BSc/BEng Examinations 1998 Theory of Computation (P220)

Question 1 [Marks]

(a) Draw a diagram of the Finite State Automaton represented by the following state transition table:

-1	*a*	*2*
-1	*b*	*3*
2	*a*	*1*
2	*b*	*4*
+3	*a*	*4*
+3	*b*	*1*
4	*a*	*3*
4	*b*	*2*

[20]

(b) Use Kleene's Theorem to derive a regular expression that defines the same language as that recognised by the FSA in part (a) [30]

(c) The FSA in part (a) recognises all strings over the alphabet {*a*,*b*} which contain an **even** number of *a*s and an odd number of *b*s.
Modify this FSA so that it recognises all strings over the alphabet {*a*,*b*} which contain an **odd** number of *a*s and an **even** number of *b*s. [20]

(c) Prove, using the Pumping Lemma and the closure properties of the regular languages, that the language consisting of all strings over the alphabet {*a*,*b*} which contain an **equal** number of *a*s and *b*s is **non-regular**. [30]

Question 2

The following questions relate to the language of arithmetic expressions defined by the following context-free grammar:

L -> L + L
L -> L - L
L -> (L)
L -> T
T -> a
T -> b
T -> λ

where the terminal symbols are =, -, (,), a, b and λ is the null symbol.

(a) Convert the grammar to Chomsky Normal Form. [30]

(b) Using the CNF grammar, construct a non-deterministic push-down automaton that recognises L. [30]

(d) State and explain the Pumping Lemma for non-context-free languages, showing how it derives from CNF. [40]

Question 3

(a) Construct a Turing Machine that recognises the language $\{a^m b^n \mid m>n\}$ using the following algorithm:

step 1. read an ‘a’;

step 2. run up the tape until the first ‘b’ is reached, if there is one;

step 3. DELETE that ‘b’ and the ‘a’ preceding it;

(Use, but **do not define**, the DELETE sub-machine that leaves the read head at the character preceding the one deleted.)

step 4. run back down the tape to the first blank cell and repeat from step 1.[30]

(b) Display a trace of your machine’s operation on each of the tapes:

(i) aaabb [10]

(ii) aabb [10]

(c) Give a BRIEF account of

(i) a coding of Turing Machines, and [20]

(ii) how that coding might be used to prove the undecidability of the Halting Problem. [30]