BE CST 6th Semester End-Term Examinife^nTApril '2010

Department of Computer Science and Technology

Theory of Computation (CS - 002)

•F.M.: 70 TIME : 3 hrs

- Attempt question no. 1 and any five from the rest.
- Answers should **Vie** in your own words as far as practicable.
- Make your own assumptions as and when necessary and state them at proper places.

1. Write short notes on any throe from the following.

    (a) Universal Turing Machine.

    (IjJ Primitive recursive functions from strings to string.

    (c) Grammar to compute functions.

    (d) Parse tree as representation of derivations under a grammar. Ill)]

2. Many authors allow a Turing machine both to move the head and to write a symbol at the same time. For thorn, a Turing machine is a quadruple (A*. L. <*), s),* whore. A'. ^ and » have their conventional meaning, but $ is a function from A" *: to *{K* **U { / * }) >** S > *{L.R. S) (S* meaning *"stay in the* MIJHC *place").*

    (a) Define carefully the relation H between configurations for these more general machines.

    (b) Explain precisely how to convert one of these machines into a Turing machine of the standard type, and vice versa.

3. A language *L* is definite if there is some *k* such that, for any string ***u.*** whether *u? L* depends only on the last *h* symbols of ***u>.***

    (a) Rewrite the definition (for definite language) more formally.

    (b) Show that every definite language is accepted by a finite automaton.

    (c) Show that the class of definite languages is closed under union and complementation. [4*3]

4. Show that for any deterministic finite automaton *M* — (A*, S. *S. s, F).* A/ accepts an infinite language if and only if *M* accepts some string of length greater than or equal to !A'| and less than 2|A'|. [12]

5. For each of the following languages construct a Pushdown Automaton that accept the language.

    (a) (« ᵐ b ⁿ | *in <n< 2m}*

    (b) ***{^e {a.by*** I =

    ' (c) ***\i + j*** = 2A-} [4*3]

G. Fur each of the following languages construct a grammar that generates the language.

(a) ***{o".r ; v*** > 0.:r e {ft, ft}* and < «}

(b) {aWr* | i../.A' > 0 and / ^ ***j*** or j / A:}

(c) {/"'"+»' j ,i, m >0} [4>.3]

7. Show from definition that the following functions are primitive recursive.

(a) A'f : ,V* - ^ At ***j*** > 0, A'|(m, n, ...n*) - j

(b) / : .V' —* A', /('M- **»2-";i-** *ii i)* = </(2. **Jia,** *n\, m).* *g* is primitive recursive

(c) ***sg : M*** — {0.**1}.** .-Tp(«) = **1** if u = **0** and .^(f)) = 1 otherwise [5+4+3]

8. (al Propose an algorithm that, generates a Pushdown Automaton *M* — (A". I\ A. ,s, Fl for a given Context Free Grammar *G* = (V,£, **J?, 5),** such that the language generated by *G* is same as the language accepted by *M.*

(b) Let *G* - (V'.E.i?. .S') be a grammar with K = {.S.n.fc.r}, £ - *{a.b.r},* *R* = {5 — *tiStt.* *S* ftSfr, *S* —» c}. Following tlie algorithm you have proposed hi the previous step, construct a Pushdown Automaton accepting the language *L[G).* [9+3]

U. (a) "Subsets of regular languages **are** not always regular," - true or false? Justify your answer.

(h) Let *L* be a context-free language and *R* be a regular language. Is *L* - *R* necessarily context-free? What about *R* — *IP.* Formally justify your answer. [4+£]