Unit 6

Parallel and Distributed Compilers

Q 1. Explain Parallel Programming models in details.

Q 2. Explain message passing in details

Q 3. What is process. Explain Process control block with neat diagram.

Q.4 Explain thread and its types.

Q 5. Write note on shared variable.

Q6. Explain marshaling and unmarshaling with respect to message passing.

Q7. What are the different issues in implementing tuple space. How to handle these issues.

Q 8. Explain dynamic compilation with its advantages and disadvantages.

Q 9.Write note on Java Virtual Machine.

Q 10. Explain in brief

1. GCC
2. g++

Q 11.Explain following commands with examples.

a)Cmake

b)Nmake

Q 12. Write note on NVCC.

UNIT I

Notion and Concepts

Q.1 With a neat diagram, explain various phases of a compiler. List various compiler writing tool

you know.

Q. 2 For following source language statement, show the output at each of the phases of a

Compiler

P=I+R\*60 Where P,I,and R are of float type.

Q .3 What is interpreter? Discuss its merits and demerits.

Q. 4 What is difference between a phase and a pass of compiler? Explain machine dependant and

Machine independent phases of compiler.

Q. 5 Explain different types of compiler.

Q. 6 Why compilation phases are divided into front-end and back-end? What are the advantages?

Q 7. Discuss the significance of lexemes with longest prefix with suitable example.

Q. 8 Write lex program to count the number of characters, words and lines in a given input text

File. Create an output file that consist of the contents of the input file as well as line

Numbers.

Q. 9 With suitable example regular expression, explain the meaning of special symbol in lex that

are used to specify tokens

Q. 10 Write lex program to count the number of words in input c file. Program also should

remove comments from the program.

UNIT I

PARSING

Q 1 Discuss the problems in top-down parsing. With suitable examples, explain how they can be

Overcome?

Q. 2 Construct LL(1) parser table for the following CFG

S-> aBDh

B->cC

C->bC|ε

D->EF

E->g| ε

F->f | ε

Q. 3 with suitable example, explain the mechanism used in YACC to resolve parser table

conflicts. Why shift-reduce conflict never occurs in LR parser?

Q. 4 Construct LL(1) parser table for following grammar

S-> iCtS | iCtSeS | a

C-> b

Is the grammar LL(1)?

Q. 5 Construct the LALR parser table for Following grammar.

S-> iCtS | iCtSeS | a

C-> b

Q. 6 Test whether the following grammar is LL(1) or not? Construct LL(1) parser table.

S ->AB | gDa

A-> ab | c

B-> dC

C-> gC | g

D-> fD | g

Q. 7 Construct SLR parsing table for the grammar given below.

S ->AB | gDa

A-> ab | c

B-> dC

C-> gC | g

D-> fD | g

Q 8 Construct LALR parsing table for the grammar given below.

S-> Aa | bAc | bc | bda

A->d

Q 9 Explain in brief the function of an operator precedence parser.

UNIT III

SYNTAX TRANSLATION SCHEMES

Q.1 Write an attributed translation grammar to parse and translate a given infix arithmetic expression into a syntax tree. Write LEX and YACC specification to implement your

syntax directed translation scheme.

Q. 2 What is marker non-terminal? Explain their significance with suitable examples.

Q. 3 Explain why every S-attributed definition is L-attributed?

Q 4. Explain following terms with suitable examples:

a)Inherited attribute b)Synthesized attribute c)Marker non terminal symbol

Q. 5 If declaration are generated by following grammar:

D->idL

L-> idL

L-> T

L->integer

T->real

Construct a translation scheme to enter the of each identifier into symbol table.

Q. 6 Explain what is inherited and synthesized attributes. With example explain how these are

Calculated.

Q.7 Write 3 address code statements for following source language statements

a) X=fun(2,Y-1) + Z, where FUN is function

b) A[B]=C[D[2]]+E\*F, where A, C, D are array variables.

Q 8. Write syntax directed definition to translate ‘switch’ statement. With suitable example

,show the translation of the source language ‘switch’ statement.

Q. 9 Write translation scheme to generate intermediate code for assignment statement with array

references.

Q. 10 List the commonly used intermediate representations. Write following expression in all

types of intermediate representations.

1. (a-b)\*(c+d)-(a+b)
2. A \* -(B+C)

Q. 11 Generate three address code for following program

While (A< C and B > D) do

if(A==3 ) then C=C+1

else

while (A<= D) do

A=A+3

Q. 12 Explain the technique of “Backpatching” for translation of flow-of-control statement:

If-then, if-then-else and while do.

Q.13.Generate 3 address code for following program fragment, where a and b are integer arrays

of size 25 x 40 each, and there are four bytes per word. Variables add, i and j are integer

and are defined.

Main()

{

add=0 ; i=j=1:

do

{

Add=add+[i][j]\*b[j][i];

i=i+1; j=j+1;

}

While (i <=25 && j<= 40);

}

UNIT IV

CODE GENERATION AND OPTIMIZATION

Q. 1 Explain dynamic programming code generation algorithm.

Q. 2 Generate code for following C program using any code generation algorithm you know.

Main()

{

int j;

int a[10];

while (j<=10)

a[j]=0;

}

Q. 3 What do you mean by ‘next-use’ information? How it is computed?

Q. 4 Explain code generation algorithm.

Q. 5 Write a note on application of DAG in code generation.

Q. 6 What are the issue in code generation process? Explain in details.

Q. 7 Explain the concept of register allocation and assignment.

Q. 8 Generate the target code using labeled tree for following 3 address code. Assume the target

machine has two registers r1 and r2. Generate the target code assuming only one register

‘r’ is available in the target machine.