

Church-Turing thesis: TMs capture our intuitive notion of computability.

TMs can:

- move data (on the tape)
- copy data
- check =, <
- increment / decrement

Moving data —

Ex: $\Sigma = \{0, 1\}$

$w = 011$
 $q_0 11 \rightarrow q_1 11, q_1 1 \rightarrow q_2 11, q_2 1 \rightarrow q_3 11, q_3 1 \rightarrow q_4 11, q_4 1 \rightarrow q_5 11, q_5 1 \rightarrow q_6 11, q_6 1 \rightarrow q_7 11, q_7 1 \rightarrow q_8 11, q_8 1 \rightarrow q_9 11$

Copying data —

Alternately: $\Gamma = \{0, 1, \#, \square, \circ, \cdot, \wedge, \vee\}$

So far — formally described TMs, i.e., explicit states & transitions.

Implementation-level description of a TM:
 Describing tape head movements, tape outputs more generally.

High-level description: mostly machine-independent description of an algo.

Equality test: (string identity)

Goal — goto an \ominus state if $w \neq x$
 " " " " \oplus state if $w = x$

"On input $w \& x$, where $w, x \in \{0, 1\}^*$:

1. Check that $|w| = |x|$ by repeatedly marking corresponding bits of w & x : if find w before x or finish x before w then goto \ominus (a move on)
2. Treat w & x
3. Repeat until all of w is examined
 - a) Mark the next bit of w (left-to-right)
 - b) compare it with the corresponding bit of x
 - c) if \neq , goto \ominus
 - d) otherwise change mark on x to its right neighbor
 - e) go back to (a)
4. Go to \oplus

Checking $<$: similar:

- pad one or the other string with leading 0's so they are the same
- scan both left to right until discrepancy found
- type of discrepancy gives the answer (if no discrepancy found, then w & x are numerically equal)

Increment/Decrement

$w \in \{0, 1\}^*$ represents a natural number (unassigned tape)

"On input w :

1. scanning w from right to left:
 - a) if see 1, change to 0 and continue scanning
 - b) if see 0, change to 1 and stop
 - c) if see $\#$, insert 1 to the right of the $\#$ & bump all the 0's one to the right.

Decrement — similar (decrementing 0 results in 0)

Adding/subtracting

While $x \neq 0$:
 decrement x
 increment w
 (invariant)