University of California, San Diego
Department of Computer Science and Engineering
Course: CSE 105 − Theory of Computability

Instructor: Adriana Palacio
2004 Summer Session I
July 14, 2004

Test 1 Solutions

The following are solutions for one of the versions of the exam. Since the other versions are very similar, separate solutions will not be provided.

Problem 1 [**12 points**]

Indicate whether the following statements are TRUE (T) or FALSE (F) by **circling** your answers. Justify your answers briefly.

a) It is possible to define a DFA with no final states.

TRUE; The set of final states of a DFA can be any subset of the set of states. In particular, it can be $\emptyset$.

b) For any languages A and B such that $A \subseteq B$, if B is regular then A is regular.

FALSE; Let $A = \{\, \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \,\}$ and $B = \{\mathtt{0}, \mathtt{1}\}^*$. Then B is regular and $A \subseteq B$, but A is nonregular.

c) If the pumping lemma for regular languages holds for L then L is regular.

FALSE; There exist nonregular languages (e.g., $\{\, a^i b^j c^k \mid i, j, k \geq 0$ and if $i = 1$ then $j = k \,\}$) for which the conditions of the pumping lemma hold.

d) All finite languages are regular.

TRUE; For any finite language $L = \{s_1, \ldots, s_k\}$, where $k \geq 1$, the regular expression $s_1 \cup \cdots \cup s_k$ describes L. The regular expression $\emptyset$ describes the language $\emptyset$.

e) An NFA without ε-transitions is not as powerful as an NFA with ε-transitions (i.e., there exists a language that can be recognized by some NFA with ε-transitions but cannot be recognized by any NFA without ε-transitions).

FALSE; For any NFA (with or without ε-transitions) there exists an equivalent DFA, which is an NFA with no ε-transitions.

f) For every NFA there exists a PDA that recognizes the same language.

TRUE; The NFA can easily be converted into a PDA that doesn't use its stack. This PDA will recognize the same language.

Problem 2 [**12 points**]

Let $L = \{\texttt{0}^{20}\texttt{1}^{20}\}$.

a) Adriana says she has a DFA with 15 states that recognizes L. Explain why she is incorrect.

Since $|\texttt{0}^{20}\texttt{1}^{20}| > 15$, any path in a DFA with 15 states labelled by the string $\texttt{0}^{20}\texttt{1}^{20}$ must revisit some state. If the DFA does not accept this string, then the language recognized by the DFA is not L. If the DFA accepts the string, then there is a cycle on a path between the start state and a final state. Therefore, the DFA accepts infinitely many strings, and the language recognized by it is not L.

b) What is the minimum number of states in a DFA that recognizes L? $\boxed{42}$

(41 states are required to accept a string of length 40. Additionally, a "black hole" state is needed to reject unwanted strings.)

c) Given a DFA, how would you determine whether the language it recognizes is infinite?

If the DFA contains a cycle that is reachable from the start state and such that a final state is reachable from some state in the cycle, then the language recognized by the DFA is infinite. Otherwise, it is finite.

Problem 3 [**14 points**]

Let $\Sigma = \{\texttt{0}, \texttt{1}\}$ and let $L = \{\, w \in \Sigma^* \mid w$ contains an even number of 0s, or exactly two 1s $\}$.

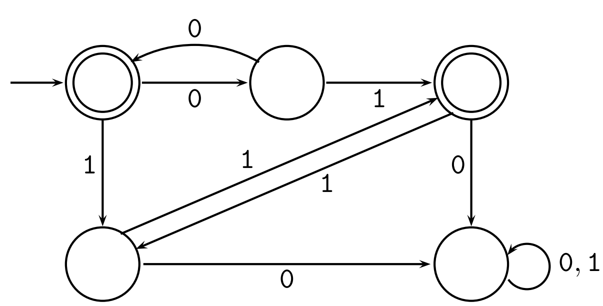
In the box below, write a regular expression that describes the language L.

$$\boxed{\texttt{1}^*(\texttt{1}^*\texttt{0}\texttt{1}^*\texttt{0}\texttt{1}^*)^* \cup \texttt{0}^*\texttt{1}\texttt{0}^*\texttt{1}\texttt{0}^*}$$

Problem 4 [**14 points**]

Let $L = \{\, \texttt{0}^m\texttt{1}^n \mid m + n \text{ is even} \,\}$.

Draw the state diagram of a DFA with *at most 6 states* that recognizes L.



Problem 5 [**10 points**]

Consider the operation & defined as follows:

$$A \,\&\, B \;=\; \{\, w \mid \text{there exist } x \in A, y \in B \text{ such that } w = xy \text{ and } |x| = |y| \,\}$$

Is the class of regular languages closed under &? If so, prove it. If not, disprove it (i.e., give an example of two regular languages A and B such that $A \,\&\, B$ is not regular).

The class of regular languages is NOT closed under &. Consider $A = \{\mathtt{0}\}^*$ and $B = \{\mathtt{1}\}^*$. Both of these languages are regular, but $A \,\&\, B = \{\, \mathtt{0}^n\mathtt{1}^n \mid n \in \mathbb{N} \,\}$ is not regular.

Problem 6 [**15 points**]

Let $L \;=\; \{\, w \in \{\mathtt{0},\mathtt{1}\}^* \mid w \text{ is an even-length palindrome} \,\}$.

Prove that L is nonregular.

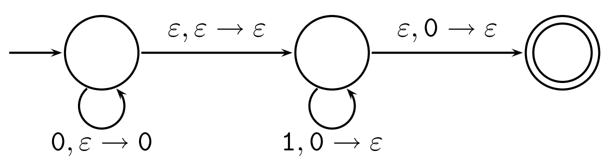
Proof by contradiction.

Assume that L is regular. Let p be given by the pumping lemma. Let $s = \mathtt{0}^p\mathtt{11}\mathtt{0}^p$. Then $s \in L$ (s is a palindrome and the length of s is even) and $|s| \geq p$. By the pumping lemma, s can be split into x, y, z such that 1) $|y| > 0$, 2) $|xy| \leq p$, and 3) for all $i \geq 0$: $xy^iz \in L$. By condition 2), $x = \mathtt{0}^j$ for some $j \geq 0$ and $y = \mathtt{0}^k$ for some $k \geq 0$. Therefore, $z = \mathtt{0}^{p-j-k}\mathtt{11}\mathtt{0}^p$. By condition 1), $k \geq 1$. If $i = 2$, then $xy^iz = xyyz = \mathtt{0}^j\mathtt{0}^k\mathtt{0}^k\mathtt{0}^{p-j-k}\mathtt{11}\mathtt{0}^p = \mathtt{0}^{p+k}\mathtt{11}\mathtt{0}^p$. Since $k \geq 1$, $\mathtt{0}^{p+k}\mathtt{11}\mathtt{0}^p$ is not a palindrome. Therefore, $\mathtt{0}^{p+k}\mathtt{11}\mathtt{0}^p \notin L$. This contradicts condition 3).

Problem 7 [**14 points**]

Let $L \;=\; \{\, \mathtt{0}^m\mathtt{1}^n \mid m > n \,\}$.

Prove that L is context-free by drawing the state diagram of a PDA with *at most 3 states* that recognizes L.



Problem 8 [**14 points**]

Let $\Sigma = \{\texttt{0}\}$ and consider the following operation defined on languages over Σ.

$$\text{DOUBLE}(L) = \{ w \in \Sigma^* \mid \text{there exists } x \in L \text{ such that } |w| = 2|x| \}$$

Prove that if L is regular then DOUBLE(L) is also regular. Describe the idea of your construction informally and provide a formal definition. You are not required to prove that your construction is correct, but do indicate what you would have to prove to establish this.

Given: L is regular, i.e., there exists a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes L.

Want: An NFA N that recognizes DOUBLE(L).

Construction:

Idea: We build an NFA N by replacing each transition in M from a state q_i to a state q_j with two transitions: one from q_i to a new state q' and another from q' to q_j. All paths from the start state q_0 to a final state in M become twice as long. Therefore, N will accept all strings that have twice the length of some string accepted by M. Notice that since M is a DFA and the alphabet Σ is unary, each state in M has exacty one transition going to another state. Thus the number of additional states required is the number of states in M.

Formal definition: Let $N = (Q', \Sigma', \delta', q'_0, F')$, where each component is defined as follows.

- $Q' = \{q_0, \ldots, q_{k-1}, q'_1, \ldots, q'_k\}$, where $Q = \{q_0, \ldots, q_{k-1}\}$ and $q'_1, \ldots, q'_k \notin Q$.
- $\Sigma' = \Sigma = \{\texttt{0}\}$
- $\delta' : Q' \times (\Sigma' \cup \{\varepsilon\}) \to \mathcal{P}(Q')$ is defined as follows:

$$\begin{array}{lcll} \delta'(q_i, \texttt{0}) & = & \{q'_{i+1}\} & \text{for } 0 \le i \le k-1 \\ \delta'(q'_{i+1}, \texttt{0}) & = & \{\delta(q_i, \texttt{0})\} & \text{for } 0 \le i \le k-1 \\ \delta'(q, \varepsilon) & = & \emptyset & \text{for all } q \in Q' \end{array}$$

- $q'_0 = q_0$
- $F' = F$

Correctness: We must show that if $w \in$ DOUBLE(L) then N accepts w, and if $w \notin$ DOUBLE(L) then N rejects w.
