CPSC 421/501 Sept. 22, 2025

Last time:

Let L_1, L_2, L_3, L_4 be a <u>partition</u> of $ASCII^*$. Then

(1) L_1, L_2, L_3, L_4 all recognizable

$\Rightarrow$ all decidable

(2) L_1 = NOT-PYTHON-INPUT

L_2 = ACCEPTANCE

L_3 = REJECTION

L_4 = LOOPING

is a partition of $ASCII^*$

(3) $L \subset \text{ASCII}^*$ is decidable

$\Longleftrightarrow$

L and L^{comp} are recognizable

(4) $T = \{ s \mid s \notin \text{LangRecBy}(s) \}$

is not recognizable.

Today: LOOPING is unrecognizable

ACCEPTANCE
REJECTION
HALTING
(= ACCEPTANCE ∪ REJECTION)

$\left.\begin{array}{l}\text{ACCEPTANCE}\\ \text{REJECTION}\\ \text{HALTING}\end{array}\right\}$ are undecidable

MPW inspires — do context-free grammars satis (1) & (3) ?

weaker

⋮

stronger

⋮

Start with

(1) Partition of $ASCII^*$ into

subsets L_1, L_2, L_3, L_4 means

each element of $ASCII^*$ lies

in exactly one of L_1, L_2, L_3, L_4

L_1, L_2, L_3, L_4 all recognizable

$\Rightarrow$ all decidable

by running L_1, L_2, L_3, L_4 "in

parallel", i.e. one step of each,

two steps of each, ...

Claim:

L_4 = LOOPING

is <u>unrecognizable</u>.

$$\left.\begin{array}{l} L_1 = \text{NOT-PYTHON-INPUT} \\ L_2 = \text{ACCEPTANCE} \\ L_3 = \text{REJECTION} \end{array}\right\} \text{are recognizable}$$

recognizing L means $\exists$ p program s.t.

$\forall s \in ASCII^*$

$s \in L \iff p$ accepts s

Pf: Assume not. Then

L_4 is recognizable.

So $L_1, \ldots, L_4$ recognizable

$L_1, \ldots, L_4$ decidable

We know

$$T = \{ p \mid p \notin \mathrm{LangRecBy}(p) \}$$

is unrecognizable...

Let's build a recognizer for T...

Given $p \in ASCII^*$ let's figure out if $p \in T$ or $p \notin T$

$p \sigma_d p$ ("p on input p")

Either

$$p \sigma_d p \in \begin{cases} L_1 \\ L_2 \\ L_3 \\ L_4 \end{cases}$$

So after finite # of Python steps of

If $p \sigma_d p \in L_1 =$ NOT-PYTHON-INPUT

is $p \in T_1$ or $\notin T_1$

$T = \{ p \mid p \notin \text{LangRecBy}(p) \}$

(Cantor $\Rightarrow$ T is not in im( )

$\Rightarrow$ T is not recognizable

$\underbrace{p}$ σ_0 p

not a valid Python program

we said

$$\text{LangRecBy}(p) = \emptyset$$

$p \notin \emptyset$?? yes $p \notin \text{LangRecBy}(p)$

$$\boxed{p \in T}$$

Case 2:

$p \, \sigma_1 \, p \in L_2$

$p\ \sigma_0\ p \in L_2 =$ ACCEPTANCE

p accepts p

$p \in \text{LangRecBy}(p)$

It's not true that $p \notin \text{LangRecBy}(p)$

$p \notin T$

Case 3: $p\ \sigma_2\ p \in L_3 =$ REJECTION

$p \in T$

p rejects p

$p \notin \text{LangRecBy}(p)$

$p \in T$

$$p\,\sigma_0 p \quad \begin{matrix} \in L_1 \\ \in L_2 \\ \in L_3 \end{matrix} \Bigg\} \text{ we know if } p \in \text{ or } \notin T$$

$$p\sigma_0 p \in L_4 = \text{LOOPING} \Rightarrow p \in T$$

$$T = \{ p \mid p \notin \text{LangRecBy}(p) \}$$

$$\{ p \mid p \text{ does not accept } p \text{ as input} \}$$

$$\text{ASCII}^* \setminus \{ p \mid p\sigma_0 p \in \text{ACCEPTANCE} \}$$

$$= \text{ASCII}^* \setminus \{ p \mid p\sigma_0 p \in L_2 \}$$

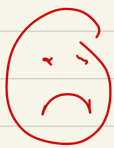
Hence we get a decider for

T. Impossible ...

Infinitely many primes. Why?

1. Assume $p_1 = 2, p_2 = 3, \dots, p_N =$ last prime

$p_1 p_2 \cdots p_N + 1 \leftarrow$ not divisible by $p_1, \dots, p_N$

contradiction ... 

2. Unique factorization of any $\mathbb{N}$

3. All primes ≥ 3 $\begin{cases} 4k+1 \\ 4k+3 \end{cases}$

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \frac{1}{13} + \cdots$$

$$\left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \cdots = 1 \right)$$

Thm $= \infty$

If finite, then

$$\sum_{i \geq i_0} \frac{1}{p_i} \leq \frac{1}{2}$$

CLASS ENDS

$$p(i,j) = \alpha_0 + \alpha_1 i + \alpha_2 j$$
$$+ \alpha_3 i^2 + \alpha_4 ij + \alpha_5 j^2$$

$\alpha_i \in \mathbb{N}$

$REJ = \{ p\sigma_0 i \mid p \text{ rejects } i \}$

L ~~aren't~~ are recognizable, $\exists q$ Python prog

$\Leftrightarrow \forall s \in ASCII^*$

if $s \in L$, then q halts, says yes

if $s \notin L$, then not

If you get

p program

i input

does "p reject i" ?

decider has to stop, say "yes"
"no"

recognizer has to stop, "yes"
if p rejects i

can stop say "no" or
do anything but stop
and say "yes"
if p ~~reject~~ i