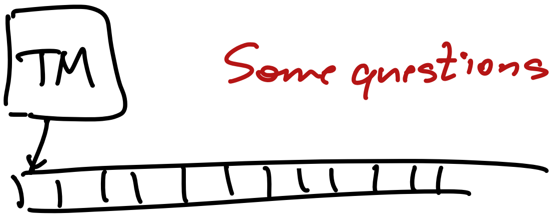
Undecidable — NO algorithm

Problem X is undecidable iff

there is no algorithm to solve arbitrary instances of X in finite time

Typically undecidable problems ask questions about code (Turing machines, Python, C, ...)



Some questions about code are decidable

Does TM M given input w ever move?

Canonical: Halting problem:

Given a program ⟨M⟩ on input string x

Does M halt when given input x?

Self Halt: Given program ⟨M⟩

Does M halt when given ⟨M⟩ as input?

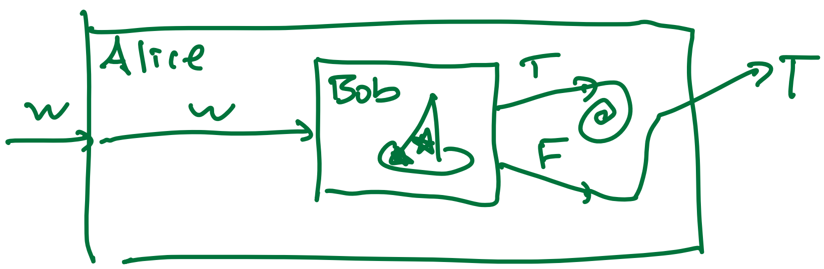
Spose Bob <u>decides</u> SELFHALT.

$\Rightarrow$ ACCEPT(Bob) = SELFHALT

Hang(Bob) = $\emptyset$

<u>Alice(w):</u>

```
IF Bob(w) accepts
    ∞ loop
else
    return True
```



ACCEPT(Alice) = REJECT(Bob)

Alice accepts⟨Alice⟩ $\Rightarrow$ Bob accepts $\boxed{\langle\text{Alice}\rangle}$ (w) by def. SELFHALT

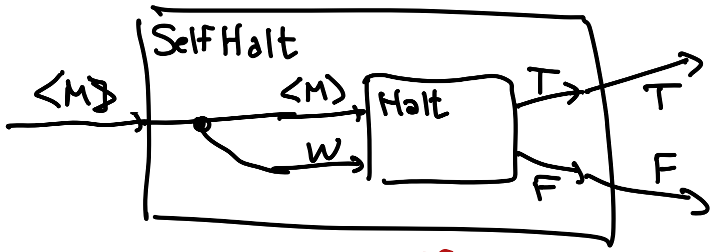
$\Rightarrow$ Alice hangs on ⟨Alice⟩ by def. Alice

$\Rightarrow$ Bob rejects⟨Alice⟩ by def SELFHALT

$\Rightarrow$ Alice accepts⟨Alice⟩ $\boxed{\Rightarrow \text{Bob doesn't exist!}}$

Thm: HALT is undecidable

Proof: reduction from SelfHalt.




```
SelfHaltDecider(<M>):
    verify that <M> encoding
    return HALT(<M>, <M>)
```

To prove problem X is undecidable
Describe a reduction from any undecidable problem to problem X

NEVERHALT: Given <M>, does M always ∞-loop?

Suppose Bunny decides NEVERHALT

Build an algorithm for HALT:

Reduction from HALT

```
HaltDecider(<M>, w):
    Write following code:
        MEOW(x):
            returns M(w)
    if Bunny(<Meow>)
        return False
    else
        return True
```

Bunny
Halt
M
Meow

Suppose M halts on w:
⟹ Meow halts on every input
⟹ Bunny rejects <Meow>
⟹ HaltD accepts <M>, w ✓

Suppose M loops on w:
⟹ Meow loops on every input
⟹ Bunny accepts <Meow>
⟹ HaltD rejects <M>, w ✓

Contradiction! Bunny doesn't exist!

Rice's Theorem:

Given $\langle M \rangle$, does M accept ________ ?

$$\text{Accept}(M) = \{ w \mid M \text{ accepts } w \}$$

Let $\mathcal{L}$ be any family of languages such that:

- There is a program Y s.t. $\text{Accept}(Y) \in \mathcal{L}$
- There is a program N s.t. $\text{Accept}(N) \notin \mathcal{L}$

Deciding, given $\langle M \rangle$, if $\text{Accept}(M) \in \mathcal{L}$ is impossible.

Does M accept ε?

- $\mathcal{L}$ = languages contain ε
- Y = accept all strings
- N = accept nothing

Rice's Theorem ✓

Does M accept all palindromes with length 2^{prime}?

- $\mathcal{L}$ is lang containing all palindromes length 2^{prime}
- Y = accept Σ^*
- N = accept $\emptyset$

Does M accept a non-regular language?

- Y = accepts $\{0^n 1^n \mid n \geq 0\}$ and nothing else.
- N = return True