Name(s)_____________________________

CS503
Homework #1

Directions: Please put your final answers on this sheet.

#0. Name some alternative notations for

a) The empty string

b)

lambda: λ, epsilon: ε

c) Union of Regular Expressions (Sets)

$\cup$, +

d) Complement of a Set

Given a set A: ~A, C(A), $\overline{A}$, $\neg$ A, not A

e) Something else related to the first 2 modules

(And it's ok to post these to the bb)

#1. (10 Points) ***True or False:***

a) Given a language (set of strings) L, the question: "Is it raining" is a decision problem:
T F

b) {ε} is the empty language T F

c) For sets A and C, ~ (A ∩ C) = ~ A ∪ ~ C T F

d) DFA's may fail to either accept or reject a string T F

e) There exist formal languages which are not regular T F

f) Given an alphabet Σ and a regular language L ⊆ Σ*, the strings in Σ* - L are not in L
T F

Proofs:

#2. (10 Points) Prove that the function f: $\mathcal{N} \rightarrow \mathcal{N}$ defined by $f(n) = n^2 + 1$ is one-to-one but not onto.

one-to-one If $n_1 \neq n_2$, then $n_1^2 + 1 \neq n_2^2 + 1$. Thus, $f(n_1) \neq f(n_2)$

onto There is no *n* such that f(n) = 3 (among others)

#3. (10 points) Prove, using induction that $(w^R)^i = (w^i)^R$
Be sure to state what you are doing the induction on.

Proof by induction on *i*

Basis When $i = 0$, $(w^R)^0 = \varepsilon$
and $(w^i)^R = (w^0)^R = (\varepsilon)^R = \varepsilon$
(both sides = ε)

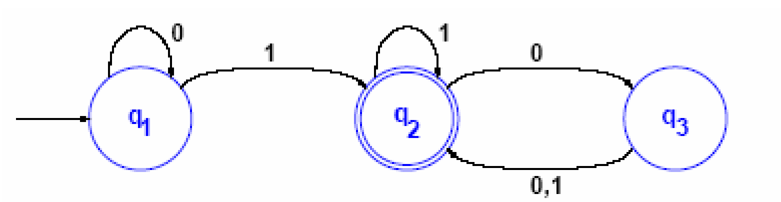
Induction Step (To show that if $(w^R)^i = (w^i)^R$, then $(w^R)^{i+1} = (w^{i+1})^R$ for $i \geq 0$)

I like to start with the left hand side ($(w^R)^{i+1}$) and show in a series of steps that it equals the right hand side ($(w^{i+1})^R$)

$(w^R)^{i+1} = (w^R)^i\ (w^R)$ **Definition of string exponentiation**
$= (w^i)^R\ (w^R)$ **Induction Hypothesis**
$= (w\ w^i)^R$ **$(uv) = v^R u^R$ (done in class)**
$= (w^{i+1})^R$ **Definition of string exponentiation**

DFA's

#4. (10 Points) What set of strings does the following automaton accept?

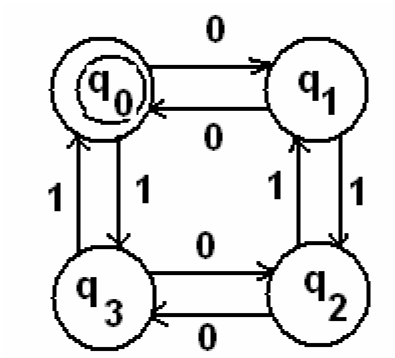


Let's see: you can have zero or more 0's to start, then a 1 brings you to a final state and as long as you have 1's, you're ok, but if you have any 0's, each must be followed by a 0 or a 1 to get back to the final state.

We hadn't had regular expressions for this homework, but here is one way to write this:

0*1 (1*(0 (0+1))*)*

#5. (10 Points) Construct a dfa to accept all strings containing an even number of zeros and an even number of ones.



This can actually be done with 3 states (can you see how?)