(Ch. 5)

Def: $L_1 \leq L_2$ if

L_1's alphabet, Σ_1

L_2's " " Σ_2

has $f: \Sigma_1^* \to \Sigma_2^*$

s.t.

① $\forall w \in \Sigma_1^*$

$w \in L_1 \iff f(w) \in L_2$

② f is computable by a Turing machine (in a finite amount time/steps on any input)

CPSC 421/501

- Midterm in one week.
- Solutions to HW 3 released "immediately" after class
- Next week: Wednesday: look at some sample exam problems
- Cover: Undecidability of A_{TM} and recognizability " "

=

Today: More abstractions...

=

Last time:

$A_{TM} \leq Halt_{TM}$

"reducibility"

So: If $Halt_{TM}$ were decidable (subjunctive)

then A_{TM} would be since

A_{TM} can be reduced to $Halt_{TM}$

i.e., $A_{TM} \leq Halt_{TM}$

↑ reduction

=

Given A_{TM} undecidable

$\rightsquigarrow$ $Halt_{TM}$ "

$\rightsquigarrow$ Many other problems undecidable

=

A_{TM} is recognizable but undecidable,

Motivation:

Given a problem: M, w

is $\langle M, w \rangle \in A_{TM}$??

(we know undecidable)

It's not hard to take

$\langle M, w \rangle \xrightarrow{\text{compute}} \langle \tilde{M}, w \rangle$

If M on w accepts, so does $\tilde{M}$

" " " " doesn't halt, - - -

- - - - rejects, $\tilde{M}$ doesn't halt

=

$\rightsquigarrow \langle M, w \rangle \in A_{TM} \iff \langle \tilde{M}, w \rangle \in Halt_{TM}$

e.g. fix alphabet Σ

$\mathcal{P} = \{$ ~~all~~ Turing machines over $\Sigma \}$

$\mathcal{A} = \Sigma^*$

$\text{Result}(p, i) = \{\text{yes}, \text{no}, \text{notHalt}\}$

$p \in \mathcal{P}$

$i \in \mathcal{A}$

—

Usually $\mathcal{P}$ and $\mathcal{A}$ are of the same size (cardinality)

If we standardise Turing machines $Q = \{1, \ldots, a\}$, $\Sigma = \Sigma$, $\Gamma = \{1, \ldots, c\}$.

Then $\overline{L} = L^{comp}$ is unrecognisable.

=

Discuss: - These ideas abstractly (§6 Hardcut)

- Variants of Turing machines (oracle, non-deterministic) ← Sipser Ch. 6 Ch. 3

=

§6 of Uncomputability & Self-referencing

Abstractly: Set of programs, $\mathcal{P}$

" " inputs, $\mathcal{A}$

Result: $\mathcal{P} \times \mathcal{A} \to \underbrace{\{\text{accept}, \text{reject}, \text{doesn't halt}\}}_{\mathcal{R}}$

Axiom { We have sets $\mathcal{P}$, $\mathcal{A}$ and Result: $\mathcal{P} \times \mathcal{A} \to \{\text{yes}, \text{no}, \text{notHalt}\}$

Definitions:

① a language is a subset of $\mathcal{A}$

② any program $p \in \mathcal{P}$ recognises the language

$$L_p = \{ i \in \mathcal{A} \mid \text{Result}(p, i) = \text{yes} \}$$

③ $p \in \mathcal{P}$ is a decider if for all $i \in \mathcal{A}$, Result(p, i) is either "yes" or "no"

$\mathcal{P} = \{$Turing machines$\}$ and $\mathcal{A}$ both countably infinite

—

Usually $\mathcal{P}$, $\mathcal{A}$ are countably infinite

—

Example 2: $\mathcal{P} = \{$C-programs$\}$

$\mathcal{A} = \{$ASCII strings$\}$

(any C-program) (any input)

result: stop, accept (Yes)
" , reject (No)
never stop

Remark: If

- $\mathcal{P}$ and $\mathcal{A}$ are countably infinite

OR

- EncodeProg : $\mathcal{P} \to \mathcal{A}$ is injective

then

$\{Languages\} = Power(\mathcal{A})$

$2^{\mathcal{A}}$, Set of subsets of $\mathcal{A}$

is larger than $\mathcal{P}$

$p \in \mathcal{P} \rightsquigarrow L_p \subset \mathcal{A}$

then there exist languages that aren't recognized since $\mathcal{P} < Power(\mathcal{A})$

$\mathcal{L}$ = Languages = $Power(\mathcal{A})$ is uncountable

Ch4: $M \rightsquigarrow \langle M \rangle$

$M, w \rightsquigarrow \langle M, w \rangle$

Axiom: There's a function

EncodeProg : $\mathcal{P} \to \mathcal{A}$

$p \mapsto \langle p \rangle$ or EncodeProg(p)

and function

EncodeProgInput : $\mathcal{P} \times \mathcal{A} \to \mathcal{A}$

$(p, i) \to \langle p, i \rangle$ or EncodeProgInput(p, i)

Axiom

$\mathcal{P}$ = programs, $\mathcal{A}$ = inputs

Define: $\mathcal{L}$, the set of languages is $Power(\mathcal{A})$, i.e. set of all subsets of $\mathcal{A}$.

If $|\mathcal{P}| \leq |\mathcal{A}|$ then $|\mathcal{L}| > |\mathcal{P}|$ so for any map $\mathcal{P} \to \mathcal{L}$, there are languages not in the image

=

$p \rightsquigarrow L_p = \{ i \in \mathcal{A} \mid Result(p, i) = yes \}$