INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Date 20.09.2006 FN / AN *Time:* 2 Hrs. *Full Marks* **30** *No. of Students:* **56**
Autumn Semester:, 2006 Department: Computer Science and Engineering Sub. No: CS 43001
3^{rd} **Yr. B. Tech.(Hons.)** Sub. Name: Compiler Construction

Instructions : Answer **all** the questions.

1. (a) Write context free grammar to detect strings over the alphabet set {a,b,c}; the string following the form $\{a^m b^n c^{m+n}, n \geq 0, m \geq 0\}$.
   (b) Write down the regular expressions to check a correct date of 21^{st} century in yyyy-mm-dd format.
   (c) What does the regular expression $((\epsilon|0)1^*)^*$ represent.
   (d) Write all the possible items that can be derived from the production rule A → BcdE.
   (e) What is phrase level error recovery strategy.
   (f) Write context free grammar to detect strings over a,b which contain a different number of a's to b's.
   (g) Represent the following set $\{\epsilon$, a, b, bb, bbb, $\cdots\}$ as a regular expression.
   (h) Explain the term *right sentential form*.
   (i) State the configuration of the stack in LR parser before and after a *Reduce* operation takes place.
   (j) Define Operator Grammar.
   (k) What is ϵ closure of a set s in NFA.
   (l) State 3 broad parts of a Lex program. $\frac{1}{2} \times 12$
2. (a) Construct the NFA for the expression (a|b)*a(a|b)(a|b) by using Thompson's Construction methodology.
   (b) Derive the DFA from the corresponding NFA. (3×2)
3. (a) Eliminate the ϵ production from the following grammar
   S → Aa | b
   A → Ac | Sd | ϵ
   (b) Consider the following grammar and test whether the grammar is LL(1) or not
   S → 1AB | ϵ
   A → 1AC | 0C
   B → 0S
   C → 1
   (c) Eliminate ambiguity from the following grammar
   S → iCtSeS | iCtS $(2 + 3 + 3)$

4. (a) Given the grammar
exp → exp ↑ exp
exp → exp + exp
exp → exp * exp
exp → **id**,
build up the operator precedence table and parse the input string **id1** + **id2** * **id3** ↑ **id4** ↑ **id5** using the operator precedence table. Show all the steps clearly.

(b) Given the grammar
E → L = R
E → R
L → *R
L → id
R → id,
form its SLR parsing table. (5 + 5)