CPSC 421/501 Sept 18, 2024

- Cantor's theorem $\Longrightarrow$

  $\text{NON-ACCEPTANCE}_{\text{Python}}$ and

  $\text{NON-HALTING}_{\text{Python}}$ are unrecognizable

- Universal Python programs

today for:

- $\text{ACCEPTANCE}_{\text{Python}}$ and

  $\text{HALTING}_{\text{Python}}$

  are (Python-) recognizable

  and (Python-) undecidable

Recall: Python has statements:

(1) i = input("Your input: ") ← input

(2) return("yes") ← we accept

(3) return("no") ← we reject

(4) other stuff ← "sufficiently expressive or powerful"

Remark: We'll need

universal Python programs.

Remark:

universal Duck programs

don't exist.

(eg. <EOF>)

Fix $\sigma_0 \in \Sigma_{ASCII}$

say $\sigma_0 = \langle FS \rangle$ file separator

(^\ ) char 28

IC

that is not allowed in a valid Python program

Theorem: Let

$$HALTING = \{ p\, \sigma_0\, i \in \Sigma^*_{ASCII}$$

s.t. p is a valid Python program and p halts on i $\}$

$= \{$ any string of the form

$p \circ \sigma_0 \circ i$ where

$p, i \in \Sigma^*_{ASCII}$ and

p is a valid Python program,

$i \in \Sigma^*_{ASCII}$ such that

p halts on i $\}$

NON-HALTING

$= \{ p \sigma_0 i \mid p \in$ VALID-PYTHON-PROG and p does not halt on $i \}$

halting = we either accept or reject

if p on input i "doesn't halt"

we say p on input i : "loops"

ACCEPTANCE = { $p \sigma_d \bar{i}$ |

$p \in$ VALID-PYTHON-PROG that accepts i }

NON-ACCEPTANCE = { $p \sigma_d \bar{i}$ |

$p \in$ VALID-PYTHON-PROG that doesn't accept i }

Rem: If s is any Σ_{ASCII}^* s.t. σ_0 appears at least once in s, then there is a unique way to write

$$s = p\,\sigma_0\,i$$

where

p does not contain σ_0

and

$i \in \Sigma_{ASCII}^*$ arbitrary

Theorem: Both

NON-HALTING and

NON-ACCEPTANCE are

unrecognizable, i.e. not

in the image of $\mathrm{LangRecBy}$,

$$\mathrm{LangRecBy} : \Sigma_{\mathrm{ASCII}}^{*} \longrightarrow \mathrm{Power}\left(\Sigma_{\mathrm{ASCII}}^{*}\right)$$

(Here $\mathrm{LangRecBy}$ we mean

$\mathrm{LangRecBy}_{\mathrm{Python}}$, not $\mathrm{LangRecBy}_{\mathrm{Duck}}$)

Pf: By Cantor's Theorem,

$$T = \{ q \in \Sigma^*_{ASCII} \mid q \notin \text{LangRecBy}_{\text{Python}}(q) \}$$

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

is unrecognizable, i.e. not in the image of $\text{LangRecBy}_{\text{Python}}$.

Say, for the sake of contradiction that, r, recognizes

NON-HALTING. Using r

let's build a Python program that

recognizes T:

Given $q \in \Sigma^*_{ASCII}$, in parallel

run

Alg 1: Check if q is a

valid Python program (☹ tedious).

If not --- q rejects any input

so $q \notin Long Rec By(q)$, so $q \in T$.

If q is a valid Python program

simulate q on input q

if q accepts q ← halt, return "no"

if q rejects q ← halt return "yes"

 tedious to write such a "simulator" or "debugger"

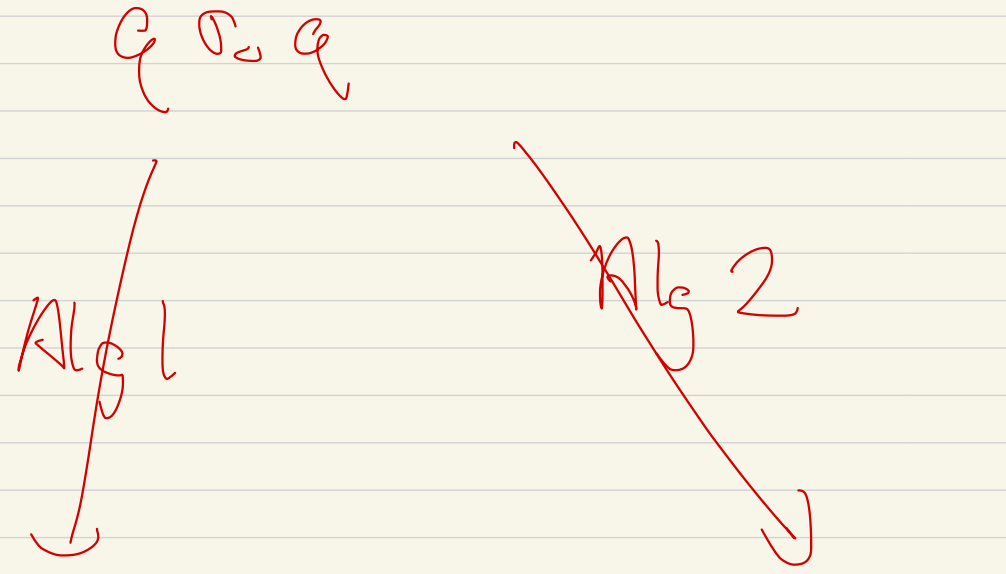
We say that u is a universal Python program if on input $p\sigma_0 i$ s.t. $p \in$ VALID-PYTHON-PROGRAM, u accepts $p\sigma_0 i \iff p$ accepts i

[u rejects $p\sigma_j i$ $\Leftrightarrow$ p rejects i]

Alg 2: Run an algorithm to see if

$q\sigma_q q \in$ NON-HALTING

(NON-ACCEPTANCE is fine)



says "no" if q accepts q

says "yes" if q rejects q

says "yes" if q does not halt on q

But q on input q { accepts / rejects / doesn't halt

(universal Python prog) (Contradiction)

If

Alg 1 Alg 2

q accepts q


q rejects q


q { loops / doesn't halt } q


q is not valid

 runs and halts after finitely many steps

AND SO ...

CLASS ENDED ...