- TA evaluations
- Wed & Friday: Go over final 2014, pick a few other practice/old exam probs,

---

Terminology is [Sip]

P^A = poly time TM that calls oracle A

NP^A = " " non-det TM " " " A

M is T.M, M^A the oracle version Turing machine with oracle A

Final 2014: Problems 2 & 4 were homework

2

[32] **5.** 4 points per part. **Briefly justify** your answers; you will not get credit for just writing "yes" or "no" (or any short answer without justification).

(a) Show that if $C_1, C_2, \ldots$ are countable sets, then $C_1 \cup C_2 \cup \cdots$ is countable.

$C_1 = \{ c_{11}, c_{12}, c_{13}, c_{14}, \ldots \}$

$C_2 = \{ c_{21}, c_{22}, c_{23}, c_{24}, \ldots$

$C_3 = \{ c_{31}, c_{32}, c_{33}, \ldots$

Let $C_i = \{ c_{i1}, c_{i2}, \ldots \}$

Then: Count $C_1 \cup C_2 \cup \ldots$ as

c_{11}, c_{12}, c_{21} c_{31}, c_{22}, c_{13} etc.

OR: Count as groups via

$c_{ij} \longrightarrow i+j$ Count $i+j=2$, $i+j=3$, ... each group is finite.

(b) What does Savitch's theorem ~~assert~~ say, and why does this show that NPSPACE (non-deterministic polynomial space) is equal to PSPACE (polynomial space)?

1 Savitch's Thm says $\mathrm{NSPACE}(n^k) \subset \mathrm{SPACE}(n^{2k})$

or $\mathrm{NSPACE}(f(n)) \subset \mathrm{SPACE}(f(n)^2)$

2 $$\mathrm{NPSPACE} \overset{\text{by def}}{=} \bigcup_{k=1,2,\ldots} \mathrm{NSPACE}(n^k) \overset{\text{Savitch}}{\subset} \bigcup_{k=1,\ldots} \mathrm{SPACE}(n^{2k}) \overset{\text{by def}}{=} \mathrm{PSPACE}$$

(c) Let A be any problem that is complete for PSPACE under polynomial time reductions. Is PSPACE contained in P^A?

By definition: A is PSPACE-complete means

1 (1) $A \in$ PSPACE, (2) $L \in$ PSPACE then $L \leq_{\text{poly time}} A$.

SNEAKY-NTM

2 So $L \in$ PSPACE, then there is a poly time function to membership in A, and you can call the oracle for A once. So $L \in \mathrm{P}^A$.

(d) Let $L_{\text{NP easy}}$ consist of all descriptions of a triple, $\langle M, i, t\rangle$ where M is a non-deterministic Turing machine that accepts input i, running in time t where t is expressed in unary. Show that any language in NP can be reduced to $L_{\text{NP easy}}$ by a polynomial time reduction.

Continued on page 7

3

Marks

[10] **1.** Give a formal description of a Turing machine—and explain how your machine works—that recognizes the language

$$L = \{0^n 1 0^n \mid n \text{ is a non-negative integer}\}.$$

You should **explicitly write** your choice of $Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}$. You may use two tapes if you like. You may either write out the values of δ, or depict these values in a state diagram.

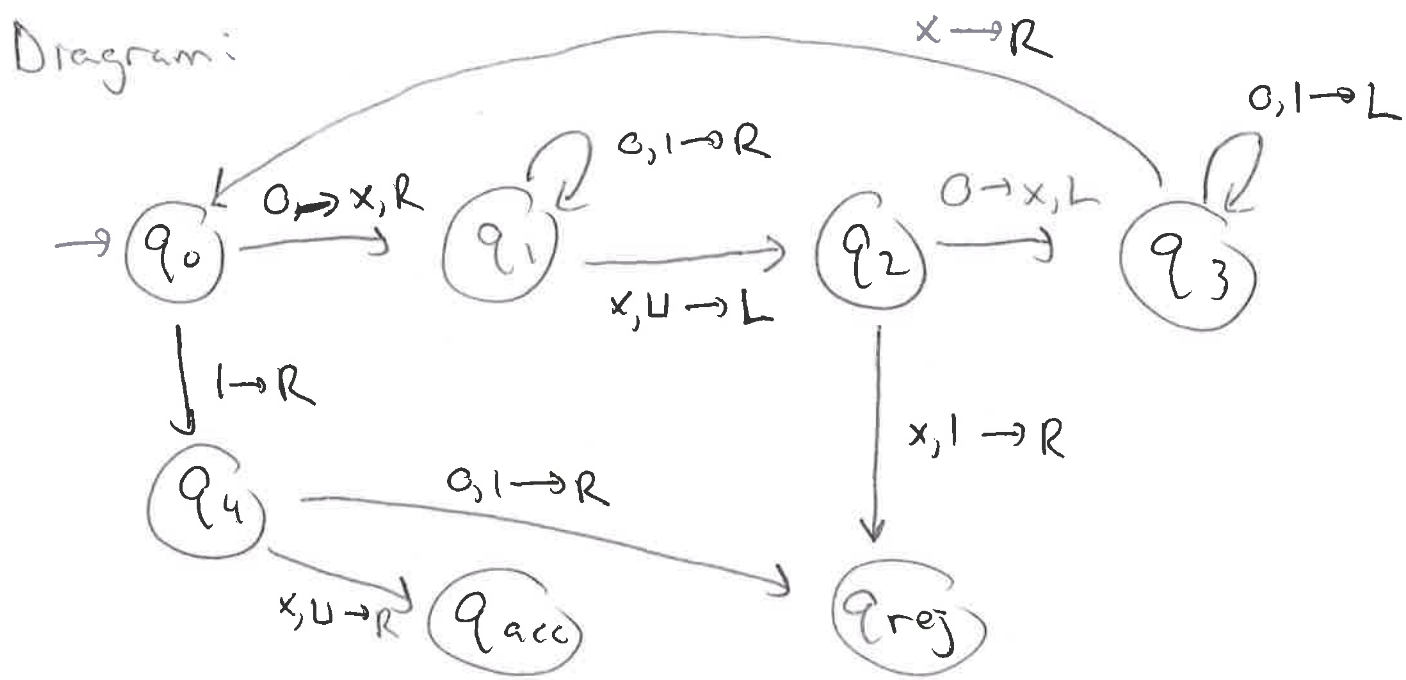
$L = \{0^n 1 0^n\}$

Continued on page 3

High-level idea:

If the first character is a "1" then accept if there is no more input. Otherwise mark the first and last character of the input as x (reject if they are not both 0's) and repeat (starting with the tape head at the start of what is left of the input.

Diagram:



q_0 = initial state: mark 0 with x and to q_1 or read 1 and to q_4

q_4 = accept if the 1 we just read is followed by a ⊔ or x reject otherwise

q_1 = keep moving R to end of input; to q_2 = mark 0 at the right end with x, move left via q_3 to start of input

$Q = \{q_0, q_1, q_2, q_3, q_4, q_{acc}, q_{rej}\}$, $\Sigma = \{0,1\}$, $\Gamma = \{0,1,x,\sqcup\}$