Undecidability of the
Halting problem

Diagonalization

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

Strings of infinite length:

0101010.....

Let $S_1, S_2, S_3$ ..... be infinite strings.

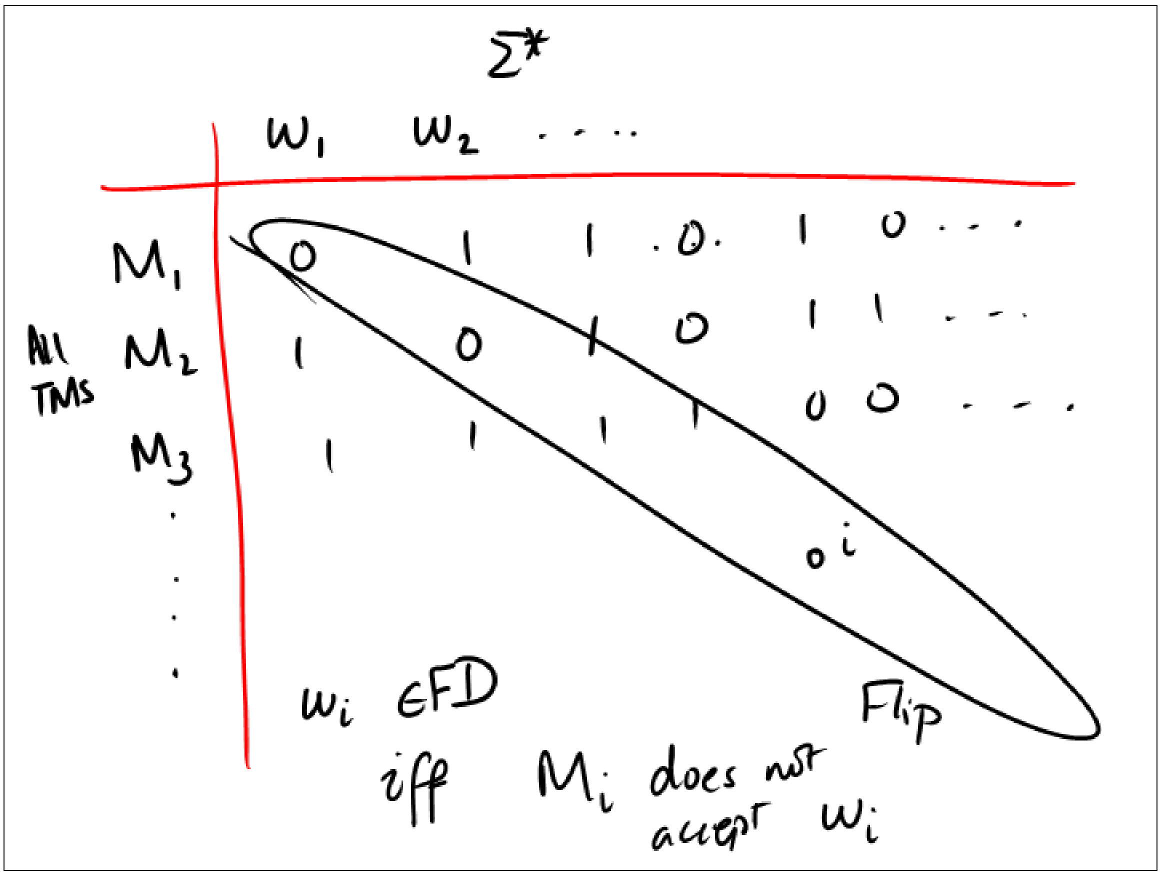
| | 1 | 2 | 3 | 4 | ... | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $S_1$ | 0 | 1 | 0 | 1 | 0 | 1 | 1 | ..... | |
| $S_2$ | 1 | 0 | 1 | 1 | 0 | 1 | 0 | ... | |
| $S_3$ | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | ... |
| $S_4$ | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | |
| ⋮ | | | | | | | | | |

Flipped diagonal string

1101 .....

Flipped diagonal is different from every $S_i$

( it differs in the i'th position wrt $S_i$)


Σ*
w1 w2 ....
M1
All TMs M2
M3
wi ∈ FD
iff Mi does not accept wi
Flip

$A_{TM} = \{\langle M, w\rangle \mid M \text{ acc } w\}$

<u>Claim</u> There is no TM that decides $A_{TM}$.

Assume H is a TM that decides $A_{TM}$.

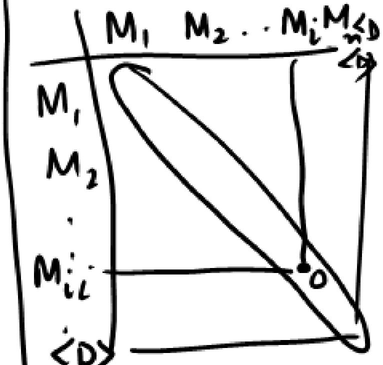
Build a TM D:

$D(\langle M\rangle)$:

Feed $\langle M, \langle M\rangle\rangle$ to H

If H answers YES, say NO

If H answers NO, say YES.



But D cannot exist!

Why?

What will happen when you feed <D> to D?

D must answer yes iff ______

<D, <D>> fed to H says NO

D does not halt on <D>

When D is fed <D>,

it will call H on <D,<D>>

If H says yes, then I know

D must accept <D>.

But D won't.

If H says no, then

D does not accept <D>.

But D will!

## Napkin-sketch of undecidability of acceptance of words of a C-pgm

Assume $H$ is a C-pgm that decides, given $\langle P, w \rangle$ whether $P$ is a C-pgm that accepts $w$.

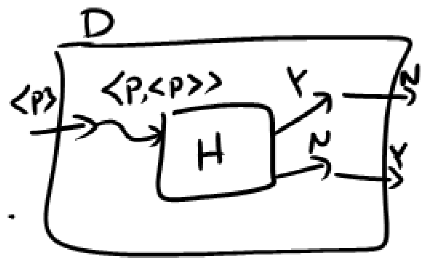
Create a new C-program $D$:

$D$ takes as input a program $\langle P \rangle$.

It feeds $\langle P, \langle P \rangle \rangle$ to $H$.

If $H$ accepts, $D$ rejects.

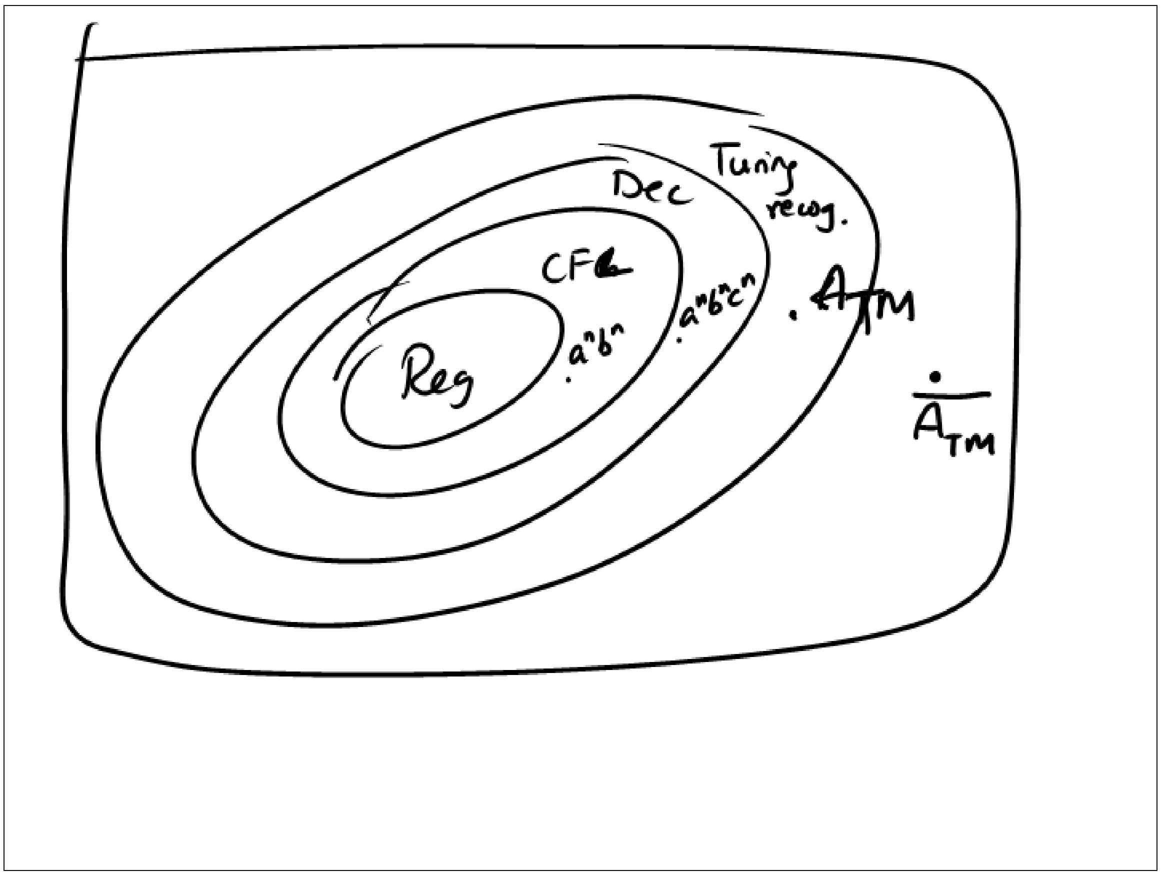
If $H$ rejects, $D$ accepts.



Now, what will D do on input <D>?

(D must say YES on <D>) iff (H says No on <D,<D>>)

iff (D says NO on <D>)


Turing recog.
Dec
CFL
Reg
$a^nb^n$
$a^nb^nc^n$
$A_{TM}$
$\overline{A_{TM}}$

**Theorem.** $L$ and $\overline{L}$ are Turing recognizable iff $L$ is Turing decidable.

($\Leftarrow$) Trivial.

($\Rightarrow$) $A$ is a Turing recognizer for $L$

$B$ is a Turing recognizer for $\overline{L}$

Input $w$: run $w$ simultaneously on $A$ and $B$. One of them will halt. And you can report the answer.

<u>Corollary</u> $\overline{A_{TM}}$ is not Turing-recognizable.

<u>Proof</u>. If $\overline{A_{TM}}$ was Turing recognizable, then since $A_{TM}$ is Turing recog., $A_{TM}$ would be Turing decidable. which is a contradiction. So $\overline{A_{TM}}$ is <u>not</u> Turing recog.