

CS341 Automata Theory (Fall 2003)
First Midterm Review Session (October 6, 2003)
Luay K. Nakhleh (nakhleh@cs.utexas.edu)

IMPORTANT:

- *The following is a list of problems that I compiled to help you practice the material for the first midterm. The sole purpose of this handout is for practice and it is not to be viewed as a representative of what's going to be on the exam, neither in content nor in format. Feel free to ask me any questions about the material in general, and the questions listed below in particular. However, any questions about the exam/grading/policies or anything else related to the structure of the course should be addressed directly to your instructor.*
- *Some problems are challenging. I added those problems for those of you who are interested in more challenging problems than what they've already seen. You can safely skip those problems, if you are not interested.*
- *I will do my best to follow the same style and terminology that your instructors use. However, if at some point the style or terminology I use differ from what your instructor uses, it is your duty to ask me for clarifications.*
- *We all make mistakes. If I write/say something that does not sound right to you, you are not convinced in, or your instructor had written/said it differently, make sure you ask.*
- *Your exams are graded by your instructors and TA's, so your writing style (especially the proofs) should conform with what they expect from you.*
- *I may sketch the solutions of some problems due to time constraints during the review session. You should be able to write the full solutions, since that's what may be expected from you on an exam.*

The Problems:

1. For each of the following languages, state whether or not the language is regular and prove your answer:
 - $L_1 = \{xy : x, y \in \{0, 1\}^*, \text{ and } \#_0(x) = \#_0(y)\}$.
 - $L_2 = \{xy : x, y \in \{0, 1\}^*, \text{ and } \#_0(x) = \#_1(y)\}$.
 - $L_3 = \{x2y : x, y \in \{0, 1\}^*, \text{ and } \#_0(x) = \#_1(y)\}$.
 - $L_4 = \{a^i b^j c^k : i, j, k \geq 0, \text{ and } i + j = k\}$.
 - $L_5 = \{x = y + z : x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z\}$.
 - $L_6 = \{a^i b^j c^k : i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k\}$.
 - $L_7 = \{w \in \{0, 1\}^* : w \text{ contains an equal number of occurrences of the substrings } 01 \text{ and } 10\}$.
 - L_8 is the set of binary representations of all non-negative integers which are the powers of 4.
 - L_9 is the set of binary representations of all positive integers which are congruent to zero modulo 5.
 - L_{10} is the set of binary representations of all positive integers whose length is congruent to zero modulo 5.
 - $L_{11} = \{ww^R : w \in \{a, b\}^+\}$.
 - $L_{12} = \{ww^R x : w \in \{a, b\}^+, x \in \{a, b\}^*\}$.
 - $L_{13} = \{a^p : p \text{ is prime}\}$.
 - $L_{14} = \{a^{n^2} : n \geq 0\}$.
 - $L_{15} = \{a^{n!} : n \geq 0\}$.
 - $L_{16} = \{a^{2^n} : n \geq 0\}$.
 - $L_{17} = \{w \in \{0, 1\}^* : \#_0(w) \neq \#_1(w)\}$.
 - $L_{18} = \{a^i b^j c^k : i, j, k \geq 0, i \neq j \text{ or } i \neq k \text{ or } j \neq k\}$.
 - $L_{19} = \{0^p 1^q 0^m 1^n : p + q = m + n, p, q, m, n \geq 0\}$.
 - $L_{20} = \{0^m 1^n : m, n \geq 0, m \neq 2n + 1\}$.
 - $L_{21} = \{0^m 1^n : 2n \leq m \leq 3n, m, n \geq 0\}$.

- L_{22} is the set of all strings w over $\{0, 1\}$ such that no 0 in w is followed by a 1.
- $L_{23} = \{w : \exists y \in \{a\}^* \text{ s.t. } w = yy\}$.
- L_{24} is the set of social security numbers of all residents in the United States.
- $L_{25} = \{w \in \{a, b\}^* : \forall x \in \{a, b\}^*, xw = w\}$.
- $L_{26} = \{xyx : x, y \in \{a, b\}^*\}$.
- $L_{27} = \{xyx^R : x, y \in \{a, b\}^+\}$.
- $L_{28} = \{xyx : x, y \in \{a, b\}^+\}$.
- L_{29} is the set of all palindromes over $\{a, b\}^*$.
- $L_{30} = \{xyzzy^R x : x, y, z \in \{a, b\}^*\}$.
- $L_{31} = \{a^i b^j : 0 \leq i < j < 2^{100}\}$.
- $L_{32} = \{a^i b^j : 0 \leq i < j\}$.
- $L_{33} = \{0^n 10^n : n \geq 1\}$.
- $L_{34} = \{w1^n : w \in \{0, 1\}^* \text{ and } |w| = n\}$.
- $L_{35} = \{w : w \in \{0, 1\}^* \text{ and between every two occurrences of 1, there are at least three occurrences of 0}\}$.
- $L_{36} = \{w : w \in \{0, 1\}^* \text{ and there are two 0's in } w \text{ that are separated by } 3i \text{ 1's, for some } i \geq 0\}$.
- $L_{37} = \{a^n b^m : n + m \equiv_3 2\}$.
- $L_{38} = \{ab^n c^m a^k : n > 0, m > 0, k > \min(n, m)\}$.
- $L_{39} = \{w : w \in \{a, b\}^* \text{ and for every prefix } x \text{ of } w, \text{ if } |x| > 0 \text{ and } |x| \text{ is even, then the last symbol of } x \text{ is 1}\}$.
- $L_{40} = \{w : w \in \{a, b\}^* \text{ and } w_i = a \text{ for every } i (\leq |w|) \text{ of the form } 2^j, \text{ where } j \geq 0\}$.

2. We can extend the definition of the transition function δ of DFA's to be of the form

$$\delta : K \times \Sigma^* \rightarrow K$$

where $\delta(p, w) = q$, for $p, q \in K$ and $w \in \Sigma^*$, if when started from state p , the DFA ends in state q after reading string w .

Given a DFA M , for each pair of states p, q , we define the language

$$L(M, p, q) = \{w : \delta(p, w) = q\}.$$

Prove or disprove the following:

- If $x \in L(M, p, q)$ and $y \in L(M, q, r)$, then $xy \in L(M, p, r)$.
- If $x \in L(M, p, p)$, then for every $y, z \in \Sigma^*$, and for every natural number i , we have

$$yxz \in L(M) \Rightarrow yx^i z \in L(M).$$

3. COMMON MISTAKES

- After having proved in class that the language $\{a^n b^n : n \geq 0\}$ is not regular, student X proves that the language

$$L = \{w \in \{a, b\}^* : \#_a(w) = \#_b(w)\}$$

is not regular as follows:

“Assume L is regular, and let N be the constant guaranteed by the Pumping Lemma. Choose the string $a^N b^N$. Clearly, the string is in the language L and its length is $\geq N$. We saw in class that $a^n b^n$ is not regular. Therefore, L is not regular.”

Is the proof that student X presented correct?

- (b) Student Y argues that the language $L = \{a^n b^n c^n : n \geq 0\}$ is regular, since L is the concatenation of the three languages $\{a^n : n \geq 0\}$, $\{b^n : n \geq 0\}$, and $\{c^n : n \geq 0\}$. Since each one of those three languages is regular and regular languages are closed under concatenation, it follows that L is regular.
Is the argument that student Y presented correct?
- (c) After having proved in class that the language $L = \{a^p : p \text{ is prime}\}$ ($\Sigma = \{a\}$) is not regular, student X proves that $L' = \{a^p : p \text{ is not prime}\}$ is not regular as follows:
“Since $L' = \overline{L}$, L is not regular, and non-regular languages are closed under complement, it follows that L' is not regular.”
Is the proof that student X presented correct?
- (d) Student X claims that if L_1, L_2, \dots (infinite set) are all regular languages, then so is $L_1 \cup L_2 \cup \dots$, since regular languages are closed under union. Is the claim of student X correct?
- (e) The Pumping Lemma states that for a regular language L , certain strings w can be written as xyz such that, under certain conditions, $xy^i z \in L$, for every $i \geq 0$. Since there are infinitely many values of i , student X claims that the Pumping Lemma applies only to infinite languages. Is the claim of student X correct?

4. CLOSURE

- (a) Given two DFA's, $M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$ and $M_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$ that accept two languages, L_1 and L_2 , respectively:
- construct a DFA, M_\cup , that accepts the language $L_1 \cup L_2$, as a product of M_1 and M_2 .
 - construct a DFA, M_\cap , that accepts the language $L_1 \cap L_2$, as a product of M_1 and M_2 .
- (b) For every language $L \subseteq \Sigma^*$, we define the following two languages:
- $$\text{double1}(L) = \{a_1 a_2 \cdots a_{2n} \text{ in } \Sigma^* \mid n \geq 0, \text{ and } a_1 a_2 a_3 \cdots a_n \in L\}$$
- $$\text{double2}(L) = \{a_1 a_2 \cdots a_{2n} \text{ in } \Sigma^* \mid n \geq 0, \text{ and } a_1 a_3 a_5 \cdots a_{2n-1} \in L\}$$
- For $\Sigma = \{0, 1\}$ and $L = 0^*$, describe in words the two languages $\text{double1}(L)$ and $\text{double2}(L)$.
 - Which of the following two statements are true and which are false?
 - If L is regular, then so is $\text{double1}(L)$.
 - If L is regular, then so is $\text{double2}(L)$.
- (c) For two strings $x, y \in \Sigma^*$, we write $x\#y$ for the string that alternates letters from x with letters from y :
- $$x\#\varepsilon = x$$
- $$\varepsilon\#y = y$$
- $$ax\#by = ab(x\#y) \text{ for all } a, b \in \Sigma.$$

For example, $abc\#defgh = adbecfgh$. For two languages $L_1, L_2 \subseteq \Sigma^*$, let

$$L_1\#L_2 = \{x\#y \mid x \in L_1 \text{ and } y \in L_2\}.$$

Given a DFA $M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$ that accepts L_1 , and a DFA $M_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$ that accepts L_2 , construct a finite automaton (not necessarily deterministic) that accepts $L_1\#L_2$.

- (d) Let L be a language over $\Sigma = \{0\}$. We define
- $$\text{PLUS}(L) = \{a^m b^n : 0^{m+n} \in L\}.$$
- $$\text{MINUS}(L) = \{a^m b^n : 0^{m-n} \in L\}.$$
- Prove or disprove the following claims.
- If L is regular, then $\text{PLUS}(L)$ is regular.
 - If L is regular, then $\text{MINUS}(L)$ is regular.
- (e) (**Challenging.**) For a set S of natural numbers, define:
- $\text{unary}(S) = \{1^n : n \in S\}$
 - $\text{binary}(S) = \{\text{binary representation of numbers in } S\} \subseteq \{0, 1\}^*$

For example, if $S = 1, 2, 3$, then:

- $unary(S) = \{1, 11, 111\}$
- $binary(S) = \{1, 10, 11\}$

Which of the following two statements are true and which are false?

S1 For all S , if $unary(S)$ is regular, then so is $binary(S)$.

S2 For all S , if $binary(S)$ is regular, then so is $unary(S)$.

- (f) (**Challenging.**) If A is any language, let $A_{\frac{1}{2}-}$ be the set of all first halves of strings in A so that

$$A_{\frac{1}{2}-} = \{x : \text{for some } y, |x| = |y| \text{ and } xy \in A\}.$$

Show that if A is regular, then so is $A_{\frac{1}{2}-}$.

- (g) (**Challenging.**) If A is any language, let $A_{\frac{1}{3}-\frac{1}{3}}$ be the set of all strings in A with their middle thirds removed so that

$$A_{\frac{1}{3}-\frac{1}{3}} = \{xz : \text{for some } y, |x| = |y| = |z| \text{ and } xyz \in A\}.$$

Show that if A is regular then $A_{\frac{1}{3}-\frac{1}{3}}$ is not necessarily regular.

5. MISCELLANEOUS

For each of the following statements, state whether it is TRUE or FALSE. Briefly justify your answer.

- There are uncountably many regular languages.
 - $\emptyset^* = \emptyset$.
 - The regular languages are closed under subtraction.
 - For every DFA M , if $\varepsilon \in L(M)$ then $s \in F$ (s is the start state of M , and F is the set of accepting states).
 - For every NFA M (with ε -transitions), if $\varepsilon \in L(M)$ then $s \in F$.
 - There exists an FA with 4 states that recognizes the language $\{w \in \{a, b\}^* : |w| \equiv_6 2\}$.
 - If $L = L_1 \cup L_2$, and L and $\overline{L_1}$ are both regular, then L_2 is regular.
 - A minimized FA always has one accepting state (i.e., $|F| = 1$).
 - If $L_1 \subseteq L_2$ and L_1 is regular, then L_2 is regular.
 - If $L_1 \subseteq L_2$ and L_2 is regular, then L_1 is regular.
6. This problem shows that, for certain languages, when converting an NFA to a DFA, we cannot avoid an exponential explosion in the number of state. Consider the following problem:

$$L = \{w \in \{a, b\}^* : |w| = n \text{ and } w_{n-m+1} = a\}$$

where w_i denotes the i^{th} symbol of w .

- Describe in English the language L .
 - Describe an NFA with $m + 1$ states that recognizes L .
 - Describe a DFA with 2^m states that recognizes L .
 - (**Challenging.**) Argue that any DFA for language L would have at least 2^m states, thus showing that 2^m is the minimum number of states needed for a DFA that recognizes language L .
7. A language $L \subseteq \Sigma^*$ is called normal if at least one of the following holds:

- $L = \emptyset$
- $L = \{\varepsilon\}$
- $L = \{a\}$, where $a \in \Sigma$
- $L = L_1 \cup L_2$, where L_1 and L_2 are normal languages
- $L = L_1 \cdot L_2$, where L_1 and L_2 are normal languages

- $L = \{ww : w \in L_1\}$, where L_1 is a normal language

Prove or disprove:

- Every normal language is regular.
- Every regular language is normal.

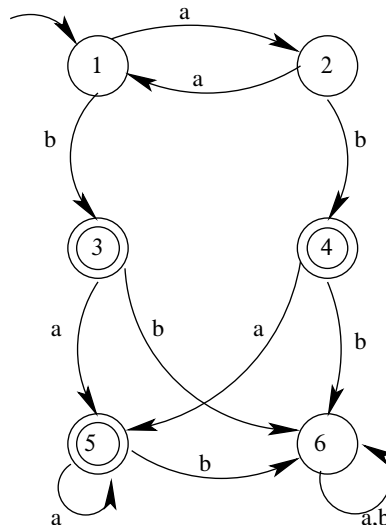
8. MINIMIZATION

- Recall the definition of the equivalence relation \approx_L :

$$x \approx_L y \text{ if and only if for all } z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L.$$

For each of the following five languages A_i over the alphabet $\{0, 1\}$, describe the \approx_{A_i} equivalence classes. If an \approx_{A_i} equivalence class is regular, describe it using a regular expression; otherwise, describe it in words (and as an extra exercise for you, do prove that it is not regular). Then determine the index of A_i . When the index is finite, give a DFA M_i that accepts A_i and has as many states as the index of A_i .

- A_1 is the language of all strings that contain the substring 00.
 - A_2 is the language of all strings that contain an odd number of occurrences of the substring 00 (possibly overlapping; e.g., 0000 $\in A_2$).
 - A_3 is the language of all strings that contain an equal number of 0's and 1's.
 - A_4 is the language of all strings that contain more 0's than 1's.
 - A_5 is the language of all strings with a square number of characters.
- Using the algorithm that you learned in class, minimize the following FA. Show all your steps.



- Given a DFA $M = (K, \Sigma, \delta, s, F)$, we define the relation E_k over the set K of states such that $(p, q) \in E_k$ if and only if for every string x , such that $|x| \leq k$, we have

$$\delta(p, x) \in F \text{ iff } \delta(q, x) \in F.$$

- Prove that E_k is an equivalence relation.
- What are the equivalence classes of E_0 ?
- Prove that $(p, q) \in E_k \Rightarrow (p, q) \in E_{k-1}$, for $k \geq 1$.

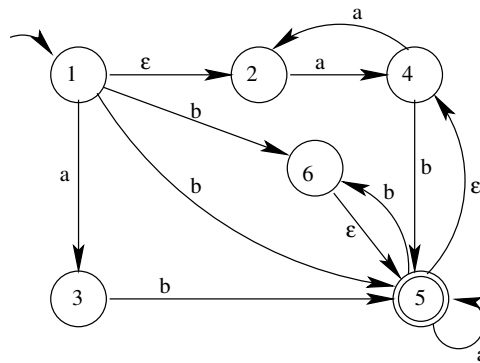
9. DECISION PROBLEMS

Give a decision procedure for each of the following questions:

- Q1** Given an FA M and a string w , is w in $L(M)$?
Q2 Given an FA M , is $L(M) = \emptyset$?
Q3 Given an FA M , is $L(M) = \Sigma^*$?
Q4 Given two FA's M_1 and M_2 , is $L(M_1) \subseteq L(M_2)$?
Q5 Given two FA's M_1 and M_2 , is $L(M_1) = L(M_2)$?
Q6 Given two regular expressions E_1 and E_2 , is $L(E_1) = L(E_2)$?
Q7 Given a regular expression E and an FA M , is $L(E) = L(M)$?
Q8 Given an FA M , does M accept a string w where $|w| \leq 2$?
Q9 Given a regular expression E , does E generate all strings w , where $|w| \equiv_3 2$?
Q10 Given an FA M , is $L(M)$ finite or infinite?

10. NONDETERMINISM

- (a) How do DFA's and NFA's compare with respect to computational power? If they are equivalent in that respect, then what are the advantages of one over the other, if any? If they are not equivalent, which is more computationally powerful?
 (b) Let M be the following NFA. Construct a DFA, M' , that accepts the complement of $L(M)$.



11. REGULAR LANGUAGE RECOGNITION

For each of the following regular languages L

- Show a regular expression that *describes* L .
 - Show an FA that *accepts* L .
 - Show a regular grammar that *generates* L .
- a.** $L = \{w \in \{a, b\}^* : \text{for every prefix } x \text{ of } w, \text{ if } |x| > 0 \text{ and } |x| \text{ is even, then the last character of } x \text{ is } 1\}$.
b. $L = \{w \in \{a, b\}^* : |w| > 5\}$.
c. L is the set of all strings over $\{a, b, c\}$ that contain at least one a and at least one b .
d. L is the set of all strings of 0's and 1's with at most one pair of consecutive 1's.

12. CONVERSIONS AMONG MODELS

- For each of the following regular grammars:
 - (a) Convert the grammar into an NFA.
 - (b) Convert the NFA you obtained in (a) to a DFA.
 - (c) Minimize the DFA you obtained in (b).

(d) Convert the DFA you obtained in (c) to a regular expression.

I.

$S \rightarrow 0A|0C|1B$
 $A \rightarrow 0A|1C|0B$
 $B \rightarrow 0C|1B$
 $C \rightarrow 0C|1C|\varepsilon$

II.

$S \rightarrow 0A|0C$
 $A \rightarrow 0B$
 $B \rightarrow \varepsilon$
 $C \rightarrow 1D$
 $D \rightarrow 0E|B$
 $E \rightarrow 1F$
 $F \rightarrow 0D$

III.

$S \rightarrow 0S|X$
 $X \rightarrow 1Y$
 $Y \rightarrow 0|S|1X$

- For each of the following regular expressions:
 - (a) Convert the regular expression into an NFA.
 - (b) Convert the NFA you obtained in (a) to a DFA.
 - (c) Minimize the DFA you obtained in (b).
 - (d) Convert the DFA you obtained in (c) to a regular grammar.

I. $(00 \cup 0)^*(00 \cup 1)^*$

II. $(1 \cup 01 \cup 001)^*(\varepsilon \cup 0 \cup 00)$

III. $(00 \cup 11 \cup (01 \cup 10)(00 \cup 11)^*(01 \cup 10))^*$

13. THE PUMPING LEMMA STRIKES AGAIN

The language $L_{GCD} = \{a^i b^j : \gcd(i, j) = 1\}$ is not regular.

- (a) What is wrong with each of the following proofs that L_{GCD} is not regular.
 - i. Let N be the constant of the Pumping Lemma. We know that any two consecutive numbers m and n larger than 1, satisfy $\gcd(m, n) = 1$. Let $w = a^{2N+1}b^{2N}$, which is obviously in L . Choose $w = xyz$ ($|xy| \leq N$ and $|y| \geq 1$), where $y = a^k$ for k odd. Then, $xy^0z = a^{2N+1-k}b^{2N} \notin L$ since both $2N+1-k$ and $2N$ are even numbers. Hence L is not regular.
 - ii. Let N be the constant of the Pumping Lemma, and let $w = a^N b^N \notin L$. Choose $w = xyz$ ($|xy| \leq N$ and $|y| \geq 1$). We know that y must be of the form a^k , $1 \leq k \leq N$. Then, $xy^{N+1}z = a^{N+kN}b^N \notin L$ since $\gcd(N+kN, N) \neq 1$. Hence, L is not regular.
- (b) Since the above two proofs failed, now it's your turn to give the correct proof (using the Pumping Lemma).
- (c) Another way of proving that L_{GCD} is not regular, is by showing that $|\approx_{L_{GCD}}| = \infty$ (Myhill-Nerode Theorem). Try this method too.

GOOD LUCK