

Compiler Construction
IInd Assignment

Short answer Questions

1. Explain how dynamic allocation is done.
2. What is an activation record? Discuss its structure.
3. What is Data Flow Analysis?
4. Define Induction variable.
5. Explain the error recovery strategies in syntax analysis.
6. What are basic blocks and flow graphs?
7. What are the properties of optimizing compilers?
8. What are the advantages of DAG Representation? Give example
9. What do you mean by machine dependent and machine independent optimization?

Long answer Questions

1. Convert the following program into 3 address code, and optimize it if required.

```
main()
{
  int a[10], i;
  for (i=0;i<10;i++)
    a[i]=i*2;
}
```

2. Convert the following program into 3 address code

```
X=0
do
{
  A=B-C*D;
  X=X+1;
}
while (X<6)
```

3. Check whether the following grammar is SLR (1) or not. Explain your answer with reasons.

$$S \rightarrow L = R \mid R$$
$$L \rightarrow *R \mid id$$
$$R \rightarrow L$$

4. Check whether the following grammar is SLR (1) or not. Explain your answer with reasons.

$$A \rightarrow aAa \mid bAb \mid ba$$

5. Check whether the following grammar is SLR (1) or not. Explain your answer with reasons.

$$S \rightarrow xAy \mid xBy \mid xAz$$
$$A \rightarrow aS \mid b$$
$$B \rightarrow b$$

6. . Check whether the following grammar is a) SLR (1) b) LALR(1) c) LR (1) / or not

$$\begin{aligned} S &\rightarrow AA \\ A &\rightarrow aA \\ A &\rightarrow b \end{aligned}$$

7. Check whether the following grammar is a) SLR (1) b) LALR(1) c) LR (1) / or not

$$\begin{aligned} S &\rightarrow Ba \mid bBc \mid dc \mid bda \\ B &\rightarrow d \end{aligned}$$

8. Explain in detail about the various issues in design of code generator.

9. Explain code generation phase with simple code generation Algorithm.

10. Explain the usage of YACC parser generator in construction of a Parser.

11. Explain the following with example:

a) Quadqaples b) Triples c) Indirect triple

12. Write a short note on:

- syntax-directed translation
- peephole optimization
- dynamic allocation
- static allocations and stack allocations
- machine dependent and machine independent optimization

* * * * *