Alexa Haushalter (alhaus)
Pablo Hernandez (pabloh2)

Problem Set #5

Problem One: Constructing DFAs (8 Points)
Please use the DFA/NFA Developer site as a basis for grading for this question. The submissions are saved under Alexa Haushalter's SUNet ID "alhaus". We have screenshots of the DFAs saved, in the event the site has a problem.

Problem Two: Constructing NFAs (6 Points)
Please use the DFA/NFA Developer site as a basis for grading for this question. The submissions are saved under Alexa Haushalter's SUNet ID "alhaus". We have screenshots of the NFAs saved, in the event the site has a problem.

Problem Three: Designing Regular Expressions (8 Points)
Please use the SimpleRegex site as a basis for grading for this question. The submissions are saved under Alexa Haushalter's SUNet ID "alhaus".
For simplicity, we will also add our answers below.

i. a*b*
ii. a*(ba+)*(b?)
iii. ((y (yd)* d) | (d (dy)* y))*
iv. **(ε | a | b | aa | bb | ba | (a | b) (a | b) ((a | b)+))**

Problem Four: Finite and Cofinite Languages (4 Points)

i. Theorem: Any finite language is regular.

Proof: Let P(n) be the statement that "any language L with |L| = n is regular." We'll then prove that P(n) is true for all n ∈ **N**.

As a base case, we prove P(0), that any language L with |L| = 0 is regular. This is the empty language ø. This is a regular language because it can be represented by the DFA:

	Σ
q0	q0

This DFA says that nothing is accepted, which is true for ø because nothing is in the language. Thus, we have shown that any language L with |L| = 0 is regular, because this is simply ø.

For the inductive step, assume that for some k ∈ N, P(k) holds. In other words, any language L0 with |L0| = k is regular. We'll prove P(k+1), that any language L1 with |L1| = k+1 is regular.

Adding any element x to L0 can be simulated by taking L1 = L0 U {x}. If we take |L0| = k, this makes |L1| = k+1.
L = {x} is a regular language because it can be represented by the regular expression a.
In lecture, we proved that the union of two regular languages is a regular language. Thus, since L1 can be formed by taking the union of regular language L0 and regular language {x}, then L1 is a regular language with |L1| = k+1. Thus, for any n ∈ **N,** P(n) holds. ■

ii. Theorem: Any cofinite language is regular.

Proof: If a language is regular, then its complement is regular as well (proved in class). A finite language's complement is a cofinite language. Since every finite language is regular (as proved in the first part of this problem), then all the complements must be regular as well. Since these complements are cofinite languages, then all cofinite languages must also be regular.■

Problem Five: ℘ (Σ*) (2 Points)

If Σ is an alphabet, then ℘(Σ*) is a set of all possible subsets that can be formed from the set of all strings composed from the letters in Σ. Since Σ* is the set of all strings that can be composed from the letters in Σ, then the ℘(Σ*) is just a set that contains all subsets of these string compositions.

Problem Six: Derivatives of Regular Languages (4 Points)

Theorem: If L is a regular language over Σ and a ∈ Σ, then ∂aL is regular.

Proof: Since L is a regular language, we know that there exists a DFA (let's call it X) that describes the strings from the regular language L. We will now show that a new DFA can be built (called X') that will describe strings from the language ∂aL.

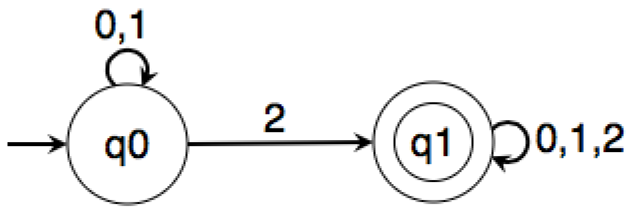
If we consider the start state of the DFA for language L to be q, then we can just move over the start state for this DFA to the node to q', which is the state after transitioning on the letter 'a'. This is the same as removing all strings in L that do not start with the letter 'a', but also taking off the first 'a' from each string in L that does begin with the letter 'a'. We can call this new DFA X'.

We now know that X' will take all of the strings (as accepted strings) in ∂aL because it is equivalent to the part of the remaining strings that were in L that started with an 'a', and then had the leading 'a' removed. We have shown that there exists a DFA (namely, X') that describes the language ∂aL. Consequently, we have shown that ∂aL is regular, given that L was a regular language over Σ and a ∈ Σ. ■

Problem Seven: Why the Extra State? (3 Points)

Let L = {w ∊ {0, 1, 2}* | w contains 2 as a substring}. An NFA for L is shown below.
Here, the NFA constructed by the second method would incorrectly accept strings such as 0, 1, or 0101. This new NFA would not be L* because it would have transitions back to the start state other than ε. This would cause some strings to be accepted (strings that mean to pass through the start state, but are inadvertently accepted) even though they are not in L*.

NFA, N, for L:



Problem Eight: Course Feedback (1 Point)
All members of our group did the course feedback.

Extra Credit Problem: Multiples of Seven (1 Point Extra Credit)
Please use the DFA/NFA Developer site as a basis for grading for this question. The submissions are saved under Alexa Haushalter's SUNet ID "alhaus". We have a screenshot of the DFA saved, in the event the site has a problem.