① Programming Project

CSCE 355
3/30/2022

Simulate a Turing machine computation

Def: A Turing machine (TM) is a tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, B, F \rangle$ where

Q is a finite set (elements called states)

Σ, Γ are alphabets:

Σ — the input alphabet (input strings are over Σ)

Γ — the tape alphabet (incl. all possible cell contents)

such that $\Sigma \subseteq \Gamma$

δ (later)

$q_0 \in Q$ (the start state),

$B \in \Gamma \setminus \Sigma$ (the blank symbol),

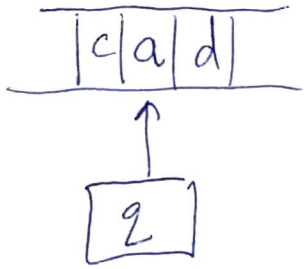
$F \subseteq Q$ (the set of accepting states)

and

δ is partial function $Q \times \Gamma \to Q \times \Gamma \times D$

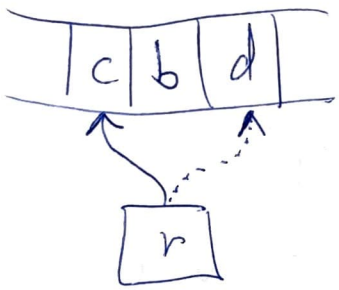
where $D := \{L, R\}$ L means left, R " right

(2)



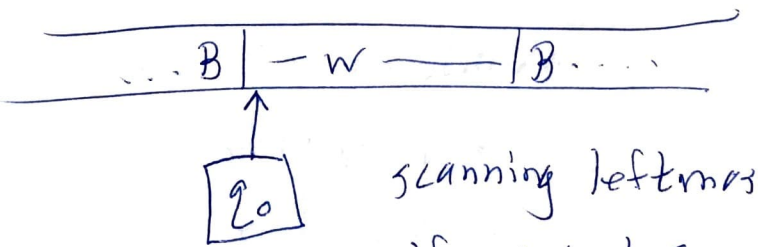
$\delta(q,a) = (r, b, \underset{R}{L})$

$\Rightarrow$ next step:



if $\delta(q,a)$ is undefined then the computation ends. (No next step). We say the machine halts.

Given TM $M = \langle Q, \Sigma, \Gamma, \delta, q_0, B, F \rangle$ and input $w \in \Sigma^*$, the initial configuration of M on input w looks like



scanning leftmost symbol of w, if $w \neq \varepsilon$.

If $w = \varepsilon$, then whole tape is blank.

If M halts on input w (after finitely many steps)

(3) M accepts or rejects w depending on ~~that~~ the state M was in when it halted (call it q_{last})

$q_{last} \in F$ means accept w	If M does not halt on input w, it neither accepts nor rejects. (it loops)
$q_{last} \notin F$ " reject w	

~~Say M is~~

Def: For any TM M with input alphabet Σ,

lang. recog. by M $\rightarrow L(M) := \{ w \in \Sigma^* : M \text{ accepts } w \}$

Def: M is total (or a decider) if M halts on all inputs. $L(M)$ is decided by M in this case.

Write a program that

1) Reads the description of a TM from a file given on the command line.
2) Read (from standard input) a series of strings, one per line.
   For each string w, give a step-by-step account of the TM's computation on the input w.
3) For each, say "accept" or "reject" at the end.

(4) Executable is tmsim, say.

$ ./tmsim TM_sample.txt
widget↵

0: (w)idget
3: x(i)dget
2: (x)jdget
1: (_)yjdget

$\delta(0,'w') = (3,'x',R)$
$\delta(3,'i') = (2,'j',L)$
$\delta(2,'x') = (1,'y',L)$

↑ space

$Q = \{0, 1, 2, \ldots, n\}$ where $n \leq 99$

0 is always the start state

Alphabet will consist ~~of only~~ only of printable chars that take up width 1, ~~and~~ (no tab, newlines) but no parentheses

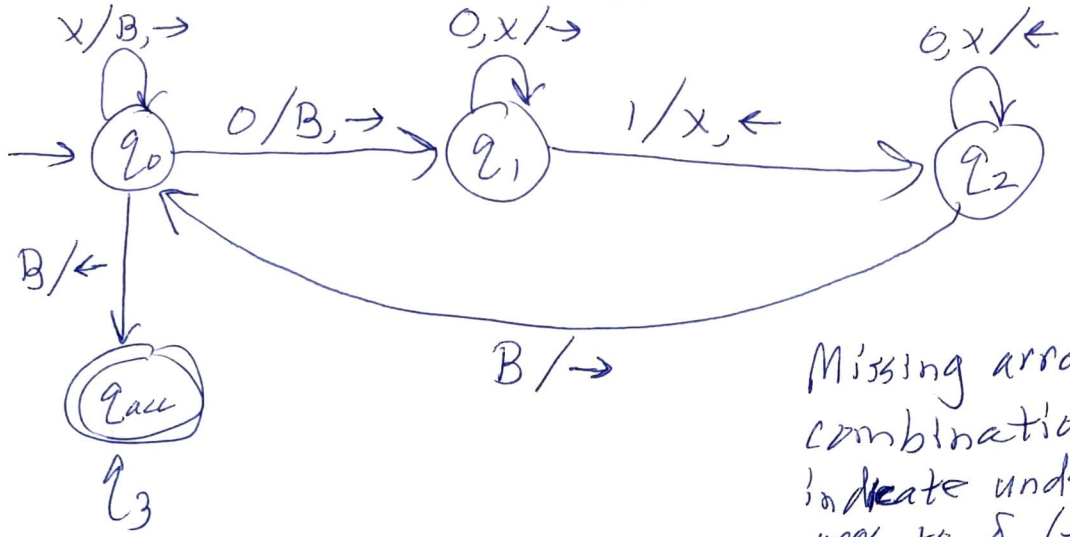
_ (underscore) is always the blank symbol

One unique accepting state: $q_{acc} = n$ (where $Q = \{0, \ldots, n\}$)

$\delta(q_{acc}, a)$ is always undefined.

Example (from the HW)
Transition diagram for a TM

(5)



Missing arrow combinations indicate undefined args to δ (thus halt)

means $\delta(q_0, x) = (q_0, B, R)$

$\delta(q_0, 0) = (q_1, B, R)$

$\delta(q_1, 0) =$ ~~$\delta(q_1, x)$~~ $= (q_1, 0, R)$

$\delta(q_1, x) = (q_1, x, R)$

File format describing this TM

TM_sample.txt:

Number of nonaccept states: 3

Alphabet: 01x_

~~0 x / 0 - >~~

0 x / 0 - >

0 0 / 1 _ >

0 _ / 3 _ <

...

(6) Input 0011

0 : (0)011

1 : (0)11

1 : 0(1)1

2 : (0)x1