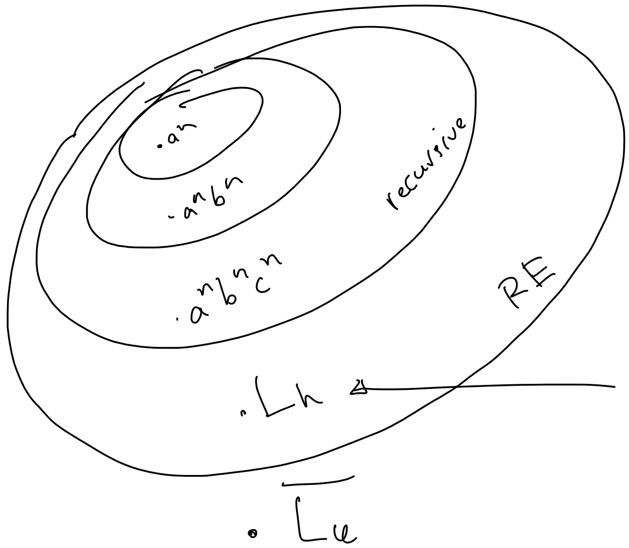

can do it on a TM, but some inputs "require ∞ time"

$$L_h = \{ M \# x \mid M \text{ is a TM which halts on } x \}$$

similarly. $L_u = \{ M \# x \mid$ - - - - - - accepts $x \}$

L_u is also RE but not recursive.

i.e: We can verify on a TM if some M accepts x, but can't always say yes/no : does some M accept x or not?

Turns out $\overline{L_u}$ is not RE.

<u>Pf</u> by contradiction, assume $\overline{L_u}$ is RE,

i.e. we can verify by TM if $x \in \overline{L_u}$.

so - - - - - - - $x \notin L_u$

So we can check for any x if $x \notin L_u$,
but since L_u is RE, we can also
check for any x if $x \in L_u$.

So we actually can say yes/no
if $x \in L_u$ or not,

$\not\Rightarrow$ L_u is RE but not recursive

It's impossible (by TM)
to tell if some python prog. will halt.

Chaitin's "Halting Probablity" Ω

$$f(5) = \frac{\text{\# of legal python programs of length 5 which halt}}{\text{total \# of progs of length 5.}}$$

$f(n)$ = proportion of programs of length n
which halt.

Consider $\lim_{n \to \infty} f(n)$ ← this exists.

$\parallel$

Ω ← the probability that
a random program
halts.

No TM can compute the digits of Ω.