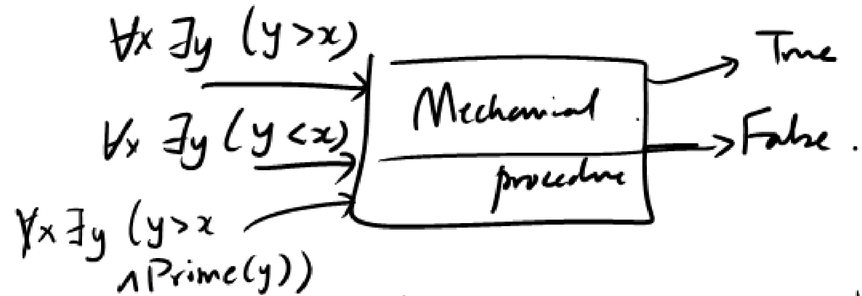
Lecture #12:

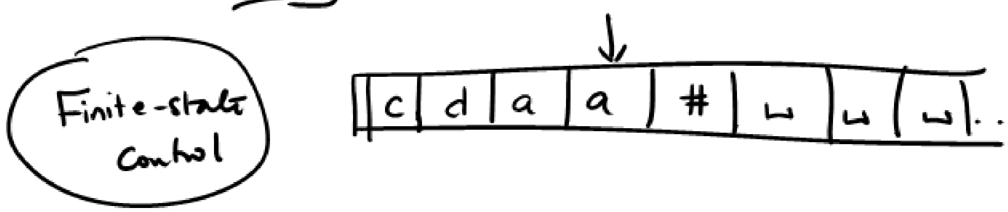
Turing and his machine.

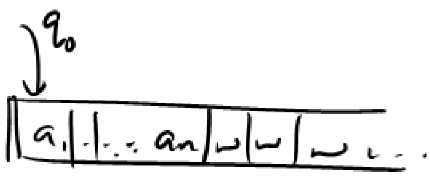
$\mathbb{N}$.



$\mathrm{Prime}(y) : \forall z \, (z \neq 1 \wedge z \neq y \Rightarrow z \nmid y)$

Turing machines





Σ - input alphabet

Γ - tape alphabet

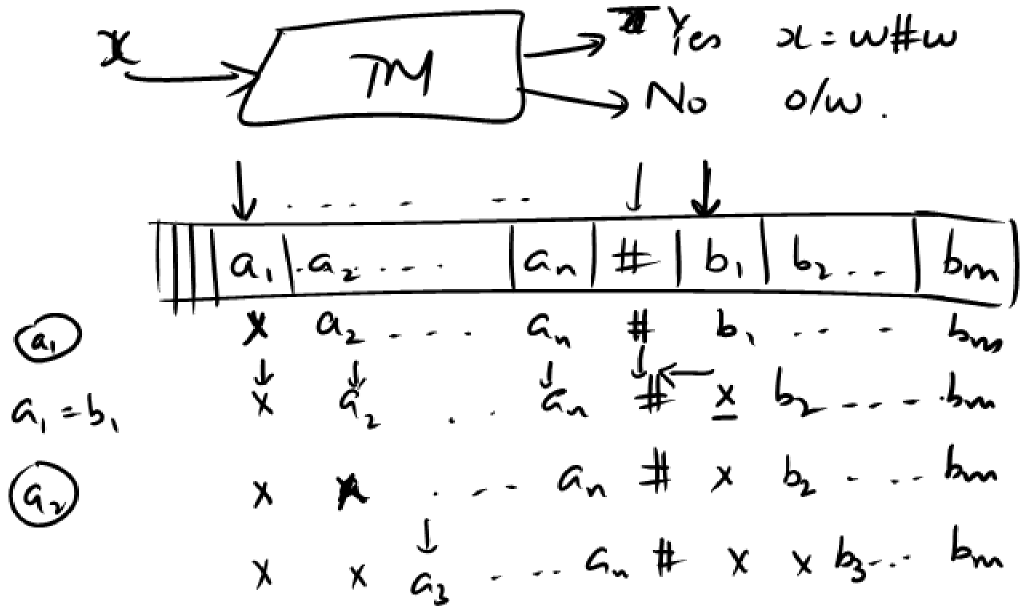
$\Sigma \subseteq \Gamma$

$␣ \in \Gamma$.

$␣ \notin \Sigma$

Eg. $\{ w\#w \mid w \in \{0,1\}^* \}$

$\Sigma = \{0, 1, \#\}$



Turing machine is a tuple $(Q, \Sigma, \Gamma, \sqcup, \delta, q_0, q_{acc}, q_{rej})$

Q - finite set (of states)

Σ - finite set (input alphabet)

Γ - ~~tp~~ finite set (tape alphabet) $\Sigma \subseteq \Gamma$

$\sqcup \in \Gamma$, $\sqcup \notin \Sigma$ (blank symbol)

$q_0 \in Q$ (initial state)

$q_{acc} \in Q$ (accepting state)

$q_{rej} \in Q$ (rejecting state), $q_{acc} \neq q_{rej}$

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

A configuration:

- tape contents — string $s \in \Gamma^*$
- head position on tape
- current (control) state. (in Q)

Configuration : uqv $\quad u, v \in \Gamma^*$

Configuration : $C \in \Gamma^*.Q.\Gamma^*$

Moves of the TM

$C \vdash C'$ "C yields C'" Only when $q \neq q_{acc}$ & $q \neq q_{rej}$.

$uaqbv \vdash uq'acv$ if $\delta(q,b) = (q',c,L)$

$\vdash uacq'v$ if $\delta(q,b) = (q',c,R)$

Special cases.

$qbv \vdash q'cv$ if $\delta(q,b) = (q',c,L)$

$\vdash cq'v$ if $\delta(q,b) = (q',c,R)$

$uaq \vdash uq'ac$ if $\delta(q,\sqcup) = (q',c,L)$

$\vdash uacq'$ if $\delta(q,\sqcup) = (q',c,R)$

TM accepts a word $w \in \Sigma^*$

if there is a sequence $C_1, \ldots, C_K$ of configurations

- C_1 is the start configuration $C_1 = q_0 w$
- $C_i \vdash C_{i+1} \quad \forall i \in [1, k-1]$
- C_K is an accepting configuration

  i.e. $C_K = u\, q_{acc}\, v$ for some $u, v \in \Gamma^*$.

$$L(M) = \{ w \in \Sigma^* \mid M \text{ accepts } w \}.$$

A TM M <u>halts</u> on w

if $\exists$ seq. of config $C_1, \ldots C_k$

- C_1 is start config
- $C_i \vdash C_{i+1}$
- C_k is either accepting of rejecting (i.e. halting)

$C_k = u\, q_{acc}\, v$ or $C_k = u\, q_{rej}\, v$.

M is a <u>decider</u> if it halts on all inputs.

A language $L \subseteq \Sigma^*$ is Turing-recognizable if there is a TM M, $L(M) = L$.

A language $L \subseteq \Sigma^*$ is Turing-decidable if there is a TM M, $L(M) = L$ and M is a decider.

Church-Turing hypothesis.
The set of real-word computable functions are precisely those that are Turing-decidable.