Recursive and recursively enumerable (r.e) Sets.

- Every recursive set is r.e
- Not every TM is equivalent to a total TM.

Recursive sets are closed under complementation.

Suppose $A \subseteq \Sigma^*$ is recursive. Then there exists a total TM M s.t $L(M) = A$.

- Switch the accept and reject states. Resulting M': $L(M') = \Sigma^* - A$.

This construction does not work for r.e. sets.

Rejecting and not accepting is not the same in a TM.

M' will still loop on the strings that M loops on.

Such strings are not accepted or rejected by either machines.

Claim. if both A and $\bar{A}$ are r.e then A is recursive

Claim. For $A \subseteq \Sigma^*$, if A is r.e. and $\overline{A}$ is r.e. then A is recursive.

Proof. Let $L(M_1) = A$ and $L(M_2) = \overline{A}$.

Construct M that on input x runs both M_1 & M_2 simulteneously on two tracks of the tape

$\vdash$	a	a	b	a	b	⊔	⊔
$\vdash$	b	b	b	a	b	b	⊔

if M_1 accepts then M accepts. if M_2 accepts then M rejects

$$\left.\begin{array}{l} x \in A \Rightarrow x \in L(M_1) \Rightarrow M_1 \text{ accepts} \Rightarrow M \text{ accepts.} \\ x \notin A \Rightarrow x \in L(M_2) \Rightarrow M_2 \text{ accepts} \Rightarrow M \text{ rejects} \end{array}\right\} \text{Total TM}$$

Decidable / Semi-decidable.

A property P of strings is decidable if the set of all strings having property P is a recursive set.

P is decidable iff $\{x \mid P(x)\}$ is recursive.

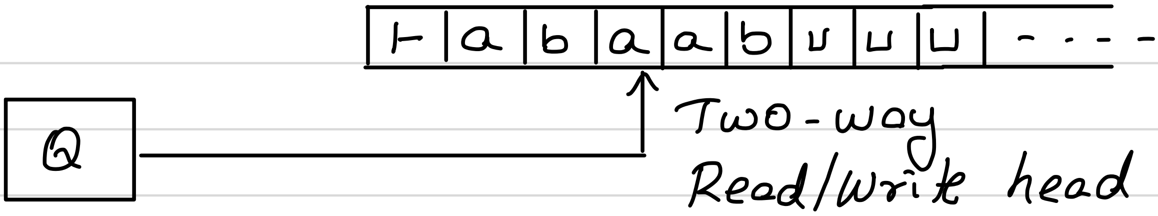
$A \subseteq \Sigma^*$ is recursive iff $x \in A$ is decidable

P is semidecidable iff $\{x \mid P(x)\}$ is r.e

$A \subseteq \Sigma^*$ is r.e iff $x \in A$ is semidecidable.

Example of a property: String x is of the form ww.

Turing machines - equivalent models.



Tape with multiple tracks.

⊢	⊢	a	a	b	a	b	⊔	⊔
	⊢	b	b	b	a	b	b	⊔
	⊢	b	a	a	a	b	a	⊔

Tape symbol is a triple $(c, d, e) \mapsto \begin{array}{|c|}\hline c \\ \hline d \\ \hline e \\ \hline\end{array}$

Claim. For $A \subseteq \Sigma^*$, if A is r.e. and $\overline{A}$ is r.e. then A is recursive.

Proof. Let $L(M_1) = A$ and $L(M_2) = \overline{A}$.

Construct M that on input x runs both M_1 & M_2 simultaneously on two tracks of its tape.

⊢	a	a	b	a	b	⊔	⊔
⊢	b	b	b	a	b	b	⊔

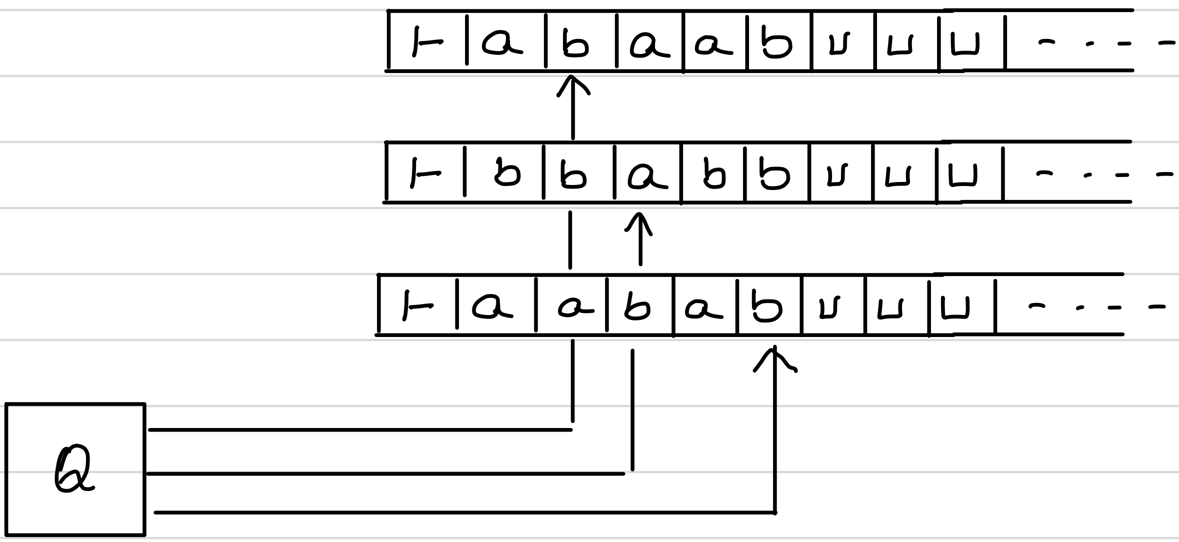
if M_1 accepts then M accepts. if M_2 accepts then M rejects

$x \in A \Rightarrow x \in L(M_1) \Rightarrow M_1$ accepts $\Rightarrow M$ accepts.

$x \notin A \Rightarrow x \in L(M_2) \Rightarrow M_2$ accepts $\Rightarrow M$ rejects

$\Big\}$ Total TM

Multiple tapes



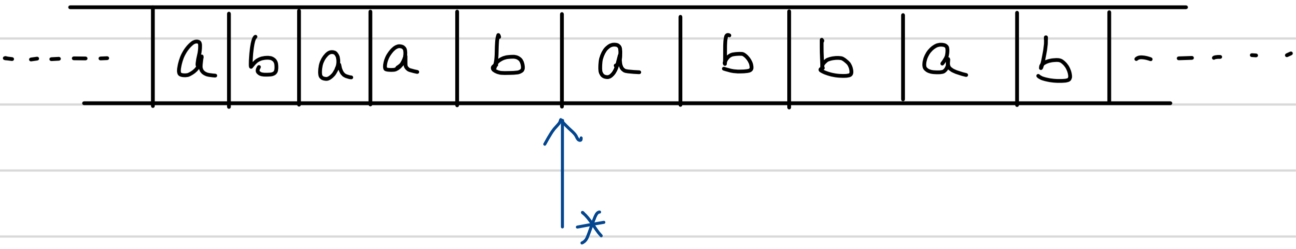
$$\delta : Q \times \Gamma^3 \to Q \times \Gamma^3 \times \{L, R\}^3$$

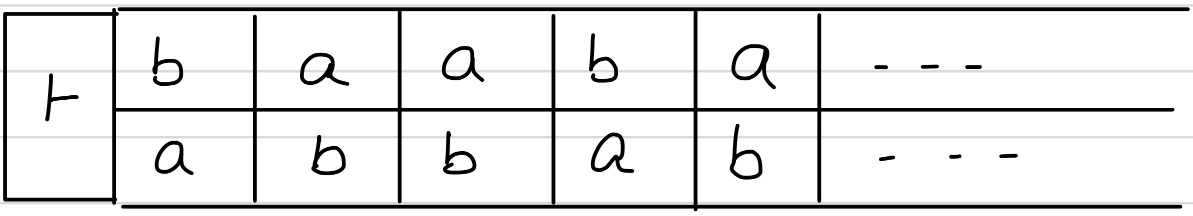
	$\vdash$	a	b	a	a	b	⊔	⊔	
$\vdash$	$\vdash$	b	b	a	b	b	⊔	⊔	
	$\vdash$	a	a	b	a	b	⊔	⊔	

Consider the tape alphabet

$$\Sigma \cup \{\vdash\} \cup (\Gamma \cup \Gamma')^3 \quad \text{where } \Gamma' = \{\hat{a} \mid a \in \Gamma\}$$

Two way infinite tape

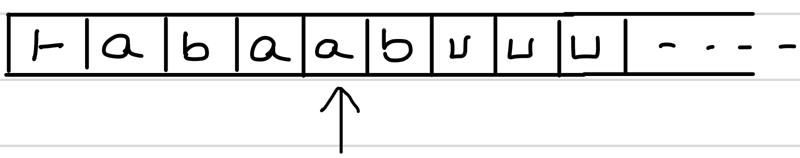


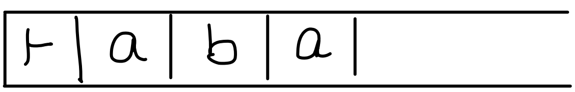


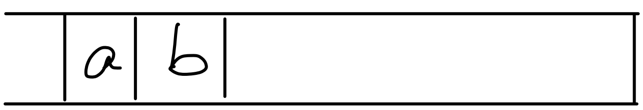
Simulate top track when head is on the left of * and simulate bottom track when head is on the right of *

Two stacks.

Claim. A finite state machine with a two-way, read only input head and two stacks is as powerful as a Turing machine.







Power of the model - Universal Turing machine.

Fix an encoding scheme of TM over some alphabet say $\{0,1\}$

Any encoding is fine as long as it is possible for another TM to take as input the encoded string and decode the description.

An example encoding scheme

$$0^n 1 0^m 1 0^k 1 0^s 1 0^t 1 0^r 1 0^u 1 0^v 1$$

M has n states

m tape symbols of which first k are input symbols

Similar encoding possible for transitions.

Important properties of the encoding scheme

- Able to encode all TMs upto isomorphism.
- Easy to interpret.

Universal Turing Machine

$$L(\mathcal{U}) = \{ \underline{M}\#\underline{x} \mid x \in L(M) \}.$$

encoding of M

encoding of input string x
x is over the input alphabet of M.

Symbol in $\mathcal{U}$'s input alphabet to delimit M and x.

How does $\mathcal{U}$ work?

1. Check if M and x are valid encodings. if not, then reject.

2. $\mathcal{U}$ does a step-by-step simulation of M on input x.

Description of M
Contents of M's tape
State of M and position of tape head

Working of the universal TM $\mathcal{U}$.

$\mathcal{U}$ takes as input an encoding of a TM M and a string x and simulates M on x.

- halts and accepts if M halts and accepts x
- halts and rejects if M halts and rejects x.
- loops if M loops on x.

$\mathcal{U}$ simulates M step by step.

Question. Can we do better than blind simulation?

Eg. If M halts on x then simulate M on x
if M does not halt on x then terminate the simulation and reject.

That is: Build $\mathcal{U}'$ that takes as input $M\#x$ and

- halts and accepts if M halts and accepts x
- halts and rejects if M halts and rejects x.
- halts and rejects if M loops on x.