CSE 396 Lecture Tue Apr 10 Spring 2018

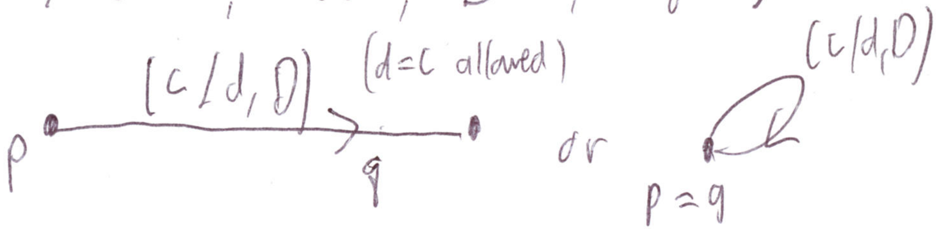
Top Hat 9034

<u>Defn</u>: A (one-tape) <u>Turing Machine</u> is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, \sqcup, s, F)$ where:

Q is a set of states like with a FA.
Σ is the <u>input alphabet</u> (or I/O alphabet if we allow functional outputs)
Γ, a superset of Σ, is the <u>work alphabet</u>.
$\sqcup$, also written 'B' or just a space, is the <u>blank</u> char, which is in $\Gamma \setminus \Sigma$.
$s \in Q$ is the start state (aka q_0). $F \subseteq Q$ is the set of final states [$= \{q_{acc}\}$ in text.
δ is a subset of $(Q \times \Gamma) \times (\Gamma \times \{L, R, S\} \times Q)$.

Typical (aka tuple) instruction: (p, c, d, D, q)

In state p, M can read c, change it to d, move in direction D (with S = stay) and go to state q.



M is <u>deterministic</u> if no two instructions have the same $(p, c, —)$ beginning.
Then we can say δ is a (partial!) function from $Q \times \Gamma$ to $\Gamma \times \{L, R, S\} \times Q$.
In the text's normal form, a <u>DTM</u> has $\delta : (Q \setminus \{q_{acc}, q_{rej}\} \times \Gamma) \to \Gamma \times \{L, R, S\} \times Q$

[Turing Kit Demo]. **Was Long**

<u>Defn</u>: An (aka <u>configuration</u> or ID) <u>instantaneous description</u> of a Turing machine is a string I that specifies

tape = ...$\sqcup\sqcup\sqcup$0101$\sqcup$1100$\sqcup$0$\sqcup\sqcup$...

- the current state q
- the current maximum span of the nonblank part of the tape w
- the current position h of the tape head, with $h=0$ where it started

$I = \langle q, w, h \rangle$

For a one-way-tape TM, cell 0 is the left end.
<u>Turing Kit</u> shows a "two-way infinite" tape.

Def^n of Computation: Given IDs I, J of a TM M

write $I \vdash_M J$ if there is an instruction that when executed in I gives the ID J.

"In configuration I, M can go to J in one step" (wording works for NTMs too)

If $I = \langle q, w, h \rangle$ and w has char c in position h then instruction must begin $[q, c / \ldots]$. Executing the d, D, r part then gives the ID J.

Then we write:

$I \vdash_M^0 I$ in 0 steps

$I \vdash_M^k J$ if there is an ID I' s.t. $I \vdash_M I'$ and $I' \vdash_M^{k-1} J$.

$I \vdash_M^* J$ if for some $k \geq 0$, $I \vdash_M^k J$.

This notation applies to multiple-tape Turing machines as well

An ID is **accepting** if its state q belongs to F **and** there is **no** instruction defined for q and c where c is in position h of w.

Different from DFA/NFA: a "crash" is the way it halts.

$L(M) = \{x \in \Sigma^*: M$ run from $I_0(x)$ enters an accepting ID$\}$

$I_0(x)$: $q = s$, $w = x$, $h = 0$. Can force $J = \langle q_{acc}, !, 0 \rangle$.