Turing machines:

A mathematical model that as far as we can tell, is equal in power to an actual computer.

A Turing is

Q - a set of states

Γ - a tape alphabet

$\Sigma \subseteq \Gamma$ - the input alphabet

$\square \in \Gamma$ - the blank symbol

$q_0 \in Q$ - the start state

$F \subseteq Q$ - final states

$$\delta : (Q \times \Gamma) \longrightarrow (Q \times \Gamma \times \{L, R\}) \cup \{\underset{\text{halt}}{\mathcal{H}}\}.$$

As a convention - we only specify transitions that don't lead to $\mathcal{H}$, and unspecified transitions go to $\mathcal{H}$.

How does a TM work?

An instantaneous description of a TM is an element of

$$\Gamma^* \times Q \times \Gamma^*$$

An instantaneous description (u, q, v) yields (u', q', v') in one step (written

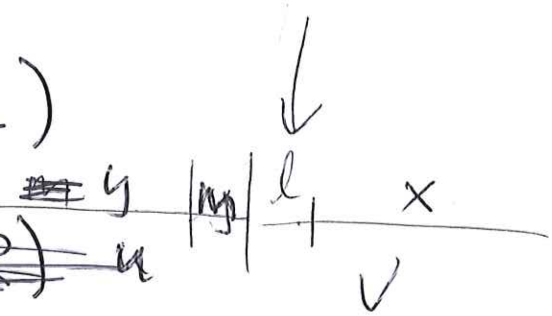
$$(u, q, v) \vdash (u', q', v')$$

)

if

$$\delta(q, \ell) = (q', \ell', L)$$

$\cancel{(q', \ell', R)}$



and

$$v = \ell x \qquad v' = m\ell' x$$

$$u = ym \qquad u' = y$$

for some $\ell', \ell, m \in \Gamma$,
$x, y \in \Gamma^*$

or if

$$\delta(q,\ell)=(q',\ell',R)$$

$$v=\ell x \qquad v'=x$$

$$\cancel{u} \qquad u'=u\ell'$$

Human description:

The TM has an infinite tape, and a head over one spot on the tape.

Given ~~our~~ the current state and what's read under the head, the transition function tells us

1) new state
2) what to write under the head
3) whether to move head L or R (on the tape)

In an inst. descr. (u,q,v),

u is stuff to left of head
q is state
v is stuff under and to right of head.

More conventions around input and acceptance -

head on first char.

1) We start with the input string, surrounded by infinitely many blanks $\square$.

2) We accept if we halt (i.e. δ goes to $\mathcal{H}$) in a final state (no matter what's on tape)

---

1) initial instant descr. is

$$(\lambda, q_0, w)$$

input.

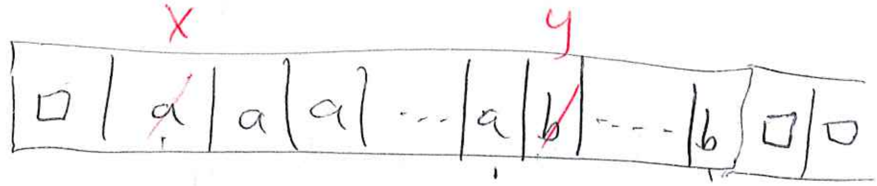
2) accept if last instant descr is

$$(u, q_f, v), \text{ where } q_f \in F,$$

and $\delta(q_f, \ell) = \mathcal{H}$, where ℓ is first letter of v (i.e. $v = \ell v'$).

Let's construct TM to accept

$\{a^n b^n \mid n \geq 1\}$



Idea: match a's & b's - mark matched items w/ x's & y's.

q_0 - initial state

$\delta(q_0, a) = (\overset{q_1}{\cancel{x}}, \overset{x}{\cancel{q_1}}, R)$

$\delta(q_1, a) = (\overset{q_1}{\cancel{a}}, \overset{a}{\cancel{q_1}}, R)$

$\delta(q_1, b) = (q_2, y, L)$

$\delta(q_2, a) = (q_2, a, L)$

$\delta(q_2, x) = (q_0, x, R)$

$\delta(q_1, y) = (q_1, y, R)$ ← needed the 2nd time through

$\delta(q_2, y) = (q_2, y, L)$

$\delta(q_0, y) = (q_3, y, R)$ ← this means we've matched all our a's

$\delta(q_3, y) = (q_3, y, R)$

q_0 - start state

q_1 - found an a, moving to right to match a b.

q_2 - matched - need to go back to find first a.

$\delta(q_3, \square) = (q_f, \square, R)$

q_f is final state.

We check that this accepts everything it chould and reject everything it should.

⋮