Midterms covers material up to Oct. 16

NOT: $\overline{A_{TM}}$ is unrecognizable

Office hours afternoon Wed, Thursday
TA: usual office hours

CPSC 421/501

- Notes may not be posted due to technological problems...
- Midterm covering:
  - Sections 1-4 Handout (Paradoxes, Counting)
  - Textbook: Ch 3: Turing machines, multitape, [Not non-deterministic]
    Ch 4: - Counting (countable vs. uncountable)
    - A_{TM} is recognizable but not decidable
    Ch 7 - Time and space that a Turing machine takes

last thing

I'll post some sample exam problems: include homework problems, old exam probs

Define (just from Axiom 1):

- a language is a subset of Σ^* (set of languages = Power(Σ^*))
- ~~A~~
- a program, P, recognizes the language $L_P = \{ i \in \Sigma^* \mid P[i] = yes \}$
- $P \in \mathcal{P}$ is a decider if $\forall i \in \Sigma^*$ P[i] is either yes or no.
- $L \subset \Sigma^*$ is decidable if $L = L_P$ for some $P \in \mathcal{P}$ that is a decider

§6 Handout: Idea: give abstract proof of undecidability of A_{TM}

abstract ↔ very general, getting to the "essence"

Axiom 1: We have a set $\mathcal{P}$, set Σ^* ("programs" and "inputs") and

Result: $\mathcal{P} \times \Sigma^* \to \{yes, no, notHalt\}$

Abbreviate: $P \in \mathcal{P}$ $i \in \Sigma^*$, use P[i] to denote Result(P, i).

Example: $\mathcal{P} = \{$C-programs$\}$, Σ^* = ASCII strings

$P \in \mathcal{P}$, $i \in \Sigma^*$, $P[i] = \begin{cases} yes - accept \\ no - reject \\ notHalt - program\ P\ never\ stops\ on\ input\ i \end{cases}$

Define

A_P = Acceptance problem

$\overset{\text{def}}{=} \{ \langle P, i \rangle \mid P[i] = \text{yes} \}$

=

Until now:

Axiom 1: set $\mathcal{P}$, set $\mathcal{A}$, $P[i]$

Axiom 2: injection $(P, i) \mapsto \langle P, i \rangle$ program U s.t. $U(\langle P, i \rangle) = P[i]$.

=

(Remark: When we get to "oracle Turing machines", those also fit these axioms)

- $L \subset \mathcal{A}$ is recognizable if $L = L_P$ for some $P \in \mathcal{P}$.

=

Example 2 (in Ch. 4 Sipser): Fix alphabet Σ, $\mathcal{P} = \{\text{Turing machines with } \Sigma\}$, $\mathcal{A} = \Sigma^*$.

=

Axiom 2: There is an injection:

EncodeBoth: $\mathcal{P} \times \mathcal{A} \to \mathcal{A}$.

(denote EncodeBoth(P, i) by $\langle P, i \rangle$)

and there is a "Universal element of $\mathcal{P}$." U, s.t.

$$U[\langle P, i \rangle] = P[i]$$

also $\forall P \in \mathcal{P} \; \exists P' \in \mathcal{P}$ s.t.

$P[i] = \text{yes}$	$P'[i] = \text{no}$
$P[i] = \text{no}$	$P'[i] = \text{yes}$
$P[i] = \text{notHalt}$	$P'[i] = \text{notHalt}$

==

Is this reasonable ??

$\forall P \; \exists P'$ s.t.

$P[i] = \text{yes}$	$P'[i] = \text{notHalt}$
$P[i] = \text{no}$	$P'[i] = \text{notHalt}$
??? $P[i] = \text{notHalt}$	$P'[i] = \text{yes}$

unreasonable (solves the Halting problem)

Remark: If U is a universal machine and $U(i') = \text{no}$ if i' is not of the form $\langle P, i \rangle$, then

$$L_U = \{ \langle P, i \rangle \mid P[i] = \text{yes} \}$$

$$= \text{Acceptance}_P$$

=

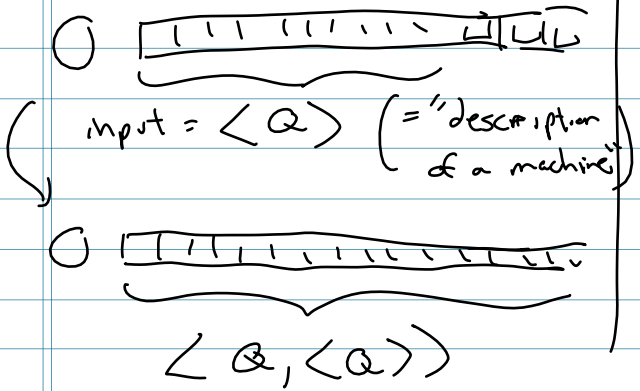
Axiom: Given $P \in \mathcal{P}$ there is a program P' s.t. for all $i \in \mathcal{A}$

$P[i] = \text{yes}$	$P'[i] = \text{no}$
$P[i] = \text{no}$	$P'[i] = \text{notHalt}$
$P[i] = \text{notHalt}$	$P'[i] = \text{notHalt}$

Axiom 3 for Turing machines, C-programs, Java programs, ... works by taking P and adding code at the end.

Axiom 4 does the same, but adds code to the beginning...

Turing machine:



Axiom 3: Let $f: \{yes, no, notHalt\} \to \{yes, no, notHalt\}$ s.t. $f(notHalt) = notHalt$ and Let $P \in \mathcal{P}'$. Then $\exists P' \in \mathcal{P}'$ s.t. $P'[i] = f(P[i])$.

Axiom 4: (Self-referential-ism)

There exists injection: $\mathcal{P} \to \mathcal{A}$ called EncodeProg, abbreviated $\langle P \rangle$ s.t. $\forall P \in \mathcal{P}$ there is a $P' \in \mathcal{P}$ s.t.

$$P'[\langle Q \rangle] = P[\langle Q, \langle Q \rangle \rangle]$$

for all $Q \in \mathcal{P}$.