Halting Problem

P1:

$HALT_{TM}$ ={<M,w>| M is a Turing machine and M halts on input w}

P2:

A_{TM} ={<M,w>| M is a Turing machine and M accepts input w} Undecidable

1

P2

- Input <M,w>
- If M does not stop on w, reject it
- If M stops, simulate M.
  If M acceps, accpets
  Otherwise

2

Halting Problem

$HALT_{TM}$ ={<M,w>| M is a Turing machine and M halts on input w}

Theorem: $HALT_{TM}$ is an undecidable language.

Proof by contraction. Assume it is decidable.

3

Undecidable Problem

A_{TM} ={<M,w>| M is a Turing machine and M accepts w}

Theorem: A_{TM} is an undecidable language.

Proof by contraction. Assume it is decidable.

4

Proof

Assume the Turing machine R decides $HALT_{TM}$

We can call R to decide A_{TM}

For input <M, w>
  Run R on <M,w>
    if R rejects, "reject"
    if R accepts, simulate M until it stops
      if M accepts, "accept"
      else "reject"

5

Empty Problem

E_{TM} ={<M>| M is a Turing machine and $L(M) = \phi$ }

Theorem: E_{TM} is an undecidable language.

Proof by contraction. Assume it is decidable.

6

Proof E_{TM}

Assume the Turing machine R decides

For input <M,w>, design another TM M_1

If $x \neq w$, reject

simulate M on input w, accepts if M accepts w

7

Proof

Use R to decide A_{TM}

Input <M,w>

Make M_1 from <M,w>

Run R on the input M_1

If R rejects (it means L(M_1) is not empty), accepts

I R accepts (it means L(M_1) is empty), rejects

8

Problem

Let

L ={M| M is a Turing machine and L(M) ={1}}

Show that L is undecidable.

9