Universal Turing Machines $\mathcal{U}$

$L(U) = \{ M\#w \mid w \in L(M) \}$

M is the description of a TM given in binary

w : encoding in binary of an input to M

$M\#w$

description of M

$\#w$

state of M : 10   10

Recursive languages — Total TM

Recursive enum — TM

co-recursively enum

Halting Problem

$HP = \{ M\#w \mid M \text{ halts on } w \}$

HP is rec. enum.

$M_\varepsilon \quad M_0 \quad M_1 \quad M_{00} \quad M_{01} \quad M_{10} \quad M_{11} \quad \ldots$

	ε	0	1	00	01	10	11	...
M_ε	H	H	H	H	H	H		
M_0	L	L	H	H	L	H		
M_1	L	H	$\boxed{H}$	L	H	H	L	H ...
M_{00}								
M_{01}								
⋮								

H - halting
L - looping

Suppose
$\exists$ Total Turing machine H
for $HP = \{ M\#w \mid M \text{ halts on } w \}$

Construct N as follows.

On input $x \in \{0,1\}^*$
Simulate H on $M_x \# x$
if H accepts then
loop forever
else (H rejects)
halt & accept.

description of N

$L(N) = \{ x \mid M_x \text{ does not halt on } x \}$

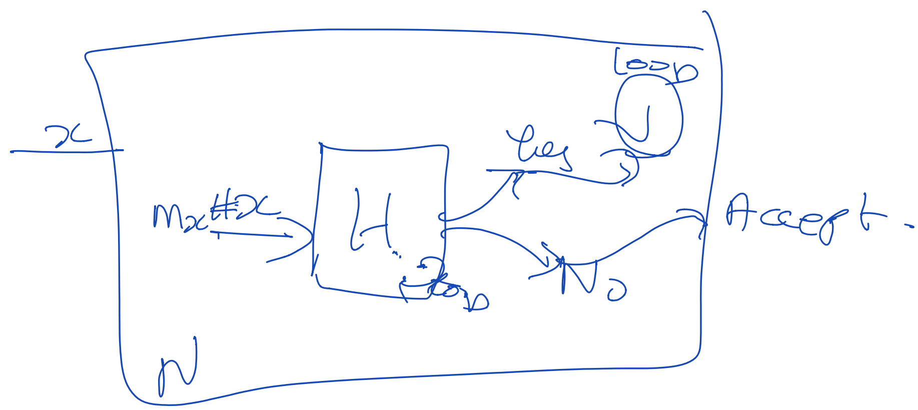
N is encoded in binary as y.

$N = M_y$.

Does N (M_y) halt on y.?

$\rightarrow\leftarrow$

N does not exist.



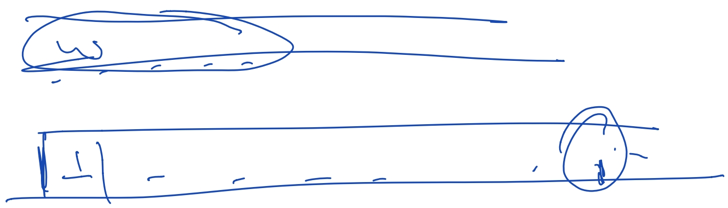
From our definition (description of N the only possible reason for this is that our assumption about H was wrong.

H is not Total.

1) HP: r.e but not rec.

2) HP is not co-re.

CFLs are recursive!



$L(M_A) = L(A)$

L

Membership Problem

$MP = \{ \underline{M\#w} \mid w \in L(M) \}$

Input: M, w

Qn: Is $w \in L(M)$?

H/W: Prove it is not recursive by Diagonalization.

<u>Proof by Reduction.</u>

we know

1) HP is not recursive

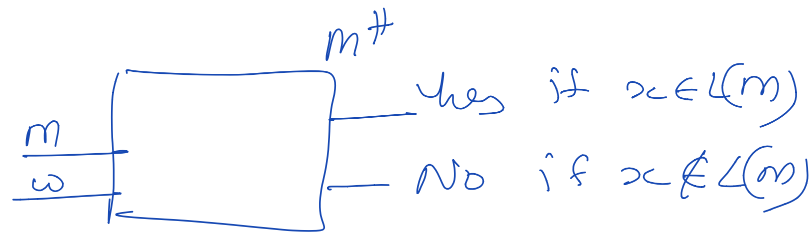
Assume MP is recursive.

$\exists$ a Total TM $M^{\#}$ for MP.

Reduce HP to MP ✓

~~MP to HP~~

N: $M\#w$



$M \to$ get N_M such that

$$L(N_M) = \{ w \mid M \text{ halts on } w \}$$

Construct H (a total TM for HP) as follows

On $M\#w$

1) save w in a tape.

2) From M, construct a TM
N_M
"N_M simulates M on any input
& accept if it halts (enter error)"

3) Check if $N_M \# w$ is
accepted by M^{H}:
Accept if M^{H} accepts
Reject o/w.