

12 Apr 2024

Turing machine examples

Announcements:

① No homework release today!

Problem Set 8 to be released mid-week next week, we expect it will have 4 problems. Will be due Thurs, Apr 25.

② Prelim 2 grading schedule.

We're getting started tomorrow.

We can't finish until after we grade the make-up prelin, which is administered Mon. evening.

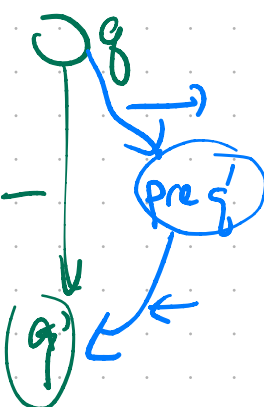
Reminder: Turing machine has.

- finite state set
- two alphabets $\Sigma \subseteq \Gamma$
input alphabet (pointing to Σ) and tape alphabet (pointing to Γ)
- infinite tape with \vdash symbol on left end that cannot be overwritten
- transition rule $\delta(q, x) = (q', y, d)$

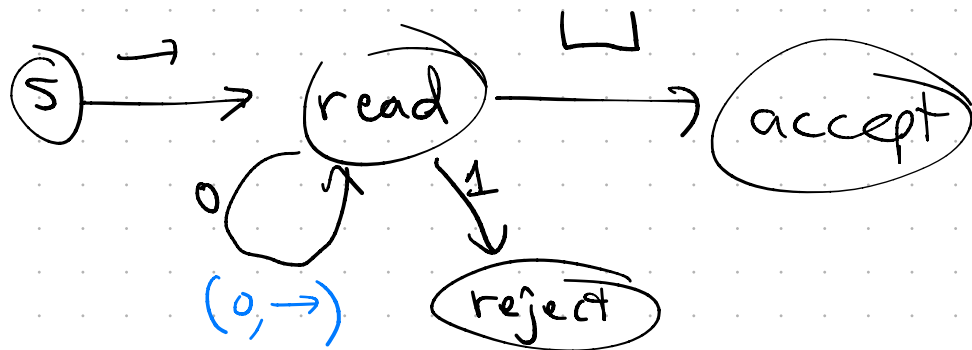
"If reading x in state q , write y , enter state q' , move in direction d ."

d is either LEFT or RIGHT, never stand still.

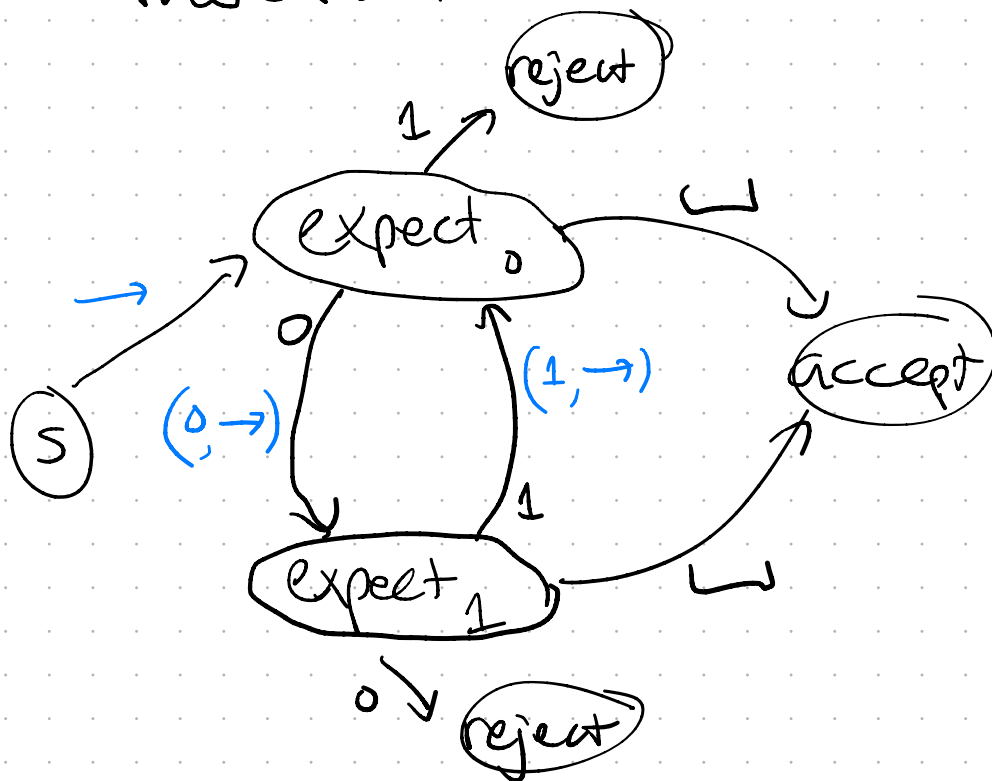
A standing still transition can be simulated using 2 steps + an extra "move left" state.



Ex. 1. Input alphabet $\{0,1\}$.
 TM should accept $\{0^n : n \in \mathbb{N}\}$.



Ex. 2 Accept binary strings that start with 0 and alternate 0 and 1 thereafter.



Ex 3. Accept binary strings that are palindromes
 (read the same backward as forward)

Tape alphabet will be $\{ \vdash, \sqcup, 0, 1, \overset{\wedge}{0}, \overset{\wedge}{1} \}$
 ↑ ↑
 symbols already checked

1: repeat
 move right
 2:
 3: $x =$ symbol on tape
 4: if x has \wedge , accept

 5: write \hat{x}
 repeat
 move right
 6:
 7:
 8: until symbol on tape = \sqcup or has \wedge
 9: move left
 10: if symbol on tape has \wedge , accept

 11: if symbol on tape $\neq x$
 reject
 12:
 13: else
 write \wedge over symbol
 repeat
 move left
 14:
 15:
 16: until symbol on tape has \wedge
 17:
 18: forever

Translation to TM:

- "marking with \wedge "
 → enlarging tape alphabet
 $\{0, 1\} \mapsto \{0, 1, \hat{0}, \hat{1}\}$

- State set $Q = [18] \times \{0, 1\}$
 ↑ ↑
 program value of x
 counter

Can have finite # of variables each taking values in a finite set.

If the pseudocode uses function calls, the call stack depth should be bounded by a finite # that depends only on the program, not the input.