COL 352 Lecture 32 April 11

$$L_\phi = \{\langle M\rangle \in \Sigma^* \mid L(M) = \phi\}$$

$$L_{n\phi} = \{\langle M\rangle \in \Sigma^* \mid L(M) \neq \phi\}$$

Using a pair generator and trying all strings in some canonical order we showed that $L_{n\phi}$ is r.e.

How about L_ϕ?

Note that if L_ϕ is r.e. $\Rightarrow$ L_ϕ is recursive

recursive / decidable same
computable

<u>Claim</u> L_ϕ is non-r.e.

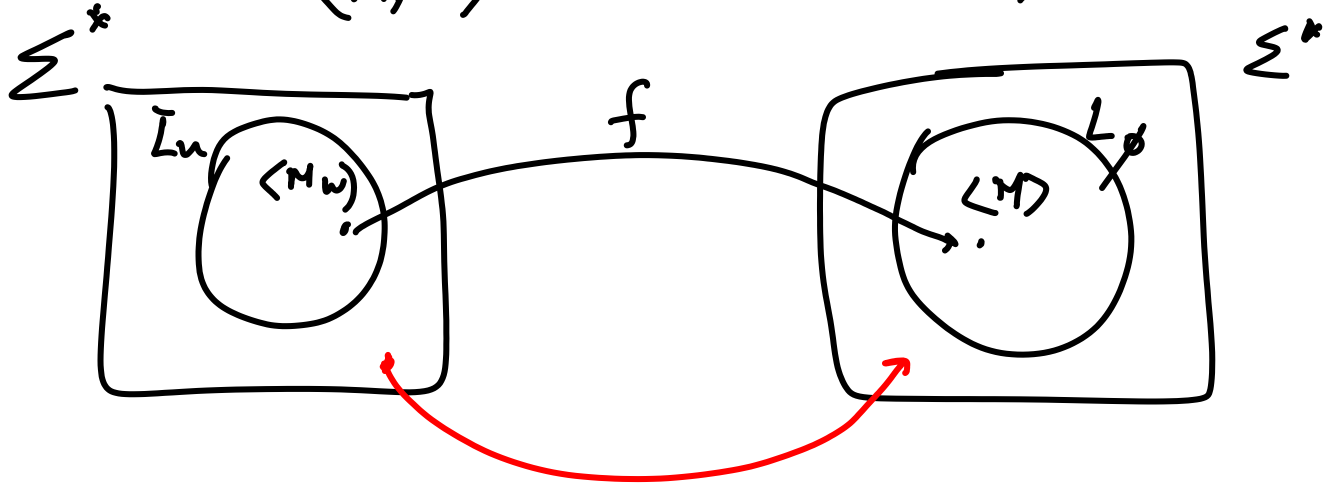
Consider some known non-r.e. lang L and $L \leq_f L_\phi$

f: many-one TM computable total function

$\{L_d, \bar{L}_u\}$

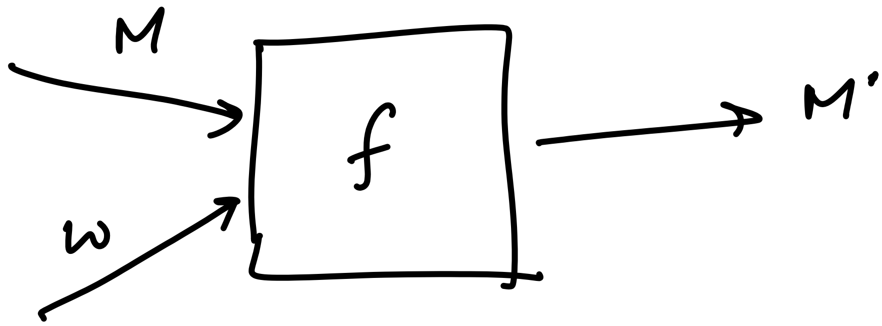
$$\bar{L}_u \overset{?}{\leq} L_\phi$$

$\langle M, w\rangle$ $\langle M\rangle$



① If M doesn't accept w then $f(M, w) = \langle M'\rangle$ must satisfy $\langle M'\rangle \in L_\phi$

② " M accepts w then $\langle M'\rangle \notin L_\phi$



Currying function

$g(x,y)$ convert it to $g_x(y)$

(transform a 2 input to a 1 input fun by fixing one parameter)

$g_x(y)$: g_{xy} with no input

$f(M,w)$ can be transformed to $f_M(w)$ and can be further transform to $f_{M,w}$

can be done using a program (Turing compu)

Initially f is similar the function for M_u and then transformed into $\boxed{M_u^{M,w}}$: code 1

Next we stitch another code 2

<u>code 2</u> : If M accept w

then read the input and

accept it

$$\boxed{\begin{array}{l}\text{code 1}\\ \text{code 2}\end{array}} : \langle M' \rangle$$

M' : 1 It initially ignores the input x

2 Runs M on w

3 If M accepts w, it accepts x

4 If M doesn't accept w, then x is not accepted

M accepts $w \rightarrow L(M') = \Sigma^*$

M doesn't accept $w \rightarrow L(M') = \phi$

Property of r.e. languages

✓ $\langle M \rangle$
- → empty
- → nonempty

undecidable

$\langle M \rangle$
- → Σ^*
- → not Σ^*

$\langle M \rangle$
- → recursive
- → non-recursive

$\boxed{\text{Any non-trivial property of r.e. languages is undecidable}}$

RICE'S THEOREM

Not all machines satisfy the property

Some $\langle M \rangle$ should satisfy and some M shouldn't satisfy