Universal TM same idea:

$\langle M, w \rangle$

M is TM $= (Q, \Sigma, \Gamma, \delta, q, q_{acc}, q_{rej}, \sqcup)$

insist:

$Q = \{1, \ldots, a\}$

$\Sigma = \{1, \ldots, b\}$

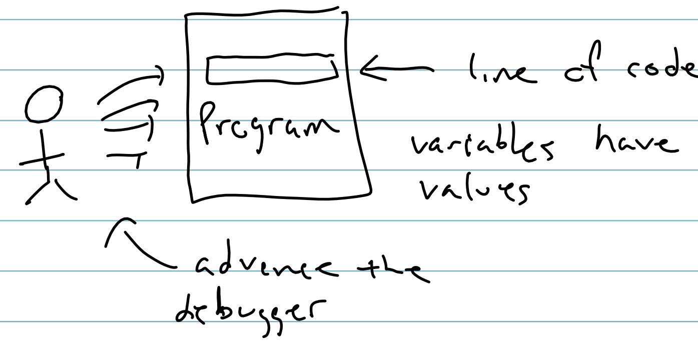
$\Gamma = \{1, \ldots, c\}$

Conventions: $q_0 = 1$, $q_{acc} = 2$, ...

Universal TM sees

(a base 10 # b base # ...
describe δ ## describe w

Debugger:



Program is written C or Java or ...

Debugger could be written in same language

=

Debugger is a sort of universal machine...

=

Debugger can run step by step

decide vs recognizing

decide: always eventually halt

recognizing: might not halt

=

$A_{TM} = \{\langle M, w \rangle$ s.t. M on input w reach $q_{acc}\}$

$= \{\langle M, w \rangle \mid M \text{ accepts } w\}$

=

Claim: A_{TM} is recognizable

Claim(!!!): " " not decidable

Claim(!!): $\overline{A_{TM}}$ is not recognizable

Universal TM

▽ input

| $\langle M \rangle$ ## $\langle w \rangle$ |

Convenient many tapes

1 tape for w
1 tape for δ description
1 tape for a,
— — b,
— — — c,

} finite # of tapes

| a base 10 | # | b .. | # | c .. | δ .. |

$\langle M \rangle$

etc.

If M on input does not halt,
U does not halt...

So
$\langle M, w \rangle \in A_{TM}$, then
U halts and {accepts / says so}

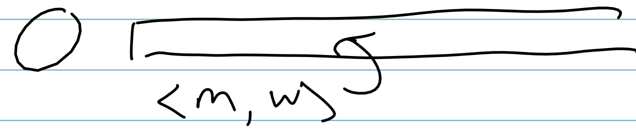
if $\langle M, w \rangle \notin A_{TM}$ then
either { U halts and rejects
{ U doesn't halt

=

For each T.m., M, we say that $\{ w \in \Sigma^*_M \mid$ M on input w is accepted $\}$

__Claim 1:__ A universal T.m. recognizes A_{TM}

U = Universal T.m.



U simulate M on input w for step 1, step 2, step 3, ...

If M on input w halts in q_{acc} of M, then U halt and accept $\langle M, w \rangle$

- - - - - - - - - q_{rej}
- - - - - - - - - reject

We will see!

A_{TM} is __recognized__ by a T.m. but can't be __decided__

=

$\{ \langle M, w, q \rangle \mid$ q is a state of M, w is input to M, M on input w at some point reaches q $\}$

= $L_{\text{state is reached}}$

is __recognizable__, but not __decidable__.

To each T.m. $M = (Q, \Sigma, \ldots$

$\text{Language}_M = \{ w \in \Sigma^* \mid M \text{ accepts } w \}$

is the language "recognized by M"

=

Old ideas: "an algorithm runs in time $O(n^4)$" means always halt, means __deciding__

New: Every T.m. __recognizes__ some language: what it accepts versus what it rejects or doesn't halt on.

Claim: A_{TM} cannot be decided.

Why not? Say that H is T.m. that decides A_{TM}.

Let D be a T.m., based on H: on input w, see if $w = \langle M \rangle$ if so, then run H on $\langle M, \langle M \rangle \rangle$

description of M, with $w = \langle M \rangle$ as input; H always halts, D output "opposite" or "negation"

If H accepts, D reject

- - - - rejects, D accept

$$\text{Halt}_{\text{Machine, Input}} = \{ \langle M, w \rangle \mid \text{on input } w, M \text{ halts}, \text{eventually reaches } q_{acc} \text{ or } q_{rej} \}$$

to recognize [Halt], run a Universal T.m., if simulation of M on input w halts, then the machine accepts, otherwise simulation never halts

So if D has input $\langle M \rangle$

D accept $\langle M \rangle$ if M on input $\langle M \rangle$ rejects

D reject - - - - - accepts

What does D do on input $\langle D \rangle$?

D on input $\langle D \rangle$ either {accepts, rejects}

-- contradiction either way.--