Computation, Machines and Formal Languages
(spring 2002)
Exam - moed B
Solution

Lecturer: Prof. Michael Ben-Or.

Time: 3 hours.

Write your answers on the exam sheet.

If the answer is yes/no, **write down the answer that you chose**.

Answer all the questions.

Unless stated otherwise, you have to add a short explanation (within the designated area for each question) for your answers. Answers that will not be accompanied by explanations will not be credited. The explanation should include the main steps in the proof, do not get into technical details, notations etc. If your explanation is given by a contradicting example, you have to state the example together with a short explanation of the contradiction. If you have an example that contradicts an unproven but a reasonable assumption (such as $P \neq NP$), state the example, the reasonable assumption, and a short explanation of the contradiction.

Definitions and Notations: (identical to those we used in class)

For a language L, we denote by $\overline{L}$, the language $\Sigma^* \setminus L$.

Complexity Classes:
REG is the class of regular languages.
CFL is the class of context-free languages.

R is the class of languages that are decidable by Turing Machines (TM).
RE is the class of languages that are recognizable by TM's.
P is the class of languages that are decidable by deterministic TM's in polynomial time.
NP is the class of languages that are decidable by non-deterministic TM's in polynomial time.
$PSPACE$ is the class of languages that are decidable by deterministic TM's in polynomial space.
L is the class of languages that are decidable by deterministic TM's in logarithmic space.
NL is the class of languages that are decidable by non-deterministic TM's in logarithmic space.
For a class of languages C, we denote by coC the class of languages that their complements are in C.

Computational problems:
$PATH = \{(G, s, t) : G$ is a directed graph, that contains a path from s to $t\}$
$CLIQUE = \{(G, k) : G$ is an undirected graph, and it contains a set of k vertices in which every two are adjacent$\}$
$STRONGLY-CONNECTED = \{G : G$ is a directed graph, where for every two vertices u, v, there is a path from u to v and vice versa$\}$
We say that a Boolean formula is a $k-cnf$ if it is of the form:

$$\bigwedge_i (a_1^i \vee a_2^i \vee \ldots \vee a_k^i)$$

where a_j^i is a variable or its negation.
$k-SAT = \{\phi : \phi$ is a satisfiable $k-cnf$ formula$\}$
$E_{TM} = \{< M >: M$ is a TM, and $L(M) = \emptyset\}$
$E_{NFA} = \{A : A$ is a nondeterministic finite automaton, and $L(A) = \emptyset\}$
$QBF = \{\phi : \phi$ is a true fully quantified Boolean formula$\}$

Reductions:
For two languages L_1, L_2, $L_1 \leq_P L_2$ denotes that there is a polynomial-time mapping reduction from L_1 to L_2.
For two languages L_1, L_2, $L_1 \leq_L L_2$ denotes that there is a logarithmic-space mapping reduction from L_1 to L_2.

1. Draw a **minimal** deterministic finite automaton for the following language (over $\Sigma = \{0, 1\}$):

$$L = \{w : \text{ the number of 1's in } w \text{ is a multiplication of } 3\}$$

Give a short proof of correctness and minimality.
(12 points)

Solution:

The solution is presented in Figure 1. The language L defines three equivalence classes, according to the number of 1's modulo 3 in the word. The automaton in Figure 1 is deterministic, and it is easy to see that it returns to the accepting state after reading three 1's, thus it accepts the language L. Since the number of states of the automaton is 3, it is minimal.

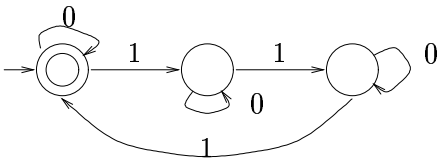

Figure 1: Minimal automaton for Question 1

2. For each one of the following languages, state the smallest class of languages that contains it from the list: REG, CFL or P. For languages in REG give a regular expression, for languages in CFL describe a context free grammar. (13 points)

   (a) $L_1 = \{0^n 1^n : n \geq 0\} \cup 0^*$

   Solution: CFL. L_1 is not in REG by Pumping Lemma: choose the word $0^{p+1}1^{p+1}$ and pump it once. The resulting word will have more 0's than 1's. From the other hand, it is easy to construct a context-free grammar that generates L_1: $S \to A|B$, $A \to 0A1|\varepsilon$, $B \to 0B|\varepsilon$.

   (b) $L_2 = \{1^{3n} : n \geq 0\}$

   Solution: REG. In fact, a slight modification of automaton from Question 1 (removing the edges labeled with 0's) gives a finite automaton for L_2.

(c) $L_3 = \{1^{3^n} : n \geq 0\}$

Solution: P. A TM that accepts L_3 first reads the input to verify that it contains only 1's. Then it divides the number of 1's each time by 3 (by crossing each third uncrossed 1). After each round it checks the number of 1's left. If it is 3, it accepts. If it is 1 or 2, rejects. L_3 is not in CFL by Pumping Lemma: let $w = 1^{3^p}$. By pumping w once we get a word $w' = 1^x$ where $3^p < x \leq 2 \cdot 3^p < 3^{p+1}$ (because the length of pumped parts is at most 3^p), thus x is not 3^y.

3. For each one of the following languages, state the smallest class of languages that contains it from the list: R, RE, $coRE$ or not in any of these classes. (13 points)

   (a) E_{NFA}
   Solution: R. Given an NFA A, we can check whether it is empty by checking whether there exists a reachable accepting state (we saw an algorithm in class). If yes - reject. Otherwise - accept.

   (b) E_{TM}
   Solution: $coRE$. First we show a reduction from $\overline{A_{TM}}$ to E_{TM}. Given M and w, we construct a machine M_w that on each input runs M on w. If M does not accepts w, then $L(M_w) = \emptyset$. If M accepts w, then $L(M_w) = \Sigma^*$. Thus, $\langle M, w\rangle \in \overline{A_{TM}}$ iff $\langle M_w\rangle \in E_{TM}$. Thus, $E_{TM} \notin RE$. From the other hand, $\overline{E_{TM}} \in RE$, because given M we can guess a word that it accepts and run M on this word.

   (c) $L = \{< M >: L(M)$ contains only words of even length$\}$
   Solution: $coRE$. First,

   $$\bar{L} = \{< M >: L(M) \text{ contains at least one word of odd length}\}$$

   is in RE, since we can guess such a word and run M on it. Now, $L \notin R$ by Rice theorem, and therefore $L \notin RE$ (because if L was in RE, then since $\bar{L}$ is in RE we would have that $L \in R$).

For each one of the languages that you answered R, state the smallest class that contains it from the list: L, NL or P.

Solution: $E_{NFA} \in NL$. Given an NFA A, we first modify it so that it would have only one initial state w_0 and one accepting state w_{acc} by adding

ε-transitions. Now, we have that $L(A) = \emptyset$ iff there is no path between w_0 and w_{acc} in A. Since $\overline{PATH} \in coNL = NL$, we have that $E_{NFA} \in NL$. From the other hand, it is unlikely that $E_{NFA} \in L$, since it would imply that $\overline{PATH} \in L$ and thus $L = NL$, which is not known. The reduction from $\overline{PATH}$ to E_{NFA} is straightforward: given a directed graph G and two nodes s and t, we regard G as NFA with the initial state s and accepting state t and transitions labeled with Σ.

4. For each one of the following functions, state if it is: $\theta(1)$, $\theta(\log n)$, or $\theta(n)$. (12 points)

   (a) $f_1(n) = \max\{|K(x) - K(x\sharp|x|)| : x \in \{0,1\}^n\}$
   The expression $x\sharp|x|$, is a string x concatenated with the length of x after a special symbol ($\sharp$).

   Answer: $\theta(1)$
   For any string x, let (M, y) be its minimal description. Define the machine M' that on every input runs M, and then concatenates to the output of M, the special symbol $\sharp$ followed by the length of the output in binary representation. Clearly, (M', y) is a description of $x\sharp|x|$, and $|(M', y)| \leq |(M, y)| + O(1)$.
   For the other direction, let (M, y) be the minimal description of $x\sharp|x|$. Consider the machine M' that on every input runs M, and then erases all the symbols that are left to the first $\sharp$ symbol in the output. Clearly, (M, y) is a description of x, and $|(M', y)| \leq |(M, y)| + O(1)$.

   (b) $f_2(n) = \max\{K(0^k) : 0 \leq k \leq n\}$

   Answer: $\theta(\log n)$
   For every k, $K(0^k) = O(\log k)$, because the description of the machine that receives a number in binary representation and outputs this number of zeros, together with k in binary representation, is a description of 0^k. Therefore $f_2(n) = O(\log n)$.
   To obtain a lower bound, suppose for contradiction that there is n, such that $f_2(n) \leq \frac{\log n}{2}$. So each string in $\{0, 0^2, \ldots, 0^n\}$, has a description of length less than $\frac{\log n}{2}$. But there are n such strings, and only $2^{\frac{\log n}{2}} = n^{\frac{1}{2}}$ descriptions of length $\frac{\log n}{2}$. So there must be a string in the list with description at least $\frac{\log n}{2}$. Therefore $f_2(n) = \Omega(\log n)$.

5. Are the following statements true or false.
   (13 points)

(a) $QBF \in NP \Rightarrow 3-SAT \in P$

Answer: No. If $QBF \in NP$, then $NP = coNP$, but this can coexist with $NP \neq P$.

(b) $\overline{PATH} \in NL \Rightarrow STRONGLY-CONNECTED \in L$

Answer: No. It is true that $\overline{PATH} \in NL$, and we also know that $STRONGLY-CONNECTED$ is NL-complete. Membership of $STRONGLY-CONNECTED$ in L would imply $L = NL$, which is not known (and not follows from the assumption).

(c) $3-SAT \in NL \Rightarrow CLIQUE \in coNP$

Answer: Yes. If $3-SAT \in NL$, then in particular $3-SAT \in P$, and thus $P = NP = coNP$, thus $CLIQUE \in P = coNP$.

6. For each one of the following languages, state the smallest class of languages that contains it from the list: NL, $PSPACE$ or P.
(13 points)

(a) $L = \{G : G$ is a directed graph with no (directed) cycles$\}$
Answer: NL. We show that $\bar{L} \in NL$, and since $NL = coNL$, claim follows. Given G, for each node s we call $PATH$ with $\langle G, s, s\rangle$. If $PATH$ accepts, accept (there is a path from s to s, which is a cycle). Otherwise, continue to the next node. If $PATH$ rejects for all s, reject.

(b) $2-SAT$ **Answer:** NL. Recall that check of whether a formula in $2-cnf$ is satisfiable can be performed by constructing a graph where each literal l is represented by a node, and there is a satisfying assignment iff there is a cycle in this graph. Since existence of a cycle is in NL, so is $2-SAT$.

(c) $3-SAT$
Answer: $PSPACE$, since $3-SAT \in NP \subseteq PSPACE$. It is not known whether $P = NP$, thus it is not known whether $3-SAT \in P$.

7. Let $A_0, A_1, A_2, \ldots$ be an infinite sequence of languages over alphabet Σ, such that for every $k \geq 1$, there is polynomial-time mapping reduction, f_k, from A_k to A_{k-1}. That is, $A_k \leq_P A_{k-1}$.
Let us denote:

$$B_r = \bigcup_{k=0}^{r} A_k \text{ and } B_\infty = \bigcup_{k=0}^{\infty} A_k$$

It is given that $A_0 \in P$.
For each one of the following languages, state the smallest class of languages that contains it from the list: P, NP, R, RE, or not in any of these classes. (12 points)

(a) A_k (for every $k \geq 0$)
Answer: P.

We showed in class that if $L \in P$ and $L' \leq_P L$ then $L' \in P$. We prove that A_k is in P, for every k, by induction. It is given that $A_0 \in P$. Assume that $A_{k-1} \in P$. It is given that $A_k \leq_P A_{k-1}$, therefore by the statement above, $A_k \in P$.

(b) B_r (for every $r \geq 0$)
Answer: P.

Above we showed that for every k, $A_k \in P$. Consider the following algorithm for B_r: on input x, run $M_0, \ldots, M_r$ on x, where M_i is the TM that decides A_i in deterministic polynomial time. If one of the M_i's accepts then accept. By definition, for $x \in B_r$, at least one M_i accepts (because $x \in A_i$ for some $0 \leq i \leq r$). And if $x \notin B_r$ then all the M_i's reject. Let n^c be the maximal running time of M_i (the maximum is taken over all the i's). Then the running time of the algorithm is at most $rn^c = O(n^c)$ (because r is a constant independent of the length of x).

(c) B_∞
Answer: not in RE.

Let L be any language not in RE (we saw a few examples for such a language). Define the language A_i to contain the single string 0^i if the string that represents i in binary representation is in L. Otherwise define A_i to contain the single string 1^i. Clearly, $A_k \in P$ for every k, because every A_k contains a single string. And for every k, $A_k \leq_P A_{k-1}$, because there is a polynomial time mapping reduction between any two languages in P. Now suppose that $B_\infty \in RE$. Then there is a machine M that recognizes B_∞. We describe an algorithm that recognizes L: on input x, let i be the number that x represents in binary. Run M on 0^i, if M accepts then accept. By definition M accepts if and only if $x \in L$. Therefore the algorithm recognizes L, but this contradicts the fact that $L \notin RE$.

8. Prove that for every language L in RE that contains infinite number of words, there is a language L' in R that also contains infinite number of

words, and $L' \subseteq L$.
Hint: Use enumerators.
(12 points)

Proof: We showed that if $L \in RE$ then there is an enumerator, E_L, that enumerates it. That is, E_L runs for infinity and every $x \in L$ is printed by E_L at some stage, while $x \notin L$ is never printed. Consider the following enumerator $E_{L'}$: run E_L, whenever it prints a string x, check if a string that comes after x in the lexicographic order was already printed by E_L. If yes, ignore x, otherwise print x and store it (to remember the highest string in the lexicographic order that was already printed). Since L contains infinite number of words, for every string x, at some stage, E_L prints a string that is higher than x (because there are only finite number of strings that are lower than x). Note that $E_{L'}$ prints its strings in a lexicographic order. Define L' to be the language that $E_{L'}$ prints. Clearly, L' is infinite, and $L' \subseteq L$ (because $E_{L'}$ prints only strings that E_L prints). We showed in class that a language is in R if and only if there is an enumerator that prints it in a lexicographic order. Therefore $L' \in R$.