

---

**CS 475: Formal Models of Computation, Fall 2005**  
**Midterm 1 — October 4, 2005**

---

**Instructions:**

- This is a closed book exam. No notes, books, calculators, etc. are allowed.
- There are 5 questions, each worth 10 points. However, all questions may not be of equal difficulty.
- Do all the work in the space provided or on the back sheets if necessary. If some your answers cannot fit into the space provided after the question, you may use the back sheets. But please tell us where to look.
- You may state and use without proof any results proved in class or in the homeworks.
- Write your name in the space provided on the top of every page.

<b>Name</b>	
-------------	--

Problem	Maximum Points	Points Earned	Grader
1	10		
2	10		
3	10		
4	10		
5	10		
Total	50		

1. This problem is True/False. Circle T if the statement is necessarily true. Circle F if the statement is not necessarily true. Provide a one line justification for your answer. Each correct answer with the right justification earns 1 point, while a correct answer without the right justification earns only  $\frac{1}{2}$  point.

- (a) For a language  $L$  and a homomorphism  $h$ , let  $h(L)$  be regular. Then  $L$  is regular because regular languages are closed under  $h^{-1}$ .

**T**      **F**

False. The main problem is that it is possible that  $L \neq h^{-1}(h(L))$ . Consider,  $L = \{0^n 1^n \mid n \geq 0\}$  and  $h(0) = \epsilon$  and  $h(1) = 1$ . Thus  $h(L) = 1^*$  which is regular but  $L$  is not regular. The main problem is that it is possible that  $L \subset h^{-1}(h(L)) = (0 + 1)^*$ .

- (b)  $L = \{0^n 1^m 0^n \mid n < 12 < m\}$  is regular.

**T**      **F**

True. There are many ways of seeing this. The simplest is to write down a regular expression for  $L$ . Let  $r = 1^{13}1^*$ , where  $1^{13}$  denotes a string of 13 1s; thus,  $r$  is regular expression for strings having at least 13 1s.  $L = L(r + 0r0 + 00r00 + \dots + 0^{11}r0^{11})$ , where  $0^{11}$  denotes a string of 11 0s.

- (c)  $M = (Q, \Sigma, \delta, q_0, F)$  be a minimal DFA. For a state  $q \in Q$ , let  $M_q = (Q, \Sigma, \delta, q, F)$ , i.e.,  $M_q$  is the same as  $M$  except that the start state is now  $q$ .  $M_q$  is a minimal DFA.

**T**      **F**

False. The trouble is that when the initial state is changed to  $q$ , some states may become unreachable and so should not be part of a minimal DFA. For example, consider the minimal DFA for  $(1 + 0)(1 + 0)^*$ , which has an initial state  $q_0$  and one final state  $q$ . If we make  $q$  the initial state then the language recognized is  $(1 + 0)^*$ , which can be accepted by a 1-state DFA.

- (d) If  $L_1 \cup L_2$  is regular and  $L_1$  is finite then  $L_2$  is regular.

**T**      **F**

True. Observe that the set  $L = L_1 \setminus L_2$  is finite and hence regular. Therefore, so is  $\bar{L} = \Sigma^* \setminus L$ . Thus  $L_2 = (L_1 \cup L_2) \cap \bar{L}$ , is also regular by closure properties.

- (e) If  $L_1 \cup L_2$  is regular and  $L_1$  is regular then  $L_2$  is regular.

**T**      **F**

False. Consider  $L_1 = (0 + 1)^*$ ,  $L_2 = \{0^n 1^n \mid n \geq 0\}$ . Clearly  $L_1 \cup L_2 = L_1$  is regular, but  $L_2$  is not.

- (f) There exists an algorithm that given two DFAs  $M_1, M_2$  determines whether  $L(M_1) = (L(M_2)L(M_1))^*$

**T**      **F**

True. Based on the constructions given in class one can construct an NFA (with  $\epsilon$  transitions)  $N$  such that  $L(N) = (L(M_1)L(M_2))^*$ . Then all we need to check is that that  $L(N) = L(M_1)$  which can be done using the algorithm presented in class.

- (g) Suppose  $L \subseteq 0^*$  and that  $L$  has the following property:

$$\exists p. \forall n. 0^n \in L \Leftrightarrow 0^{n+p} \in L$$

Then  $L$  is regular.

**T**      **F**

True. We can show this by proving that the number of equivalence classes of  $\equiv_L$  (in the Myhill-Nerode theorem) is finite. Observe that by definition, if  $i = j \pmod p$  then  $0^i \in L$  iff  $0^j \in L$ . Hence, if  $i = j \pmod p$  then  $0^i \equiv_L 0^j$  because  $0^i 0^k \in L$  iff  $0^j 0^k \in L$  as  $i + k = j + k \pmod p$ . Hence the index of  $\equiv_L$  is at most  $p$  which is finite.

- (h) The set  $\{L \mid L \text{ is a regular language over the alphabet } \{0, 1\}\}$  is countably infinite.

**T**      **F**

True. Any regular expression over  $\{0, 1\}$  can be seen as a string over a finite alphabet, and therefore the number of regular expressions is countably infinite.

- (i) Any infinite regular language properly contains another infinite regular language.

**T**      **F**

True. If  $L$  is an infinite regular language then consider  $w \in L$  for some string  $w$ . Clearly,  $\{w\}$  is regular (because it is a finite) and so is  $L_1 = \{x \mid x \neq w\}$  (because regular languages are closed under complementation). Thus,  $L \cap L_1$  is regular, infinite and is a strict subset of  $L$  (does not contain  $w$ ).

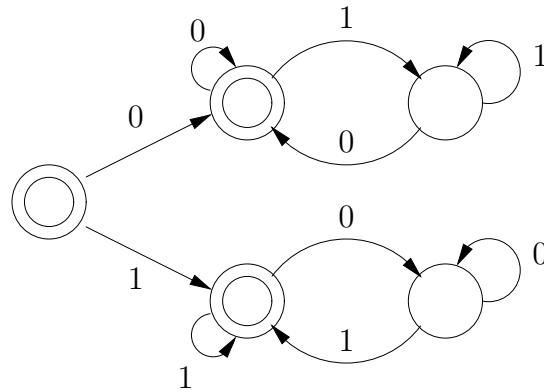
- (j) If  $L$  is regular then every subset of  $L$  is regular.

**T**      **F**

False. Consider  $L = (0 + 1)^*$ . Clearly  $\{0^n 1^n \mid n \geq 0\}$  is a non-regular subset of  $L$ .

2. Let  $L = \{w \in \{0,1\}^* \mid w \text{ has an equal number of } 01 \text{ and } 10 \text{ substrings}\}$ . For example,  $101, 1001 \in L$  because there is occurrence of  $01$  and one occurrence of  $10$  in both  $1001$  and  $101$ . Construct a finite automaton (deterministic or nondeterministic) that recognizes  $L$ . (You need not *prove* the correctness of your construction; but your construction should be self explanatory and clear.)

At first glance  $L$  might not seem regular because we need to keep track of two unbounded quantities: number of occurrences of  $01$  and the number of occurrences of  $10$ . However, the crucial observation which makes  $L$  regular is the fact that the occurrences of  $01$  and  $10$  must alternate in every string over  $0$ s and  $1$ s. The automaton is shown below



3. Prove that  $L = \{0^p 1^q \mid p \geq q^2 \text{ or } q \geq p^2\}$  is not regular.

We will prove that  $L$  is not regular by using the pumping lemma. Let  $n > 0$  be the constant of the pumping lemma. Consider the string  $w = 0^{n+1} 1^{(n+1)^2} \in L$ . Let  $x, y, z$  be some partition of  $w$  such that  $w = xyz$ ,  $|xy| \neq n$  and  $v \neq \epsilon$ . Now since  $|xy| \leq n$ , we know that  $x = 0^i$ ,  $y = 0^j$  and  $z = 0^k 1^{(n+1)^2}$ , where  $i + j \leq n$ ,  $i + j + k = n + 1$  and  $k \geq 1$ . Consider the string  $xy^2z = 0^{(n+1)+j} 1^{(n+1)^2}$ . Since  $1 \leq j \leq n$ , we have  $(n + 1)^2 < (n + 1 + j)^2$  and  $(n + 1 + j) < (n + 1)^2 < ((n + 1)^2)^2$ . Thus  $xy^2z \notin L$  and so  $L$  does not satisfy the pumping lemma. *Aside:* Observe that if we start with the string  $0^n 1^{n^2}$  then we will fail to show that the pumping lemma is not satisfied when  $n = 1$ .

4. For two strings  $x, y$ , with  $|x| = |y|$ , the *hamming distance* between  $x$  and  $y$  (denoted by  $d_H(x, y)$ ) is defined to be the number of positions where  $x$  and  $y$  have a different symbol; if  $|x| \neq |y|$  then  $d_H(x, y) = \infty$ . For example  $d_H(1001, 1100) = 2$ , because 1001 and 1100 differ in position 2 and 4. Let  $H_3(L) = \{y \mid \exists x \in L. d_H(x, y) \leq 3\}$ . In other words,  $H_3(L)$  consists of all strings that are at most distance 3 away from some string in  $L$ . Show that if  $L$  is regular then so is  $H_3(L)$ .

Consider the alphabet  $\Sigma_2 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$ . Let  $N_3$  be the set of strings over  $\Sigma_2$  with at most 3 occurrences of  $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$  and  $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ ;  $N_3$  is clearly regular. Thus,  $N_3$  contains all strings whose top row and bottom row are within Hamming distance 3.

Define a homomorphism  $h : \Sigma_2 \rightarrow \{0, 1\}$  such that  $\text{top}\left(\begin{bmatrix} a \\ b \end{bmatrix}\right) = a$  and  $\text{bot}\left(\begin{bmatrix} a \\ b \end{bmatrix}\right) = b$ .

Consider  $L' = \text{bot}(\text{top}^{-1}(L) \cap N_3)$ . Now  $\text{top}^{-1}(L)$  is the set of strings whose top row belongs to  $L$ .  $\text{top}^{-1}(L) \cap N_3$  is the set of strings whose top row belongs to  $L$  and whose bottom row is at most hamming distance 3 from the top row. Thus  $L' = H_3(L)$ . Furthermore,  $L' = H_3(L)$  is regular because it is obtained by applying regularity preserving operations to regular languages  $L$  and  $N_3$ .

Observe that the above construction can be generalized to show that the set of all strings that are within hamming distance  $k$  (for some fixed  $k$ ) from a regular languages  $L$  is regular.

**Alternate Solution:** It is also possible to give a direct construction of an NFA  $N$  recognizing  $H_3(L)$ . Since  $L$  is regular there is a DFA  $M = (Q, \Sigma, \delta, q_0, F)$  recognizing  $L$ .  $N$  will simulate  $M$  as it reads the input, except it might for some steps ignore the input symbol it reads and non-deterministically guess some symbol on which to simulate  $M$ . However,  $N$  will make such “non-deterministic jumps” only 3 times. Formally,  $N = (Q', \Sigma, \delta', q'_0, F')$  where

- $Q' = Q \times \{0, 1, 2, 3\}$ , where the second component stores how many “non-deterministic jumps” have been made,
- $q'_0 = (q_0, 0)$  (initial state of  $M$  and 0 non-deterministic jumps have been made),
- $F' = F \times \{0, 1, 2, 3\}$  (any final state of  $M$  is reached no matter how many jumps have been made), and
- $\delta'$  is defined as follows

$$\delta'((q, i), a) = \begin{cases} \{(\delta(q, a), 3)\} & \text{if } i = 3 \\ \{(\delta(q, a), i)\} \cup \{(q', i + 1) \mid \exists a'. \delta(q, a') = q'\} & \text{otherwise} \end{cases}$$

5. Let  $\Sigma$  be a finite alphabet, and let  $h : \Sigma^* \rightarrow \Sigma^*$  be a homomorphism. For any language  $L$ , define  $h^*(L)$  to be

$$h^*(L) = L \cup h(L) \cup h(h(L)) \cup \dots$$

- (a) If  $L$  is regular, is  $h^*(L)$  necessarily regular? Prove your answer. **[5 points]**

$h^*(L)$  is not necessarily regular, even if  $L$  is regular. Let  $L = \{0\}$  and define  $h(0) = 00$ . Then  $h^*(L) = \{0^{2^n} \mid n \geq 0\}$  which was shown to be non-regular in homework 2.

- (b) Let  $h$  be such that for all  $a \in \Sigma$ ,  $h(a) \in \Sigma$ , i.e.,  $h$  maps each symbol to another symbol, instead of a string in general. If  $L$  is regular then is  $h^*(L)$  necessarily regular? Prove your answer. **[5 points]**

$h^*(L)$  is necessarily regular for such homomorphisms. The number of maps from  $\Sigma$  to itself is bounded above by  $|\Sigma|^{|\Sigma|}$  (finite) and  $h^k$  for any  $k$  is one such map. Thus, the infinite union describing  $h^*$  boils down to just a finite number of terms. We already know that regular languages are closed under finite union and homomorphism.