Recall: An ID of a TM $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej} \rangle$ is a string

$$\alpha q \beta \in (Q \cup \Gamma)^*$$

where $\alpha\beta \in \Gamma^*$
$q \in Q$

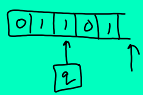
Initially: on input $w \in \Sigma^*$, w is on the left portion of the tape (cells $0, \ldots, n-1$, where $n := |w|$), all other cells blank (containing $\sqcup$).

$\alpha\beta$ represents contents of cells $0, \ldots, |\alpha\beta| - 1$, where all other tape cells are $\sqcup$.

α = tape contents to left of the scanned cell, β is the rest

q is current state of M.

Ex:



$= 01q101$

Ex: $01101q = 01101q\sqcup$
$= 01101q\sqcup\sqcup$

Can freely pad any ID by appending $\sqcup$ to it (considered the same ID)
Can also remove a $\sqcup$ as the last symbol of an ID.

How the comp proceeds

Initial ID of M on input $w \in \Sigma^*$ is

$q_0 w$

Define the successor of an ID $\alpha q \beta$:

Def: Let $ID = \alpha q \beta$ be an ID of M. We define the successor ID' of ID as follows:
Let $\beta := a\gamma$ for $a \in \Gamma$.

If $\delta(q, a) = (r, b, R)$, then $ID' = \alpha b r \gamma$

If $\delta(q, a) = (r, b, L)$

Case 1: If $\alpha = \mu c$ $(c \in \Gamma)$, then

$ID' = \mu r c b \gamma$

Case 2: $\alpha = \varepsilon$ $(ID = q\beta)$
then $ID' = r b \gamma$

If $\delta(q, a)$ is undefined (because q is a halting state — q_{acc} or q_{rej})

then ID has no successor
We write $ID \vdash ID'$

$ID_0 \vdash ID_1 \vdash ID_2 \vdash \cdots$

where

ID_0 is the initial ID, and

$\forall i \geq 0$, ID_{i+1} is the successor of ID_i (if it exists).

If ID has no successor, call it a halting ID.

3 types of computations:

accepting: $ID_n = \alpha q_{acc} \beta$ (some α, β, $n \geq 0$)

"M accepts (input) w"

rejecting: $ID_n = \alpha q_{rej} \beta$ (some α, β, n)

"M rejects w"

The computation is infinite (no halting ID).

"M loops on w"

(otherwise "M halts on w")

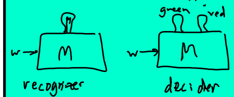
Def: M a TM as above, The language recognized by M is

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

Def: M is a decider if M halts on all inputs. $L(M)$ is decided by M in this case.

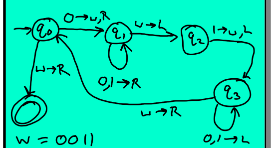
Def. L is Turing-recognizable (T-rec) if $L = L(M)$ for some TM M.

L is decidable if $L = L(M)$ for some decider M



Last week: defined a TM that decides the language

$\{0^n 1^n : n \geq 0\}$



$w = 0011$

computation on w:

$q_0 0011 \vdash \sqcup q_1 011 \vdash \sqcup 0 q_1 11 \vdash \sqcup 01 q_1 1$
$\vdash \sqcup 011 q_1 \vdash \sqcup 01 q_2 1 \vdash \sqcup 0 q_3 1$
$\vdash \sqcup q_3 01 \vdash q_3 \sqcup 01 \vdash \sqcup q_0 01$
$\vdash \sqcup \sqcup q_1 1 \vdash \sqcup \sqcup 1 q_1 \vdash \sqcup \sqcup q_2 1$
$\vdash \sqcup q_3 \underbrace{\sqcup \sqcup}_{\text{optional}} \vdash \sqcup \sqcup q_0 \sqcup$
$\vdash \sqcup \sqcup \sqcup q_{acc}$ halts & accepts

$w = 0$

$q_0 0 \vdash \sqcup q_1 \vdash q_2 \sqcup$
$[\vdash \cdots q_{rej}]$ optional

M [halts &] rejects

Church-Turing thesis:

Our intuitive notion of computation is captured by TMs:

- TMs are clearly simulatable with no specialized knowledge or intuition
- TMs powerful enough to compute anything we think can be computed.