

6.045 Final Exam Solutions

May 18, 2004

Susan Hohenberger

Name: _____

- Please write your name on each page.
- This exam is open book, open notes.
- There are two sheets of scratch paper at the end of this exam.
- Questions vary substantially in difficulty. Use your time accordingly.
- If you can not produce a full proof, clearly state partial results for partial credit.
- Good luck!

Problem	Points	Grade
1	40	
2	20	
3	20	
4	20	
5	20	
6	20	
7	20	
Total	160	

Problem 1: Multiple Choice Questions. (40 points, 4 points for each question)

For each question, any number of the listed answers may be correct. **Clearly** place an “X” in the box next to each of the answers that you are selecting.

1. Which of the following are known to be true?

- If language A is recognized by an NFA, then the complement of A must also be recognizable by an NFA.
- The complement of every Turing-decidable language is Turing-decidable.
- $NP = coNP$.
- $NL = coNL$.

2. Which of the following are true statements about the sizes of various kinds of representations of regular languages?

- Every language recognizable by a DFA with n states is recognizable by some NFA with n states.
- Every language recognizable by an NFA with n states is recognizable by some DFA with n states.
- Every language describable by a length n regular expression is recognizable by an $O(n)$ -state NFA.
- If two languages A, B are recognized by two (potentially different) DFAs with n states, then the language $A \cup B$ can be recognized by a DFA with at most $2n + 1$ states.

3. Which of the following languages are Turing-recognizable?

- $\{\langle M \rangle \mid M \text{ is a (deterministic) Turing machine and } M \text{ accepts } 010\}$.
- $\{\langle M \rangle \mid M \text{ is a nondeterministic Turing machine and } M \text{ accepts } 010\}$.
- $\{\langle M \rangle \mid M \text{ is a Turing machine and } M \text{ does not accept } 101\}$.
- $\{\langle M \rangle \mid M \text{ is a Turing machine and } L(M) = \Sigma^*\}$.

4. Which of the following languages can be shown to be undecidable by a direct application of Rice's theorem?

- $\{\langle M \rangle \mid M \text{ is a DFA and } M \text{ accepts } 010\}$.
- $\{\langle M \rangle \mid M \text{ is a Turing machine and } M \text{ accepts } 010\}$.
- $\{\langle M \rangle \mid M \text{ is a Turing machine and } M \text{ accepts } 010 \text{ and does not accept } 101\}$.
- $\{\langle M \rangle \mid M \text{ is a minimal Turing machine, that is, no Turing machine with a smaller representation recognizes the same language}\}$.

5. Which of the following are decidable relative to the Post Correspondence Problem (PCP)? (That is, which are decidable by an oracle Turing machine that uses an oracle for PCP?)
- The acceptance problem for oracle Turing machines relative to PCP.
 - A_{TM} , the acceptance problem for ordinary Turing machines.
 - The problem of whether $L(M)$ contains 010 and does not contain 101.
 - The emptiness problem (that is, does $L(M) = \emptyset$?) for ordinary Turing machines.
6. Which of the following are known to be true?
- $\text{CLIQUE} \leq_P \text{VERTEX-COVER}$.
 - $\text{CLIQUE} \leq_P \text{3SAT}$.
 - $\text{TQBF} \leq_P \text{HAMCYCLE}$.
 - $\text{PATH} \leq_P \{6045\}$.
7. Which of the following are known to be in NP?
- $L_1 - L_2$, for all L_1, L_2 in NP.
 - $L_1 \cap L_2$, for all L_1, L_2 in NP.
 - $\overline{A_{TM}}$, the complement of the acceptance problem for Turing machines.
 - $\overline{\text{PATH}}$.
8. Which of the following are known to be true statements about log space reducibility?
- Any log space transducer runs in polynomial time.
 - \leq_L is transitive.
 - If $A \leq_L B$ and $B \in \text{NL}$, then $A \in \text{NL}$.
 - For all languages A and B , if $A \leq_P B$, then $A \leq_L B$.

9. Which of the following are true statements about Savitch's theorem and its proof?

- Savitch's theorem implies that $\text{NSPACE}(\log n) = \text{SPACE}(\log n)$.
- Savitch's theorem implies that $\text{NSPACE}(n^2) \subseteq \text{SPACE}(n^4)$.
- In the proof of Savitch's theorem, when the simulating Turing machine computes CANYIELD recursively, it chooses the midpoint configuration nondeterministically.
- When the simulating Turing machine computes CANYIELD recursively, it uses space approximately equal to the sum of the space bounds used by the two recursive calls to CANYIELD, plus space to record the midpoint configuration.

10. Consider a language L , and a probabilistic polynomial time Turing machine M such that M always accepts words in L , and for any word w not in L , M rejects w with probability at least $1/10$. Which of the following must be true?

- $L \in \text{BPP}$
- $L \in \text{RP}$
- $L \in \text{coRP}$
- $L \in \text{NP}$

Problem 2: Regular Languages. (20 points) Provide solutions with brief justifications.

1. Find regular languages L_1, L_2 over $\{a, b\}$ for which $L_1 \not\subseteq L_2$, $L_2 \not\subseteq L_1$ (i.e., they are not equal and neither is a subset of the other), and $(L_1 \cup L_2)^* = L_1^* \cup L_2^*$.

Let $L_1 = \{a\}$, $L_2 = \{aa\}$, then neither is a subset of the other and $(L_1 \cup L_2)^* = L_1^* \cup L_2^* = a^*$.

2. Find a regular language L_1 and a non-regular language L_2 such that $L_1 \cap L_2$ is non-regular and yet $L_1 \cup L_2$ is regular.

Let $L_1 = \{0\}^*$, $L_2 = \{0^i \mid i \text{ is prime}\}$, then $L_1 \cap L_2 = L_2$ and $L_1 \cup L_2 = L_1$.

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Problem 3: Undecidability. (20 points) Let $L = \{\langle M \rangle \mid M \text{ is a basic Turing machine that accepts } 11 \text{ and does not accept } 00\}$. Use the Recursion Theorem to prove that L is undecidable. Fill in the blanks in the proof below.

Proof: For the sake of contradiction, assume that D is a decider Turing machine for L ; that is, D accepts $\langle M \rangle$ if M accepts 11 and does not accept 00, and D rejects $\langle M \rangle$ if M does not accept 11 or does accept 00.

Then define a new Turing machine R , as follows:

$R =$ "On input w ,

Obtain own description $\langle R \rangle$ via the Recursion Theorem.

Run D on input _____.

If this computation accepts then

accept w if _____ and

reject w if _____.

On the other hand, if this computation rejects then

accept w if _____ and

reject w if _____."

If R _____

then D _____,

which means R _____,

which is a contradiction.

On the other hand, if R _____

then D _____,

which means R _____,

which is again a contradiction. ■

Solution 3:

R : On input w :

Obtain $\langle R \rangle$

Run D on input $\langle R \rangle$.

If this computation accepts then

accept if $w = 00$

reject if $w \neq 00$

On the other hand, if this computation rejects then

accept if $w = 11$

reject if $w \neq 11$

If R accepts 11 and does not accept 00, then D on input $\langle R \rangle$ accepts, which means R accepts 00 and does not accept 11, which is a contradiction.

On the other hand,

If R does not accept 11 or does accept 00, then D on input $\langle R \rangle$ rejects, which means that R accepts 11 and R does not accept 00, which is again a contradiction.

Problem 4: Solitaire. (20 points) Consider the following solitaire game. You are given an $m \times k$ board where each one of the mk positions may be empty or occupied by either a red stone or a blue stone. Initially, some configuration of stones is placed on the board. Then, for each column you must remove either all of the red stones in that column or all of the blue stones in that column. (If a column already has only red stones or only blue stones in it then you do not have to remove any further stones from that column.) The objective is to leave at least one stone in each row. Finding a solution that achieves this objective may or may not be possible depending upon the initial configuration. Let

$$\text{SOLITAIRE} = \{ \langle G \rangle \mid G \text{ is a game configuration with a solution} \}.$$

Prove that SOLITAIRE is in NP.

Solution 4: It is easy to see that SOLITAIRE is in NP. The guess would be the (ordered) sequence of stones to be removed. Checking involves determining if each sequence is valid (i.e., that all the blue/red stones in a column have been removed), and determining if the resulting board has at least one stone in each row.

Prove that SOLITAIRE is NP-hard.

Solution 4: This problem smells like 3SAT !!

We reduce 3SAT to SOLITAIRE. Given a 3SAT instance with m clauses C_1, C_2, \dots, C_m and k variables x_1, x_2, \dots, x_k , we create an $m \times k$ SOLITAIRE board as follows:

- Each column of the board corresponds to a variable, and each row corresponds to a clause.
- For every cell (i, j) , place a red stone in the cell, if variable x_j occurs in clause C_i . Place a blue stone in the cell if $\overline{x_j}$ occurs in clause C_i . Else, place nothing on the cell.

We claim that the resulting solitaire board has a solution iff the 3SAT formula is satisfiable.

If the formula is satisfiable, we solve the solitaire board as follows: First remove the stones corresponding to all the literals that were set to false by the satisfying assignment. i.e, if in the satisfying assignment, x_j is set to true, then remove all the blue stones, otherwise remove all the red stones. It is clear that each column has either only red stones or only blue stones. Now, look at each row: we know that there is at least one literal in each clause that was set to true. Since we did *not* remove any stones corresponding to true *literals*, it follows that each row has at least one stone left (either red or blue).

Now, suppose there is a solution to the SOLITAIRE game. Consider each column – it has to consist of either only red stones or blue stones or neither. If column j has only red stones, set $x_j = true$. If it has only blue stones, set $x_j = false$, else set x_j to some value (we dont care what it is). Now, look at the i^{th} row. We know that it has at least one stone left. Let the stone be in the j^{th} column, and suppose the stone is red. This means clause C_i has literal x_j in it. But, since it is a red stone, we just set x_j to true. Thus clause C_i is satisfied. On the other hand, if the stone is blue, C_j has literal $\overline{x_j}$, and we just set x_j to false. Thus, again C_j is satisfied. It follows that the assignment we just constructed satisfies the formula. ■

Problem 5: NFA Equality. (20 points) Define EQ_{NFA} to be the equivalence problem for NFAs, that is,

$$EQ_{NFA} = \{(M_1, M_2) \mid M_1 \text{ and } M_2 \text{ are NFAs and } L(M_1) = L(M_2)\}.$$

Show that EQ_{NFA} is in PSPACE.

Solution 5: We show that EQ_{NFA} is in NPSpace. i.e, there is a non-deterministic TM M that uses polynomial space and decides EQ_{NFA} . Since $NPSpace = PSPACE$, we are done.

Consider the DFAs D_1 and D_2 corresponding to the NFAs M_1 and M_2 . (We will never actually construct them, since they are going to be exponentially larger). How do we test if D_1 and D_2 are equivalent? In a theorem we proved in the class, we showed that it is enough to test all strings of length up to $|D_1||D_2|$ to see if there is some such string which D_1 accepts, but D_2 does not (or vice-versa). Since $|D_1| \leq 2^{|M_1|}$ (the same for D_2), it is enough to test all strings of length upto $2^{|M_1|}2^{|M_2|}$.

Now, assume that M_1 and M_2 are not equivalent. Then, from the above discussion, there exists a string x of length at most $\ell = 2^{|M_1|+|M_2|}$ such that the machines M_1 and M_2 decide differently on x . Let $M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$. Our TM M does the following :

- Set $States_1 = \phi, States_2 = \phi$.
- For $i = 1$ to ℓ , do:
 - Guess the i^{th} bit x_i of the string x .

Note: i.e, If $x = x_1x_2 \dots x_\ell$, then at any point of time, the machine guesses just one of the bits x_i of x . Thus, M does not have to store all of x in its tape.
 - Compute $States_1 = \delta_1(States_1, x_i)$ and $States_2 = \delta_2(States_2, x_i)$.
 - If $States_1 \cap F_1 = \phi$ and $States_2 \cap F_2 \neq \phi$, (or) $States_2 \cap F_2 = \phi$ and $States_1 \cap F_1 \neq \phi$, then reject. Else continue to the next iteration of the for loop.

The correctness of this procedure, and the fact that it takes up only polynomial space are easy, and are left as exercise to the reader.

Problem 6: Right on Target. (20 points) We define the following language

TARGET = $\{\langle G, t \rangle \mid G \text{ is a directed graph, } t \text{ is a node in } G, \text{ and } t \text{ is reachable from every other node in } G \text{ via a directed path}\}$.

Show that TARGET is NL-hard.

Solution 6: To show that TARGET is NL-hard, we show that $\text{PATH} \leq_L \text{TARGET}$. Informally, the idea is to add directed edges from every node v in the graph to s . Let $G = (V, E)$. Convert an instance $\langle G, s, t \rangle$ of PATH to an instance $\langle G', t \rangle$ of TARGET, such that $G'(V', E')$ is such that $V' = V$, and $E' = E \cup \{(v, s) \mid v \in V\}$.

Now, if there is a path from s to t in G , there is certainly a path from every node to t in G' . Just use the new edges, in conjunction with the existing path from s to t . Conversely, if there is a path from all the nodes in G' to t , then, in particular, there is a path from s to t in G' . Wlog, assume that this is a simple path (that is, it traverses no vertex twice). Which means, it does not use any of the “new edges” (since all the new edges point to s). Thus, the same path is a path from s to t in G too. ■

Problem 7: Random World. (20 points) A language L has a probabilistic polynomial time Turing machine M that accepts words in L with probability at least $2/3$, rejects words not in L with probability at least $2/3$. Further, on any input w , M makes at most $\log_2(|w|)$ coin tosses (that is, in every computation path for input w , all but at most $\log_2(|w|)$ of the steps are deterministic).

Prove that $L \in P$.

Solution 7: If there are at most $\log_2(|w|)$ coin-tosses that the machine makes, then there are at most $2^{\log_2(|w|)} = |w|$ possible values for the coins. Simulate the machine on each of these coin-tosses. Tally the number of yes-answers and the number of no-answers. If the number of yes-answers is more than $\frac{2}{3}$ of the total number of answers, then accept w . If the number of no-answers is more than $\frac{2}{3}$ of the total number of answers, then reject w . This takes time $|w|T_M(|w|)$, which is polynomial in $|w|$ (here, $T_M(|w|)$ is the running time of M on w). Thus $L \in P$.

Note: What about the intermediate cases (for instance, when the number of yes-answers is equal to the number of no-answers) ? We *know* that such cases cannot arise, because of the property of M (that it either gives a large number of yes-answers, or a large number of no-answers, never an equal number of them). ■

END OF EXAM.

SCRATCH PAPER 1

SCRATCH PAPER

SCRATCH PAPER 2