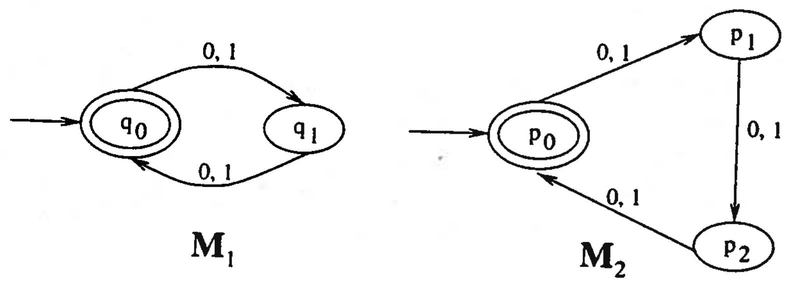
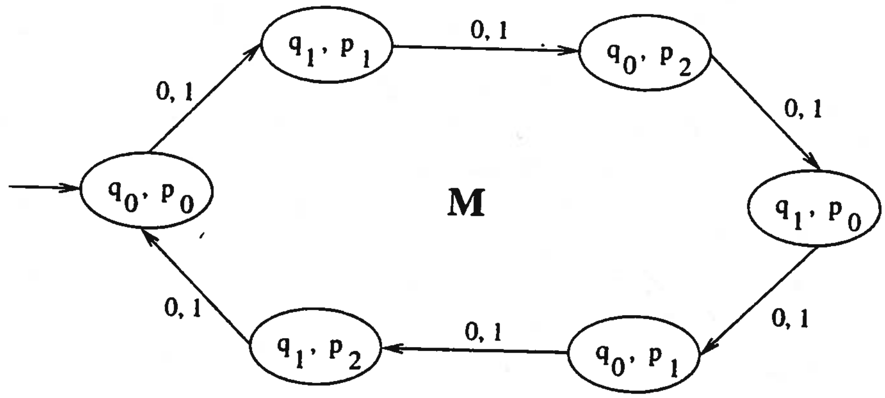
Comprehensive Exam: Autumn 1998-99

Automata and Formal Languages (60 points)
Sample Solutions

Problem 1. [10 points]
Consider the following DFAs (deterministic finite-state automata) called M_1 and M_2 over the alphabet $\{0, 1\}$. Let L_1 be the language of M_1 and L_2 be the language of M_2.



a). [4 points]
Give succinct descriptions of the languages L_1 and L_2.
b). [6 points]
Consider the machine M given below that is the cross-product of the two machines M_1 and M_2. (Running the machine M on an input string corresponds to running M_1 and M_2 together in parallel on that input string. Each state in the machine M corresponds to a pair of states, one each from M_1 and M_2. Each transition in M is the combination of the state transitions from M_1 and M_2.)



For each of the following languages, specify the choice of final states in M that would cause M to accept that language.
(i). $L_1 \cup L_2$
(ii). $L_1 - L_2$ (the set of strings in L_1 that do not belong to L_2)
(iii). $\overline{L_1}$

Solution:

The language L_1 contains all strings of even length from $\{0,1\}^*$. The language L_2 contains all strings of length divisible by 3.

(i). For $L_1 \cup L_2$, make (q_0, p_0), (q_0, p_2), (q_1, p_0) and (q_0, p_1) the final states.

(ii). For $L_1 - L_2$, make (q_0, p_2) and (q_0, p_1) the final states.

(iii). For $\overline{L_1}$, make (q_1, p_1), (q_1, p_0), and (q_1, p_2) the final states.

Problem 2. [10 points]

In the following, R denotes a *regular* language, and C, C' denote *context-free* languages. Classify each of the following statements as being TRUE or FALSE. *You will receive 2 points for each correct answer and −1 point for each incorrect answer.*

a). There must exist a *deterministic* push-down automata that accepts R.

b). There must exist a *deterministic* Turing machine that accepts $C \cap C'$.

c). $R \cap C$ must be regular.

d). $C \cup C'$ cannot be regular.

e). $\overline{C}$ must be recursive.

Solution:

a). TRUE

b). TRUE

c). FALSE

d). FALSE

e). TRUE

Problem 3. [20 points]

Classify each of the following languages as being in one of the following classes of languages: *empty, finite, regular, context-free, recursive, recursively enumerable.* You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the language of a DFA M could be *empty* or *finite*, and must always be *context-free*, but the smallest class that is appropriate *regular.*

a). $L = \{a^i b^j c^k d^l \mid i = j \text{ or } k = l\}$.

b). $L = \{a^i b^j c^k d^l \mid i = j \text{ and } k = l\}$.

c). $L = \{a^i b^j c^k d^l \mid i = j \text{ and } j = k\}$.

c). $L = \{a^i b^j c^k d^l \mid i = l \text{ and } j = k\}$.

e). $L = \{a^i b^j c^k d^l \mid i \times j \times k \times l \text{ is divisible by } 5\}$.

f). The language of a push-down automaton with only one state.

g). The set of strings encoding all push-down automata that accept non-recursive languages.

h). The language of a PDA with two stacks.

i). The complement of a language in NP.

j). The intersection of a recursive language and a recursively enumerable language.

Solution:

a). context-free

b). context-free

c). recursive

d). context-free
e). regular
f). context-free
g). empty
h). recursively enumerable
i). recursive
j). recursively enumerable

Problem 4. [10 points]
Suppose we have languages A and B such that A has a *polynomial-time reduction* to B. Classify each of the following statements as being TRUE or FALSE. *You will receive 2 points for each correct answer and* -1 *point for each incorrect answer.*
a). If B is recursively enumerable, then A must be recursively enumerable.
b). If A is recursive, then B must be recursive.
c). If B is NP-hard, then A must be NP-hard.
d). If A is NP-complete, then B must be NP-complete.
e). It is possible that A is solvable in polynomial time but B is not even in NP.

Solution:
a). TRUE
b). FALSE
c). FALSE
d). FALSE
e). TRUE

Problem 5. [10 points]
For any language L define its reversal as follows:

$$L^R = \{w^R \mid w \in L\}.$$

Recall that for a string $w = w_1w_2 \dots w_k$, we define its reversal as $w^R = w_n \dots w_2w_1$.

Suppose that L is an NP-complete language. Then, is L^R also an NP-complete language? For full credit, you must sketch a proof for your answer.

Solution: Indeed, L^R is NP-complete. To prove this, we need to show two things: that L^R is in NP and that it is NP-hard. First observe that a Turing machine given an input w can reverse its input and obtain w^R in polynomial time. Thus, given any Turing machine for a language L we can construct another one for the language L^R with only a polynomial overhead in the running time. Since L must be in NP, there exists a nondeterministic Turing machine M for L that runs in polynomial time. If M is modified to first reverse its input (from w to W^R), then it accepts exactly the language L^R in polynomial time, implying that L^R is also in NP. To show NP-hardness, we give a reduction from the NP-hard language L to L^R. The reduction merely takes a string w and outputs w^R; clearly, this is a polynomial-time reduction. The correctness of the reduction follows from the definition: $w \in L$ if and only if $w^R \in L^R$.