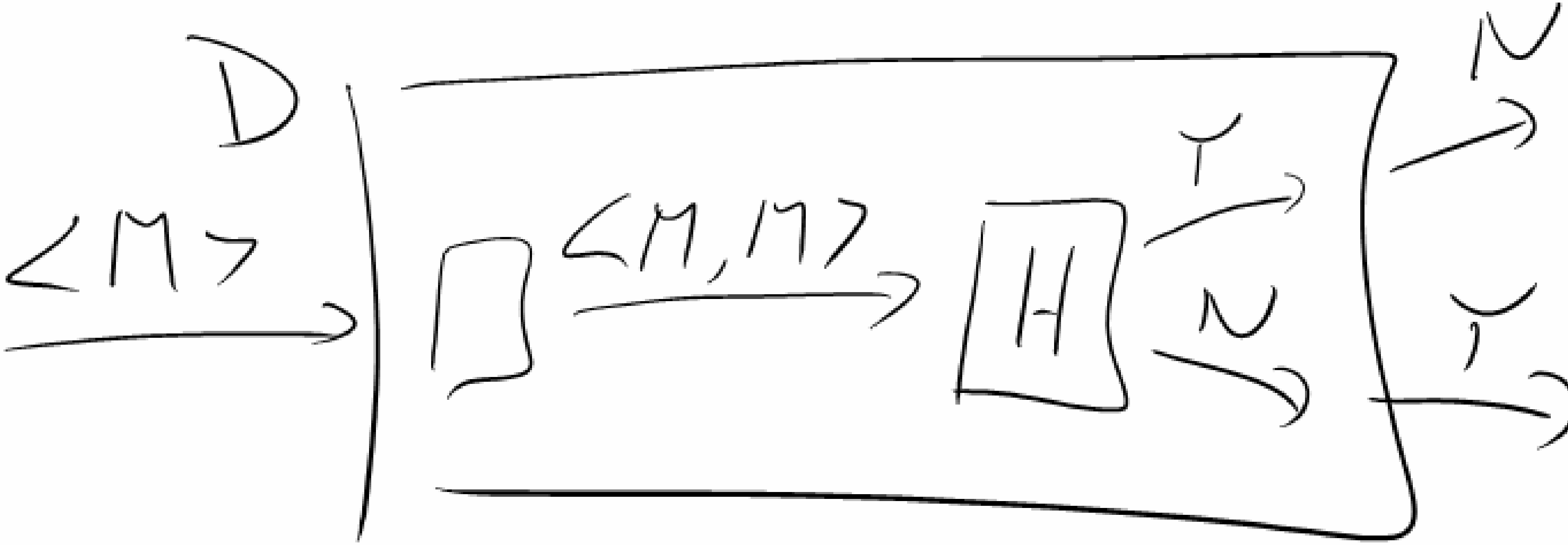
Note Title 11/9/2006

A_{TM} is undecidable



$D(\langle D \rangle)$

Reductions

P_1 reduces to P_2

$\Rightarrow$ a solution to P_2 can be converted to a solution to P_1

$\Rightarrow$ P_1 is no harder than P_2

L_1 reduces to L_2

means

a decider for L_2

can be converted to

a decider for L_1

So if L_2 is decidab

Then L_1 is decidable

So if L_1 not decidable

Then L_2 is not decidable

$Halt_{TM}$ is undecidable

$$Halt_{TM} = \{ \langle M, w \rangle : \begin{array}{l} M \text{ halts} \\ \text{on } w \end{array} \}$$

Proof: Suppose $Halt_{TM}$ is decidable. Let H be a TM deciding $Halt_{TM}$. Construct H' deciding A_{TM}

... contradiction

H' = inputs $\langle M, w \rangle$

run $H(\langle M, w \rangle)$

if H accepts

return $M(w)$

else

reject

input <M>

or <M, w>

rewrite code for M

or w

or both

feed results to H

$L_{uiuc} = \{ \langle M \rangle : L(M)$ contains the string "uiuc" $\}$

reduce A_{TM} to L_{uiuc}

Proof: Suppose H decides L_{uiuc}. Construct H' that decides A_{TM}. A contradiction. So L_{uiuc} isn't decidable

H' = input $\langle M, w \rangle$

  Construct new TM M_w

    M_w = input x

      ignores x

      runs M on w

return $H(\langle M_w \rangle)$

Code for M_w

= Code for M

consts w = " "

copy from input to H'

```
main (x) {
    return M(w)
}
```

$L(M_w) = \Sigma^*$ if M accepts w

or $\emptyset$ if M doesn't accept w

so $M_w \in L_{uinc}$ if M accepts w

$M_w \notin L_{uinc}$ if M doesn't accept w

$$E_{TM} = \{\langle M \rangle : L(M) = \emptyset\}$$

is undecidable

Proof: Assume H decides E_{TM}

Construct H' decides A_{TM}

$H' =$ • input $\langle M, w \rangle$

• construct new TM M'

$M'_w =$ input is x

if $(x = w)$ return $M(w)$

else reject

• $H(\langle M'_w \rangle)$ and return ~~result~~

opposite of result

M' code

const w = "......"

copied from input for H'

$L(M'_w)$ can contain

w

w might or might not be in $L(M'_w)$

other strings definitely not in $L(M'_w)$

$L(M'_w) =$

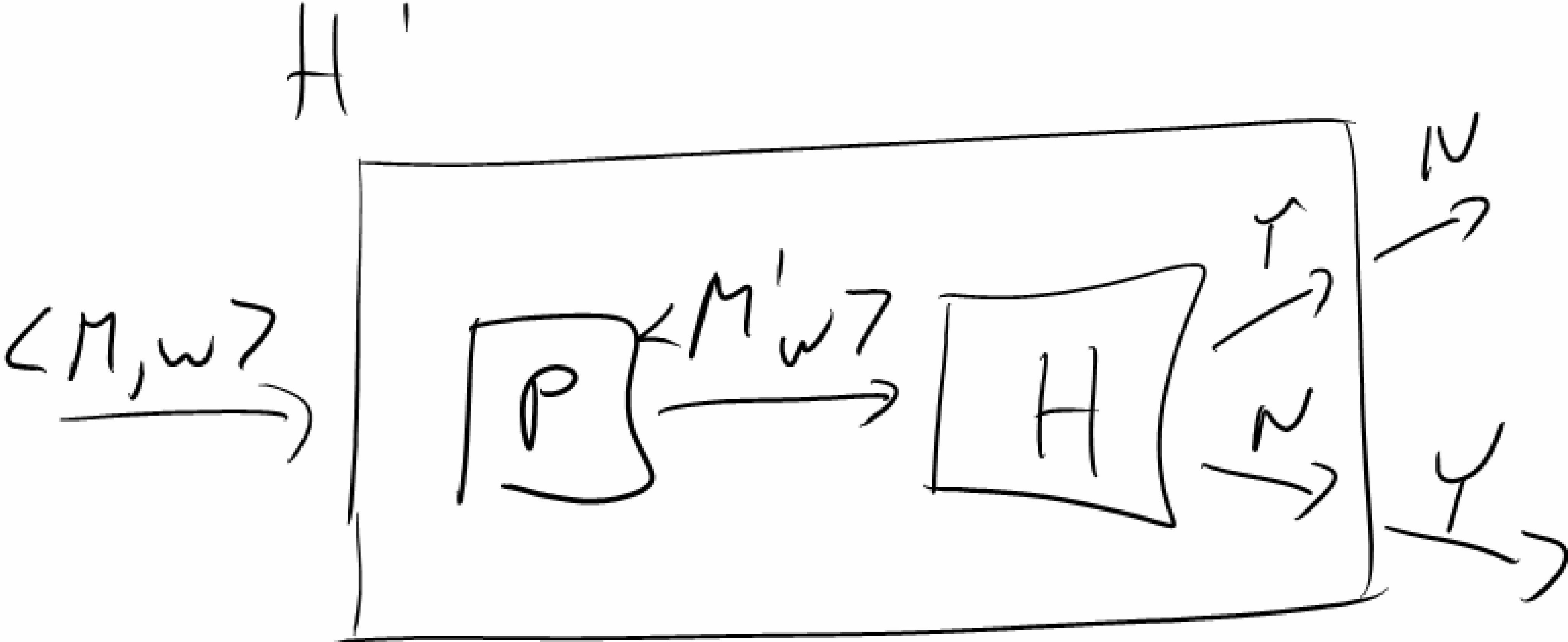
$\emptyset$ if $w \notin L(M)$

or $\{w\}$ if $w \in L(M)$

$H(M'_w)$ accepts

exactly when $w \notin L(M)$

i.e, when M doesn't ~~accepts~~ w


H'
<M,w>
P
<M',w>
H
Y
N
N
Y

Is E_{TM} recognizable? — no

Is $\overline{E_{TM}}$ recognizable?

yes

Dovetailing

Fixed M (input)

Hunt for a input sty w

such That M accepts w

Enumerate all strings w
in sensible order

For string w
run M on w and
See if M accepts w

Loop $i = 0$ to ∞

- generate i^{th} string w_i
- start simulate of M on w_i
- Advance by 1 step simulations of M on $w_0, \ldots, w_i$
- If any simulation accepts return accept