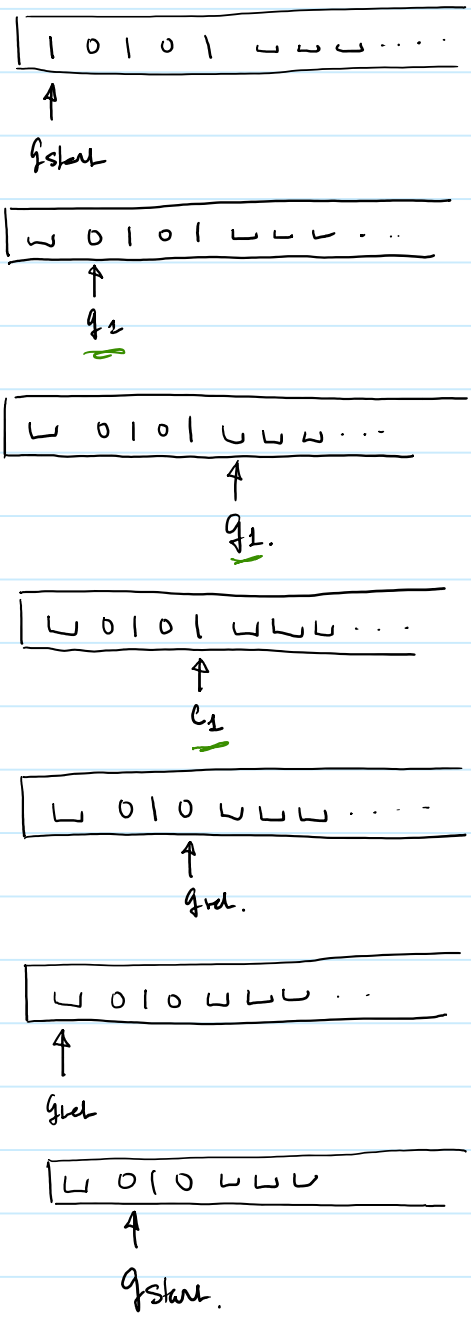
Wednesday, April 4, 2018 8:07 AM

Turing Machine Example: recognizing palindromes. ($x = x^R$).

$Q = \{ q_{start}, q_0, q_1, q_{ret}, c_0, c_1, q_{acc}, q_{rej} \}$

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



$\delta(q_{start}, 1) = (q_1, \sqcup, R)$
$\delta(q_{start}, 0) = (q_0, \sqcup, R)$
$\delta(q_{start}, \sqcup) = (q_{acc}, -, -)$

$\delta(q_1, 0) = (q_1, 0, R)$
$\delta(q_0, 0) = (q_0, 0, R)$
same for 1.

$\delta(q_0, \sqcup) = (c_0, \sqcup, L)$
$\delta(q_1, \sqcup) = (c_1, \sqcup, L)$

$\delta(c_1, 0) = q_{rej}$
$\delta(c_0, 1) = q_{rej}$

$\delta(c_1, 1) = (q_{ret}, \sqcup, L)$
$\delta(c_0, 0) = (q_{ret}, \sqcup, L)$

$\delta(c_1^{c_0}, \sqcup) = q_{acc}$

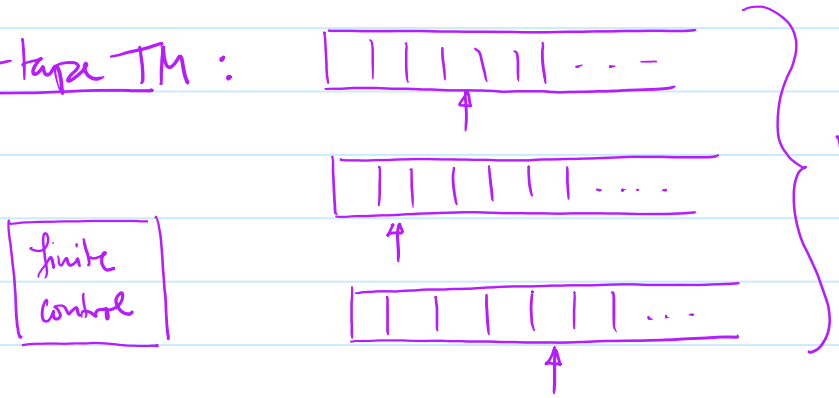
$\delta(q_{ret}, 0/1) = (q_{ret}, 0/1, L)$
$\delta(q_{ret}, \sqcup) = (q_{start}, \sqcup, R)$

Wednesday, April 4, 2018 8:07 AM

TM is robust to variation:

For Example A multi-tape TM can be simulated by a single-tape TM with only polynomial overhead.

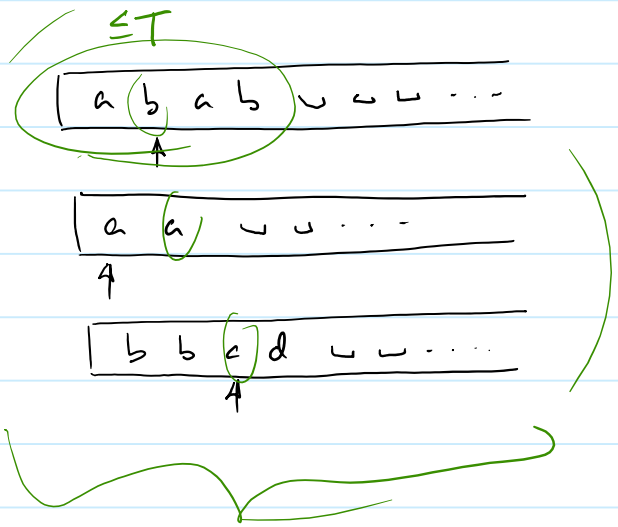
Multi-tape TM:



$\delta: Q \times \Sigma^k \to Q \times \Sigma^k \times \{L, R, -\}^k$

Can simulate a k-tape TM using a single tape TM

for each $a \in \Sigma$ add $\boxed{a}$ to Σ.
Also add # to Σ.



M'

a [b] a b # [a] a # b b [c] d # ....

live

Wednesday, April 4, 2018 8:08 AM

M k-tape TM
Σ

M' 1-tape TM
$\Sigma \cup \Sigma^{\square} \cup \{\#\}$

One step of M:

head scans entire tape remembering each symbol at head (stored in state).
Then scans back to implement the step on each tape.

If a # is reached then the contents of the tape are moved over to make room.

M' accepts or rejects if M accepts or rejects.

If M takes T steps on input x:

Non-blank contents of each tape $\leq T+1$ cells.

Non-blank contents of M's tape $\leq k(T+1)$ cells.

1 step of M takes: $O(T)$ steps in M'.

M' takes $O(T^2)$ on x.

Wednesday, April 4, 2018 8:08 AM

Def

A multi-tape TM M computes language L in time $t(n)$ if

$\forall x \in \Sigma^*$ on input x M "halts" (i.e reaches q_{acc} or q_{rej}) in at most $t(|x|)$ steps.

$x \in L \iff M$ accepts x.

The "Extended" Church-Turing Thesis

Every language that can be computed in time $t(n)$ on any physical computing device can be computed by a Turing Machine in time $[t(n)]^c$ for some constant c.

Wednesday, April 4, 2018 8:08 AM

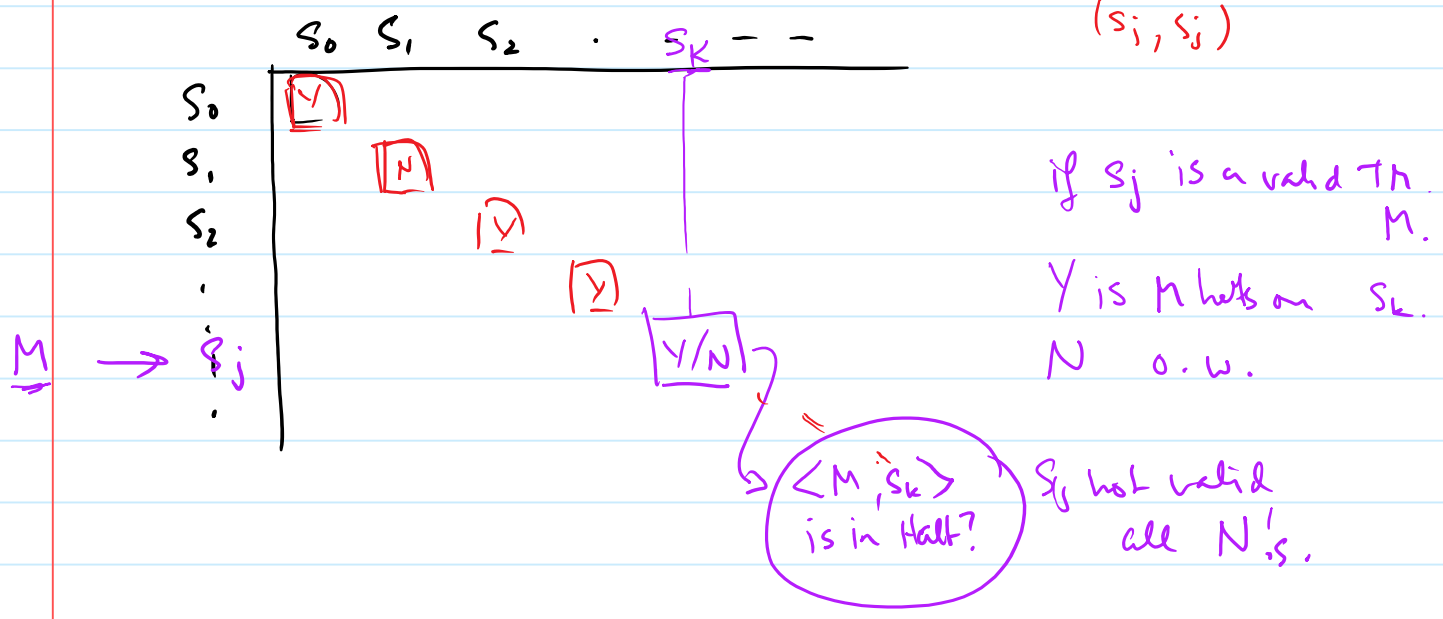
L is decided by a TM M if
$\forall x$ M halts on input x after a finite # of steps
(finite — no other restriction.)

$x \in L \iff M$ accepts x

Halt = $\{ \langle M, x \rangle : M \text{ halts on input } x \}$.
($\langle M, x \rangle$ — encoding using alphabet Σ.)

Theorem: Halt is undecidable.

enumerate all strings in Σ^*
$s_0\ s_1\ s_2 \ldots$
0 1 00 01 10 ...



⇒ Suppose H decides Halt.
On input (s_j, s_k) outputs cell of table.

Wednesday, April 4, 2018 8:08 AM

Use H to construct TM H'.

On input s_j:

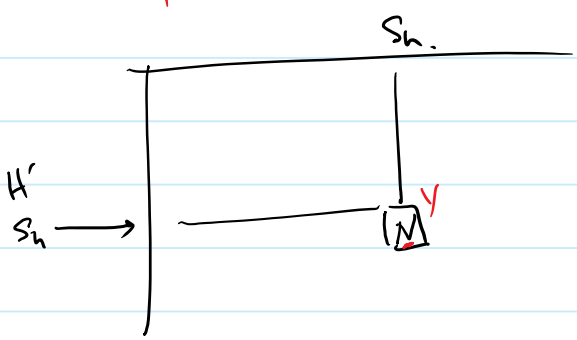
Simulate H on (s_j, s_j)

if H transition to qacc → H' goes into an inf loop.

If H " qrej → H' goes qacc.

H' is a TM.

corresponds to some s_h.



H' halts on input s_h.

H' loops forever on s_h.

⇒ H that decides Halt can not exist.

Complexity Classes: TIME($f(n)$) $f: \mathbb{N} \to \mathbb{N}$.

Set of languages decided by a multi-tape TM in $\leq f(n)$ steps.

SPACE $(f(n))$ uses $\leq f(n)$ tape squares.

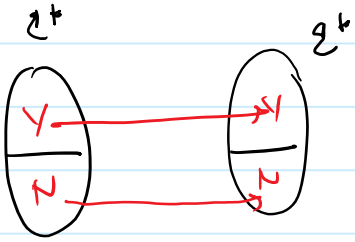
$P = \bigcup_{k \geq 1} TIME(n^k)$

Wednesday, April 4, 2018 8:08 AM

Two languages L_1 & L_2 L_1 "reduces to" L_2.
$L_1 \propto L_2$

if $\exists$ an efficient (poly-time) alg that computes f such that

$x \in L_1 \iff f(x) \in L_2$.
$x \notin L_1 \iff f(x) \notin L_2$



If $L_1 \propto L_2$ & $L_2 \in P \Rightarrow L_1 \in P$

If $L_1 \propto L_2$ & $L_1 \notin P \Rightarrow L_2 \notin P$.

Example 3SAT $\propto$ Indep Set.

3SAT = $\{\phi : \phi$ is 3-CNF formula that have a sat assign$\}$

$(x_1 \vee x_7 \vee \neg x_9) \wedge (\neg x_2 \vee x_7) \wedge (x_5) \wedge \ldots$

Indep Set: Input: undirected graph G & k.
Is there an indep set of size $\geq k$?