Date completed: ____________________ Login: ______________________

CS60 – Homework 12 – Written Portion (No online component) – 35 points

When should you finish this homework assignment?

- You should finish this homework by Friday December 12th at 11:59pm.
- If you use a euro you'll have until Saturday December 13th at 11:59pm.

How do you turn in this homework assignment?

- You can turn this homework assignment under Colleen's office door, in class, or before the final exam.
- The last time you can turn this assignment in is December 15 (before the final exam)

1. Draw a DFA for the language:
   `{w | The number of 0's in w is a multiple of 2 or a multiple of 3 or both.}`

For example, the empty string, `0101`, and `01000100` are in the language but `01` is not. Your DFA should have at most 6 states. The empty string, with zero `0`s *should* be accepted.

2. Prove that any DFA for the language in problem 1 MUST have at least 6 states. You will need to use the idea of *pairwise distinguishable sets of strings* as we covered in class. You might use the attached example proof as both starting guide and template. Write a clear and precise proof below. Note, too, that since your DFA for this language had 6 states, you have proven that that machine is ***the most-efficient-possible DFA for this language***!

```
+--------------------------------------+
+ Question 2:  a minimum-state proof
+--------------------------------------+

++ Claim:  A DFA for the language

L = {w | The number of 0's in w is a multiple of 2 or a multiple of 3 or both.}

requires at least 6 states. (you fill in the rest!)
```

3. Next, you'll prove two languages are not regular by using the Nonregular Language Theorem. **Recall that we say that a language is "Nonregular" if there cannot exist a DFA for it.** First, consider the *palindrome* language:

```
{w | w is made of 0's and 1's and is the same forwards as backwards}
```

For example, `0`, `11`, `101`, and `00100` are all in the language but `01`, `100` and `010111` are not in the language. Prove that that this language is not regular by using the Nonregular Language Theorem.

You might use the attached example proof as both starting guide and template. Notice that simply giving an infinite set `S` does not suffice. You must also argue that an arbitrary pair of strings from `S` are always distinguishable!

```
+--------------------------------------+
+ Problem 3:  a nonregular language proof
+--------------------------------------+

++ Claim:  The language

L = {w | w is made of 0's and 1's and is the same forwards as backwards}

is not a regular language (no DFA accepts it).
This is the "language of palindromes" for binary strings. (you fill in the rest!)
```

4. *Another non-regular language* Now consider the language

```
{w | w is a string of 0's whose length is a power of 2}
```

For example, the strings `0`, `00`, `0000`, `00000000` are all in the language since their lengths are 1, 2, 4, and 8, all of which are powers of 2. On the other hand, `000`, `00000`, any string containing a character other than `0`, and the empty string are all **not** in this language. Prove that this language is not regular by using the Nonregular Language Theorem. Your proof must be clear and precise.

```
+--------------------------------------------+
+ Problem 4:  another nonregular language proof
+--------------------------------------------+

++ Claim:  The language

L = {w | w is a string of 0's whose length is a power of 2}

is not a regular language (no DFA accepts it). (you fill in the rest!)
```

+--------------------------------------+
+ Example 1: a minimum-state proof
+--------------------------------------+

++ Claim: A DFA for the language

L = {w | w's first bit == w's final bit}

requires at least 5 states. We consider
the empty string to be accepted, too.

We will use λ as shorthand for lambda, the empty string.

++ Strategy: We use the Distinguishability Theorem.

We need to show a set S of 5 strings, each
pair of which is distinguishable.

++ Proof:

Let S be this set of five strings:

{ λ, 0, 1, 10, 01 }

There are 10 possible pairs from this set S.

We need to show all 10 of these possible
are distinguishable: for each pair, we need to show a
suffix z that makes one of the strings accepted and
makes the other string rejected.

First, we note that the empty-string suffix, z = λ
distinguishes the strings in S that are currently
accepted (λ, 0, and 1) from those currently
rejected (10 and 01). This handles six of the 10 cases.

Only four cases remain:

10 vs 01: adding the suffix z = 0 to both yields 100 and 010, and
the former is rejected and the latter is accepted

λ vs 0:   adding the suffix z = 1 to both yields 1 and 01, and
the former is accepted and the latter is rejected

λ vs 1:   adding the suffix z = 0 to both yields 0 and 10, and
the former is accepted and the latter is rejected

0 vs 1:   adding the suffix z = 1 to both yields 01 and 11, and
the former is rejected and the latter is accepted

Thus, we've shown all 10 possible pairs from S are distinguishable.

As a result, the DFA for our language L requires at least 5 states. QED.

[[Aside: "QED" is sometimes used as a proof-ending abbreviation. It stands
for the Latin phrase "Quod Erat Demonstrandum," i.e., "which was to
be shown."]]

+--+
+ Example 2: a nonregular-language proof
+--+
++ Claim: The language

L = {w | w contains an equal number of 0's and 1's in any order.}

is not regular.

++ Strategy: We will use the Nonregular Language Theorem.

We must show an infinite set S of strings, such that S's strings are all pairwise distinguishable.

++ Proof:

Let S = {w | w contains n 0's, n is any positive integer.}

(Aside: Another way to write this is S = {0^n | n is any positive integer.}
0^n stands for a string of n 0's.)

Put another way, S = {0, 00, 000, 0000, ...}

Clearly, S is infinite. Now we must show ANY pair of strings chosen from S are distinguishable.

We use the names u and v to designate two _different_ strings from S:

Let u = 0^i (that means i consecutive 0's)
and v = 0^j (that means j consecutive 0's)
... where i and j are different.

So, u and v represent two arbitrary strings in S.

To show that u and v are distinguishable, we must find some suffix z that can be added to both u and v so that

uz (u with z concatenated to the end of it) is accepted in L, but
vz (v with z concatenated to the end of it) is rejected by L

or vice-versa (uz can be rejected and vz can be accepted). The key is that if they have different fates, then u and v must have been in different states.

So, we choose

z = 1^i   (which means i consecutive 1's)

Then,

uz = 0^i 1^i   which has an equal # of 0s and 1s,
and thus is _accepted_

but

vz = 0^j 1^i   which has a different # of 0s and 1s (because i != j),
and thus is _rejected_

Since u and v were arbitrary strings in the infinite set S, any DFA for our language L would have to have infinitely many states - but this can't happen. Thus, by the Nonregular Language Theorem, L is not a regular language, i.e., there is no DFA that accepts it. Q.E.D.