CS 154 - Introduction to Automata and Complexity Theory

Spring Quarter, 2000

INFORMATION ON FINALS

Final Exam: FRI JUN 2, 2000 12:15-3:15 p.m. TCSEQ200

The final examination will be held from 12:15–3:15 pm on Friday, June 2 in TCSEQ 200. The finals will be **open-book** and **open-notes** exam, i.e., you will be allowed to consult both the text-book and your class notes. You are not permitted to refer to any other material during the exam.

Exam Syllabus: We will cover the entire course in the exam, with roughly 2/3 of the questions being on the material not already covered by the midterm. With regard to the theory of NP-completeness, there will be some general questions about basic definitions but no real problems to be solved.

Homework 6 Deadline and Solutions: While this homework is due on today, some of you may exercise your late credit to extend the deadline till Friday. For the latter, the homework is due by 5:00 pm on Friday. We will be handing out the solutions at that time on Friday, so the deadline is extremely strict. The solutions will be handed out in hard-copy as well as on the class web page.

Review Session: Tentatively, the review session for the final examination will be held from 6:00-7:30 on Thursday, June 1. The review session will be held in Gates B03. In case of any change, the details will be posted on the newsgroup and the web page for this class.

SITN Students: We expect the SITN students to come to campus for the finals. If this presents any problems, please let me know as soon as possible so we can make alternate arrangements.

Sample Final Exam

Instructions: Answer all 9 questions in the space provided. The questions are NOT organized in order of difficulty; if you cannot solve a problem, move on to the next one. You have a total time of 3 hours and the maximum possible score is 360. The point allocation is roughly proportional to the time required to solve a problem, i.e., 2 points for each minute. We strongly recommend that you use 10-15 minutes to go over the examination and plan your strategy. It is important that you be brief in your answers and ensure that it fits in the space provided. *You will lose points for a complicated solution, even if it is correct.*

This exam is an **open-book** and **open-notes** exam, i.e., you are allowed to consult both the text-book, your class notes and homeworks, and any of the handouts from us. You are not permitted to refer to any other material during the exam.

CS154N Students: You are required to solve problems 4–10 only for a maximum possible score of 240. You have 2 hours time to work on this exam.

Problem 1. [40 points] Decide if the following statements are TRUE or FALSE. *You do not need to justify your answer. You will receive 5 points for each correct answer, −2 (that is NEGATIVE 2) for each incorrect answer, and 0 points if you choose not to answer at all.* Think carefully before answering.

1. A PDA with two stacks can recognize any recursively enumerable language.
2. If L_1 is regular and L_2 is context-free, then $L_1 - L_2$ is context-free.
3. If L is not context-free and F is finite language, then $L - F$ is not context-free.
4. Let M be a PDA whose stack has size bounded by k, for some constant k. Then, $L(M)$ must be regular.
5. The language $\{a^i b^j a^i b^j \mid i, j \geq 0\}$ is context-free.
6. The union of a context-free and a regular language must be context-free.
7. The context-free grammar $S \rightarrow a \mid ab \mid Sa \mid SbS$ is ambiguous.
8. Every regular language without ϵ has a context-free grammar in the Chomsky normal form.

Problem 2. [40 points] Consider the PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where $Q = \{q_0, q_1, q_2, q_3, f\}$, $\Sigma = \{a, b\}$, $\Gamma = \{Z_0, A, B\}$, $F = \{f\}$, and δ defined as follows:

$$\begin{aligned}
\delta(q_0, a, Z_0) &= \{(q_1, AAZ_0)\} \\
\delta(q_0, b, Z_0) &= \{(q_2, BZ_0)\} \\
\delta(q_0, \epsilon, Z_0) &= \{(f, \epsilon)\} \\
\delta(q_1, a, A) &= \{(q_1, AAA)\} \\
\delta(q_1, b, A) &= \{(q_1, \epsilon)\} \\
\delta(q_1, \epsilon, Z_0) &= \{(q_0, Z_0)\} \\
\delta(q_2, a, B) &= \{(q_3, \epsilon)\} \\
\delta(q_2, b, B) &= \{(q_2, BB)\} \\
\delta(q_2, \epsilon, Z_0) &= \{(q_0, Z_0)\} \\
\delta(q_3, \epsilon, B) &= \{(q_2, \epsilon)\} \\
\delta(q_3, \epsilon, Z_0) &= \{(q_1, AZ_0)\}
\end{aligned}$$

(a). Give an execution trace (using instantaneous descriptions) of the PDA M showing that input string bab is in $L(M)$.
(b). Give an execution trace (using instantaneous descriptions) of the PDA M showing that input string abb is in $L(M)$.
(c). Describe the contents of the stack after the PDA has scanned b^7a^4 from the input.
(d). Describe the *final state* language $L(M)$ in English.

Problem 3. [40 points] Decide whether the following language over $\Sigma = \{0, 1\}$ is a context-free language or not. Prove your answer by either specifying a context-free grammar or using the pumping lemma for context-free grammars. $L = \{\, ww \mid w \in \Sigma^* \,\}$.

Problem 4. [40 points] Consider the following classes of languages numbered as **1–7**.

1. Empty.

2. Finite.

3. Regular.

4. Context-free.

5. Recursive.

6. Recursively enumerable.

7. All possible languages.

For each of the following languages specify the *lowest-numbered* class to which it belongs; you *do not* need to provide any justification for your answers. For example, for a context-free language L that is not regular, the right answer is 4 (even though L clearly belongs to all the classes of number larger than 4). Similarly, suppose we know that L is recursively enumerable and that, although it could possibly be recursive, the available information does not guarantee that it is recursive, then the right answer is 6.

1. The complement of a recursively enumerable language.
2. The intersection of two recursive languages.
3. The intersection of two recursively enumerable languages.
4. The set of strings rejected by a DPDA (deterministic PDA).
5. The set of strings accepted by a Turing machine that does not halt on any input.
6. A language that is accepted by both a PDA and a Turing machine.
7. The language of a regular expression that does not use the $*$ operation.
8. The language of a nondeterministic Turing machine (NTM).

Problem 5. [30 points] Decide if the following statements are TRUE or FALSE and give a *brief justification* of your answer.

(a). Let $L = \{w\}$ be a language containing only *one* string w where $w = 000$ if *God exists* and $w = 111$ if *God does not exist.* Then, L is decidable. (*Remark:* The answer to this question does not depend on your religious convictions!)

(b). Suppose that L is recursively enumerable but not recursive. Then, any Turing machine for L must fail to halt on an infinite number of strings.

(c). Let $L_1, L_2, \ldots$ be an *infinite* sequence of languages such that for $1 \leq i \leq \infty$, each L_i is recursively enumerable. Then, the infinite union of these languages $L_1 \cup L_2 \cup \ldots$ is also recursively enumerable.

(d). The complement of a context-free language need not be recursively enumerable.

(e). It is possible that an undecidable language is NP-complete.

Problem 6. [30 points] Recall the *halting problem* as defined in the following language:

$$L_H = \{<M, w>|\text{ Turing machine } M \text{ halts on input } w\}.$$

Show that if L_H is recursive then *every* recursively enumerable language must also be recursive. (*Hint:* You need to show that given an algorithm for L_H, you can construct an algorithm for any recursively enumerable language L.)

Problem 7. [30 points] For any two languages L_1 and L_2, define the *difference* language $L_1 - L_2$ as the set of strings in L_1 that *do not* belong to L_2.
(a). Suppose L_1 and L_2 are both *recursive*. Show that $L_1 - L_2$ must be *recursive*.
(b). Suppose L_1 is *recursive* and L_2 is *recursively enumerable*. Show that $L_1 - L_2$ need not be *recursively enumerable*. (You need to identify specific examples of L_1 and L_2 for this purpose.)
(c). Suppose L_1 is *recursive* and L_2 is *recursively enumerable*. Classify $L_2 - L_1$ as being one of the following: *recursive*, *recursively enumerable but need not be recursive*, or *need not be recursively enumerable*. Show that your answer is correct.

Problem 8. [40 points] Consider the following two languages over the alphabet $\Sigma = \{0, 1\}$.

$$L_H = \{<M, w>|\text{ TM } M \text{ halts on input } w\}$$

$$L_{ODD} = \{<\widehat{M}>|\ L(\widehat{M}) \text{ contains a string of odd length}\}$$

The language L_H corresponds to the following decision problem: given a Turing machine M and an input string w, does M halt on input w? Similarly, the language L_{ODD} corresponds to the following decision problem: given a Turing machine $\widehat{M}$, does $L(\widehat{M})$ contain at least one string of odd length?

Your goal is to show that L_{ODD} is undecidable, assuming that L_H is undecidable. We propose the following reduction from L_H to L_{ODD}.

> Given a Turing machine M and input w, construct a Turing machine $\widehat{M}$ which behaves as follows on being given input $\widehat{w}$.
>
> 1. $\widehat{M}$ simulates the behavior of M on input w;
> 2. if M halts on w, then $\widehat{M}$ determines whether its input $\widehat{w}$ has odd or even length, halting in a final state if $\widehat{w}$ has an odd length and halting in a non-final state otherwise.

(a). In the case where M does not halt on w, what is $L(\widehat{M})$?
(b). In the case where M does halt on w, what is $L(\widehat{M})$?
(c). Using the reduction describe above, show that L_{ODD} is undecidable.

Problem 9. [60 points] Consider the following languages.

$$\begin{aligned}
L_B &= \{<M>|\text{ TM } M \text{ accepts on a blank input tape}\} \\
L_1 &= \left\{<M> \;\middle|\; \begin{array}{l}\text{TM } M \text{ starting with a blank input tape will} \\ \text{eventually write the symbol 1 on its tape}\end{array}\right\} \\
L_* &= \left\{<M> \;\middle|\; \begin{array}{l}\text{TM } M \text{ starting with a blank input tape will} \\ \text{eventually write a non-blank symbol on its tape}\end{array}\right\}
\end{aligned}$$

You may assume that all Turing machines provided as input are *deterministic*.
(a). [30 points] Prove that L_* is decidable.

(b). [30 points] Assuming that L_B is undecidable, show that L_1 is undecidable.