Deterministic Turing Machines

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$, where

- Q: set of states
- Σ: input alphabet
- Γ: tape alphabet
- $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\}$
- $q_0 \in Q$: start state
- $q_a \in Q$: accept state
- $q_r \in Q$: reject state

q_a and q_r halting states

$q_a \neq q_r$

Deterministic Turing Machines

Input $w = w_1 w_2 \cdots w_n$
A configuration on input w: current state, current head location, tape contents

a string uqv over $(Q \cup \Gamma)^*$ where $u, v \in \Gamma^*$, $q \in Q$

tape contents: uv, current head location: the first symbol of v

initial configuration: $q_0 w_1 w_2 \cdots w_n$

accepting configuration: $uq_a v$ (Halting configuration)

rejecting configuration: $uq_r v$ (Halting configuration)

Deterministic Turing Machines

current configuration: $uaqbv$, $a, b \in \Gamma$

$\delta(q, b) = (p, c, L)$ Next configuration: $upacv$

$\delta(q, b) = (p, c, R)$ Next configuration: $uacpv$

current configuration: qbv, $b \in \Gamma$

$\delta(q, b) = (p, c, L)$ Next configuration: pcv

current configuration: uq, current symbol is blank

Deterministic Turing Machines

A DTM M accepts input w iff $\exists$ a sequence of configurations $C_0, \cdots, C_t$ such that:

1. C_0 is the initial configuration of M on input w
2. For all $1 \leq i \leq t$, the machine M moves from C_{i-1} to C_i
3. C_t is an accepting configuration

$L(M)$ is the set of all strings accepted by M.

A language L is recognizable iff there is a DTM M such that $L = L(M)$.

As defined, M need not halt on strings NOT in the language $L(M)$.

Decider: A DTM that halts on all inputs.

A decider is a recognizer.

Not all recognizers are deciders (later).

Algorithm = Decider