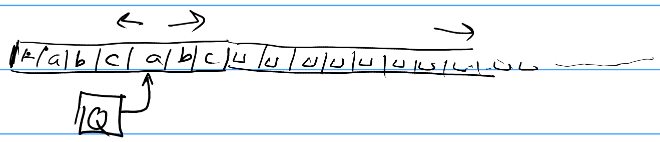
# Turing Machines



In addition to be able to read the input by moving its head in both directions, the TM can also delete a character on the tape and write another character.

$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$

where

$Q$ is the set states

$\Sigma$ is the input alphabet

$\Gamma$ is the tape alphabet, $\Sigma \subseteq \Gamma$

$\vdash \in \Gamma - \Sigma$ is the left endmarker

$\sqcup \in \Gamma - \Sigma$ is the blank symbol

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

$s \in Q$ is the start state

$t \in Q$ is the accept state

$r \in Q$ is the reject state

$\delta(p, a) = (q, b, d) \iff$ the head scans symbol $a$ on the input tape while $M$ is in state $p$, and then $M$ goes to state $q$, the head writes $b$ in place of $a$, and moves one step in direction $d \in \{L, R\}$

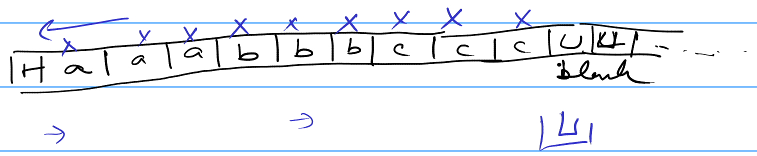
We forbid the TM to delete $\vdash$ and move to the left beyond the tape. So $\forall\ p \in Q\ \exists\ q \in Q$ s.t.

$\delta(p, \vdash) = (q, \vdash, R)$

Once M reaches the accept or reject state, it never leaves that state. So $\forall\, b \in \Gamma\ \exists\ c, c' \in \Gamma$ and $d, d' \in \{L, R\}$ s.t.

$$\delta(t, b) = (t, c, d)$$
$$\delta(r, b) = (r, c', d')$$

$a^n b^n c^n$ is not a CFL




i) Checks if input is of the form $a^x b^y c^z$

ii) In each of its left to right or right to left run, it deletes an equal no. of a, b, c.

iii) If it finds that one character is exhausted and others are not, then it rejects. Otherwise it accepts.

---

Configuration of a TM:

$$Q \times \{ y \sqcup^{\omega} \mid y \in \Gamma^* \} \times \mathbb{N}$$

$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$

$(p, z, n)$ index marking the position of the header

start configuration : $(s, \vdash x \sqcup^{\omega}, 0)$

M is said to accept $x \in \Sigma^*$ if

$$(s, \vdash x \sqcup^{\omega}, 0) \xrightarrow[M]{*} (t, y, n)$$, for some $y$ and $n$

M is said to reject $x \in \Sigma^*$ if

$$(s, \vdash x \sqcup^{\omega}, 0) \xrightarrow[M]{*} (r, y, n)$$, for some $y$ and $n$.

M is said to halt on input $x$ if M either accepts or rejects $x$.

If M neither accepts nor rejects $x$, then M is said to loop on $x$.

A set $S \subseteq \Sigma^*$ is called recursively enumerable (r.e.) if $S = L(M)$ for some T.M. M.

S is called co-r.e. if the complement of S is r.e.

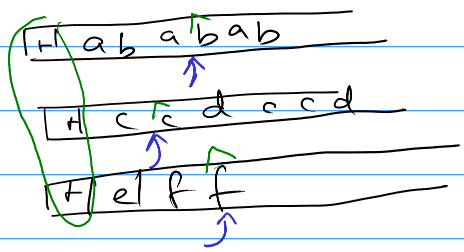
If a TM halts on all inputs, it is called a total TM

S is recursive if $S = L(M)$ for a total TM M.

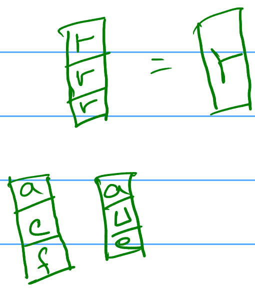
---

## The power of a Turing Machine.

1) TM with multiple tapes.

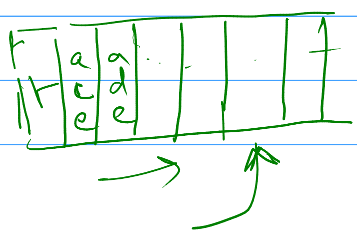


← paste the tapes together



$\Gamma_1 \times \Gamma_2 \times \Gamma_3 \to \Gamma_4$

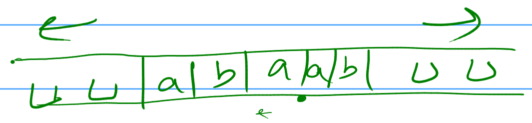
$2|\Gamma_1| \times 2|\Gamma_2| \times 2|\Gamma_3| \simeq |\Gamma_4|$

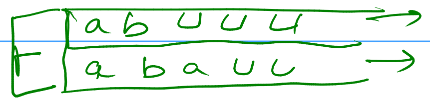


$(\hat{a}, d, e) \to x_1 \in \Gamma_4$

$(b, d, e) \to y_1 \in \Gamma_4$

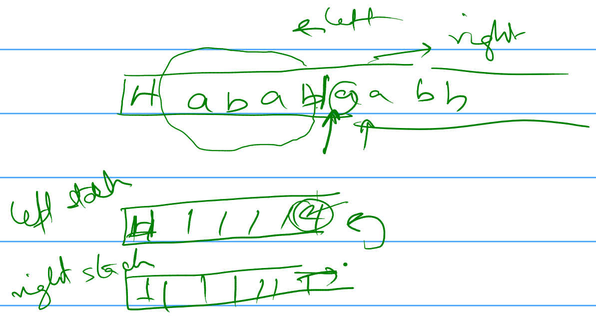
two way infinite tapes





---

PDA with two stacks and head that moves both ways.




---

Can a TM be written in such a way that another TM can simulate it?

Is there a particular TM U and a standard way of encoding any TM M s.t. U can simulate M?

Universal Turing Machine

$$(Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

$\Sigma = (a, b)$ $\quad$ $\Gamma$ has $a, b, c, d$

$(0, 1)$ $\quad$ 101 1001 10001 10000

$\boxed{a}$

$\boxed{1\,0\,1}$

$0^n \, 1 0^m \, 1 0^k \ldots$

n states — m tape symbols — k of them are input symbols ..

$(p, a), (q, b, L)$

$0^p 1 0^a 1 0^q 1 0^b 1 0$ — 0 left, 00 right

given an encoding of a TM and an input

M # x (input)

101 001 111 1001 . .