CPCS 421/501 Oct 18, 2023

Today start Turing machines:

Chapter 3 of [Sip].

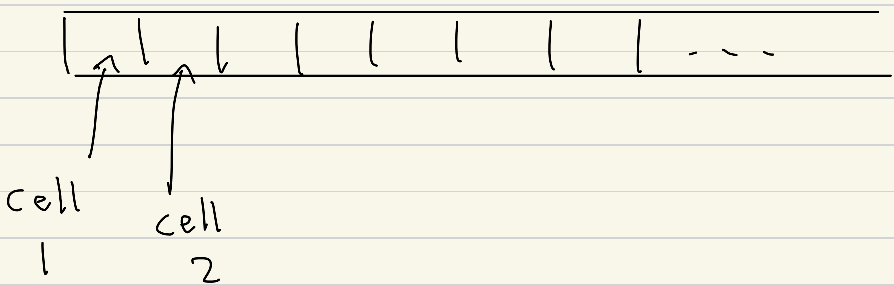
DFA is $(Q, \Sigma, \delta, q_0, F)$

TM is $(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej}, \sqcup)$

or simply:

$$(Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$$

Idea: a Turing machine has a tape



Cell contains a "tape symbol":

Tape alphabet:

Γ — contains Σ,

Σ = input alphabet

— contains blank symbol, $\sqcup$

$\sqcup \notin \Sigma$

— contains any other symbols

you want

M = Turing machine, over $\Sigma = \{a, b\}$

on input $i = abba \in \Sigma^*$

then initially the tape looks like

a	b	b	a	⊔	⊔	⊔	⊔	...

tape head – can move right but
also to the left

tape is a read/write tape

Mechanics:

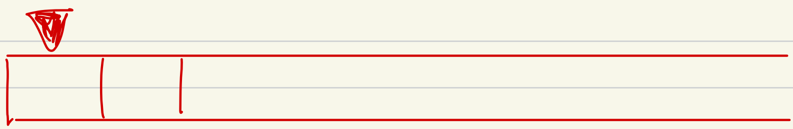
...	b	b	⊔	b	a	b	a	⊔	⊔	⊔	...

δ function tells us: (1) what is

written over "a" — any symbol in Γ

(2) do we move L or R

Q: What happens when you get an instruction to move L = left when



Let's say (most standard): in this case you don't move, tape head remains over cell 1.

Q: What about S = stay still versus L, R

For now (but not for 2-tape or multi-tape TM) we don't allow S, or L, R

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

Formally, a Turing machine is a 7-tuple

$$(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$$

Set of states

input alphabet

tape ~~ta~~alphabet

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

intuitively

$$\delta(q, \gamma) = (q', \gamma', \text{L or R})$$

q' = new state, γ' = symbol written over γ where ▼ is situated

q_{accept}, q_{reject} states s.t. when you reach them, the T.M. "halts"

Any T.M., on input $i \in \Sigma^*$ either

accepts i — you transition to q_{acc}

rejects i — " " " q_{rej}

"loops" — doesn't halts, i.e. never reaches q_{acc}, q_{rej}

A T.M, M, <u>recognizes</u> the language

$$\{ i \in \Sigma^* \mid M \text{ accepts } i \}$$

A T.M., M, is a <u>decider</u> if it always halts, i.e. for any $i \in \Sigma^*$, M either accepts or rejects i

A language $L \subset \Sigma^*$, is

<u>(Turing-) recognizable</u> if some

T.M., M, recognizes L,

and is

(Turing-) <u>decidable</u> if some

T.M., M, that is a decider

recognizes L.

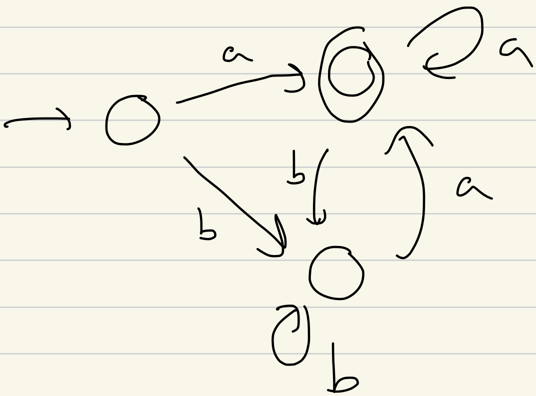
Remark: Unlike Python (C, APL, ...)

programs, T.M.'s have only a

single accept state, q_{acc}, and a single reject state

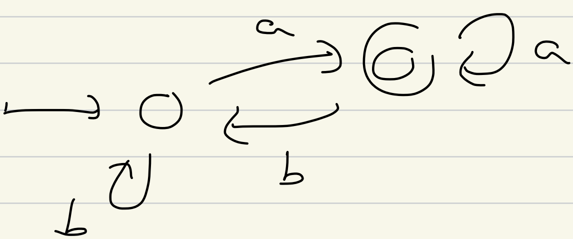
$$L = \left\{ w \in \{a,b\}^* \;\middle|\; \begin{array}{l} w \text{ is non-empty, and} \\ w \text{ ends in an ``a''} \end{array} \right\}$$

DFA:

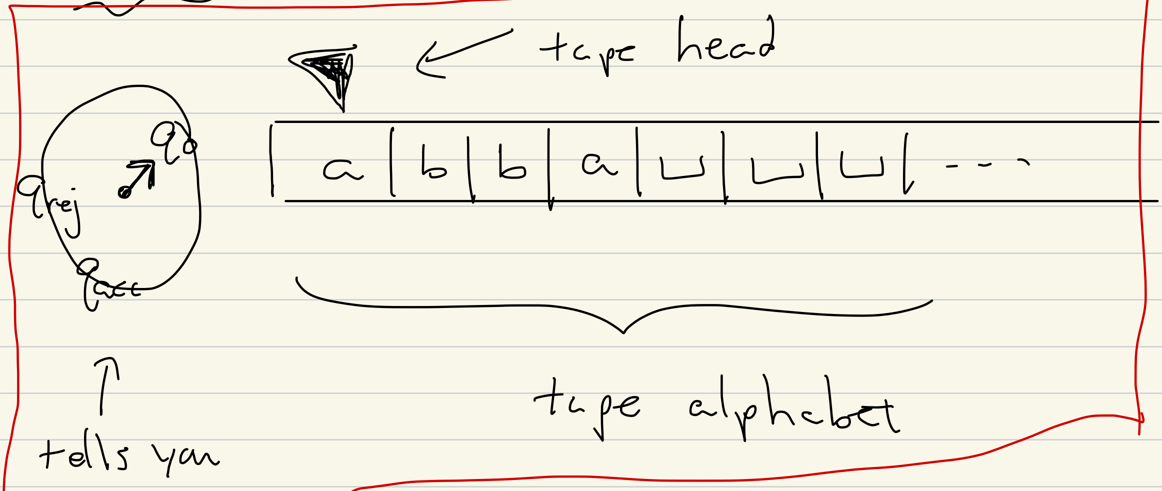


AccFut(ε) = AccFut(b) and therefore

also



T.M. (technically, although $q_{acc} \neq q_{rej}$, q_0 can be one of q_{acc}, q_{rej})



OR

this indicates tape head is over cell 1

$$\begin{matrix} \_\_ & & & \\ a & b & b & a \\ q_0 & & & \end{matrix}$$

q_0abba or $q_0abba\sqcup\sqcup$