

Spring 2004 Midterm Exam. Time: 150 minutes

This exam is closed book. It contains five problems. Complete as many problems as you can. Show steps of your work. **Justify your answers.**

Keep your answers clean. Do not include anything irrelevant. You will be graded not only on the correctness of your answer, but also on the clarity you express it.

Problem	Points	Grade
1	20	
2	20	
3	20	
4	20	
5	20	
Total	100	

I have abided by the Academic Honor Code on this exam.

Name (please print)

Signature

Date

Problems

- (20 points) Let $M_i = (Q_i, \Sigma, \delta_i, q_0^i, F_i)$, $i = 1, 2$, be two DFAs. Construct a new DFA $M = (Q, \Sigma, \delta, q_0, F)$ such that $L(M) = L(M_1) \cap L(M_2)$.

Solution: See Du-Ko on product DFA on pages 33 and 34 and change F to $F_1 \times F_2$, or see lecture notes if you took notes.

- (20 points) Let L be a language. Show that if every subset of L is regular, then L must be finite.

Proof. We will prove it by contradiction. Assume that L were infinite. Let w_0 be an arbitrary string in L . Let $w_i \in L$ with $|w_i| > 2|w_{i-1}|$, where $i = 1, 2, \dots$. Since L is infinite, such strings w_i 's exist. Let $L_0 = \{w_0, w_1, \dots\}$. Then L_0 is a subset of L , and L_0 is infinite. We note that L_0 so constructed has the following property: For any two strings $u \in L_0$ and $v \in L_0$, if v is longer than u , then

$$|v| > 2|u|. \quad (1)$$

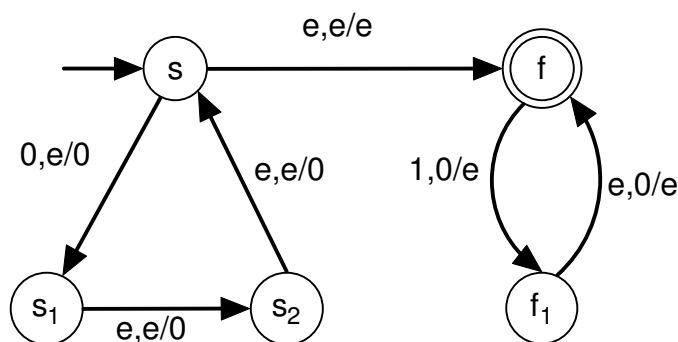
By assumption, L_0 is regular. It follows from the pumping lemma that there exists a positive integer K such that for any $w \in L_0$ with $|w| \geq K$, there must be strings x, y, z such that $w = xyz$, $y \neq \epsilon$, and $xy^2z \in L_0$. But $|xy^2z| = |w| + |y| \leq 2|w|$, and so Inequality 1 is violated, which implies that $xy^2z \notin L_0$. This is a contradiction. Thus, L must be finite. This completes the proof.

3. (20 points) Let $L = \{0^i 1^j \mid 3i = 2j\}$. Select one of the following two problems; only one problem will be graded. (**Note:** If you try both and do not indicate which problem is to be graded, problem 2(a) will be graded by default.)

- (a) Construct a context-free grammar that generates L .
 (b) Construct, using a state diagram, a PDA that accepts L .

Solution: (a) We note that for non-negative integers i and j to satisfy $3i = 2j$, it must be true that $i = 2k$ and $j = 3k$ for some $k \geq 0$. Construct a context-free grammar $G = (Q, \Sigma, R, S)$ for L as follows: $Q = \{S\}$, $\Sigma = \{0, 1\}$, and $R = \{S \rightarrow 00S111 \mid \epsilon\}$.

(b) Construct a PDA as follows: For each 0 in the input, push three 0's in the stack. Then for each 1 in the input, pop two 0's off the stack. The following is a state diagram of this PDA, where e represents the empty string ϵ (I apologize that my drawing software does not have ϵ).



4. (20 points) Assume that $G = (V, \Sigma, R, S)$ is a context-free grammar in which every production rule is in one of the following two forms: $A \rightarrow BC$ or $A \rightarrow a$, where $A, B, C \in V$ and $a \in \Sigma$. Let $x \in L(G)$ with $|x| = n > 1$. What is the exact number of times you need to apply production rules to generate x ? Defend your answer.

Solution: Note that a rule of the form $A \rightarrow BC$ generates one more nonterminal in each application, while a rule of the form $A \rightarrow a$ simply replace a nonterminal to a terminal, where $a \in \Sigma$. Thus, to generate a string of length n , we must apply rules of the form $A \rightarrow BC$ exactly $n - 1$ times and apply rules of the form $A \rightarrow a$ exactly n times. The exact total number of times of applying rules is therefore $2n - 1$.

5. (20 points) Describe how to use a one-tape DTM to simulate a deterministic PDA that has two unbounded stack memories.

Solution: Let M_1 be deterministic PDA with two stacks. Construct a one-tape deterministic Turing machine M to simulate M_1 as follows: Divide the tape into three

sections, the first section holds input x , the second section is used to simulate the first stack, and the third section is used to simulate the second stack. Note that the second and the third sections may shrink and grow dynamically. The initial configuration of M looks like this:

$$(s, BxS_1S_2\underline{B}),$$

where S_1 is a special symbol representing stack 1 and its top, and S_2 a special symbol representing stack 2 and its top.

M will first move to the rightmost symbol of x , then simulate the transition of the PDA as follows: For each transition move $\delta_1(p, a, \alpha_1, \alpha_2) = (q, \beta_1, \beta_2)$ of M_1 , M marks the current symbol a with \hat{a} , moves to stack 1, depending on α_1/β_1 , either pops a new symbol by left shifting everything of the right-hand side of S_1 , or right shifting everything of the right-hand side of S_1 and then pushes a new symbol into stack 1; moves to stack 2, depending on α_2/β_2 , either pops a new symbol by left shifting everything of the right-hand side of S_2 , or right shifting everything of the right-hand side of S_2 and then pushes a new symbol into stack 2. M then moves back to the rightmost symbol of x that has not been marked and repeats the above process. When M finishes reading x , i.e., when M reads S_1 , M checks whether the last state of M_1 being simulated is a final state of M_1 and both stacks are empty, if so, M moves to the halting state. Otherwise, M enters a loop state.

This completes the construction.
