shwetatiwari08@recabn.ac.in
shwetatiwari08aug@gmail.com

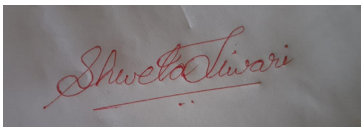
CD: UNIT-3,4
UNIT-3 SYNTAX-DIRECTED TRANSLATION
UNIT-4 SYMBOL TABLE

FALL SEMESTER, YEAR (V/VI, 3rd)

Published: August, 2022

FALL SESSION: 2022-23

—



UNIT-3&4 CD
COMPILER DESIGN (KIT-052)

UNIT III – SYNTAX-DIRECTED TRANSLATION

UNIT IV – SYMBOL TABLE

PREPARED FOR
Engineering Students
All Engineering College

PREPARED BY
SHWETA TIWARI

Guest Faculty

Rajkiya Engineering College | Ambedkar Nagar, UP, India

Faculty Name: Miss. Shweta Tiwari, Subject: CD-COMPILER DESIGN (KIT-052), Year- 3rd Year, Semester- 5th/6th Sem, Branch- CS and IT, Session- Odd/Even Semester (2022-23)

CD-COMPILER DESIGN (KIT-052)

This **(Question Bank Part-1 Easy to Advanced Level (Belong your Syllabus))** only attempts to discover some questions with proper explanation that can be generated in "CD-COMPILER DESIGN" with the answer to all these questions. There can be some errors to these answers. If you find any errors then please do write to us.

CD-COMPILER DESIGN (KIT-052)

UNIT III – SYNTAX-DIRECTED TRANSLATION

UNIT IV – SYMBOL TABLE

(SYNTAX DIRECTED TRANSLATION & RUN TIME ENVIRONMENT)

1. List the different storage allocation strategies.

The strategies are:

- Static
- allocation
- Stack
  allocation
  Heap
  allocation

2. What are the contents of the activation record?

- The activation record is a block of memory used for managing the information
- needed by a single execution of a procedure. Various fields f activation record
- are:
- Temporary
- variables Local
- variables
- Saved machine
  registers Control link
  Access link
  Actual
  parameters
  Return values

3. What is dynamic scoping?

In dynamic scoping a use of non-local variables refers to the non-local data declared in the most recently called and still active procedure. Therefore each time new findings are set up for local names called procedure. In dynamic scoping symbol tables can be required at run time.

4. Define symbol table.

Symbol table is a data structure used by the compiler to keep track of semantics of the variables. It stores information about scope and binding

information about names.

5. **What is code motion?**

   Code motion is an optimization technique in which the amount of code in a loop is decreased. This transformation is applicable to the expression that yields the same result independent of the number of times the loop is executed. Such an expression is placed before the loop.

6. **What are the properties of optimizing a compiler?**

   The source code should be such that it should produce a minimum amount of target code.
   There should not be any unreachable code.
   Dead code should be completely removed from source language.
   The optimizing compilers should apply following code improving transformations on source language.

   i) common subexpression elimination
   ii) dead code elimination
   iii) code movement
   iv) strength reduction

7. **What are the various ways to pass a parameter in a function?**

- Call by value
- Call by
- reference
  Copy-restore
  Call by name

8. **Suggest a suitable approach for computing hash function.**

Using a hash function we should obtain exact locations of names in the symbol table. The hash function should result in uniform distribution of names in the symbol table.
The hash function should be such that there will be a minimum number of collisions. Collision is such a situation where a hash function results in the same location for storing the names.

9. **Define bottom up parsing?**

   It attempts to construct a parse tree for an input string beginning at leaves and working up towards the root (i.e.) reducing a string „w" to the start symbol of a grammar. At each reduction step, a particular substring matching the right side of a production is replaced by the symbol on the left of that production. It is a rightmost derivation and it" s also known as shifts reduce parsing.

10. **What are the functions used to create the nodes of syntax trees?**

- Mknode (op, left, right)

- Mkleaf (id,entry)
- Mkleaf (num, val)

11. What are the functions for constructing syntax trees for expressions?

i) The construction of a syntax tree for an expression is similar to the translation of the expression into postfix form.

ii) Each node in a syntax tree can be implemented as a record with several fields.

12. Give a short note about call-by-name?

Call by name, at every reference to a formal parameter in a procedure body the the name of the corresponding actual parameter is evaluated. Access is then made to the effective parameter.

13. How parameters are passed to procedures in call-by-value method?

This mechanism transmits values of the parameters of call to the called program. The transfer is one way only and therefore the only way to return can be the value of a function.

```
Main ( )
{ print (5);
  } Int
  Void print (int n)
   { printf (“%d”, n); }
```

14. Define static allocations and stack allocations

Static allocation is defined as laid out for all data objects at compile time. Names are bound to storage as a program is compiled, so there is no need for a run time support package.

Stack allocation is defined as a process which manages the run time as a Stack. It is based on the idea of a control stack; storage is organized as a stack, and activation records are pushed and popped as activations begin and end.

15. What are the difficulties with top down parsing?

a) Left recursion
b) Backtracking
c) The order in which alternates are tried can affect the language accepted
d) When failure is reported. We have very little idea where the error actually occurred.

16. Define top down parsing?

It can be viewed as an attempt to find the leftmost derivation for an input string. It can be viewed as attempting to construct a parse tree for the input starting from the

root and creating the nodes of the parse tree in preorder.

17. Define a syntax-directed translation?

Syntax-directed translation specifies the translation of a construct in terms of Attributes associated with its syntactic components. Syntax-directed translation uses context free grammar to specify the syntactic structure of the input. It is an input- output mapping.

18. Define an attribute. Give the types of an attribute?

An attribute may represent any quantity, with each grammar symbol, it associates a set of attributes and with each production, a set of semantic rules for computing values of the attributes associated with the symbols appearing in that production. **Example:** a type, a value, a memory location etc.,

i) Synthesized attributes.

ii) Inherited attributes.

19. Give the 2 attributes of syntax directed translation into 3-address code?

i) E.place, the name that will hold the value of E and

ii) E.code , the sequence of 3-addr statements evaluating E.

20. Write the grammar for flow-of-control statements?

The following grammar generates the flow-of-control statements, if-then, if- then-else, and while-do statements.

S -> if E
then S1
| If E then
S1 else
S2
| While E do S1.

PART B

1) Discuss in detail about the run time storage arrangement.
2) What are different storage allocation strategies? Explain.
3) Write in detail about the issues in the design of the code generator.
4) Explain how declarations are done in a procedure using syntax directed translations.
5) What is a three address code? Mention its types. How would you implement these address statements? Explain with suitable examples.
6) Write syntax directed translation for arrays.

**********THE END**********