CS6013 Assignment 1

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

   (a) The language of all strings over the alphabet {a, b} where every 'a' is immediately followed by at least one 'b'.

   (b) The language of all strings over the alphabet {0, 1} in which the number of consecutive 1s is divisible by 3.

   (c) The language of all strings over the alphabet {a, b} where each 'a' is followed by an even number the number of 'b's.

   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 double the number of 0s than 1s$\}$.

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

   (c) L=$\{w \in \{lock\_x, unlock\_x, access\_x\}^* | w$ denotes a sequence of valid accesses over a shared location and x can be any integer.$\}$.

3. **Parsing**
   LL(1) Grammar (30), Parser Implementation (40).
   Consider the grammar
   `stmt ...= id(); | stmt stmt | { stmt } | if (id) 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 non-terminal symbol, and the predictive parsing table, together with an argument that the new grammar is LL(1).