Joshua Reed

915743782

Project 7

1.

L₁ and L₂ are infinite, regular languages. Use the closure properties of regular languages to help prove that L₁\* and (L₁ ⋃ L₂) are also both infinite and regular.

Assume that we have the language L₁ and L₂ and they are regular languages. Lets suppose L₁ and L₂ are α and β respectively. α + β is a regular language as well and can be defined as α ∪ β or L₁ ⋃ L₂.

Assume L is a regular language αβ. Then L\* can be represented as α\*.

2.

L(M₁) and L(M₂) are both finite languages. Use the bounded simulation technique presented on p. 190 of the textbook to decide |L(M₁)| < |L(M₂)|.

1.   If *L*(*M*1) is infinite, return *False*.

2.   If *L*(*M*2) is infinite, return *True*.

3. Run every string of through that is < || and count the number accepted, denoted

4. Run every string of through that is < || and count the number accepted, denoted

5. If return true (|L(M₁)| < |L(M₂)|), otherwise return false.

3.

ß and µ are two regular expressions

L = { w : |w| is even ∧ w Є L(ß) ∧ w ∉ L( µ) }

Complete the procedure below for deciding L = Ø

1. LEVEN = {w Є Σα\* : |w| is even }﻿
2. Get LEVEN and make a minimal deterministic FSM denoted as
3. Create a minimal deterministic FSM of |w| is even denoted
4. If and are considered equivalent return true, otherwise continue
5. L(ß') = L(ß) ∩ LEVEN
6. Let L(ß’) be a FSM denoted
7. Create FSM of L(ß) ∩ LEVEN denoted
8. See if L( is empty and equivalent to
9. If both are true then L(ß') = L(ß) ∩ LEVEN = Ø, else return false