York University CSE 2001 September 10, 2007

Homework Assignment #1
Due: September 24, 5:30 p.m.

1. In the first lecture, I said we would restrict our attention in this course to *decision problems*, where the answer is always yes or no.

Consider the following decision problem. The input to the problem is a pair of locations in Toronto and a natural number. The output is yes if and only if the distance between the two locations (in kilometres) is less than the number. For example, if the input was "City Hall, CN Tower, 20", the answer would be yes, because the distance between City Hall and the CN Tower is less than 20 km.

Given two locations in the city, you would like to be able to *compute* the distance between them. Suppose somebody gave you a computer programme that solves the decision problem, but you do not know how it works (because you cannot look at the source code). How could you use it to compute distances? Your solution should produce answers that are accurate within half a kilometre of the true distance.

Do not worry too much about making your solution efficient, as long as it works correctly.

2. Nine people are arranged in a circle, facing the centre, to play a game. At any time during the game, each person can either be standing up or sitting down. There is a clock on the wall. When the game starts at noon, at least one person is standing and at least one person is sitting. During the game, each person follows these instructions repeatedly:

- Look to your left.
- If you are sitting and the person to your left is standing, then stand up when the clock reaches the next minute.
- If you are standing and the person to your left is standing, then sit down when the clock reaches the next minute.
- Otherwise, do not change your position when the clock reaches the next minute.

The game ends when everyone is sitting down at the same time. (It is not a very fun game.)

For example, here is one possible way the game could begin, where 0 represents a person sitting down and 1 represents a person standing up. The people are listed in clockwise order around the circle.

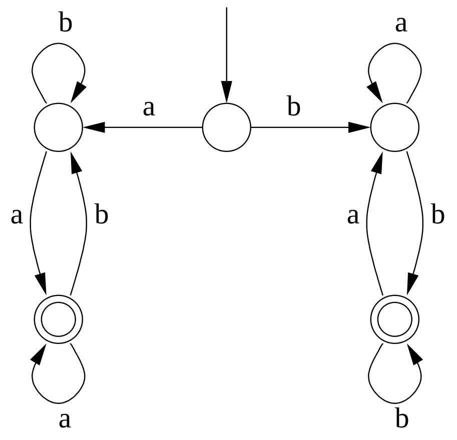
```
12:00 101101000
12:01 110111001
12:02 011001010
12:03 101011110
12:04 111100011
12:05 000100100
12:06 001101100
12:07 010110100
   ⋮
```

Prove that, no matter how the nine people are arranged at noon, the game will never end.

3. Consider the alphabet $\left\{\binom{0}{0}, \binom{0}{1}, \binom{1}{0}, \binom{1}{1}\right\}$. Let L be the language of all strings that represent correct steps of the game described in the previous question (for any number of players, not just nine). The top row of bits represents the people before the step, and the bottom row represents the people after the step (one minute later). For example, the string $\binom{1}{0}\binom{1}{0}\binom{1}{0}\binom{1}{1}\binom{0}{0}\binom{0}{0}\binom{0}{1}\binom{1}{0}\binom{1}{0}$ is in L because it describes the step that took place in the example game above at 12:05. The string $\binom{1}{0}\binom{1}{0}\binom{1}{0}\binom{1}{0}$ is also in the language, because if four people are standing up one minute, they will all be sitting down the next minute.

Draw a DFA that accepts L. The simpler your DFA is, the better. Briefly explain why your answer is correct. In particular, you should explain what each state of the DFA represents.

4. Describe, in plain but precise English, the languages accepted by the following automaton.



5. This question is about a standardized way of writing down non-negative decimal numbers. Large numbers are written down using spaces to divide the digits into groups of 3, starting from the decimal point. For example, one billion is written down as "1 000 000 000". A similar convention is used for readability if there are many digits to the right of a decimal point. For example, we might say π is approximately "3.141 592 65". There should not be a decimal point unless there are digits to the right of it. There is always at least one digit to the left of a decimal point. The first digit should never be 0 unless it is the only digit to the left of the decimal point.

Draw an NFA that accepts a string if and only if it follows the conventions described above for writing down a number. The alphabet for this problem is $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ., ␣\}$, where ␣ is used to indicate a space. For example, "23␣234.281␣1", "0.24" and "1␣000␣000" should be accepted, but "1987", ".99", "207␣01" and "007" should not be accepted. To save space, you can use d to label an edge that can be traversed by any of the digits $1, 2, 3, 4, 5, 6, 7, 8, 9$. *Briefly* explain why your answer is correct. In particular, you should indicate what each state represents.