Mid-Term Exam
Due by 11:15am on Feb 8, 2012

Prof. Amit Chakrabarti
Computer Science
Dartmouth College

Submission Instructions

- Submit exactly like a homework. Late penalties apply in the same way as for a homework.
- Please write or type your solutions neatly and **staple together your sheets of paper**. The course staff is not responsible for sheets lost due to lack of stapling.

Honor Code

- **You must work on the exam alone**; you may not collaborate with anyone.
- You may consult your textbook (Sipser), your notes, and anything posted on the CS 39 website. Consulting anything else (e.g., last year's notes, your friend's notes, other websites) is a violation of the Honor Code.
- You may speak to others about the exam only in *complete generality* (e.g., "The exam is hard", "I'm almost finished with the exam", "I'll be working on the exam tonight"). You may not speak about the exam in any detail whatsoever (e.g., "Problem 3 is hard", "Problem 5 is easy", "That pumping lemma problem is tough").
- Since this is an exam, we (the TAs and the instructor) cannot help you with the particular problems on it, nor can we check if you are on the right track with a problem.

Important General Instructions

- You may use, without proof, any results proven in class, in a homework, or in the textbook. Simply cite the result you are using.
- Please read each question carefully. If you misread and answer a different question than the one asked, unfortunately, you will not get credit.
- **Good luck!**

Part 1: Short Answers

For the problems in this part, you don't have to prove anything. Just answer these questions as briefly and directly as possible.

1. Consider the language $L_1 = (a^3)^* \cup (a^5)^* = (aaa)^* \cup (aaaaa)^*$.

1.1. Describe a DFA that recognizes L_1 and has as few states as possible. You may either draw the DFA or specify it formally. [5 points]

1.2. Draw an NFA that recognizes L_1 and has six fewer states than the above DFA. [5 points]

2. Let $L_2 \subseteq \{a, b, c\}^*$ be the language generated by the following context-free grammar (CFG):

$$\begin{aligned} S &\longrightarrow AB \\ A &\longrightarrow AA \mid ab \\ B &\longrightarrow CA \\ C &\longrightarrow c \mid \varepsilon \end{aligned}$$

2.1. Give a simple regular expression for L_2. [5 points]

2.2. Show that the above CFG is ambiguous. [5 points]

2.3. Give an unambiguous CFG that generates L_2. [5 points]

3. Design a CFG that generates $\{0^m 1^n : m \neq n \text{ and } 2m \neq n\}$. Explain the logic behind your CFG at a high level, by describing the roles of the key variables. [10 points]

4. Draw a PDA that recognizes $\{w_1 \# w_2 : w_1, w_2 \in \{0,1\}^* \text{ and } w_1 \neq w_2\}$. Note that this language is over the alphabet $\{0, 1, \#\}$. Explain, at a high level, how your PDA works. [10 points]

5. Let $M = (P, \Sigma, \Gamma, \delta_P, p_0, F_P)$ be a PDA and let $D = (Q, \Sigma, \delta_D, q_0, F_D)$ be a DFA over the same alphabet Σ. Formally describe a PDA that recognizes $\mathcal{L}(M) \cap \mathcal{L}(D)$. No justification required. [10 points]

Part 2: Problems on Proof

The problems in this part require you to write proofs. If your proof involves constructing an automaton (DFA, NFA, or PDA), you *must* specify it formally and you *must* give some justification that your construction is correct, though this justification need not be strictly formal.

6. The *balanced concatenation* of the languages A and B, denoted $A \odot B$, is defined as follows.

$$A \odot B \;=\; \{xy : x \in A, y \in B, \text{and } |x| = |y|\}.$$

Suppose A and B are regular languages over some alphabet Σ.

6.1. Is it always the case that $A \odot B$ is regular? Prove your answer. [5 points]

6.2. Is it always the case that $A \odot B$ is context-free? Prove your answer. [10 points]

7. Let L_{Java} denote the language consisting of all syntactically correct Java programs. Is L_{Java} a regular language? Prove your answer, and take care to ensure that you are giving a rigorous proof. (If you are not familiar with the Java programming language, you may instead use any other general purpose prorgamming language, e.g., C, C++, Python, Haskell, ...) [10 points]