Homework 4

Antonio Zea Jr

November 12, 2022

Problems

0.1 Given implementation-level description of a Turing machine M that decides the language $A = \{w_1 \sim w_2 | w_1, w_2 \in \{0,1\}^*$, w_2 is bitwise complement of$w_1\}$. For example, M should accept "$101 \sim 010$" and reject "$101 \sim 101$". Hint: see the Turing machine M_1 in the book

Scan the across the tape to corresponding positions on either side of the $\sim$ symbol to check whether these positions contain opossite symbols. If they do not, or if no $\sim$ is found, *reject.* Cross off symbols as they are checked to keep track of which symbols correspond.

When all symbols to the left of the $\sim$ have been crossed off, check for any remaining symbols to the right of the $\sim$. If any symbols remain, *reject* ; otherwise, *accept.*

0.2 Give a formal description of M including a state diagram for δ.

$Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_A, q_R\}$

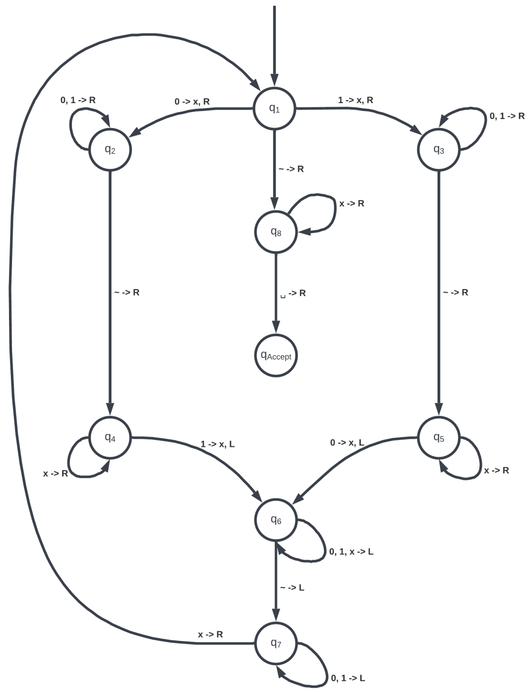
$\Sigma = \{0, 1, \sim\}$

$\Gamma = \{0, 1, \sim, x, \_\}$

$q_0 = q_1$

$q_{\textbf{Accept}} = q_A$

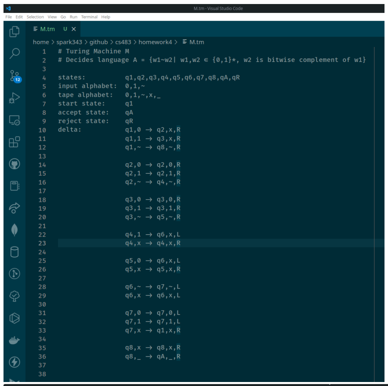
$q_{\textbf{Reject}} = q_R$



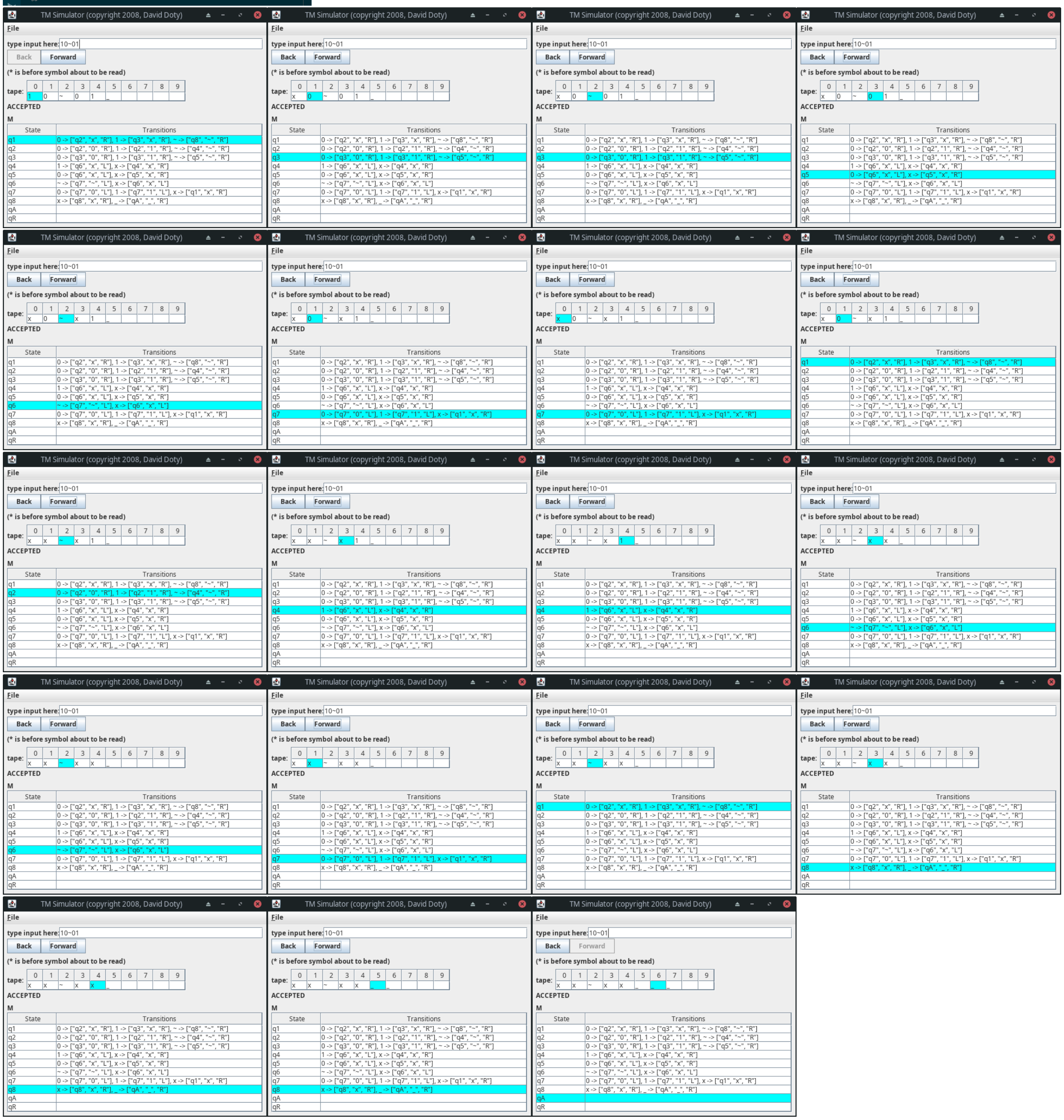
$\delta =$

	0	1	$\sim$	x	$\llcorner\lrcorner$
q_1	q_2, x, R	q_3, x, R	$q_8, \sim, R$	ϕ	ϕ
q_2	$q_2, 0, R$	$q_2, 1, R$	$q_4, \sim, R$	ϕ	ϕ
q_3	$q_3, 0, R$	$q_3, 1, R$	$q_5, \sim, R$	ϕ	ϕ
q_4	ϕ	q_6, x, L	ϕ	q_4, x, R	ϕ
q_5	q_6, x, L	ϕ	ϕ	q_5, x, R	ϕ
q_6	q_6, x, L	q_6, x, L	$q_7, \sim, L$	ϕ	
q_7	$q_7, 0, L$	$q_7, 1, L$	ϕ	q_1, x, R	ϕ
q_8	ϕ	ϕ	ϕ	q_8, x, R	$q_A, \llcorner\lrcorner, R$

0.3 Implement M (show the code) for the TMSimulator. Run the machine on the simulator and show the sequence of configurations that M enters when started on the string "$10 \sim 01$".



```
# Turing Machine M
# Decides language A = {w1~w2| w1,w2 ∈ {0,1}*, w2 is bitwise complement of w1}

states:          q1,q2,q3,q4,q5,q6,q7,q8,qA,qR
input alphabet:  0,1,~
tape alphabet:   0,1,~,x,_
start state:     q1
accept state:    qA
reject state:    qR
delta:           q1,0 → q2,x,R
                 q1,1 → q3,x,R
                 q1,~ → q8,~,R

                 q2,0 → q2,0,R
                 q2,1 → q2,1,R
                 q2,~ → q4,~,R

                 q3,0 → q3,0,R
                 q3,1 → q3,1,R
                 q3,~ → q5,~,R

                 q4,1 → q6,x,L
                 q4,x → q4,x,R

                 q5,0 → q6,x,L
                 q5,x → q5,x,R

                 q6,~ → q7,~,L
                 q6,x → q6,x,L

                 q7,0 → q7,0,L
                 q7,1 → q7,1,L
                 q7,x → q1,x,R

                 q8,x → q8,x,R
                 q8,_ → qA,_,R
```



0.4 Show that if languages L1 and L2 are decidable, then the intersection of L1 and L2 is also decidable.

For any two decidable languages L_1 and L_2, let M_1 and M_2 be the TMs that decide them. We construct a TM M' that decides the intersection of L_1 and L_2 :
"On input $w \in L_1 \cap L_2$:

1. Run M_1 and M_2 on w. Because M_1 and M_2 are deciders the TMs do not need to be run in lockstep unison.
2. If either rejects, then M' rejects.
3. If both accept, then M' accepts.

0.5 Show that if languages L1 and L2 are decidable, then concatenation of L1 and L2 is also decidable.

For any two decidable languages L_1 and L_2, let M_1 and M_2 be the TMs that decide them. We construct a TM M' that decides the concatenation of L_1 and L_2 :
"On input $w \in L_1 \circ L_2$:

1. Divide w into two substrings where $w = w_1 \circ w_2$, where $w_1 \in L_1$ and $w_2 \in L_2$.
2. Run M_1 on w_1. Run M_2 on w_2. Because M_1 and M_2 are deciders the TMs do not need to be run in lockstep unison.
3. If M_1 and M_2 accept then M' accepts.
4. This process needs to be repeated for every possible subdivision of w. All of those subdivisions need to be run against M_1 and M_2 in parallel.
5. M' rejects if all subdivisions reject."

0.6 Show that if languages L1 and L2 are recognizable, then the intersection of L1 and L2 is also recognizable.

For any two Turing-recognizable languages L_1 and L_2 , let M_1 and M_2 be the TMs that recognize them. We construct a TM $M\prime$ that recognizes the intersection of L_1 and L_2 :
"On input $w \in L_1 \cap L_2$:

1. Run M_1 and M_2 alternately on w step by step. If both accept, then M' accepts. If either halt and reject, then M' rejects."

If both M_1 and M_2 accept w, $M\prime$ accepts w because the accepting TM arrives to its accepting state after a finite number of steps. Note that if either M_1 or M_2 reject and either of them does so by looping, then $M\prime$ will loop.

0.7 Show that if languages L1 and L2 are recognizable, then the concatenation of L1 and L2 is also recognizable.

For any two Turing-recognizable languages L_1 and L_2, let M_1 and M_2 be the TMs that recognize them. We construct a TM M' that recognizes the concatenation of L_1 and L_2 :
"On input $w \in L_1 \circ L_2$:

1. Divide w into two substrings where $w = w_1 \circ w_2$, where $w_1 \in L_1$ and $w_2 \in L_2$.
2. Run M_1 and M_2 alternately on w_1 and w_2 in a step by step fashion.
3. If M_1 and M_2 accept then M' accepts.
4. This process needs to be repeated for every possible subdivision of w. All of those subdivisions need to be run against M_1 and M_2 in parallel.
5. M' rejects if all subdivisions reject."

0.8 Show that language $B = \{\langle A\rangle | A$ is a DFA and $L(A) = \Sigma^*\}$ is decidable.

A DFA accepts some string iff reaching an accept state from the start state by traveling along the arrows of the DFA is possible. To test this condition, we can design a TM T that uses a marking algorithm.
$T =$ "On input $\langle A\rangle$, where A is a DFA:

1. Mark the start state of A.
2. Repeat until no new states are marked:
3. Mark any state that has a transition coming into it from any state that is alread marked.
4. If no accept state is marked, reject; otherwise accept."

0.9 Show that language $C = \{\langle D, R\rangle | D$ is a DFA, R is a regular expression and $L(D) = L(R)\}$ is decidable.

By theorem 1.40, A language is regular if and only if some NFA recognizes it. Therefore, $\exists E$, an NFA that recognizes R.
By Theorem 1.39, Every NFA has an equivalent DFA. Therefore, $\exists F$, an DFA that recognizes R.

$L(D) = L(R) = L(F)$

$L(D) = L(F)$

By Theorem 4.5, two DFAs that recognize the same language is decidable, therefore C is decidable.