**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:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Question** | **Marks** | **Blooms Taxonomy** | Mapped | | |
| **CO** | **PO** | |
| **Set-1**  **Roll numbers 1602-21-733-013,1602-21-733-036,1602-21-733-032,1602-21-733-037,1602-21-733-012,1602-21-733-026,1602-21-733-063,1602-21-733-005,1602-21-733-0201602-21-733-029answer the following Questions** | | | | | | |
| 1 | Define Tokens and identify the number of tokens in the following input.  int max(x,y)  int x,y; /\*find maximum of x and y \*/  {  return(x>y? x:y);  } | 1 | 3 | 1 | 1,2 | |
| 2 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  S=(a+b+c)/2  Assume that all variables are double. | 1 | 3 | 1 | 1,2 | |
| 3 | Consider two binary operators ‘’ and ‘↓’ with the precedence of operator ↓ being lower than that of the operator . Operator is right associative while ↓ is left associative. Then draw the parse tree for the following expression.  (7↓343↓2) | 2 | 3 | 1 | 1,2 | |
|  | Write a lex program to add line numbers to each line in a file | 1 | 3 | 1 | 1,2 | |
| **Set-2**  **Roll numbers 1602-21-733-001,1602-21-733-002,1602-21-733-003,1602-21-733-004,1602-21-733-006,1602-21-733-007,1602-21-733-008answer the following Questions** | | | | | | |
| 1 | Define Tokens and identify the number of tokens in the following input.  printf(“%d”, &x); | 1 | 3 | 1 | 1,2 | |
| 2 | Write regular expressions for the following informally described languages:  a. All strings of a’s and b’s with the subsequence abb.  b. All strings of a’s and b’s with an even number of a’s and an odd number of b’s. | 2 | 3 | 1 | 1,2 | |
| 3 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  X=25+(b+c/d\*e)/(f\*g-h\*i)  Assume that all variables are double | 1 | 3 | 1 | 1,2 | |
| 4 | Let L be a language defined over = {0, 1} and L consists of all strings with the same, and even number of 0’s and 1’s.  a. Give a context free grammar for L.  b. Show a parse tree for the string 01110100.  c. Give the leftmost derivation the string 01110100. | 1 | 3 | 1 | 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 | Define Tokens and identify the number of tokens in the following input.  printf(“Total = %d\n”, score); | 1 | 3 | 2 | 1,2 | |
| 2 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  a=b+c\*2;  Assume that all variables are double. | 2 | 3 | 1 | 1,2 | |
| 3 | Consider the following grammar defined over = {0, 1}.  S->0S11  S->S1  S->@  (a) Briefly describe the language generated by this grammar.  (b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways and showing the two corresponding parse trees.  (c) Rewrite the grammar to eliminate the ambiguity | 1 | 3 | 1 | 1,2 | |
|  | Eliminate left recursions for the following grammar.  S->A E B  A->Ax | Ay | Ba | a  E->=  B->Ab | b | 1 | 3 |  |  | |
| **SET-IVRoll 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 | Define Tokens and identify the number of tokens in the following input.   * 1. if(a == b) a=x+y; | 1 | 3 | 1 | 1,2 | |
| 2 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  X=25+e/(f-h\*i)  Assume that all variables are double. | 2 | 3 | 1 | 1,2 | |
| 3 | Consider the following grammar defined over = {a, b}.  S->aS  S->Sa  S->a  (a) Briefly describe the language generated by this grammar.  (b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways and showing the two corresponding parse trees.  (c) Rewrite the grammar to eliminate the ambiguity. | 1 | 3 | 1 | 1,2 | |
| 4 | Consider the following grammar defined over = {True, False}.  bExp->bExp or bExp | bExp and bExp | not bExp | True | False (a) Briefly describe the language generated by this grammar.  (b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways and showing the two corresponding parse trees.  (c) Rewrite the grammar to eliminate the ambiguity | 1 | 3 | 1 | 1,2 | |
| **SET-VRoll 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 | Define Tokens and identify the number of tokens in the following input.  delta = epsilon + 1.23 sqrt(ZZ); | 1 | 3 | 2 | 1,2 | |
| 2 | Discuss various phases of compiler. Explain the result of each phase for the example  Position=initial+ rate\*60  Assume that all variables are double. | 1 | 3 | 1 | 1,2,3 | |
| 3 | Let L be a language defined over = {0, 1} and L consists of all strings with the same, and even number of 0’s and 1’s. a. Give a context free grammar for L. b. Show a parse tree for the string 01110100. c. Give the leftmost derivation for (b) | 1 | 3 | 1 | 1,2 | |
| 4 | Consider the context-free grammar: S->S(S)S|@ and the string ( () () )  a. Give a left most derivation for the string  b. Give a right most derivation for the string  c. Give a parse tree for the string  d. Describe the language generated by this grammar | 2 | 3 | 1 | 1,2 | |
| **SET-VIRoll 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 | Define Tokens and identify the number of tokens in the following input.  print(q); | 1 | 3 | 1 | 1,2 | |
| 2 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  X=25/(g-h)  Assume that all variables are double.. | 1 | 3 | 1 | 1,2 | |
| 3 | Consider the following grammar. Note that id, +, [, ], and “,” are terminals. E-> E + T | T T-> id | id[] | id[X] X E-> , E | E (a) Eliminate left recursion in the grammar. (b) Perform left factoring for the grammar. | 2 | 3 | 1 | 1,2 | |
| 4 | Eliminate left recursions for the following grammar.  S->A E B  A->Ax | Ay | Ba | a  E->=  B->Ab | b | 1 | 3 | 1 | 1,2 | |
| **SET-VIIRoll 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 | Define Tokens and identify the number of tokens in the following input.  printf(“%d”, a); | 1 | 3 | 1 | 1,2 | |
| 2 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  X=(b+d\*5)/(h\*i)  Assume that all variables are double. | 1 | 3 | 1 | 1,2 | |
| 3 | Consider the following grammar defined over = {a, b}.  S->aS  S->Sa  S->a  (a) Briefly describe the language generated by this grammar.  (b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways and showing the two corresponding parse trees.  (c) Rewrite the grammar to eliminate the ambiguity. | 1 | 3 | 1 | 1,2 | |
| 4 | Explain algorithms to find FIRST and FOLLOW and find FIRST and FOLLOW of following grammar  S🡪aBbSA/d  A🡪eS/ ε  B🡪 f  .. | 2 | 3 | 1 | 1,2 | |
| **SET-VIIIRoll 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 1602-21-733-060answer the following Questions** | | | | | | |
| 1 | Define Tokens and identify the number of tokens in the following input.  printf("i = %d, &i = %x", i, &i); | 1 | 3 | 1 | 1,2 | |
| 2 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  S=(a+b+c)/2  Assume that all variables are double. | 1 | 3 | 1 | 1,2 | |
| 3 | Consider the following grammar S->aSbS S->bSaS S->@ (a) Briefly describe the language generated by this grammar. (b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways and showing the two corresponding parse trees. (c) Rewrite the grammar to eliminate the ambiguity. | 1 | 3 | 2 | 1,2,3 | |
| 4 | Generate LL(1) Passing table for grammar  E🡪TE1  E1🡪 +TE1 / ε T🡪FT1 T1🡪 \*FT1/ ε F🡪(E)/id | 2 | 3 | 2 | 1,2,3 | |
| **SET-IXRoll numbers 1602-21-733-0611602-21-733-0621602-21-733-0641602-21-733-0651602-21-733-066**  **1602-21-733-0671602-21-733-135answer the following Questions** | | | | | | |
| 1 | Define Tokens and identify the number of tokens in the following input.  int main()  {  int a = 10, b = 20;  printf("sum is :%d",a+b);  return 0;  } | 1 | 3 | 2 | 1,2 | |
| 2 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  S=(a+b+c)/2  Assume that all variables are double. | 1 | 3 | 2 | 1,2,3 | |
| 3 | Consider the context-free grammar: S->SS+ | SS\* | a and the string aa+a\* a)Give a left most derivation for the string b) Give a right most derivation for the string c) Give a parse tree for the string d) Describe the language generated by this grammar | 2 | 3 | 1 | 1,2 | |
| 4 | Show that following grammar in not LL(1)  S🡪 iEtSS1/a  S1🡪eS/ ε  E🡪 b | 1 | 3 | 1 | 1,2 | |
| **SET-XRoll numbers 1602-21-733-1361602-21-733-3011602-21-733-3021602-21-733-3031602-21-733-3041602-21-733-3051602-21-733-3061602-21-733-307answer the following Questions** | | | | | | |
| 1 | Describe the phases of a complier and show the transformation that takes after each phase in translating the statement:  S=(a+b+c)/2  Assume that all variables are double. | 1 | 3 | 2 | 1,2,3 | |
| 2 | Consider the following grammar defined over = {a, b}. R->R+R R->RR R->a/b/c (a) Briefly describe the language generated by this grammar. (b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways and showing the two corresponding parse trees. (c) Rewrite the grammar to eliminate the ambiguity | 1 | 3 | 1 | 1,2,3 | |
| 3 | Consider the following grammar for symbols a and b rexpr->rexpr + rterm | rtermrterm->rtermrfactor | rfactorrfactor->rfactor \* |rprimaryrprimary ->a|b a) Left factor this grammar b) Does left factoring make the grammar suitable for top-down parsing? c) Eliminate left recursion from the original grammar | 1 | 3 | 1 | 1,2 | |
| 4 | Consider the following grammar defined over = {id}. E->E+E E->E\*E E->id (a) Briefly describe the language generated by this grammar. (b) Show that this grammar is ambiguous by giving a string that can be parsed in two different ways and showing the two corresponding parse trees. (c) Rewrite the grammar to eliminate the ambiguity. | 2 | 3 | 1 | 1,2 | |