- Eg. $100(n+n^2) = O(n^2),\ \Theta(n^2)$.

  " $= o(n^{2.001}),\ \omega(n^2/\log n)$.

  $n\log n = O(n^{1.01}),\ o(n^{1+\varepsilon})$.

▷ $n^c = o(2^n)$, for all constant c.

- Some remarks:

  (1) The input/output-tape is only for writing the output string (preceded by the input string).

  For all the other stuff use the work-tape.

  (2) Though you can search a name in a directory in #steps = "$\log_2 N$", it does not mean that the problem has time complexity $O(\log n)$!

  Because, here you are assuming N to be <u>fixed</u> & the directory already <u>indexed</u>.

- Church-Turing thesis:
  Every realizable computing device — silicon, DNA, neuron, quantum, alien technology — can be simulated on a TM.

- Problems, solvable on a TM, are called computable, decidable, or recursive.

- Defn.: We say that a fn. f is polynomial time computable, if there is a TM M computing f in time $T(n)$, where $T(n) = O(n^c)$ for a constant $c \in \mathbb{R}_{\geq 0}$.

- Roughly, we say that f has an efficient algorithm!

- Exercise: Any C-program that runs

efficiently can be converted to an efficient TM, and vice versa.

- Thus, if a problem is hard for TMs then it is also intractable in "real"-life!

- This might change if somebody can build a "real" quantum computer!

<u>Feeding TMs to themselves</u>

- Since, a TM has a <u>finite</u> description (T, Q, δ), we can encode it as a boolean string.

  Let us fix some encoding.

- For every string $\alpha \in \{0,1\}^*$, we write

$$M_\alpha := \begin{cases} \text{the TM that } \alpha \text{ encodes, if } \alpha \text{ does encode.} \\ \text{otherwise the TM computing the zero fn.} \end{cases}$$

▷ There is a <u>universal</u> TM U that on input (α, x) simulates M_α on x.

- <u>Pf. idea</u>: U reads the transition fn. δ from α & decides the next step.

<u>The Halting problem</u>

- $\text{HALT} := \{(\alpha, x) \mid M_\alpha$ halts on x & outputs a bit $\}$.

- Its (un)computability was proven by Church (1936) & much simplified by Turing.

<u>Theorem</u>: HALT is not TM computable.

<u>Proof</u>: • Suppose it is decidable by M_0.

Say, it outputs 1 on (α, x) if $M_\alpha(x)$ halts, else 0.

- We define a TM M' as:

On input α:

(1) If $M_0(\alpha, \alpha) = 0$ then <u>output</u> 1.

(2) Simulate $M_\alpha(\alpha)$ & <u>output</u> the negation of $M_\alpha(\alpha)$.

- Let β be an encoding of M'. So, $M' = M_\beta$.
- What is $M_\beta(\beta)$?

- If $M_\beta(\beta) = 1$ then,

  either $M_0(\beta, \beta) = 0$, so $M_\beta(\beta)$ should not halt!

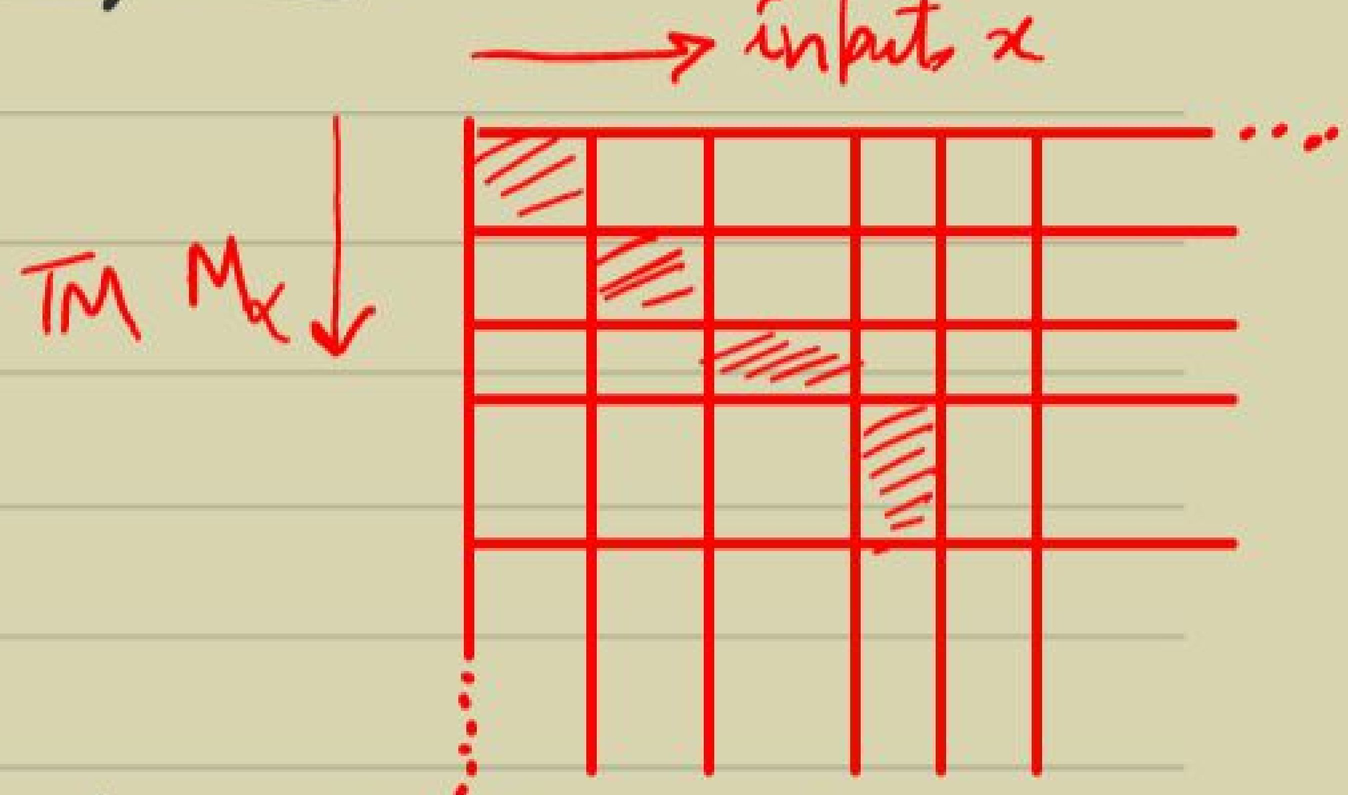
  or $M_\beta(\beta)$ halts & outputs 0!

- If $M_\beta(\beta) = 0$ then,

  $M_\beta(\beta)$ halts & outputs 1!

- In all the three cases we have a contradiction. So M_0 does not exist. $\square$

- This proof technique is called <u>diagonalization</u>.



- A TM is designed that disagrees with $M_\alpha(\alpha)$, $\forall \alpha$.

- This technique was used in other foundational results as well:

  - <u>Cantor's pf.</u> of the uncountability of the reals (1891).
  - <u>Russell's</u> paradox (1901).
  - <u>Gödel's</u> incompleteness theorem (1931).