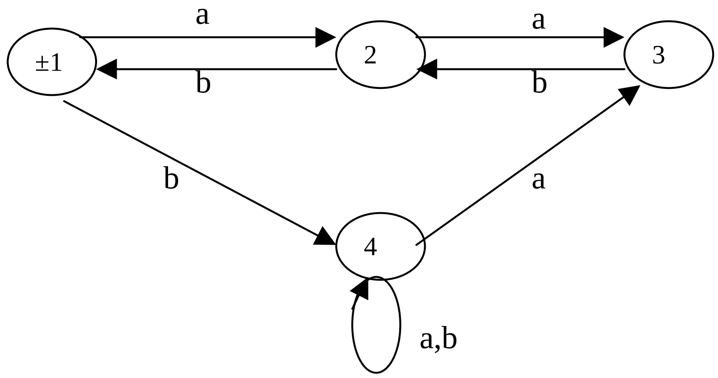
P222 Theory of Computation
Examination 1999

Answer 3 Questions

1. (a) Define the five components of a Deterministic Finite State Automaton [20]
   (b) Let M be a Deterministic Finite State Automaton, *w* be a string of symbols and L(M) be the language accepted by M . Under what circumstances is $w \in$ L(M)? [20]
   (c) Using Kleene's Theorem, derive a regular expression that defines the same language as that recognised by the following FSA: [30]



   (d) 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]

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

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]

4. (a) Define the class of languages **DTIME**(n^i) [20]

(b) Define the classes of problem ***NP***, ***NP*-hard**, and ***NP*-complete** [30]

(c)

- (i) Describe the ‘Propositional Satisfiability Problem’ and discuss its importance in relation to the question of computational tractability. [30]
- (ii) Describe two other problems that are ***NP*-complete**. [20]