CS 5000: Theory of Computation
Assignment 2: Closure Properties of Regular Languages & Implementing DFAs in Java

Vladimir Kulyukin
Department of Computer Science
Utah State University

Learning Objectives

1. Closure Properties of Regular Languages
2. Implementation of DFAs in Java

Problem 01: Closure Properties of Regular Language (2 points)

Let L_1 and L_2 be two languages. The concatenation of L_1 and L_2 is defined as $L_1 \cdot L_2 = \{xy | x \in L_1 \; and \; y \in L_2\}$. The Kleene closure of a language L is defined as $L^* = \{x^* | x \in L\}$, i.e., the language consists of 0 or more occurrences of strings in L. The differences between L_1 and L_2 is defined as $L_1 - L_2 = \{x | x \in L_1 \; and \; \neg(x \in L_2)\}$, i.e., strings that are in L_1 but not in L_2. Finally, the quotient of L_1 and L_2 is defined as $L_1/L_2 = \{x |$ there exists $y \in L_2$ such that $xy \in L_1\}$.

Sketch proofs of the following statements:

1. Show that regular languages are closed under concatenation.
2. Show that regular languages are closed under the Kleene closure.
3. Show that regular languages are closed under difference.
4. Show that regular languages are closed under quotients with final languages, i.e., languages that consist of finite numbers of strings.

Problem 02: Implementing Finite State Machines in Java (3 points)

Figure 1 shows a DFA that processes strings of 0's and 1's that are binary representations of numbers divisible by 3. Implement this DFA in Java. Define a

class Mod3DFA with a public method processString(String x). Mod3DFA.processString(String x) should return true if x is the binary representation of a number divisible by 3 and false otherwise. Save your solution in Mod3DFA.java and submit it via Canvas along with your solution to Problem 1.

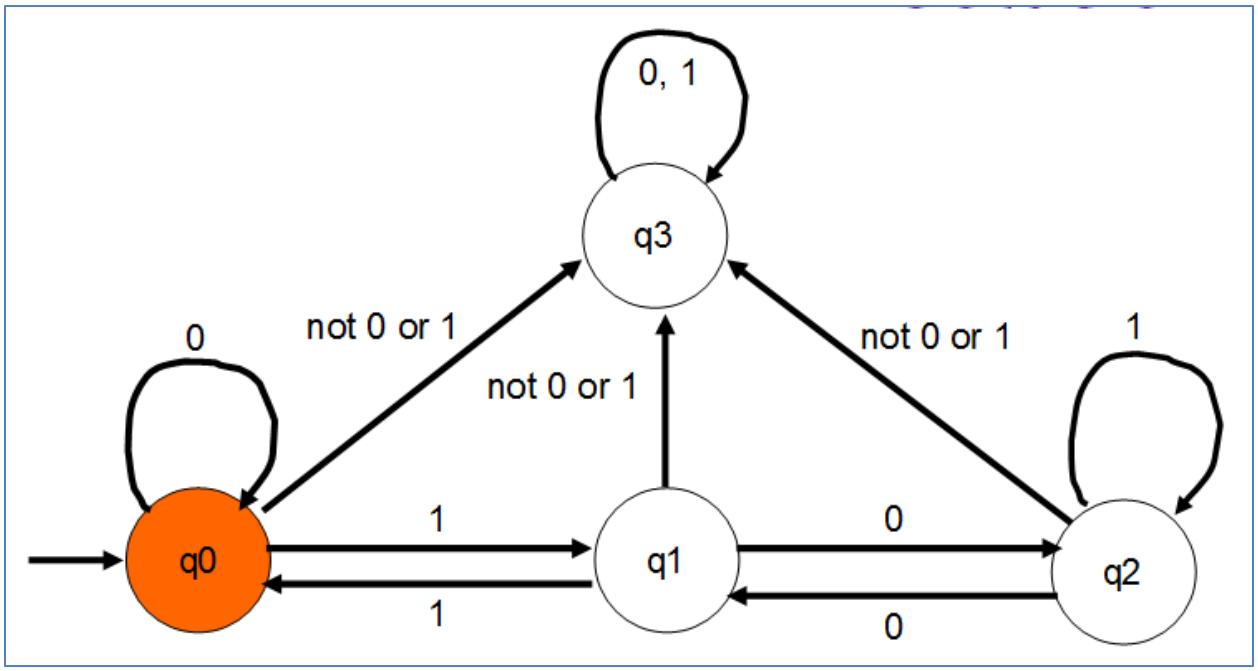

Figure 1. Modulo 3 DFA

Define a static method runTests() that tests your solution, e.g.,

```
public static void runTests() {
        final String s1 = “0”;
        final String s2 = “10”;
        final String s3 = “11”;
        final String s4 = “110”;
        final String s5 = “1001”;
        // etc
        System.out.println(s1 + “ : ” + processString(s1));
        System.out.println(s2 + “ : ” + processString(s2));
        System.out.println(s3 + “ : ” + processString(s3));
        System.out.println(s4 + “ : ” + processString(s4));
        System.out.println(s5 + “ : ” + processString(s5));
}
```