CS 154 - Introduction to Automata and Complexity Theory

Spring Quarter, 2001

Information on Midterm Exam

Midterm Exam. The midterm examination will be held from 3:15–4:30 pm on Monday, May 7.

> **Please note that the exam will be held in Kresge Auditorium, and not in the regular lecture room.**

You need to be present and seated by 3:10 pm so that we can distribute the exam in an orderly fashion. At exactly 4:30 pm, when the TAs announce that the time is up, everyone should stop writing. Those who continue writing will be penalized.

The exam will be open-book and open-notes, i.e., you will be allowed to consult any of the class handouts, your notes, and the textbook. You are not permitted to refer to any other source during the exam. The exam will cover the material presented in class up to Wednesday, May 2.

SITN Students. If you are an SITN student, we strongly recommend that you come to campus for the exam. In case you are unable to do so, please send an email to Ananth and Rajeev with your contact information (email, phone, and fax). We will either email or fax the exam to you around the time it begins.

Special Office Hours for Midterm. The TAs will hold special office hours for the midterm exam from 3pm to 7pm on Sunday, May 6 in Gates 100.

Homework 3 Solutions. Solutions to Homework 3 will go up on the course web page late Friday evening or early Saturday morning.

CS 154N Students

We expect to start discussing the topics from CS 154N, starting with Turing machines, around Monday, May 14. We suggest that you start attending the lectures starting now.

SAMPLE MIDTERM EXAM

The following is a sample midterm exam along with the solutions.

Problem 1. [25 points] Each of the following statements concerns languages over a finite alphabet Σ. Decide if these statements are TRUE or FALSE. To receive any credit you must provide a brief justification or a counter-example.

(a). [5 points] Non-regular languages are closed under intersection.

False. Consider a non-regular language L_1 and its complement, L_2. The intersection of L_1 and L_2 is the empty language, which is regular.

(b). Suppose L_1 is finite and $L_1 \cup L_2$ is regular. Then L_2 must be regular.

True. This would not necessarily hold if L_1 were regular, but the more restrictive condition that L_1 is finite allows the above statement.

(c). [5 points] The class of ϵ-NFAs which do not allow ϵ-moves from a state to itself can accept any regular language.

True. An ϵ transition from a state to itself is useless and may be removed without changing the language of the FA.

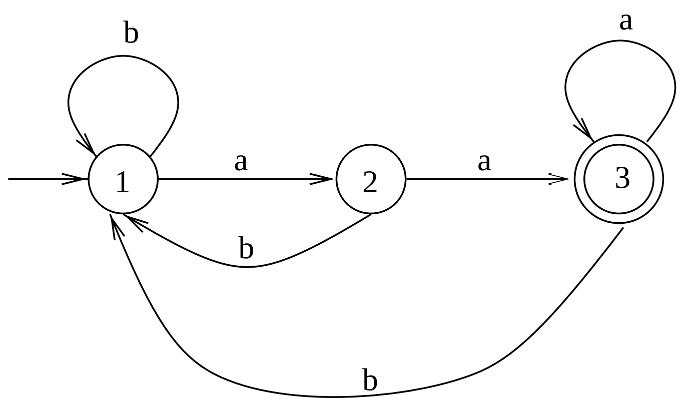
(d). [5 points] The language of an ϵ-NFA can contain ϵ if and only if $q_0 \in F$.

False. An ϵ-NFA may have an ϵ transition from q_0 to a final state.

(e). [5 points] In a Chomsky Normal Form grammar, the sentential form $aBcDe$ must have a derivation sequence of length exactly 9.

False. The derivation of the above sentential form is of length seven.

Problem 2. [20 points] Recall the construction given in class for converting a DFA into a regular expression. Given a DFA, we had defined the notion of a regular expression $R_{ij}^{(k)}$ for the language $L_{ij}^{(k)}$ consisting of all strings that go from state q_i to state q_j without visiting a state numbered larger than k along the way.



For the DFA given above, specify the following regular expressions. You do not need to justify your answer and are free to use any method to determine the answer (including "reasoning it out").

a) $R_{11}^{(0)}$ $\qquad\epsilon + b$

b) $R_{21}^{(1)}$ $\qquad b^{+}$

c) $R_{21}^{(2)}$ $\qquad b(b + ab)^{*}$

d) $R_{21}^{(3)}$ $\qquad (a + b)^{*}b$

e) $R_{13}^{(3)}$ $\qquad (a + b)^{*}aa$

Problem 3. [20 points] Prove that the following language is non-regular.

$$L = \{0^i 1^j 2^k \mid k \geq i + j\}$$

Suppose L is regular. Then there is a constant n satisfying the PL condition. Choose $w = 0^n 2^n$, $w \in L$ ($i + j = n + 0 \geq n = k$) and $|w| \geq n$. Then $w = xyz$, where $|xy| \leq n$ and $y \neq \epsilon$. This means y consists of solely zero's. By PL, $xy^2z \in L$. However, $xy^2z = 0^{n+|y|}2^n$ and $i + j = n + |y| > n = k$, which is not in L. This is a contradiction, so L is not regular.

Problem 4. [25 points] A *simple grammar* is a context-free grammar $G = (V, T, P, S)$ in which all productions are of the following two types:

$$\begin{aligned} A &\rightarrow bB \\ A &\rightarrow \epsilon \end{aligned}$$

for A, $B \in V$ and $b \in T$. You will sketch the main idea behind the following claim: *a language is regular if and only if it has a simple grammar.*

(a). Given a simple grammar G, explain how you can construct an NFA M such that

$L(M) = L(G)$. Just explain your construction — a proof is not required.

Let the grammar G be (V, T, P, S). We will construct an NFA M as the following:

$M = (Q, \Sigma, \delta, q_0, F)$, where $Q = V$, $\Sigma = T$, $q_0 = S$, $F = \{B \in V \mid B \to \epsilon \text{ is in } G\}$.

The transitions of the NFA is the following:

$\delta(A, b) = \{B \mid A \to bB \text{ is in } G\}$.

(b). In your construction for (a), when will M be a DFA?

When in the production rules, we don't have the following:

$A \to bB$ and $A \to bC$ for some A,B,C,b.

(c). Given an NFA M, explain how you can construct a simple grammar G such that $L(M) = L(G)$. Just explain your construction — a proof is not required.

Given $M = (Q, \Sigma, \delta, q_0, F)$, construct the grammar $G = (V, T, P, q_0)$, where $V = Q$, $T = \Sigma$, and P consists of the following:

For each transition $\delta(A, b) = B$, add rule $A \to bB$. For each final state $F_i \in F$, add rule $F_i \to \epsilon$.

(d). In your construction for (c), how would the resulting grammar change if M were an ϵ-NFA?

We would have added the rules of the form $A \to B$.

(e). What class of languages is generated by simple context-free grammars which are allowed, in addition, productions of the type $A \to a$? Give a brief justification of your answer.

Still regular expressions. Because rules of the type $A \to a$ can be written as the following: $A \to aB$ and $B \to \epsilon$.

Problem 5. [20 points] Consider the following language over the alphabet $\{a, b\}$

$$L = \{w \mid \text{every prefix of } w \text{ has no more } a\text{'s than } b\text{'s}\}$$

You are to construct a PDA $M = (Q, \Sigma, , , \delta, q_0, Z_0, \phi)$ such that $N(M) = L$. The starting point for your construction will be the following three transitions:

1. $\delta(q_0, b, Z_0) = (q_0, BZ_0)$.

2. $\delta(q_0, b, B) = (q_0, BB)$.

3. $\delta(q_0, a, B) = (q_0, \epsilon)$.

(a). [4 points] Explain what these three transitions do.

Rule 1: when in state q_0 and the input symbol is b and the top symbol is Z_0 on the stack, stay in state q_0, leave Z_0 on the stack, and in addition, push symbol B onto the stack.

Rule 2: when in state q_0 and the input symbol is b and the top symbol is B on the stack, stay in state q_0, leave B on the stack, and in addition, push symbol B onto the stack.

Rule 3: when in state q_0 and the input symbol is a and the top symbol is B on the stack, stay in state q_0, pop B off the stack.

In high level description, rule 1 and 2 push an extra "B" onto the stack once it sees a "b", and rule 3 pops a "B" off the stack (if there is any) once it sees an "a".

(b). [10 points] Write all the additional transitions needed to completely define the machine M.

$\delta(q_0, \epsilon, Z_0) = (q_0, \epsilon)$

$\delta(q_0, \epsilon, B) = (q_0, \epsilon)$

(c). [6 points] Provide a context-free grammar $G = (V, T, P, S)$ such that $L(G) = L$.

$S \rightarrow bAS|\epsilon$

$A \rightarrow a|bAA|\epsilon$