**VASAVI COLLEGE OF ENGINEERING (AUTONOMOUS)**

**IBRAHIMBAGH, HYDERABAD-31**

B.E 4/4 (CSE-A) I-SEMESTER

**Department of Computer Science and Engineering**

Name of the Subject: Compiler Construction

**Assignment** –I **DOS: 1-4-2023**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no** | **Question** | **Marks** | **Blooms Taxonomy** | Mapped | |
| **CO** | **PO** |
| **Set-1**  **Roll numbers 1602-21-733-013,1602-21-733-036,1602-21-733-032,1602-21-733-037,1602-21-733-012,1602-21-733-026,1602-21-733-063,1602-21-733-005,1602-21-733-020 1602-21-733-029answer the following Questions** | | | | | |
| 1 | |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | For the grammar below, a partial LL(1) parsing table is also presented along with the grammar. | | | | | | | | | | | | | Entries that need to be filled are indicated as | | | | | | | | | | | | | E1,E2,E3. @ is the empty string, $ indicates end of input,  and | separates alternate right hand sides of productions. | | | | | | | | | | | | | S->aAbB|bAaB|@ | | | | | |  |  |  |  |  | | | A->S | | | | |  |  |  |  |  |  | | | B->S | | | | |  |  |  |  |  |  | | |  | A | b | $ | | S | E1 | E2 | S->@ | | A | A->S | A->S | Error | | B | B->S | B->S | E3 | | 1. Evaluate the FIRST and FOLLOW sets for the   non-terminals A and B | | | | | | | | | | | | | 1. Mark the appropriate entries for E1,E2,E3 | | | | | | | | | |  |  | | 2 | 3 | 2 | 1 |
| 2 | Eliminate left recursion from the following grammar and construct LL(1) parser and parse the string *(a,a)*  S->(L)/a  L->L,S /S | 2 | 3 | 2 | 1,2 |
| 3 | Test whether the grammar below is   1. CLR(1) 2. LL(1) 3. SLR(1)   S->Aa/bAc/dc/bda  A->d | 1 | 3 | 2 | 1,2 |
| **Set-2**  **Roll numbers 1602-21-733-001,1602-21-733-002,1602-21-733-003,1602-21-733-004,1602-21-733-006,1602-21-733-007,1602-21-733-008 answer the following Questions** | | | | | |
| 1 | Find the first and follow set for the following grammar.  S->As  A->BCA  A->BCa  B->b  C-> c  And Show that the grammar is not LL(1) | 1 | 3 | 2 | 1 |
| 2 | Test whether the grammar below is   1. LL(1) 2. SLR(1)   S->dA/aB  A->bA/c  B->bB/c | 2 | 3 | 2 | 1 |
| 3 | Test whether the grammar below is   1. CLR(1) 2. LALR(1)   S->AA  A->aA/b | 2 | 3 | 2 | 1,2 |
| **SET-3 Roll numbers 1602-21-733-009 1602-21-733-010 1602-21-733-011 1602-21-733-014 1602-21-733-015 1602-21-733-016 1602-21-733-017 answer the following Questions** | | | | | |
| 1 | Test whether the grammar below is   1. LL(1) 2. SLR(1) 3. CLR(1)   S->AaAb/BbBa  A->ℇ  B-> ℇ | 1 | 3 | 2 | 1 |
| 2 | Construct bottom-up parsers for the following input string and grammar:  Input String: 000111  Grammar: S-> 0S1/01 | 2 | 3 | 2 | 1 |
| 3 | Construct LL(1) parsing table for grammar  E->TE’  E’->+TE’ / ℇ  T->FT’  T’-> \*FT’/ ℇ  F-> E/ id  Show the error recovery moves made for input : (id++id)\*id++id | 2 | 3 | 2 | 1,2 |
| **SET-IV Roll numbers 1602-21-733-018 1602-21-733-019 1602-21-733-021 1602-21-733-022 1602-21-733-023 1602-21-733-024 1602-21-733-025 answer the following Questions** | | | | | |
| 1 | Test whether the grammar below is   1. LL(1) 2. SLR(1) 3. CLR(1)   S->AS/b  A->SA/a | 1 | 3 | 2 | 1 |
| 2 | Construct CLR parser using the following grammar and parse (id+id)\*id string  E🡪E+T /T  T🡪 T\*F /F  F🡪(E)/id | 2 | 3 | 2 | 1 |
| 3 | Construct LALR parsing table for the following grammar  E🡪E+T /T  T🡪 T\*F /F  F🡪(E)/id | 2 | 3 | 2 | 1,2 |
| **SET-V Roll numbers 1602-21-733-027 1602-21-733-028 1602-21-733-030 1602-21-733-031 1602-21-733-033 1602-21-733-034 1602-21-733-035 answer the following Questions** | | | | | |
| 1 | Test whether the grammar below is   1. LL(1) 2. SLR(1)   E->E+T/T  T->TF/F  F->F\*/a/b | 1 | 3 | 2 | 1 |
| 2 | Test whether the grammar below is   1. CLR(1) 2. LALR(1)   S->Aa/bAc/Bc/bBa  A->d  B->d | 2 | 3 | 2 | 1 |
| 3 | Construct LL(1) parsing table for grammar  E->TE’  E’->+TE’ / ℇ  T->FT’  T’-> \*FT’/ ℇ  F-> E/ id  Show the error recovery moves made for input : (id++id)\* | 2 | 3 | 2 | 1,2 |
| **SET-VI Roll numbers 1602-21-733-038 1602-21-733-039 1602-21-733-040 1602-21-733-041 1602-21-733-042 1602-21-733-043 1602-21-733-044 answer the following Questions** | | | | | |
| 1 | Construct LL(1) parsing table for grammar  E->TE’  E’->+TE’ / ℇ  T->FT’  T’-> \*FT’/ ℇ  F-> E/ id  Show the error recovery moves made for input : (id++id)\*id++id | 1 | 3 | 2 | 1 |
| 2 | Test whether the grammar below is   1. CLR(1) 2. LALR(1)   S->AS/b  A->SA/a | 2 | 3 | 2 | 1 |
| 3 | Construct SLR parser for the below grammar and parse id+id\*id string  E🡪E+T /T  T🡪 T\*F /F  F🡪(E)/id | 2 | 3 | 2 | 1,2 |
| **SET-VII Roll numbers 1602-21-733-046 1602-21-733-047 1602-21-733-048 1602-21-733-049 1602-21-733-051 1602-21-733-052 1602-21-733-053 answer the following Questions** | | | | | |
| 1 | Test whether the grammar below is   1. LL(1) 2. SLR(1) 3. CLR(1)   E->E+T/T  T->T\*F/F  F->i | 1 | 3 | 2 | 1 |
| 2 | Construct the FIRST and FOLLOW for the following grammar  S->aBDh  B->cC  C->bC/ ℇ  D->EF  E->g/ ℇ  F->f/ ℇ | 2 | 3 | 2 | 1 |
| 3 | Test whether the grammar below is   1. CLR(1) 2. LALR(1)   S->AA  A->aA/b | 2 | 3 | 2 | 1,2 |
| **SET-VIII Roll numbers 1602-21-733-054 1602-21-733-055 1602-21-733-056 1602-21-733-057 1602-21-733-058 1602-21-733-059 1602-21-733-060 answer the following Questions** | | | | | |
| 1 | Test whether the grammar below is   1. LL(1) 2. SLR(1) 3. CLR(1)   S->A  A->AB/ℇ  B->aB/b | 1 | 3 | 2 | 1 |
| 2 | Construct bottom-up parsers for the following input string and grammar:  Input String: 000111  Grammar: S-> 0S1/01 | 2 | 3 | 2 | 1 |
| 3 | Test whether the grammar below is   1. LL(1) 2. SLR(1) 3. CLR(1) 4. LALR(1)   S->Aa/bAc/Bc/bBa  A->d  B->d | 2 | 3 | 2 | 1,2 |
| **SET-IX Roll numbers 1602-21-733-061 1602-21-733-062 1602-21-733-064 1602-21-733-065 1602-21-733-066**  **1602-21-733-067 1602-21-733-135 answer the following Questions** | | | | | |
| 1 | Test whether the grammar below is   1. LL(1) 2. SLR(1) 3. CLR(1) 4. LALR(1)   S->Aa/bAc/Bc/bBa  A->d  B->d | 1 | 3 | 2 | 1 |
| 2 | Construct the FIRST and FOLLOW for the following grammar  S->ACB/CbB/Ba  A->da/BC  B->g/ ℇ  c->h/ ℇ | 2 | 3 | 2 | 1 |
| 3 | Construct bottom-up parsers for the following input string and grammar:  Input String: aaa\*a++  Grammar: S-> SS+/SS\*/a | 2 | 3 | 2 | 1,2 |
| **SET-X Roll numbers 1602-21-733-136 1602-21-733-301 1602-21-733-302 1602-21-733-303 1602-21-733-304 1602-21-733-305 1602-21-733-306 1602-21-733-307 answer the following Questions** | | | | | |
| 1 | Consider the grammar with non-terminals N={S,C,S1} , terminals  T={a,b,i,t,e} , with S as the Start symbol and the following set of production rules.  S-> iCtSS1 |a  S1->eS | @  C->b  Test whether this grammar is LL(1)? | 1 | 3 | 2 | 1 |
| 2 | Test whether the grammar below is   1. LL(1) 2. SLR(1)   E->E+T/T  T->i | 2 | 3 | 2 | 1 |
| 3 | Test whether the grammar below is   1. CLR(1) 2. LALR(1)   S->AaAb/BbBa  A-> ℇ  B-> ℇ | 2 | 3 | 2 | 1,2 |