MA2C03: TUTORIAL 19 PROBLEMS
TURING MACHINES

1) Consider the language over the binary alphabet $A = \{0, 1\}$ given by

$$L = \{(01)^m \mid m \in \mathbb{N}\} = \{\epsilon, 01, 0101, 010101, \dots\}.$$

(a) Draw a finite state acceptor that accepts L. Be sure to carefully label the initial state, the accept states, and all the transitions.

(b) Write down the algorithm of a Turing machine M that recognizes L.

(c) Draw the transition diagram of the Turing machine M from part (b). How is it different from the finite state acceptor you drew in part (a)?

2) Consider the language over the decimal digits

$$A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

given by $L = \{3m \mid m \in \mathbb{N}\}$. Write down the algorithm of a Turing machine that **decides** L. Process the following strings according to your algorithm: 0, 1, 5, and 9.

Solution: 1 (a) The finite state acceptor is drawn on last page.

(b)

(1) If ␣ is in the current cell, then ACCEPT. If 1 is in the current cell, then REJECT. If 0 is in the current cell, then erase 0 and move right.
(2) If ␣ or 0 is in the current cell, then REJECT. If 1 is in the current cell, then erase 1 and move right.
(3) Go to step 1.

(c) See attached diagram on last page. The Turing machine is an elaboration of the finite state acceptor with more complicated transitions ('character → character, direction' instead of just 'character'), separate transitions for ␣, and separate accept and reject states as opposed to having all states of the finite state acceptor be either accepting or rejecting states.

2) Note that $L = \{3m : m \in \mathbb{N}\} = \{m \equiv 0 \mod 3 : m \in \mathbb{N}\}$.

We can tell if a number is divisible by three by just examining its digits. We will use modular arithmetic heavily - recall the Michaelmas lectures and tutorials!

Note that for all $n \in \mathbb{N}, n \geq 1$, $10^n \equiv 1 \mod 3$ (also written $10^n \equiv_3 1$).

Consider 279. Is this divisible by 3?

$$279 = 200 + 70 + 9 = 2 \cdot 100 + 7 \cdot 10 + 9 \equiv_3 2 \cdot 1 + 7 \cdot 1 + 9.$$

To determine if 279 is divisible by 3, we just need to determine the conjugacy classes of its digits:

$$2 \equiv_3 2, 7 \equiv_3 1, 9 \equiv_3 0 \quad \to \quad 2 + 7 + 9 \equiv_3 2 + 1 + 0 \equiv_3 3 \equiv_3 0.$$

Therefore 279 is indeed divisible by 3. We see to determine this, we need only know the conjugacy classes of the digits $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$. Our Turing machine will reflect that.

Let $T = (\{s_0, s_1, s_2, s_{\text{rej}}, s_{\text{acc}}\}, A, A, t, s_0, s_{\text{rej}}, s_{\text{acc}}\}$. The states s_0, s_1, s_2 correspond to "congruent to 0, 1, 2 mod 3". Note that we begin in state s_0.

Here is the algorithm:

(1) If there is a blank in the first cell, REJECT. Otherwise move to step 2.

(2) If in state s_0, remain in s_0 if the digit is $0, 3, 6, 9$, then move right and go to step 3. If the digit is $1, 4$ or 7, move to state s_1, move right, and go to step 3. If the digit is $2, 5$ or 8, move to state s_2, move right, and go to step 3.

If in state s_1, remain in s_1 if the digit is $0, 3, 6, 9$, then move right and go to step 3. If the digit is $1, 4$ or 7, move to state s_2, move right, and go to step 3. If the digit is $2, 5$ or 8, move to state s_0, move right, and go to step 3.

If in state s_2, remain in s_2 if the digit is $0, 3, 6, 9$, then move right and go to step 3. If the digit is $1, 4$ or 7, move to state s_0, move right, and go to step 3. If the digit is $2, 5$ or 8, move to state s_1, move right, and go to step 3.

(3) Suppose there is a blank cell here. If in s_0, ACCEPT, otherwise, REJECT.

If the cell is not blank, move right and go to step 4.

(4) Move left and go to step 2. Do not change the contents of the current cell.[1]

[1] If we allow our Turing machine the option to 'stay in place' after reading the contents of a cell, this step is not needed.

Note that this is indeed a **decider** as there are no loops.
Here is how the following strings are treated:

- Begin in s_0. Read 0, remain in s_0. ACCEPT.
- Begin in s_0. Read 1, move to s_1. REJECT.
- Begin in s_0. Read 5, move to s_2. REJECT.
- Begin in s_0. Read 9, remain in s_0. ACCEPT.
- For the purposes of our example, suppose the input is 279. The configurations are as follows:

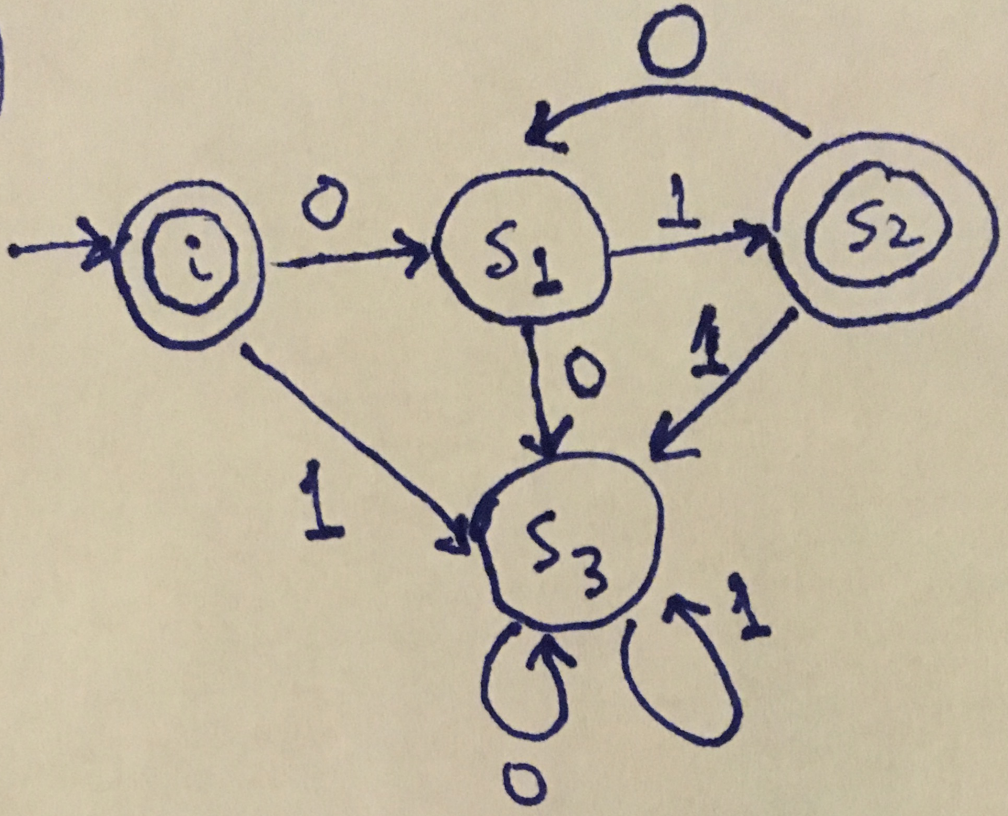
$$\epsilon s_0 279 \to 2 s_2 79 \to 27 s_0 9 \to 279 s_0 \to \text{ACCEPT.}$$

This answer is technically correct, but maybe difficult to think up of in an exam. Another solution is an algorithm for a two-tape Turing machine as follows:

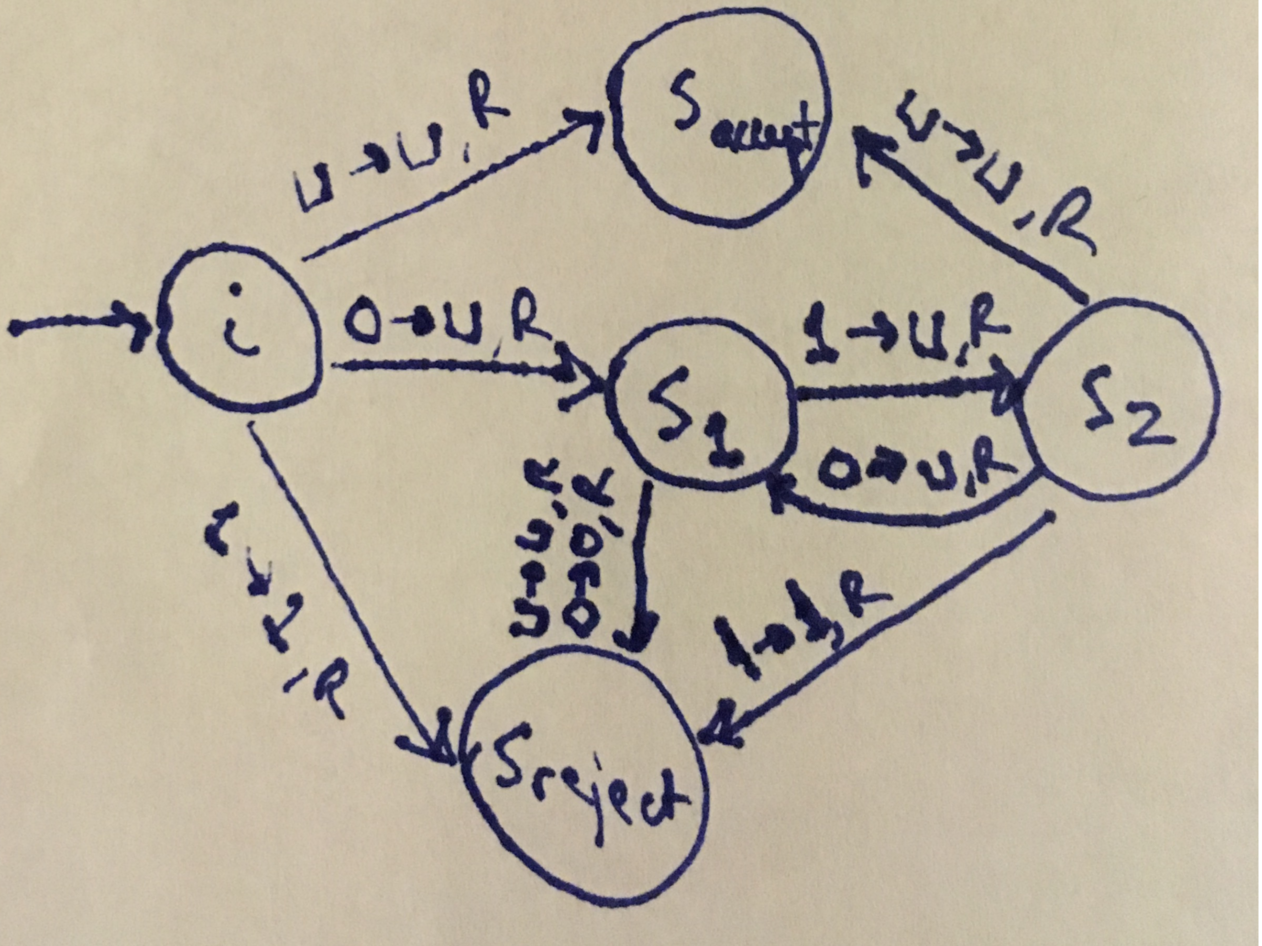
(1) The input is on the first tape (T1), and 0 is on the second tape (T2) initially.
(2) If the number on T2 is equal to the number on T1, ACCEPT. If the number on T2 is bigger than the number on T1, REJECT. If the number on T2 is smaller than the number on T1, add 3 to the number on T2 and write the result on T2.
(3) Go to step 2.

Here we might need to write as a subroutine the process via which a Turing machine ‘understands’ what the number on a tape is - how a sequence of cells like $\|5\|0\|1\|$ becomes the number 501 in the machine.

1) (a)


0
0
1
i
S1
S2
1
0
1
S3
1
0

(c)


Saccept
⊔→⊔,R
⊔→⊔,R
i
0→⊔,R
S1
1→⊔,R
S2
0→⊔,R
⊔→⊔,R
0→0,R
1→1,R
1→1,R
Sreject