Marks

[8] **1.** Give a formal description of a Turing machine—and explain how your machine works—that takes as input, $x \in \{0,1\}^*$, and (1) accepts x if it has an even number of 1's, and (2) rejects x if not. You should **explicitly write** your choice of $Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}$. For example, you should write $\Sigma = \{0,1\}$, since this is the input alphabet.

Answer: Here is one algorithm (there could be variants of this): scan to the right, each time toggling states between q_0 **and** q_1 **when we see a** 1**; when we hit the first blank cell, we accept if we are in state** q_0**. Formally we have** $Q = (q_0, q_1, q_{\text{acc}}, q_{\text{rej}})$**,** $\Sigma = \{0,1\}$**,** $\Gamma = \{0,1,blank\}$**, and** δ **given by:**

toggle on a 1**:** $\delta(q_0, 1) = (q_1, ., R)$**,** $\delta(q_1, 1) = (q_0, ., R)$**, with** $.$ **irrelevant;**

no toggle on a 0**:** $\delta(q_0, 0) = (q_0, ., R)$**,** $\delta(q_1, 0) = (q_1, ., R)$**, with** $.$ **irrelevant;**

decide upon encountering a blank: $\delta(q_0, blank) = (q_{\text{acc}}, ., .)$**,** $\delta(q_1, blank) = (q_{\text{rej}}, ., .)$**, with** $.$ **irrelevant.**

Continued on page 3

[9] **2.**

(a) In class we showed that $|S| < |Power(S)|$ for any set S, where $Power(S)$ is the set of all subsets of S. We argued that otherwise there is a $f: S \to Power(S)$ such that each element of $Power(S)$ is in the image of f, and then we considered:

$$T = \{s \in S \mid s \notin f(s)\}.$$

How do we obtain a contradiction? Explain.

Answer: Since f is surjective, there exists a $t \in S$ such that $f(t) = T$. So either (1) $t \in T$, or (2) $t \notin T$; but we will now show that either of these situations is impossible. However, in case (1), by definition of T, we have $t \notin f(t)$; since $f(t) = T$ this means that $t \notin T$, which contradicts the assumption that $t \in T$. Similarly, in case (2) we have that $t \in f(t)$ and $f(t) = T$, contradicting that fact that $t \notin T$.

Hence our assumption that f is surjective leads to a contradiction; hence f is not surjective.

(b) Let $f: \{1,2,3\} \to Power(\{1,2,3\})$ be the function given by $f(1) = \{1,2\}$, $f(2) = \{1,3\}$, and $f(3) = \{1,2,3\}$. What is the set T constructed above?

Answer: We have $2 \notin f(2)$, so T contains 2, but $1 \in f(1)$ and $3 \in f(3)$ so T does not contain 1 or 3. Hence $T = \{2\}$.

(c) Using your answer to (b), explain why the proof in (a) is called "diagonalization."

Answer: One can think of T as being constructed from the diagonal of the relation of a grid with $1,2,3$ versus $f(1), f(2), f(3)$:

	$f(1)$	$f(2)$	$f(3)$
1	$\in$	$\in$	$\in$
2	$\in$	$\notin$	$\in$
3	$\notin$	$\in$	$\in$

Continued on page 4

[16] **3.** (4 points for each part) Briefly justify your answers **without references to nondeterministic Turing machines**:

(a) Let Σ be a fixed alphabet (i.e., a finite, nonempty set). Is the set of languages over Σ that are recognizable by a Turing machine countable or uncountable? Explain.

Answer: A Turing machine—when standardized (i.e., the set of states is of the form $\{1, \ldots, a\}$, and similarly for the set of tape symbols not in Σ)—can be described by a string over some finite alphabet. The set of strings over a finite alphabet are countable, and so each Turing recognizable language is recognized by one of countably many Turing machines. Hence the set of recognizable languages is countable.

(b) Is the set of (Turing-)recognizable languages closed under union? Explain.

Answer: Yes. Let L_1 and L_2 be recognized by Turing machines M_1 and M_2. Consider a Turing machine, M, that works as follows: given an input, M runs one step of M_1 and one of M_2, then two of M_1 and M_2, etc. If either M_1 and M_2 halts and accepts, then M accepts; if both M_1 and M_2 halt and reject then M rejects; otherwise M_1 and M_2 don't halt, and M doesn't halt. This M recognizes $L_1 \cup L_2$.

(c) Is a union of countably many countable sets necessarily countable? Explain.

Answer: If c_{ij} is the j-th element of set i (where i and j range over the integers, then we can list the elements of c_{ij} as

$$c_{11}; c_{12}, c_{21}; c_{13}, c_{22}, c_{31}; \ \ldots$$

(i.e., we run through the c_{ij} with $i+j$ equal to 2, then 3, then 4, etc.) Hence a countable union of countable sets is countable.

(d) Let L be a language decidable by a multi-tape Turing machine in time $n + 10$. For which constants a and b are we guaranteed that there is a one-tape machine that decides L in time at most an^b? Explain.

Answer: We claim that for $b = 2$ and any $a > 0$ we know that we can decide L on a one-tape machine in time an^b, but for any $b < 2$ (and any a) we cannot decide L in time an^b.

We know that we can simulate a machine that runs in time order n in time order n^2.

We also know that for any Turing machine, M, that takes time $f(n)$, and any integer, $r > 1$, we can form a new Turing machine M' that combines r tape cells

Continued on page 5

of M into a single tape cell of M' (which adds to the tape alphabet size and the number of states); we first need to read the input and convert it into the cells of M', which takes time $O(n)$. Hence for any $r > 0$ we can take a machine running in time $f(n)$ and form a new running in time $O(n)+f(n/r)$. We can also introduce extra states so that the machine halts immediately after reading a string of small size. It follows that if we can recognize a language in time (n^2), we can recognize the language in time an^2 for any real constant $a > 0$.

If L is the language consisting of words of the form wu where w is a palindrome over $\{0,1\}$ and u is a string of 2's with $|u| = 4|w|$, then we can recognize L in time $n+1$ on a three tape Turing machine (we use tapes two and three to copy w onto both as we move to the right, and then send, say, the head on tape two to the beginning, and then move tape head on tape two to the right while moving tape head three to the left). But we know that palindrome of length $|w| = n/5$ cannot be recognized in time order n^b for and $b < 2$, and hence the same is true of this L.

Continued on page 6

Continued on page 7

The End

Be sure that this examination has 6 pages including this cover

The University of British Columbia

Midterm Examinations - October 2015

Computer Science 421/501

Closed book examination

Time: 50 minutes

Name ____________________

Signature ____________________

Student Number____________________

Instructor's Name ____________________

Section Number ____________________

Special Instructions:

Calculators, notes, or other aids may not be used. Answer questions on the exam. This exam is two-sided!

Rules governing examinations

1. Each candidate should be prepared to produce his library/AMS card upon request.

2. Read and observe the following rules:

No candidate shall be permitted to enter the examination room after the expiration of one half hour, or to leave during the first half hour of the examination.

Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.

CAUTION - Candidates guilty of any of the following or similar practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.

(a) Making use of any books, papers or memoranda, other than those authorized by the examiners.

(b) Speaking or communicating with other candidates.

(c) Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

3. Smoking is not permitted during examinations.

1		8
2		9
3		16
Total		33