CS6013 Assignment 1

1. **Regular Expressions and DFA**
   Write DFAs for the following languages [5 + 5 + 5]:

   (a) L=$\{w \in \{x,\ y,\ z\}^* |$ at most one letter in w is repeated more than once$\}$

   (b) L=$\{w \in \{lock,\ unlock,\ sharedAccess\}^* | w$ denotes a sequence of valid accesses to shared data$\}$

   (c) L=$\{w \in \{1, 2, 3\}^* |$ Number of 2s modulo 2 doesn't match the number of 3s modulo 3$\}$

   [Bonus] Write the equivalent REs. [10]

2. **CFG**
   Which of the following grammars is ambiguous. Justify. [5 + 5 + 5]

   (a) $S \to\ S\ S\ +\ |\ S\ S\ *\ |\ a$

   (b) $S \to\ +\ S\ S\ |\ *\ \ S\ S\ |\ a$

   (c) $S \to\ S\ +\ S\ |\ S\ *\ S\ |\ a$

   Write the CFGs for the following languages. [5 + 5]

   (a) L=$\{w \in \{push, pop, top\}^* | w$ denotes a sequence of valid stack operations$\}$.

   (b) The set of all strings of 0 and 1 that are *palindromes.*

3. **Parsing**
   LL(1) Grammar [15], Parser Implementation [20].
   Consider the grammar
   $S \to\ S\ +\ S\ |\ S\ S\ |\ (\ S\ )\ |\ S\ *\ |a$
   a is a terminal symbol. Rewrite the grammar into a grammar which is LL(1), and use the rewritten grammar as the basis for implementing a recursive descent parser; write the LL(1) grammar, the FIRST and FOLLOW sets for each nonterminal symbol, and the predictive parsing table, together with an argument that the new grammar is LL(1).