CS6013 Assignment 1

1. **Regular Expressions and DFA**
   Draw DFAs for the following languages (5 + 5 + 5)

   (a) L=$\{w \in \{a, b, c, d\}^* | w$ has no repeated letters$\}$

   (b) L=$\{w \in \{1, 2, 3\}^* |$ Number of 2s modulo 2 equal the number of 3s modulo 3$\}$

   (c) L=$\{w \in \{lock, unlock, tryLock\}^* | w$ denotes a sequence of valid lock operations in Java on a lock$\}$.

   Bonus: Write the equivalent REs. (10)

2. **CFG**
   Write the CFG for the following language: (5 + 5 + 5)

   (a) L=$\{w \in \{0, 1\}^* | w$ contains equal number of 0s and 1s$\}$.

   (b) L=$\{w \in \{0, 1\}^* | w$ contains unequal number of 0s and 1s$\}$.

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

3. **Parsing**
   LL(1) Grammar (30), Parser Implementation (40).
   Consider the grammar
   `stmt ...= id(); | stmt stmt | { stmt } | if (id) stmt else stmt`
   where `stmt` is the only non-terminal symbol, `stmt` is the start symbol, and
   `id ( ) ; { } if else`
   is the list of terminal symbols. The terminal symbol `id` is defined using the regular expression (letter+) where letter is an ascii character in the interval a...z. The grammar generates a subset of the Java statements. 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).