Cairo University
Faculty of Computers and Information

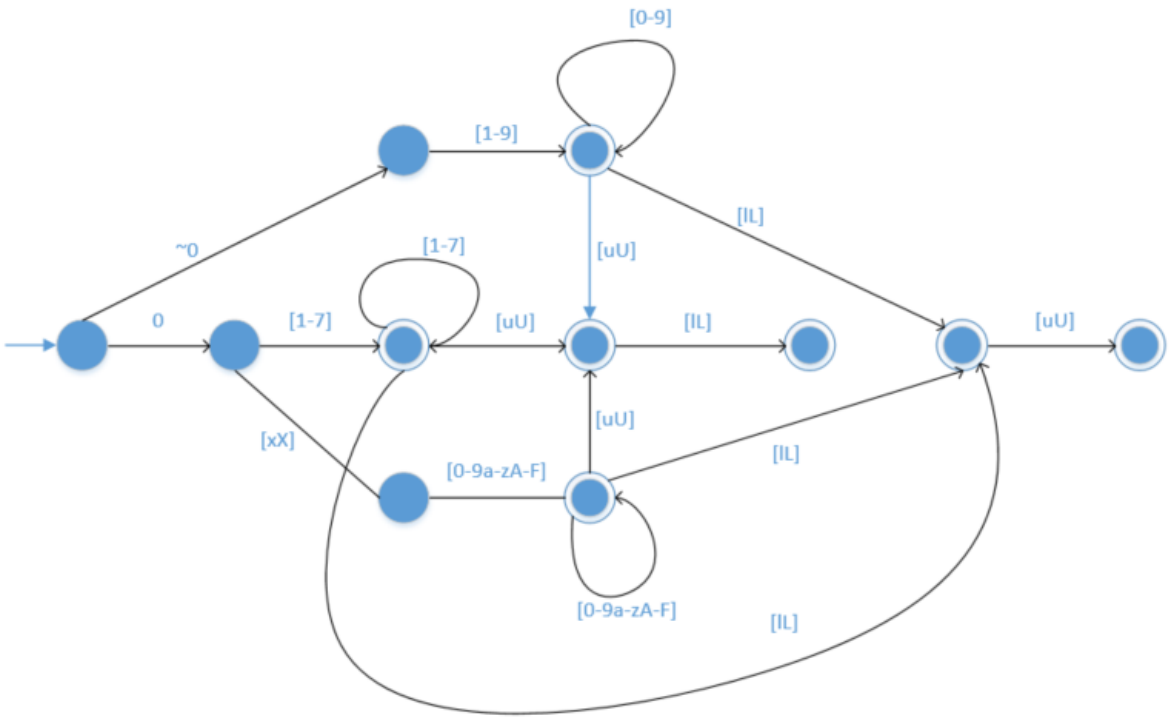
Course: Compilers
Duration: 1 hour
Date: 19 November 2022

Student Name: ——————————
Student ID: ——————————

Question 1 (10 points)

In The C Programming Language, an integer constant is defined as follows:
An integer constant consisting of a sequence of digits is taken to be octal if it begins with 0 (digit zero), decimal otherwise. Octal constants do not contain the digits 8 or 9. A sequence of digits preceded by 0x or 0X (digit zero) is taken to be a hexadecimal integer. The hexadecimal digits include a or A through f or F with values 10 through 15. It may be suffixed with the letter u or U to specify it's unsigned and at the same time could be suffixed with the letter l or L to specify it's a long. If both suffixes are present, they may appear in either order.

A. **Write a regular expression that generates C integer constants as described above.**
   (0[1-7]+)|(~0[1-9][0-9]+)|(0(x|X)[0-9a-fA-F]+)((u|U)(l|L)|(l|L)(u|U))?
B. **Draw a DFA that recognizes integer constants as defined by your solution to point (A).**



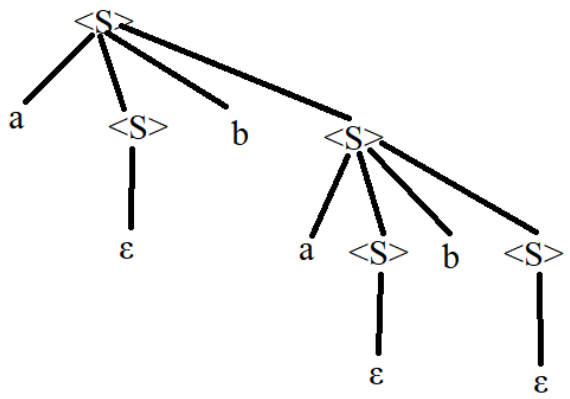
Question 2 (10 points)

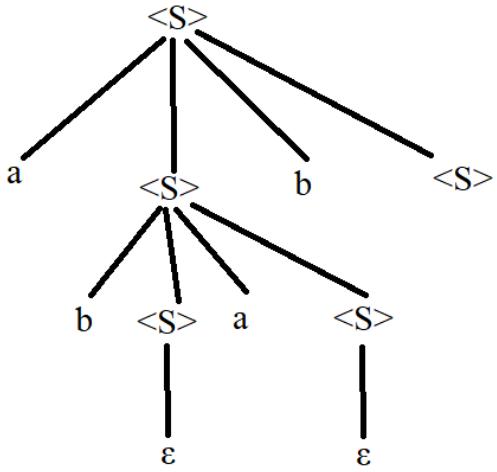
Consider the grammar:

<S> —> a<S>b<S> | b<S>a<S> | ε

A. **Is that grammar ambiguous ? YES**
B. **IF the answer is Yes, Prove a string that shows such ambiguity and prove it through providing two distinct leftmost derivations for the match of the above string.**
   YES: String example is abab
   First derivation:
   <S> ————> a<S>b<S>
   <S>————> ab<S>
   <S>————> aba<S>b<S>
   <S>————> abab<S>
   <S>————> abab
   Second derivation:
   <S> ————> a<S>b<S>
   <S>————> ab<S>a<S>b<S>
   <S>————> aba<S>b<S>
   <S>————> abab<S>
   <S>————> abab

C. **Draw the corresponding parse trees for your derivations**.





Question 3 (5 points)

Given Σ={ a,b,c} write a regular expression that matches the following language if possible and if not, Please explain why then suggest another way that could be used to match it:

A. **A set of all strings that contains at most one b.**
   (a|c)*(b|ε)(a|c)*
B. **A set of all strings that contains no two consecutive b's.**
   (a|c|ba|bc)*(b|ε)
C. **A set of strings of two b's surrounded by the same number of a's.**
   It's not possible to express such language with regular expression since the regular expression couldn't count we could use BNF or Context free grammar to express it as follow:
   <G> ————> a <G> a | bb