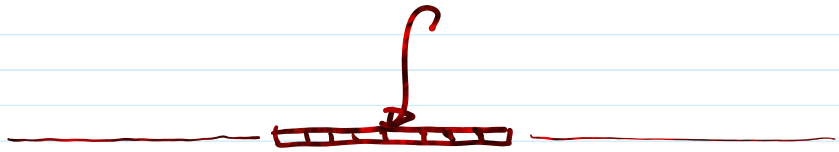
Church-Turing thesis and universal computation

Friday, April 25, 2025 1:38 PM



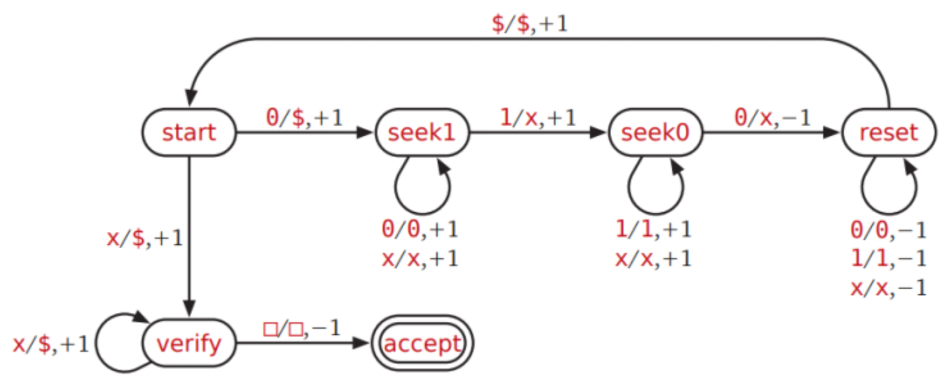
Church-Turing Thesis. [1936]

Any problems solvable by machines can be solved by TMs. Surprisingly powerful!

Q. What are the next big-picture questions?

1. Benefit of simple description? emulation.
2. Problems not solvable by TM? computability
3. Intermediate levels of machine power? complexity

$\{0^n 1^n 0^n : n \in \mathbb{N}\}$



Encoding TM. $\{0, 1, \$, x, □\}$!!

Let $M := (\Gamma, Q, \text{start}, \text{accept}, \text{reject}, \delta)$ be an arbitrary TM.

Construct encoding over $\{0, 1, [,], \bullet, |, 0, !\}$

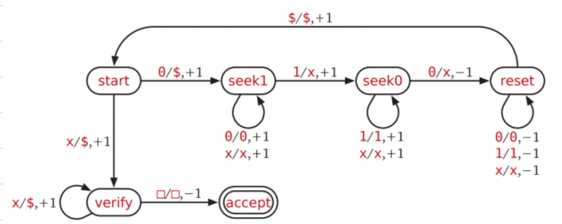
- $\langle 0 \rangle = 001$
  $\langle 1 \rangle = 010$
  $\langle \$ \rangle = 011$
  $\langle x \rangle = 100$
  $\langle □ \rangle = 000$

$\langle 001100 \rangle = [001 \bullet 001 \bullet 010 \bullet 010 \bullet 001 \bullet 001]$

- $\langle start \rangle = 001$
  $\langle accept \rangle = 110$
  $\langle reject \rangle = 000$
  $\langle seek1 \rangle = 010$
  $\langle seek0 \rangle = 011$
  $\langle reset \rangle = 100$
  $\langle verify \rangle = 101$

- $\langle \delta \rangle :=$ concatenation of $[\langle p \rangle \bullet \langle a \rangle \mid \langle q \rangle \bullet \langle b \rangle \bullet \langle \Delta \rangle]$
  for each transition $p \xrightarrow{a/b,\Delta} q$

- $\langle M \rangle := [\langle reject \rangle \bullet \langle \square \rangle][\langle \delta \rangle]$



$\Rightarrow$

```
[000•000][[001•001|010•011•1][001•100|101•011•1]
          [010•001|010•001•1][010•100|010•100•1]
          [010•010|011•100•1][011•010|011•010•1]
          [011•100|011•100•1][011•001|100•100•1]
          [100•001|100•001•0][100•010|100•010•0]
          [100•100|100•100•0][100•011|001•011•1]
          [101•100|101•011•1][101•000|110•000•0]]
```

$\mathcal{U}$:

<u>Universal TM</u>: Take $\langle M \rangle$ & $\langle w \rangle$ and run $M(w)$.

- encode <u>configuration</u> (curr state, tape info) on work tape.

  $(start,\ \hat{0}01100) : [\hat{0}01][\hat{0}01 \bullet 001 \bullet 010 \bullet 010 \bullet 001 \bullet 001]$

  $(reset,\ \$0x\hat{1}x0) : [100][001 \bullet 001 \bullet 010 \bullet 010 \bullet 001 \bullet 001]$

- UTM has an 1. <u>input tape</u> to store $\langle M \rangle . \langle w \rangle$
  2. <u>data tape</u>, blank at first.
  3. <u>state tape</u>, "

- initialization: write $\overset{001}{\langle start \rangle}$ on <u>state tape</u>.
  write $\langle w \rangle$ on <u>data tape</u>, replace 0 w/ $\hat{0}$
  $\langle 001100 \rangle = [\hat{0}01 \bullet 001 \bullet 010 \bullet 010 \bullet 001 \bullet 001]$

- each step of $M(w)$:
  1. find $\hat{0}$ or $\hat{1}$ in <u>data tape</u>

1. find ◬ or ! in data tape
2. scan through input tape to find $[\langle p\rangle \cdot \langle a\rangle \mid \langle q\rangle \cdot \langle b\rangle \cdot \langle \Delta\rangle]$

   s.t. $\langle p\rangle$ is on state tape
   $\langle a\rangle$ is the word containing ◬ or !
3. replace $\langle p\rangle$ w/ $\langle q\rangle$ on state tape.

   $\langle a\rangle$ w/ $\langle b\rangle$ on data tape.
4. move to adjacent word on data tape, add ▲

- Accept/reject when ever state tape says so. ☒

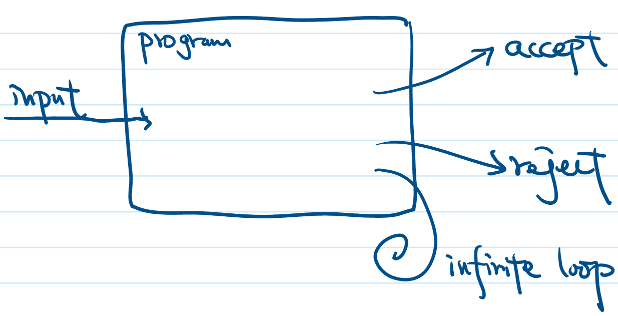
Corollary. Given $\langle M\rangle$, write M in TM design is safe!

$\underline{UTM(\langle M\rangle, \langle w\rangle):}$

> compile $\langle M\rangle$ into M.
> return $M(w)$

using UTM ☠

You have no idea what we have unleashed. BWAHAHAHA.

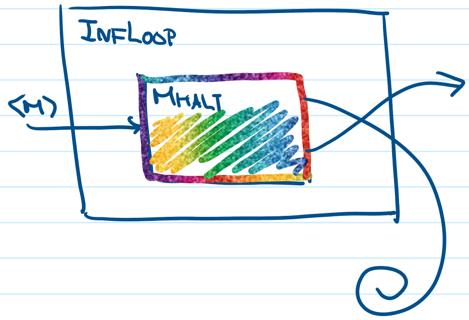


$$\boxed{\text{HALT}(\langle M\rangle): \text{Does } M(\langle M\rangle) \text{ halt?}}$$

Q. Is HALT computable?

Q. Is HALT computable?

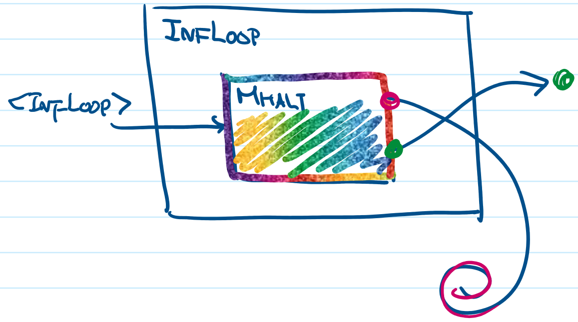
Say TM $\boxed{M_{HALT}}$ solves HALT.



InfLoop run $M_{HALT}(\langle M \rangle)$.

need UTM! $\langle M_{HALT} \rangle$.

Q. What if we feed ⟨InfLoop⟩ into InfLoop? InfLoop(⟨InfLoop⟩).

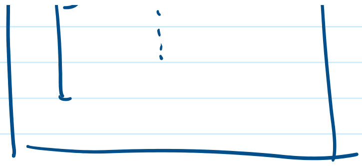


- Say InfLoop(⟨InfLoop⟩) halts,
  ⇒ InfLoop(⟨..⟩) does not halt.
- Say InfLoop(⟨..⟩) loops forever.
  ⇒ InfLoop(⟨ ⟩) does halt.

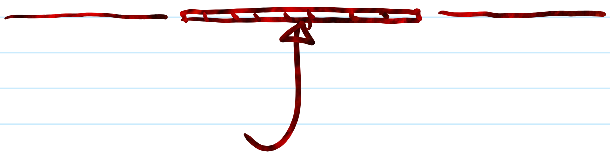
Conclusion. $\boxed{M_{HALT}}$ does not exist! HALT uncomputable.

```
⋮
If M(w) halts :
    ⋮
```

not all pseudocodes are safe!
at least not for arbitrary program M.



at least not for arbitrary program M.

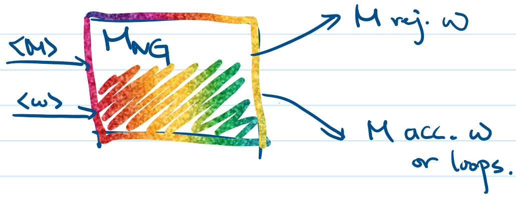


Other examples:

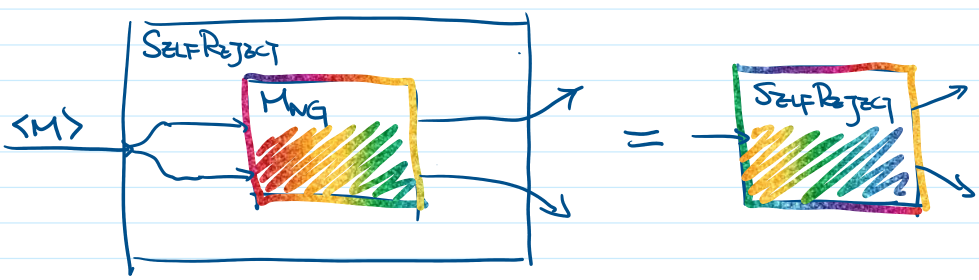
NeverGonnaGiveYouUp $:= \{\langle M\rangle, \langle w\rangle : M \text{ rejects input } w\}$

Claim. NG is uncomputable.

Pf. Assume $\boxed{M_{NG}}$ exists.



SelfReject(⟨M⟩): Does M rejects ⟨M⟩?



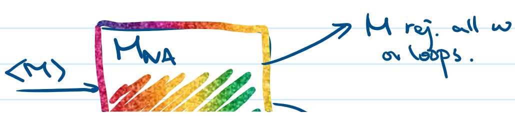
- If SelfReject accepts ⟨SelfReject⟩

$\Leftrightarrow$ SelfReject rejects ⟨SelfReject⟩

Example. NeverAccept $:= \{\langle M\rangle : M \text{ never accepts any } w\}$

Claim. NeverAccept is uncomputable.

Pf. Assume $\boxed{M_{NA}}$ exists.



Pf. Assume $\boxed{M_{NA}}$ exists.