BITS Pilani, Dubai Campus

I - Semester 2011-12

Course	: CS C351 Theory of Computation
Component	: Comprehensive Examination
Year	: III Year CS
Date	: 12-1-2012 (F.N.)
Duration	: 180 mins.
Weightage	: 40 % (40 marks)
No. of Pages	: 02 Pages

Note: Answer all the questions. Give your design without overwriting.

1. Give the finite state machine that distinguishes among the 3 input objects: identifiers, unsigned integers, and other strings. The input alphabets or set of events in I = {l, d, b, $}, where l denotes an alphabet character, d is a digit, i.e., d in {0,1,...,9}, b is the "blank" or space character, and $ represent any special character, such as @, #, or ! . An identifier is any string of alphabetic characters and surrounded by one or more blanks. Your FSM design should have a separate final state for each of the three possible input objects. (3 Marks)
2. Design a Non-deterministic finite state machine; consider a common coffee vending machine that can dispense regular and decaffeinated coffee with or without sugar and milk. After accepting the customer's coin check for the validity of the coin and remitting any change owed, the machine reads the buttons pressed by the customers for options like accept choice, dispense cup, make beverages by mixing the sugar status like with sugar or without sugar. Once the beverage is ready it displays the message "your drink is ready... help yourself". (4 Marks)
3. Convert the given regular expressions into its equivalent finite state machine design. (3 Marks each)

   i) $(a(a+b)^*a + b(a+b)^*b)^*$.

   ii) $(aa+bb)(a+b)^*(aa) + (aa+bb)\ (a+b)^*\ (bb)$
4. Given the CGF grammar rules : S → SS | XaXaX | null (3 Marks each)

   X → Xa | Xb | null

   i) Prove that XaXaX can generate any number of $b^*ab^*ab^*$

   ii) Prove that S can generate $(b^*ab^*ab^*)^*$
5. Construct a derivation tree using the grammar rules S → aa | bX | aXX and X → ab | b

   The derivation tree should accept the sting (4 Marks)

   aababbbaababaabbabababbaabababababaabbabb

6. Design and check whether the PDA design constructed using the CFG grammar rules will accept the language L={y=$a_n b_n$ | where $n_a(y)=n_b(y)$}. The CFG grammar production rules are: S → aSb | ab. (3 Marks)

7. Design a PDA for the language L = { a^n b^n, n=1,2,3} and also give the formal definition of the PDA. Use the given input string for checking the correctness of the PDA Design.
(String: aaabbb). (3 Marks)

8. If the CFG grammar rules are: S1→ S1+T | T ; T→T*a; T→ a. Using bottom-up parsing prove that the input string a + a * a $ is accepted. (3 Marks)

9. Design a Turing machine that converts the given binary input "111" to " 011". (3 Marks)

10. Design a Turing Machine which can compute the Binary 1's complement for the given input "1100110". (3 Marks)

11. Differentiate between P and NP Completeness? (2 Marks)

***** All the Best*****

BITS Pilani, Dubai Campus
I Semester 2010-11

Course	:	**CS C351 Theory of Computation**
Year	:	**III Year Computer Science**
Component	:	**Test 2 (Open Book)**
Date	:	**22-12-2011 (Thursday)**
Duration	:	**50 mins**
Weightage	:	**20% (20 Marks)** **No. of Pages: 02 Pages**

Note: Answer all the Questions.
Only Text Book and Class Notes are allowed.

1. Construct a Turing machine that accept L (a* b* c*) using 4 states (including the halt state); give the generalized Turing machine that can accept all possible types of inputs like #a#, #aaabbb#, #ab#, #aaaaccc#, #ac# and after reading the inputs from the magnetic tape storage media it has to write back the same symbols before moving to the next reading operation. (5 Marks)

2. Construct a Turing Machine that accept the Language L ($a^nb^nc^n$) using 7 states including the halt state. Give the Turing machine design and machine has to read and write the symbol back into the magnetic tape storage media as follows:
   i) Read the 1st occurrence of the symbol ***a*** and overwrite it as X and move right; if the next symbol is ***a*** then read it and overwrite the same symbol ***a*** at that place and move right to read the next symbol. If the next symbol is an ***a*** again then read it and overwrite the same symbol ***a*** at that place and move right to read the next symbol. if the next symbol is ***b*** and it is the 1st occurrence overwrite it with a symbol Y move right; if the next symbol is ***b*** then read it and overwrite the same symbol ***b*** at that place and move right to read the next symbol. If the next symbol is an ***b*** again then read it and overwrite the same symbol ***b*** at that place and move right to read the next symbol; Similarly for reading the symbol ***c*** and overwrite it as Z and move right; if the next symbol is ***c*** then read it and overwrite the same symbol ***c*** at that place and move right to read the next symbol. If the next symbol is a ***c*** again then read it and overwrite the same symbol ***c*** at that place and move right to read the next symbol.

ii) Once it reaches the extreme end of the input string and identifies a blank symbol(#) the Turing machine has to move backward to the starting location of the first symbol X and overwrite it with the same symbol and read the next input symbol, if it other than X , Y and Z the Turing reads them and changes the symbols into X | Y | Z and reaches the end of the string by identifying the # symbol.

iii) What is the function preformed by the Turing machine (after reading the input content by the read / write head) that have designed? (5 Marks)

3. Construct a PDA that accept the language L = $\{a^n b^n c^m d^m \mid n, m \geq 1\}$ using 5 states. Check your PDA design with a sample input string that satifies the given language. (5 Marks)

4. Let G have production (5 Marks)

S → S1$

S1 → S1[S1] | S1[ ] | [S1] | [ ] | ∧

Give the transition table for a deterministic bottom-up parser obtained from G. Also check that the grammar and the transition function of the PDA accepts the input string [[[[ ]]]

..........oOo..........

BITS PILANI, DUBAI CAMPUS

I SEMESTER 2011-12

Course	**:**	**CS C351 Theory of Computation**
Component	**:**	**Test 1 (Closed Book)**
Year	**:**	**III Year CS**
Date	**:**	**3-11-2011, Thursday**
Duration	**:**	**50 mins**
Weightage	**:**	**25 % (25 Marks)**

Note: **Answer all Questions. Do not overwrite the design and it should clear and neat.**

1. Check whether the given Context-Free Grammar is ambiguous or not for the input string $a^2b^2a^2$. Production rules are: S → aAS | a , A → SbA | SS | ba. Give the ambiguous Derivation Trees for the input string. (3 Marks)

2. Give a deterministic finite automaton to recognize each of the following languages. (3 * 2 = 6 Marks)

   a. L1 = {w {a,b} * | each *a* in w is immediately preceded and immediately followed by a *b*}.

   (Use 5 states)

   b. L2 = { w {a,b} * | w has both *ab* and *ba* as substrings}. (Use 6 states)

3. Construct a CFG that will generate the sting 01^40^2. Also give the formal definition for the CFG. And what is the language does the grammar represent. (3Marks)

4. Design a finite state automaton for the language given below: (3 Marks)
   L = {w | w contains equal number 0's and 1's}.

   Your design should include the state called for **reject state** to terminate the computation process if the input string does not satisfy the above language.

5. Convert the regular expression into finite automaton (0 + 11 + 01)*0*(01)* (3 Marks)

6. Design a Finite State Machine for the following problem definition: Consider the behavior and associated events of a gate at a train crossing. Suppose that there is a road crossing railway track, and a gate that opens and closes over the road. When a train approaches the crossing, the gate should close. More than one train can be crossing area at once, for example, a convoy of trains, each with a single engine and no cars. When the last train has left and the area is empty of trains, the gate should open. The gate could be in one of four states: **open, closed, opening,** and **closing.** The relevant events are: **cg** and **og** which are commands to close and open the gate, respectively; and **o-o** and **c-c** indicating from sensor input that the gate has completed opening (and thus changed from the opening to the opened state), and that the gate has completed closing, respectively. The initial state is closed and there are no stop states. The Finite state machine accepts the **cg** and **og** commands in all its states; thus, for example, the gate may be commanded to close while it is opening. (4 Marks)

7. In each case, say what language is generated by the CFG with the indicated productions:
   (a) S→ aS | aSbS | N
   (b) S→ SS | bS | a (3 Marks)
   Your answer should be more generalized while defining the Language.

--oOo—

BITS Pilani, Dubai Campus

I Semester 2011-12

Quiz – 2

B

Course	:	**CS C351 Theory of Computation**
Year	:	**III Years CS**
Duration	:	**20 mins** **(Closed Book)**
Date	:	**14-12-2011**
Marks	:	**7 Marks**

Name: ______________________ Id. No.: ______________________

1. Give the Pushdown Automata with empty store formal Definition? (1 Mark)

2. Give the PDA transition function for accepting the balanced Parenthesis and check for the input: ({ }). The Pushdown Automata is an empty store and the number of states used will be ONE. (3 Marks)

3. Convert the CFG to Pushdown Automata, if the grammar rules are (3 Marks)

$S \rightarrow E$

$E \rightarrow E+T \mid T$

$T \rightarrow T*F \mid F$

$F \rightarrow (E) \mid a$

Check whether the PDA transition functions accept the string (a+a)*(a+a).

A

BITS Pilani, Dubai Campus
I Semester 2011-12

Course: **CS C351 Theory of Computation** Date: **05-10-2011**
Year: **III Year CS** Duration: **20 mins** <u>Quiz – I</u> Max. Marks: 8 Marks

Name: ______________________________ **Id. No.:** ____________________

1. Draw Deterministic Finite Automata to accept the following sets of strings over the alphabet {0,1}:
   (i) All strings that start with 0 and has odd length or start with 1 and has even length.

   (ii) All strings that don't contain the substring 110.

   (iii) All strings that contain exactly 4 0's.

2. Draw Non-Deterministic Finite Automata to accept the following sets of strings over the alphabet {a,b,c,d}:
   Input string: caaad, caaaaaab, cad and cab

3. Give the formal definition for DFA and NFA and how do they differ each other?