE \rightarrow SINGLE SUBSUP CURLYB

NESTED \rightarrow VARIABLE SUBSUP "(" SINGLE ")"

SUBSUP \rightarrow "_" \mid "^"

CURLYB \rightarrow "(" VARIABLE ")"

VARIABLE \rightarrow ID \mid INT \mid REAL

REL_OP \rightarrow ...etc...

#4b. HOW WOULD THE ABOVE GRAMMAR BE CHANGED?
#2  \[ S \rightarrow SS \mid (S) \mid \varepsilon \]

**STEP 1:** REMOVE LR.

\[
S \rightarrow (S)S' \mid S' \\
S' \rightarrow SS' \mid \varepsilon
\]

**STEP 2:** REMOVE E-MOVES - \(\overline{\varepsilon} S'\)

\[
S \rightarrow (S)S' \mid S' \mid (S) \mid \varepsilon \quad \text{CYCLE STILL POSSIBLE} \\
S' \rightarrow SS' \mid S
\]

**STEP 3:** REMOVE E-MOVES - \(\overline{\varepsilon} S\)

\[
S \rightarrow (S)S' \mid S' \mid (S) \mid (S)S' \mid (S') \mid (S)S' \mid (S)
\]

\[
S' \rightarrow S' \mid \varepsilon \quad \text{REDUNDANT, SO OMIT!}
\]