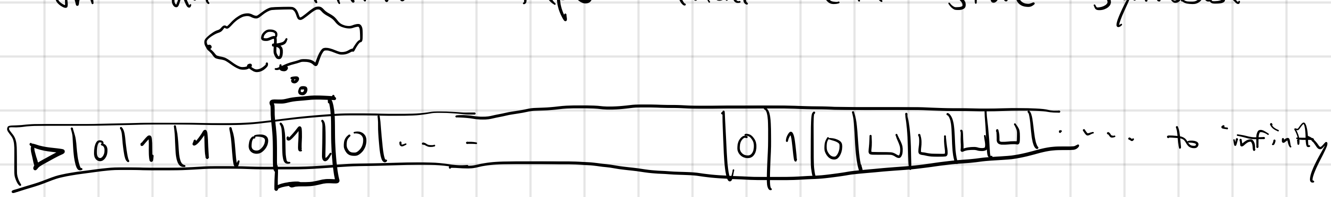
9 April 2018: Turing Machines

A formal definition of "algorithm" that is computationally equivalent to every other definition that has been proposed.

A Turing machine consists of a read/write head with a finite amt of internal state, moving on an infinite tape that can store symbols.



A Turing machine is specified by...

1. An alphabet: a finite set Σ containing $\triangleright$ and $\sqcup$.
2. A state set: a finite set K containing "s" (start state).
3. A transition function (i.e. "program")

$$\delta : K \times \Sigma \longrightarrow \underbrace{(K \cup \{halt, yes, no\})}_{\text{next state}} \times \underbrace{\Sigma}_{\text{symbol to write}} \times \underbrace{\{\leftarrow, \rightarrow, -\}}_{\text{direction to move.}}$$

Three special states {halt, yes, no} represent 3 ways a program could terminate.

{yes, no} for answering decision problems

{halt} for answering problems where the answer is written on the tape, e.g. adding two binary numbers.

How a Turing machine works, in plain English: run an infinite loop, starting at left edge of tape in state s. Every loop iteration uses the function δ to decide on next state, what symbol to write, what direction to move, until state is {halt, yes, no}. Then terminate.

How a Turing machine works, in math.

Let Σ^* denote the set of all finite length strings over Σ.
For $x \in \Sigma^*$ let $|x|$ denote length of x, i.e. # of symbols.

A <u>configuration</u> of a TM is a triple (x, q, k)
where $x \in \Sigma^*$ (tape contents)
$q \in K$ (current state)
$k \in \mathbb{N}$ $\quad 0 \leq k < |x|$ (position on tape)

A <u>transition</u> of a TM is a pair of configs that could occur consecutively during its operation.

$(x, q, k) \xrightarrow{M} (x', q', k')$ is a transition of M if

- $x'_i = x_i \quad \forall\, i \neq k$
  $x'_k = \sigma$ where $\underbrace{\delta(q, x_k) = (q', \sigma, d)}$
- q' is the state def'd by this equation.
- $k' = \begin{cases} k-1 & \text{if } d = \leftarrow \\ k+1 & \text{if } d = \rightarrow \\ k & \text{if } d = - \end{cases}$

Exceptions for $k = 0$, $x_k = \triangleright$, we constrain a TM program by saying that $\delta(q, \triangleright) = (q', \triangleright, d)$ where $q' \in K \cup \{\text{halt}, \text{yes}, \text{no}\}$, $d = \{-, \rightarrow\}$.

Exception for $k = |x| - 1$, the transition $(x, q, k) \xrightarrow{M} (x', q', k')$ appends an extra blank symbol to x', i.e.
$|x'| = |x| + 1 = k + 2$, $\quad x'_{k+1} = \sqcup$.

A <u>computation</u> of a TM is a sequence of transitions starting with $(x, s, \emptyset)$ s.t. $x_0 = \triangleright$
and either ends with (y, q, k) s.t. $\delta(q, y_k) \in \{\text{halt}, \text{yes}, \text{no}\}$
or consisting of an infinite sequence of transitions.

These are <u>my</u> conventions for TMs. Other people use different conventions that differ from these in insignificant ways.