ECS120 Introduction to the Theory of Computation
Fall Quarter 2007

Homework 1 Help
Due Friday October 05, 2007

Problem 1.1

You can find definitions for **reflexive** and **symmetric** in your book (in Chapter 0). If friendship is anti-reflexive and symmetric, then we know:

- Friendship involves two different people, meaning you are not "friends" with yourself.
- Friendship is mutual. If Xander is friends with Yasmin, then Yasmin is friends with Xander.

The **pigeonhole principle** basically states that if you have more pigeons than holes, then at least two pigeons will have to be placed in the same hole.

If you are stuck, try breaking the problem into two separate cases:

- Case 1: One person at the party has no friends.
- Case 2: Everyone at the party has at least 1 friend.

For each of the cases, prove that at least 2 people have the same number of friends for each case.

Problem 1.2

How to do **Proof by Induction** is also covered in Chapter 0 of your book. Make sure you include your base case!

If you are stuck, try thinking about the hint. It states that:

> "The lines will divide the plane in the maximum number of regions if no two lines are parallel, and if there are never more than two lines crossing at each point."

Therefore, if we have a plane with 2 lines, how many more regions can adding a 3rd line create? Can you generalize this?

Problem 1.3

Before trying to prove that your DFA is minimal, you may want to check that it is minimal! You can do this in JFLAP (as shown in discussion).

To prove that it is minimal, we will be using the proof by contradiction and the pigeonhole principle. See the example in the discussion notes.

Problem 1.4

For parts (a) through (d), try using the union and complement operations covered in your book. For example, (b) is the complement of (a).

If you are stuck on part (c), create DFAs for the following languages:

- $L_1 = \{w \in \Sigma^* \mid w \text{ has an even number of 0s}\}$
- $L_2 = \{w \in \Sigma^* \mid w \text{ has an even number of 1s}\}$

Then, you can use the union operation in your book to find the DFA for $L_1 \cup L_2$.

For part (d), the language you want accepts all strings except 11 and 111. Try using both the union and complement operations here!

Part (e) is challenging. Remember, 101 and 0101 represent the same number. Your DFA should allow for any number of 0s at the start of the binary encoding. Also, you should not accept ϵ!

If you are stuck, think about this as a mod problem. Any number mod 5 is either 0, 1, 2, 3, or 4. Make these the name of your states, and try to build your transitions from there!

Also, I will take off points if your DFA is missing transitions!