

CPSC 421/501

Sept 23, 2024

- Our proof that NON-ACCEPTANCE is unrecognizable is a reduction:

- Examples of undecidable and unrecognizable problems

- Note: We are only collecting (1-3) on Homework 3 this week. Problems (4, 5) will be collected Thursday, Oct 3

Reduction: We know

$$T = \left\{ q \in \Sigma_{ASCII}^* \mid q \notin \text{LangRecBy}(q) \right\}$$

is unrecognizable. We show

NON-ACCEPTANCE is recognizable

$\Rightarrow$   $T$  is recognizable

Hence, by contradiction ---

- Each string in  $\Sigma_{ASCII}^*$  is in one of:

$NOI-PYTHON-INPUT = \{ \text{not of the form } p \sigma_0 i \text{ with } p \text{ a valid Python program} \}$

ACCEPTANCE =  $\{ p \sigma_0 i \mid p \text{ accepts } i \}$

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

LOOPING =  $\{ p \sigma_0 i \mid p \text{ loops on } i \}$

$\sigma_0 = \langle FS \rangle$

$q \in T?$

Now: Either

(1)  $q \sigma_0 q \in \text{NOT-PYTH-IMP}$ :  $q \in T$

(2)  $q \sigma_0 q \in \text{ACCEPTANCE}$ :  $q \notin T$

(3)  $q \sigma_0 q \in \text{REJECTION}$ :  $q \in T$

(4)  $q \sigma_0 q \in \text{LOOPING}$ :  $q \in T$

Alg 1: Uses universal Py prog

Alg 2: Uses recognizer for

NON-ACCEPTANCE!

Input to Alg 1, 2:  $q \sigma_0 q$

If  $q$  is not valid P<sub>1</sub> prog

$$\text{LangRecBy}_1(q) \stackrel{\text{defined}}{=} \emptyset$$

$$q \notin \emptyset = \text{LangRecBy}_1(q)$$

$$q \notin \text{LangRecBy}_1(q)$$

$$\Rightarrow q \in T$$



$$q \in q \in \text{ACCEPTANCE}$$

$$q \in \text{LangRecBy}_1(q)$$

Cantor's Theorem

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

is not recognizable

- unrecognizable

- Not in the image of  $\text{LangRecBy}$

$a \text{ Py Prog}$     Rec For  
 $WON \cdot ACC$

$q \in a$  in  $\mathbb{Z}$

Alg 1

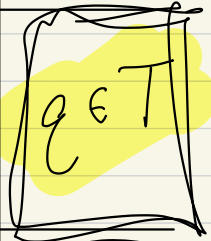
Alg 2

in  $T$

NON-PYTH-IND



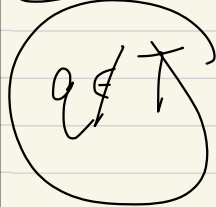
?



ACCEPTANCE



?



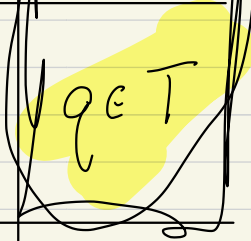
REJECTION



LOOPING



not fooled, but  
may never terminate



$$T = \{ q \mid q \text{ does not accept } q \}$$

$$= \{ q \mid q \notin \text{Lang RecBy}(q) \}$$

Decider: A Py program that  
always accepts (return "yes")  
or rejects (return "no")

---

Decidable: PRIMES, PALINDROME,  
NON-PYTH-INP, ...

---

Thm:  $L$  is decidable  $\Leftrightarrow$   
 $L, L^{\text{comp}}$  are recognizable

---

(sometimes  $\overline{L} = \sum^* \setminus L = L^{\text{comp}}$ )



Pf:

Say that  $L$  is decidable.

$L$  is recognizable.

But also  $L^{\text{comp}} = \bar{L}$  is decidable

(in Python prog

return "yes"  $\rightsquigarrow$  return "no"

return "no"  $\rightsquigarrow$  return "yes"

Rem:  $L$  is Duck recognizable

$\Leftrightarrow L = \text{finite union:}$

$$\sum^{k_1} \cup \sum^{k_2} \cup \dots \cup \sum^{k_m}$$

$m \geq 0$

e.g.,

duck duck 6 duck 3

$$\text{LangRegBy}(\quad) = \Sigma^3 \cup \Sigma^6$$

is  $\Sigma^*$   $\searrow$   $(\Sigma^3 \cup \Sigma^6)$

$$= \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^4 \cup \Sigma^5 \cup \Sigma^7 \cup$$

$$\Sigma^8 \cup \Sigma^9 \cup \Sigma^{10} \cup \dots$$

not Duck recog.

Is Python stronger 4 Let Duck

(1) Image (Lang Rec By Duck)

$\subsetneq$  Image (Lang Rec By Python)

(2)  $L \in \text{Image (Lang Rec By Python)}$

$\Rightarrow L \in \text{Comp}$

$\Leftarrow$

Conversely, say that

$L$  and  $\bar{L} = L^{\text{comp}}$  are

recognizable ...

|            |   |  |
|------------|---|--|
|            | Alg 1                                     | Alg 2                                      |
| $s \in L?$ | if<br>$s \in L$<br>it<br>returns<br>"yes" | if<br>$s \notin L$<br>it<br>returns<br>yes |

1 step ,  
2 steps ,  
1 step  
2 steps

SEL

Alg 1

Alg

SEL



SEL



Q: Could ACCEPTANCE be  
decidable??

Corollary! ACCEPTANCE and  
HALTING are undecidable.

---

Corollary:  $\{ p \text{ is a valid } P_y$   
prog that accepts some input }  
is undecidable but recognizable.