

CPSC 421/501

Nov 18, 2021

Today!

- Universal TM's
- Delightful TM's
- $ACCEPTANCE_{TM}$, $HALTING_{TM}$, ...
undecidable
- "Abstructify"
 - Program-Input Systems
Results: $P \times \mathcal{I} \rightarrow \{\text{yes, no, loops}\}$
 - Enhanced P-I Systems
- Oracle TM's

← Defined Ch 6
← Ch 9

$$\Sigma_{\text{word!}} = \{ @, !, \#, L, R \}$$

↑ ↑
here for
convenience

There's a map:

$$\text{Standardized TM} \rightarrow (\Sigma_{\text{word!}})^*$$

"Reasonable description
of a TM"

Standardized TM:

blank

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej}, \sqcup)$$

subject to:

$$(1) \quad Q = \{1, \dots, q\} \text{ some } q \in \mathbb{N},$$

$$(2) \quad \Sigma = \{1, \dots, s\} \text{ some } s \in \mathbb{N},$$

$$(3) \quad \Gamma = \{1, \dots, s, s+1, \dots, \gamma\}$$

\uparrow
blank

$$\gamma \in \mathbb{N}$$

$$(4)$$

$$(5) \quad q_{acc} = 2, q_{rej} = 3, q_0 = 1$$

$$\text{and } q_0 \neq \{q_{acc}, q_{rej}\}$$

$$[\gamma \geq s+1, q \geq 3, \dots]$$

Def A standardized TM is
one that satisfies the above,

a standardized alphabet is

a $\Sigma = \{1, \dots, s\}$ for some $s \in \mathbb{N}$.

=

So $\Sigma = \{a, b\}$ is not standardized

can we solve

PALINDROME $\Sigma = \{a, b\}$?

Take bijection

$\{a, b\} \rightarrow \{1, 2\}$

ask same question of PALINDROME $\Sigma_{\{1, 2\}}$

Remark: Any TM over $\Sigma = \{1, \rightarrow, \epsilon\}$

has an "equivalent TM"

that is standardized:

i.e.

Result(some
TM, input)

should be {yes, no, loops}

\mathcal{P} = set of programs some subset of $\left(\sum w_{\omega_i}! \right)^*$

\mathcal{I} = set of inputs " "

$\sum w_{\omega_i}! = \{0, !, \#, L, R\}$

Abstractly, an algorithm is

Result: $P \times \mathcal{I} \rightarrow \{ \text{yes, no, loops} \}$

Compare Prof \times Icecream \rightarrow $\left\{ \begin{array}{l} \text{yes} = \text{like} \\ \text{no} = \text{doesn't} \\ \text{like} \end{array} \right\}$

e.g.

TM last class:

$\langle q \text{ in binary} \rangle \# \langle 5 \text{ in binary} \rangle \#$

$\langle \gamma \text{ in binary} \rangle \# \underbrace{5(1,1)}$

$\# 5(1,2)$
 \vdots

TM & input:

$\langle q \text{ in binary} \rangle \# \langle s \text{ in binary} \rangle \#$

$\langle \gamma \text{ in binary} \rangle \# \underbrace{\delta(1,1)}$

$\# \delta(1,2)$
:

$\# \delta(q, \gamma)$

$\# \#$ 1st symbol/letter of input

$\#$ 2nd " "

:

$\#$ last symbol/letter of input

So this gives

$$\text{EncodeProg: } \mathcal{P} \rightarrow \sum_{w \in \mathcal{L}}^k$$

\mathcal{L}

Encode Both Prog And Input

$$\mathcal{P} \times \mathcal{L} \rightarrow \sum_{w \in \mathcal{L}}^k$$

\mathcal{L} subset of $\{0, 1, \#\}^k$

For simplicity

$$\mathcal{L} = \sum_{w \in \mathcal{L}}^k$$

if input is not what we expect, say not a

valid input

⊆

Back to TM's:

$$\text{ACCEPTANCE}_{TM} = A_{TM} =$$

$$\stackrel{\text{def}}{=} \left\{ \langle p, i \rangle \mid \begin{array}{l} \text{Result}(p, i) \\ = \text{accept} \end{array} \right\}$$

$$[Sip] = \left\{ \langle p, i \rangle \mid \begin{array}{l} p(\langle i \rangle) \\ = \text{accept} \end{array} \right\}$$

$$\stackrel{\text{def}}{=} \left\{ \text{EncodeBoth}(p, i) \mid \begin{array}{l} \text{Result}(p, i) = \text{yes} \\ \text{Result}(p, i) = \text{accept} \end{array} \right\}$$

Helps me to think about what is going on

Thm: $A_{TM} = \text{ACCEPTANCE}_{TM}$

is undecidable (in the context of TM's).

Thm: HALT_{TM}

$= \left\{ \langle p, i \rangle \mid p(\langle i \rangle) = \begin{matrix} \text{accept} & \text{reject} \\ \uparrow & \uparrow \\ \text{yes or no} \end{matrix} \right\}$

$= \left\{ \text{Encoding}(p, i) \mid \text{Result}(p, i) = \begin{matrix} \text{yes or no} \end{matrix} \right\}$

$=$

Proof:

Break 10:15 - 10:20

Define: For each TM, M , over Σ

① $LANGUAGE_{\Sigma}(M)$

$= \{ i \in \Sigma^* \mid M \text{ accepts } i \}$

we say M recognizes

② We say M "halts on i "

if ---

M accepts i , or

M rejects i , but

M doesn't loop on i

③ We say M is a decider

if on any input, i , M

either rejects or accepts, i.e. M

halts.

I.e.

$\text{Result}(\overset{a}{T_M}, \overset{an}{input}) \in \{\text{yes, no, loops}\}$

"to decide", M is decider,

means

$\text{Result} : \text{Program} \times \text{Input} \rightarrow \{\text{yes, no, loops}\}$
(T_M)

also a map

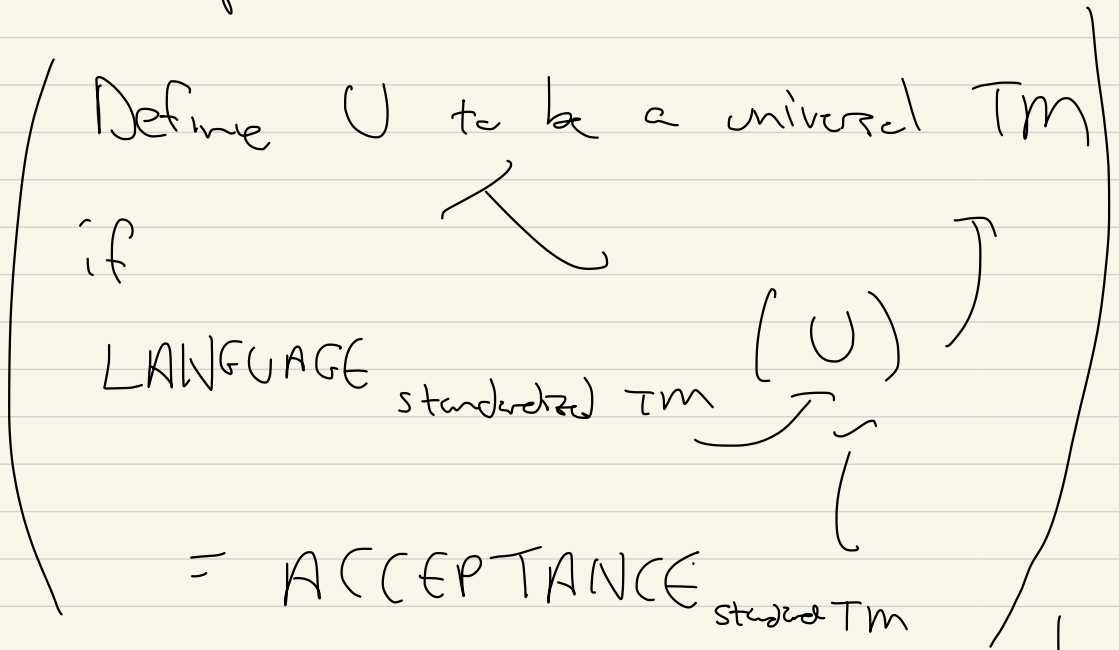
$\rightarrow \{\text{yes, no}\}$
(as a subset of)

i.e. reach accept / yes

reject / no

after finitely many steps

Example!

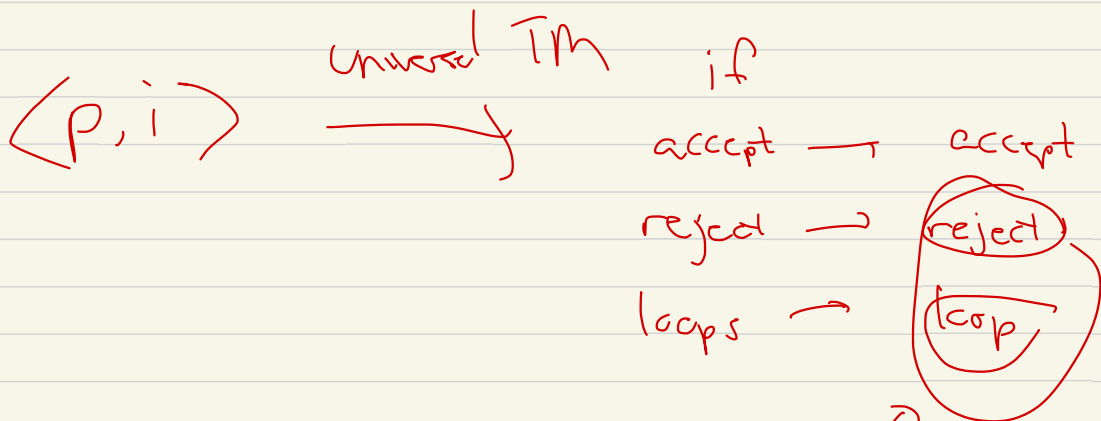


Really a universal TM is an
enhanced debugging tool

very mild simplification!

Universal TM as simplifier,
yes \rightarrow yes
no \rightarrow no
loops \rightarrow loops

build a U as above



as in [Sip] §4.2

but in either reject or loops

Def: A TM D is delightful if

$$\forall M \text{ TM} \quad \text{Result}(D, \langle M \rangle) = \neg \text{Result}(M, \langle M \rangle)$$

$$[Sip] \quad D(\langle M \rangle) = \neg M(\langle M \rangle)$$

$$\text{or} \\ \text{Result}(D, \text{EncodeProg}(M))$$

$$= \neg \text{Result}(M, \text{EncodeProg}(M))$$

Thm: If D is delightful,
then

$$\textcircled{1} \text{Result}(D, \text{EncodeProg}(D))$$

$$= \neg \text{Result}(D, \text{EncodeProg}(D))$$

$$\textcircled{2} \text{Hence } \text{Result}(D, \text{EncodeProg}(D)) = \text{loops}$$

Cor: Hence there is no \cup TM
that decides ACCEPTANCE, i.e.
ACCEPTANCE is undecidable.

