CPSC 4N2A/501 Nov 27

- Chapter 3: Oracle Turing machines
- Chapter 8: NPSPACE = PSPACE
  - there are problems complete for PSPACE
- Chapter 9: there is an oracle, A, for which $P^A = NP^A$

Final Exam: Dec 8, 8:30am (Tuesday)

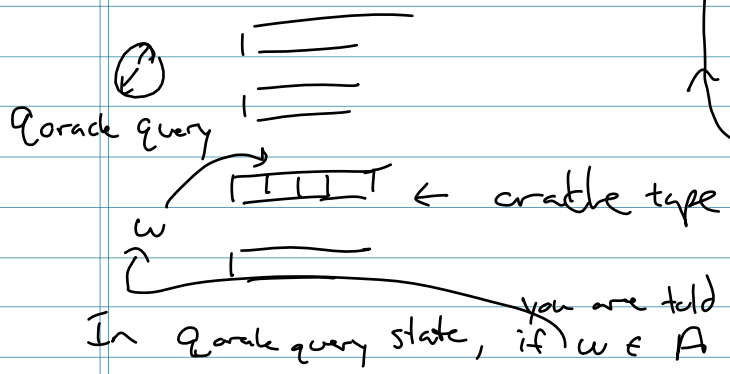
Next week: finish, solve sample exam problems.

Ch 3: Oracle Turing machines:

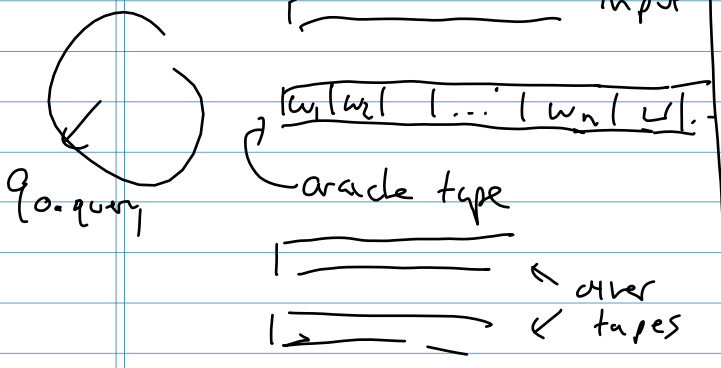
Fix Σ_A alphabet,

fix $A \subseteq \Sigma_A^*$. A Turing machine, M, with oracle A:

idea: you have a "button on your Turing machine"



Oracle T.m. with oracle A works like T.m. with one added feature



if $w = w_1 \dots w_n \in A$

$q_{0.query} \to q_{0.says\ yes}$

Formally:

Multi-tape T.m.

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej}, q_{oracle\ query}, q_{oracle\ says\ yes}, q_{oracle\ says\ no})$

- one of M's tapes is designated "oracle tape"
- $\Sigma_A \subseteq \Gamma$

e.g.

T.m. with oracle HALT

" " " A for any language A

HALT, SAT, TQBF, - -

=

Here's the problem with almost all theorems in Sipser's textbook regarding "complexity theory"

- We know there languages not Turing - { decidable / recognizable

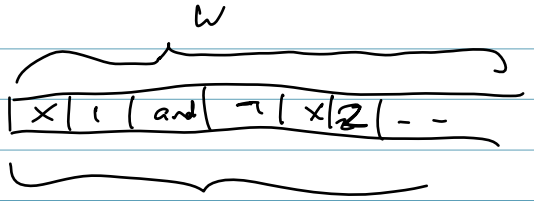
and if

$w = w_1 \ldots w_n \notin A$

q_0. query → q_0. says no .

=

E.g. T.m. with oracle SAT



can ask oracle if this is is SAT

So in one step, no additional memory, you see if $w \in$ SAT or not

=) there are languages that cannot be { decided / ~~accepted~~ recognize } by any T.m.

=

Fix oracle, $A \subset \Sigma_A^*$.

Standardize any T.m. with oracle A

$Q = \{q_0, q_{acc}, q_{rej}, q_{0.q.}, q_{0.y.}, q_{0.n}, \ldots\}$

if we want { 1, 2, 3, 4, 5, 6, 7, 8, - -, 9 } other states

{ languages over fixed alphabet }

is uncountable.

Each T.m. $(Q, \Sigma, \Gamma, \delta, \ldots)$

(even crazy T.m.s)

have a standardized form:

$Q = \{1, \ldots, a\}$

$\Sigma = \{1, \ldots b\}$

{ ⟨ any Turing machine in standard form ⟩ } is countable

the halting problem for "T.m. with oracle A" is undecidable by "Turing machines with oracle A"

=

A = HALT:

"T.m. with oracle HALT" (can solve usual halting problem) cannot decide "Halting problem for T.m. with oracle HALT"

= Etc.

Same theorem is true fix an oracle, A:

- Set of languages is uncountable,
- Set of T.m. recognizable languages with oracle A is countable

≥

The halting problem is undecidable -- acceptance " " -- -.

The complement of halting problem is not even recognizable..

=

for any fixed oracle A,

$SPACE^A(f(n)) = \{$ .. space ... $\}$

$NSPACE^A(f(n)) = \{$ .. non-det ... $\}$

$P^A = \bigcup_{k=1,2,\ldots} TIME^A(n^k)$

$NP^A = \ldots NTIME^A \ldots$

for given $A \subset \Sigma_A^*$ set

$TIME^A(f(n)) \stackrel{def}{=}$

{languages decided within time $O(f(n))$ by a Turing machine with oracle A}

$NTIME^A(f(n)) \stackrel{def}{=} \{$ .. $O(f(n))$ .. non-deterministic .. oracle A$\}$

Theorem: for all $x,y,z > 0$ integer $n > 0$

$$x^n + y^n \neq z^n$$

if $n \geq 3$ (with x,y,z are integers)

[with x,y,z are reals]

false

=

So if "proof" $P \neq NP$ also shows $P^A \neq NP^A$ for any A, then "proof" is wrong.

Baker-Gill-Solovay Thm:

There is an oracle, A, for which

$$P^A = NP^A$$

(very powerful)

and an oracle B s.t.

$$P^B \neq NP^B.$$

=

Say have proof $P \neq NP$ (or $P = NP$)

Proof: blah blah Turing blah blah diagonalize blah blah Turing - -

- -

A- PSPACE Complete Problem

$$NP^A \subset NPSPACE$$

Savitch

$$= PSPACE \subset P^A$$

=

Next Monday:

Review: SPACE, PSPACE

- -

Old Exam Problems