Automata Qual Exam (Fall 2012)
Answer ALL questions (Closed Book Exam)

Question 1 (25 points)

Show that the following problem is undecidable.

`Given a Turing machine` M`, is` $L(M)$ `recursive (Turing-decidable)?`

You can assume without proof that the halting problem for Turing machines is undecidable.

Answers: Let M_1 be a TM that accepts A_{TM} (see Sipser's book), which is a Turing-recognizable (recursively enumerable) language. Given M and w, we design a TM M_2 that behaves like M_1 while simulating M on w in the background using time sharing. If M_1 accepts, M_2 accepts. Also, if M halts on w, M_2 accepts irrespective of how M_1 behaves on the input. So, either M_2 accepts Σ^* (that is, $L(M_2)$ is recursive), or M_2 behaves the same as M_1 which recognizes a recursively enumerable (but not recursive) language. If we can solve the given problem, we could have solved the halting problem.

Question 2 (15 points)

Given a language L, let $\mathrm{Perm}(L) = \{y \mid x \in L$ and y is a permutation of $x\}$. Is the set of context-free languages closed under the operation Perm? Justify your answer.

Answers: $L_2 = (abc)^*$ is context-free, but $\mathrm{Perm}(L_2) = \{w \mid |w|_a = |w|_b = |w|_c\}$ is not.

Question 3 (15 points)

Use the pumping lemma to show that $\{a^i b^j c^k \mid i * j = k\}$ is not context-free.

Answers: (Sketch) Pick the string $w = a^p b^p c^{p^2}$. For each case of breaking w into $uvwxy$ where $|vxy| \leq p$, we can show that $uv^0xy^0z \notin L$.

Question 4 (15 points)

Give an unambiguous context-free grammar for the set of strings with the same number of a's and b's. (Hint: Your grammar should reflect the design of a one-state deterministic PDA for the language. But there is no need to go through the steps given the textbook's algorithm for converting a general PDA to a CFG.)

Answers: $S \longrightarrow \epsilon \mid aAS \mid bBS, \quad A \longrightarrow aAA \mid b, \quad B \longrightarrow bBB \mid a$

Question 5 (5 points + 25 points)

(a) Give an inductive definition for regular expression over Σ.

Answers: $\emptyset$ and ϵ are both regular expressions. a is a regular expression for $a \in \Sigma$. If r_1 and r_2 are regular expressions, then $(r_1 \cup r_2)$, $(r_1 \cdot r_2)$ and r_1^* are regular expressions.

(b) Given a regular expression r, we are interested in constructing a regular expression r' such that $L(r')$ denotes the set of suffices of strings in $L(r)$. You are asked to give a recursive procedure that constructs r' given a regular expression r. Your recursive procedure should be designed based on the inductive definition of regular expression.

Answers: If $r = \emptyset$, then $r' = \emptyset$. If $r = \epsilon$, then $r' = \epsilon$. If $r = a$ for $a \in \Sigma$, then $r' = x \cup \epsilon$. If $r = r_1 \cup r_2$, then $r' = r_1' \cup r_2'$. If $r = r_1 r_2$, then $r' = r_1' r_2 \cup r_2'$. If $r = r_1^*$, then $r' = r_1' r_1^*$.