

**HW 3 CMSC 452. Morally DUE Feb 20
THIS HOMEWORK IS THREE PAGES**

USE our convention for representing:

- a number ($n = [0\ 0\ 0\ 0\ \dots\ 0\ 1\ * * * \dots]$, where there are n 0s)
- a tuple of numbers
- a tuple of numbers and finite sets

SOLUTIONS

1. (0 points) What is your name? Write it clearly. What day is the midterm? Staple your HW.
2. (40 points) For each of the following languages (lettered a-d), draw a DFA. Make sure it has ACCEPT states, REJECT states, and STUPID states. Then, answer each of these questions about the DFA you drew:
 - (i) How many ACCEPT states does it have?
 - (ii) How many REJECT states does it have?
 - (iii) How many STUPID states does it have?
 - (a) $\{(x, y) : x = y + 2\}$.
 - (b) $\{(x, y) : x \neq y + 2\}$. (Note the \neq here!)
 - (c) $\{(x, y) : x = y + 100\}$. (For this one, you can and should use DOT DOT DOT rather than have LOTS of states.)
 - (d) $\{(x, y) : x \neq y + 100\}$. (For this one, you can and should use DOT DOT DOT rather than have LOTS of states.)

SOLUTION TO PROBLEM 2

(Hard to draw in LaTeX so I'll describe.)

I'll just do the last two.

For $x = y + 100$:

So long as the input is 00 (that should be 0 on top of 0 but if I did that in LaTeX it would be hard to read) stay in the start state. The Start State is STUPID.

If 10 is seen that means that means that $x < y$ IF y ever shows up. So go to a STUPID state. Stay in that state unless \$1 is the symbol in which case goto a REJECT state. The REJECT state, on any input, stays in the same state.

If 11 is seen then goto a REJECT state and stay there on any input.

If 01 is seen THEN this is interesting. Use around 100 states to keep track of x not coming in yet. If it comes in early, then REJECT. If it comes in exactly 100 states later then ACCEPT. Until it does either its just STUPID.

The states:

- START, which is STUPID.
- See a 11 which is a REJECT. We only need one REJECT state.
- See a 10 which is a STUPID (Not the same as START)
- See a 01 which is STUPID.
- See a 01 then a 0\$, which is stupid
- See a 01 then a 0\$ then a 0\$, which is stupid
- \vdots
- See a 01 then a 0\$ then a 0\$... 99 of these, which is stupid
- on the 100th go to an accept state.

(I may be off by 1)

104 states

102 stupid, 1 reject, 1 accept.

To recognize the complement of this lang just swap the ACC and REJ but STUPID is still STUPID.

THERE ARE TWO MORE PAGES

3. (30 points) Consider the sentence

$$(\exists X)(\forall x)(\exists y)[(x \in X) \wedge (y \notin X) \wedge (x = y + 100)]$$

I would want to ask you to build the DFA's needed to decide if this sentence is true or false. That would be madness! Instead, I'll ask you about parts of the process and about number-of-states.

- (a) Draw a DFA for $\{(x, X) : x \in X\}$ How many states does it have?
- (b) Draw a DFA for $\{(y, X) : y \notin X\}$ How many states does it have?

For the rest of this question, we will assume that

$$\{(x, y) : x = y + 100\}$$

can be done with 100 states (OK, it's really more like 104, but what's 4 states among friends?).

- (c) Consider the language

$$\{(x, y, X) : [(x \in X) \wedge (y \notin X) \wedge (x = y + 100)]\}$$

DO NOT DRAW THE DFA! For that way lies madness!

But: how many states would the DFA for this have if you were to draw it? DO NOT be clever! Just take the answers to the prior 3 problems and use them. (You may be able to do better by looking at this particular problem, but I am trying to make a more general point.)

- (d) Consider the language

$$\{(x, y) \text{ st } (\exists X)[(x \in X) \wedge (y \notin X) \wedge (x = y + 100)]\}$$

Give an upper bound on how many states a DFA for this has, based on the prior problem. (You may be able to do better by looking at this particular problem, but I am trying to make a more general point. Use the NFA to DFA construction.)

SOLUTION TO PROBLEM 3

a) Start state is stupid state, on input 11 go to an accept, on input 10 goto a reject, on 00 or 01 stay stupid. THREE STATES

b) Start state is stupid state, on input 11 go to an accept, on input 10 goto an accept, on 00 or 10 stay stupid. THREE STATES

c) $3 \times 3 \times 100 = 900$ states.

d) Going from NFA to DFA increases the number of states exp, so 2^{900} is the bound. $n_a(w) \equiv i \pmod{n_1}$

$n_b(w) \equiv j \pmod{n_2}$

The number of states is $n_1 n_2$.

The number of accept states is 1- just the one (a_1, a_2) .

**END OF SOLUTION TO PROBLEM THREE
THERE IS ONE MORE PAGE**

4. (30 points) A *Sekora DFA* is a DFA which we intend to run on infinite strings. We DEFINE a Sekora DFA M to ACCEPT $x \in \{0,1\}^\omega$ if, when you run x through the M and get an infinite sequence of states, an infinite number of them are FINAL states.

Give an algorithm that will, given a Sekora DFA, determine if there exists ANY infinite string that it accepts.

Hint: You may find it helpful to think of a DFA as a (finite) directed graph - think about what has to happen for us to be able to visit the same vertex repeatedly in the same path.

(NOTE - we DO NOT ever actually run a DFA on an infinite string.)

SOLUTION TO PROBLEM FOUR

By using Breadth first search determine if there is SOME path from the start state to SOME final state.

Let A be the set of all final states that can be reached from the start state.

For every $q \in A$ determine if there is SOME path from q to itself.

IF there exists some q such that can get from s to q , and from q to itself, then YES there is some infinite string that is accepted.

IF NOT then NO infinite string is accepted.

END OF SOLUTION TO PROBLEM FOUR