**Indian Institute of Engineering Science and Technology, Shibpur**

**B.Tech (CST) 7th Semester Mid-term examination, 2021**

**Compiler Design (CS-701)**

**Full Marks: 50 Time: 45 minutes+15 minutes for uploading**

**Answer any two questions**

**1. Consider the following augmented grammar:**

**E’🡪E$**

**E🡪 E+T |T**

**T🡪 T F |F**

**F🡪 F\* |a|b**

1. **Construct FIRST and FOLLOW for all non-terminal symbols. 6**
2. **Construct LR(0) collection of items and draw LR(0) parsing machine. 6**
3. **Is there any inadequate state in the LR(0) parsing machine? 1**
4. **How will you resolve shift- reduce conflict and reduce-reduce conflict in a SLR(1) parser? 2**
5. **Construct the SLR parsing table for the grammar. 7**
6. **Discuss the relative advantages and disadvantages of SLR, Canonical LR(1) and LALR parser. 3**

**2.**

**(a) Define LL(1) grammar. 2**

**Given the grammar G(S) with the following productions**

**S🡪AaAb | BbBa**

**A🡪 Ԑ**

**B🡪 Ԑ**

1. **Construct FIRST and FOLLOW of all non-terminal symbols. 6**
2. **Show that the grammar is LL(1). 2**

**(b) Define a left recursive grammar. How will you eliminate left recursion from a context-free grammar? 2+3**

**(c) What is regular expression? How will you construct a non-deterministic finite automata from a regular expression? Why lookahead is necessary for designing a lexical analyzer? 2+6+2**

**3.**

**(a) Write an algorithm to convert a given grammar into an equivalent Ԑ -free grammar. (Hint: First determine all non-terminals that can generate the empty string.) 8**

**Apply your algorithm to the following grammar and generate equivalent Ԑ-free grammar.**

**S🡪aSbS | bSaS | Ԑ 4**

**(b) Construct both the leftmost and the rightmost derivation for the sentence abab. 4**

**(c) Construct the corresponding parse trees for abab. 4**

**(d) Why does left-recursion create a problem in Top down parsing? How can it be 2+3 eliminated?**