Comprehensive Exam Solutions Autumn 2003-04

Automata and Formal Languages (60 points)

Problem 1. [10 points]

Consider the following grammar G over the alphabet $\Sigma = \{0,1\}$, where S is the start symbol of the grammar.

$S \rightarrow \varepsilon \mid 0T \mid 1U$
$T \rightarrow 0T \mid 1S$
$U \rightarrow 0S$

1. [2 points] Give a derivation of the string 1001.

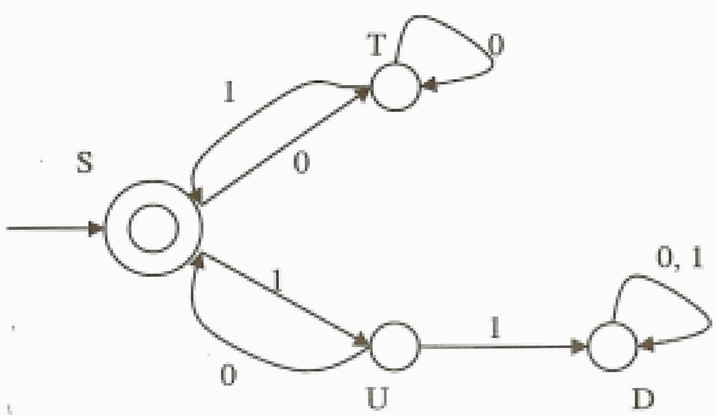
2. [4 points] Give a deterministic finite automaton for the language of G.

3. [4 points] Give a regular expression for the language of G.

Solution.

1. $S \Rightarrow 1U \Rightarrow 10S \Rightarrow 100T \Rightarrow 1001S \Rightarrow 1001$

2. The DFA is shown in the following figure. It is ok to omit state D.



3. (10 + 00*1)*

Problem 2. [10 points]

For each of the following statements, write TRUE if the statement is true for all languages L,M that satisfy the hypothesis; otherwise, write FALSE. You will receive 2 points for each correct answer and –2 points for each incorrect answer.

1. If L is a nonregular language and M is a regular language then their concatenation LM is not regular. (Recall that the concatenation is $LM = \{xy \mid x \in L, y \in M\}$.)

2. All finite languages L are regular.

3. All context-free languages L are in the class P (Polynomial Time).

4. If L, M are languages in PSPACE then their difference L-M is also in PSPACE. (Recall that $L - M = \{x \mid x \in L \text{ and } x \notin M\}$.)

5. If L, M are recursively enumerable languages then L-M is also recursively enumerable.

Solution.

1. FALSE

2. TRUE

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, all languages*. 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. The set of strings over $\{0,1\}$ in which the number of 0's is divisible by 5.

2. The set of strings over $\{a,b,c\}$ that contain the same number of a's, b's and c's.

3. $L = \{a^i b^j c^k \mid j = i + k,\ i, j, k \geq 0\}$.

4. The set of encodings of deterministic Turing machines M such that $L(M) \neq \varnothing$.

5. The language $L = \{x \in \{0,1\}^* \mid xx \in M\}$ where M is a regular language.

Solution.

1. regular

2. recursive

3. context-free

4. recursively enumerable

5. regular

Problem 4. [12 points]
Classify each of the following problems as being in one of the following 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*: Context free grammar G and deterministic finite automaton A.
   Question: $L(G) \cap L(A) = \varnothing$?

2. *Input*: Context free grammar G and deterministic finite automaton A over alphabet Σ.
   Question: $L(G) \cup L(A) = \Sigma^*$?

3. *Input*: Deterministic finite automaton A.
   Question: Is $L(A)$ finite ?

4. *Input*: Encoding of a deterministic Turing machine M, input string w of length n.
   Question: Does M use more than n^2 space on input w ?

Solution.

1. in P

2. undecidable

3. in P

4. decidable but not known to be in P

Problem 5. [13 points]

a. [10 points] Give a polynomial time reduction from the SATISFIABILITY problem to the following LINEAR INTEGER PROGRAMMING PROBLEM

Input: A set of linear inequalities of the form $\sum_{i=1}^{n} a_i x_i \geq c$ or $\sum_{i=1}^{n} a_i x_i \leq c$, where the a_i's and c are integer constants and $x_1, x_2, \cdots x_n$ are variables.

Question: Does there exist an assignment of integers to each of the variables that makes all the inequalities true?

Justify briefly the correctness of your reduction.

b. [3 points] Can you conclude from part (a) that LINEAR INTEGER PROGRAMMING is an NP-complete problem? Justify briefly your answer.

Solution.

a. We are given an instance F of the SATISFIABILITY problem consisting of m clauses $C_1,...,C_m$ over n Boolean variables $y_1,\cdots,y_n$, where each clause is a disjunction of literals. Construct an instance L of the LINEAR INTEGER PROGRAMMING problem as follows. We have an integer variable x_j for each Boolean variable y_j, $j=1,\cdots,n$ of F. There are two inequalities for each variable and one inequality for each clause of F. The inequalities corresponding to variable y_j are $x_j \geq 0$ and $x_j \leq 1$.

The inequality corresponding to clause C_i is a follows:

$\sum\{x_j \mid y_j \text{ appears positively in } C_i\} + \sum\{(1-x_j) \mid y_j \text{ appears negated in } C_i\} \geq 1$.

We claim that F is satisfiable *iff* L has an integral solution.

(*if*) Suppose that F is satisfiable and let τ be a satisfying truth assignment. Set each variable x_j equal to 1 if y_j is true in τ and set x_j=0 otherwise. Since τ satisfies F, it follows that all inequalities of L are satisfied.

(*only if*) Suppose that L has an integral solution. Then every variable x_j is set to 0 or 1. Define a truth assignment for the variables of F by setting y_j to true if x_j=1 and to false if x_j=0. Every clause C_i of F contains a true literal because otherwise the corresponding inequality of L would be violated (the left hand side would be 0).

b. Part a is not enough to conclude that LINEAR INTEGER PROGRAMMING is NP-complete; it shows that the problem is NP-hard. For NP-completeness we need to show also that LINEAR INTEGER PROGRAMMING is in NP.

This is in fact one of the few problems where membership in NP is not completely trivial. (The reason it's nontrivial is that an argument has to be made that if there exists a solution, then there is one where all the variables are assigned integers whose number of bits is polynomially bounded in the size of the input.)

But the problem in the comp did not ask to prove actually this part.