Comprehensive Exam: Autumn 2002-03

Automata and Formal Languages (60 points)
Sample Solutions

Problem 1. [10 points]

Recall that a string x is called a *substring* of another string w if x appears consecutively within w (i.e. $w = yxz$ for some strings y,z).

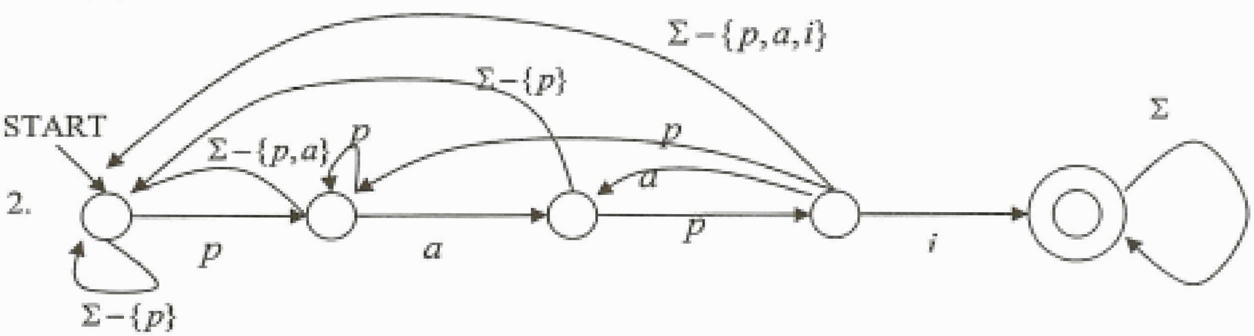
1. [3 points] Give a regular expression for the language $L = \{ w \in \Sigma^* \mid$ the string *papi* is a substring of $w\}$, where Σ is the English alphabet.

2. [7 points] Give a deterministic finite automaton for L. (Partial credit for a nondeterministic finite automaton if you cannot get a DFA.)

Solution:

1. $\Sigma^* papi \Sigma^*$

2.



Problem 2. [10 points]

Decide whether the following statements are TRUE or FALSE. You will receive 2 points for each correct answer and –2 points for each incorrect answer.

1. Nondeterministic and deterministic finite automata recognize the same set of languages.

2. Nondeterministic and deterministic pushdown automata recognize the same set of languages.

3. Nondeterministic and deterministic Turing machines recognize the same set of languages.

4. The intersection of two recursive languages is recursive.

5. The intersection of two context-free languages is context-free.

Solution:

1. True

2. False

3. True

4. True

5. False

Problem 3. [15 points] Classify each of the following languages as being in one of the following classes of languages: *regular, context-free, recursive, recursively enumerable.* You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the appropriate class for the language of a PDA is context-free. You will receive 3 points for each correct answer and –2 points for each incorrect answer.

1. Call a string w over the alphabet $\Sigma=\{s,r\}$ *well-formed* if every prefix of w contains at least as many occurrences of letter s as of letter r. What is the appropriate class for the language of well-formed strings ?

2. The language of well-formed strings w over $\Sigma=\{s,r\}$ such that in each prefix of w the number of occurrences of s does not exceed the number of occurrences of r by more than 10.

3. The language of strings over $\{s,r,s',r'\}$ that are well-formed with respect to both pairs (s,r) and (s',r'), i.e. each prefix contains at least as many occurrences of s as of r, and at least as many occurrences of s' as of r'.

4. The set of encodings of Turing machines M whose time complexity is not bounded by n^2; i.e., $L = \{ M \mid$ there exists an input string w such that M performs more than $|w|^2$ steps on input $w\}$

5. The language $L = \{w \in \Sigma^* \mid \exists x \in L_1, \exists y \in L_2 \text{ such that } x = wy\}$, where L_1 is regular and L_2 is recursively enumerable.

Solution:

1. context-free
2. regular
3. recursive
4. recursively enumerable
5. regular

Problem 4. [12 points]
Classify each of the following problems as being in one of the following three classes: *P (polynomial-time solvable), decidable but not known to be in P* (this class includes eg. NP-complete and PSPACE-complete problems) , *undecidable*. You will receive 3 points for each correct answer and –2 points for each incorrect answer.

1. *Input*: Deterministic finite automata *A, B*. *Question*: $L(A) \subseteq L(B)$?
2. *Input*: Pushdown automaton *A*, deterministic finite automaton *B*. *Question*: $L(A) \subseteq L(B)$?
3. *Input*: Deterministic finite automaton *A*, pushdown automaton *B*. *Question*: $L(A) \subseteq L(B)$?
4. *Input*: Nondeterministic finite automata *A, B*. *Question*: $L(A) \subseteq L(B)$?

Solution:

1. P
2. P
3. undecidable
4. decidable but not known to be in P

Problem 5. [13 points]
Prove that the FEEDBACK NODE SET problem is NP-complete, using the fact that the NODE COVER problem is NP-complete.
The definitions of these problems are recalled below.

FEEDBACK NODE SET
Input: A directed graph H and a positive integer k.
Question: Is there a set F of at most k nodes such that removing from the graph the nodes of F and their incident edges leaves an acyclic graph ? (Such a set F is called a *feedback node set* of H).

NODE COVER
Input: An undirected graph G and a positive integer k.
Question: Is there a set C of at most k nodes such that every edge of G is incident to at least one node of C ? (Such a set C is called a *node cover* of G.)

Solution:

First show that the FEEDBACK NODE SET problem is in NP. To see this, note that we can guess a set F of at most k nodes, remove them from the graph along with their incident edges, and then check in polynomial time that the resulting graph is acyclic.

To show that the FEEDBACK NODE SET problem is NP-hard, we give a polynomial time reduction from the NODE COVER problem. Given an instance (G,k) of the NODE COVER problem, we construct an instance (H,k) of the FEEDBACK NODE SET problem as follows. The directed graph H has exactly the same nodes as G. For each undirected edge $[u,v]$ of G, the directed graph H has two directed edges $u \to v$ and $v \to u$. We claim that G has a node cover of size k if and only if H has a feedback node set of size k.

(if) Suppose that H has a feedback node set F of size k. Note that for each edge $[u,v]$ of G, the corresponding two directed edges $u \to v$ and $v \to u$ that were introduced in H form a cycle. Therefore, for each edge $[u,v]$ of G, the feedback node set F of H must contain at least one of the nodes u,v. Thus, the set F is a node cover of G.

(only if) Suppose that G has a node cover C of size k. Then every directed edge of H is incident to at least one node of C. Therefore, removing the set C of nodes and their incident edges from H, will leave a graph with no edges, which is of course acyclic. Thus, C is a feedback node set of H.