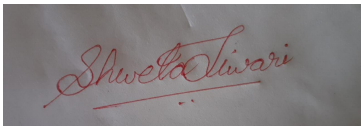
shwetatiwari08@recabn.ac.in
shwetatiwari08aug@gmail.com

CD: UNIT-2
BASIC PARSING TECHNIQUES
FALL SEMESTER, YEAR (V/VI, 3rd)

Published: August, 2022

FALL SESSION: 2022-23



UNIT-2 CD
COMPILER DESIGN (KIT-052)

UNIT II – BASIC PARSING TECHNIQUES

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 II BASIC PARSING TECHNIQUES - *(SYNTAX ANALYSIS)*

PART A

1. **Define parser.**

Hierarchical analysis is one in which the tokens are grouped hierarchically into nested collections with collective meaning.
Also termed as Parsing.

2. **Mention the basic issues in parsing.**

There are two important issues in parsing.

- Specification of syntax
- Representation of input after parsing.

3. **Why are lexical and syntax analyzers separated out?**

Reasons for separating the analysis phase into lexical and syntax analyzers:

- Simpler design.
- Compiler efficiency is improved.
- Compiler portability is enhanced.

4. **Define context free grammar.**

A context free grammar G is a collection of the following

- V is a set of non terminals
- T is a set of terminals
- S is a start symbol
- P is a set of production rules

G can be represented as G = (V,T,S,P)

Production rules are given in the

following form Non terminal → (V U T)*

5. **Briefly explain the concept of derivation.**

Derivation from S means generation of string w from S. For constructing derivation two things are important.

i) Choice of non terminal from several others.

ii) Choice of rule from production rules for corresponding non terminal.

Instead of choosing the arbitrary non terminal one can choose

i) either leftmost derivation – leftmost non terminal in a sentinel form

ii) or rightmost derivation – rightmost non terminal in a sentinel form

6. **Define ambiguous grammar.**

A grammar G is said to be ambiguous if it generates more than one parse tree for some sentence of language L(G).

i.e. both leftmost and rightmost derivations are the same for the given sentence.

7. **What is an operator precedence parser?**

A grammar is said to be operator precedence if it possess the following properties:

1. No production on the right side is ε.
2. There should not be any production rule possessing two adjacent non terminals at the right hand side.

8. **List the properties of LR parser.**

1. LR parsers can be constructed to recognize most of the programming languages for which the context free grammar can be written.
2. The class of grammar that can be parsed by LR parser is a superset of a class of grammars that can be parsed using predictive parsers.
3. LR parsers work using non backtracking shift reduce technique yet it is an efficient one.

9. **Mention the types of LR parser.**

- SLR parser- simple LR parser
- LALR parser- lookahead LR parser
- Canonical LR parser

10. What are the problems with top down parsing?

The following are the problems associated with top down parsing:

- Backtracking
- Left recursion
- Left factoring
- Ambiguity

11. Write the algorithm for FIRST and FOLLOW. FIRST

1. If X is terminal, then FIRST(X) IS {X}.
2. If X → ε is a production, then add ε to FIRST(X).
3. If X is non terminal and X → Y1,Y2..Yk is a production, then place a in FIRST(X) if for some i , a is in FIRST(Yi) , and ε is in all of FIRST(Y1),...FIRST(Yi-1);

FOLLOW

1. Place $ in FOLLOW(S),where S is the start symbol and $ is the input right endmarker.
2. If there is a production A → αBβ, then everything in FIRST(β) except for ε is placed in FOLLOW(B).
3. If there is a production A → αB, or a production A→ αBβ where FIRST(β) contains ε , then everything in FOLLOW(A) is in FOLLOW(B).

12. List the advantages and disadvantages of operator precedence parsing. Advantages

This typeof parsing is simple to implement.

Disadvantages

1. The operator like minus has two different precedence(unary and binary).Hence it is hard to handle tokens like minus sign.
2. This kind of parsing is applicable to only a small class of grammars.

13. What is the dangling else problem?

Ambiguity can be eliminated by means of dangling-else grammar which is

show below: stmt → if expr then stmt

| if expr then stmt else stmt

| other

14. Write short notes on YACC.

YACC is an automatic tool for generating the parser program.

YACC stands for Yet Another Compiler Compiler which is basically the utility available from UNIX.

Basically YACC is a LALR parser generator.

It can report conflict or ambiguities in the form of error messages.

15. What is meant by handle pruning?

A rightmost derivation in reverse can be obtained by handle pruning.

If w is a sentence of the grammar at hand, then w = γn, where γn is the nth right- sentential form of some as yet unknown rightmost derivation

S = γ0 => γ1...=> γn-1 => γn = w

16. Define LR(0) items.

An LR(0) item of a grammar G is a production of G with a dot at some position of the right side. Thus, production A → XYZ yields the four items

A→**.**XYZ

A→X**.**YZ

A→XY**.**Z

A→XYZ**.**

17. What is meant by viable prefixes?

The set of prefixes of right sentential forms that can appear on the stack of a shift-reduce parser are called viable prefixes. An equivalent definition of a viable prefix is that it is a prefix of a right sentential form that does not continue past the right end of the rightmost handle of that sentential form.

18. Define handle.

A handle of a string is a substring that matches the right side of a production, and whose reduction to the nonterminal on the left side of the production

represents one step along the reverse of a rightmost derivation.

A handle of a right – sentential form γ is a production A→β and a position of γ where the string β may be found and replaced by A to produce the previous right-sentential form in a rightmost derivation of γ. That is , if

S =>αAw =>αβw,then A→β in the position following α is a handle of αβw.

19. What are kernel & non-kernel items?

Kernel items, which include the initial item, S'→ **.**S, and all items whose dots are not at the left end.

Non-kernel items, which have their dots at the left end.

20. What is phrase level error recovery?

Phrase level error recovery is implemented by filling in the blank entries in the predictive parsing table with pointers to error routines. These routines may change, insert, or delete symbols on the input and issue appropriate error messages. They may also pop from the stack.

PART B

1) Construct a predictive parsing.table for the grammar E -> E + T / F

   T -> T * F / F

   F -> (E) / id

2) Give the LALR parsing table for the grammar.

   S -> L = R / R

   L -> * R / id R -> L

3) Consider the grammar

   E -> TE'

   E' -> + TE' / E T -> FT'

   T' -> *FT' / E

   F -> (E) / id

   Construct a predictive parsing table for the grammar shown above. Verify whether the input string

   id + id * id is accepted by the grammar or not.

4) Consider the grammar.

   E -> E + T

   E -> T

   T -> T * F T -> F

   F -> (E) / id

   Construct an LR parsing table for the above grammar. Give the moves of the LR parser on

   id * id + id

5) For the grammar given below, calculate the operator precedence relation and

the precedence functions

E -> E + E | E – E |

| E * E | E | E |

| E ^ E | (E) |

| -E | id

6) Compare top down parsing and bottom up parsing methods.

7) What are LR parsers? Explain with a diagram the LR parsing algorithm.

8) What are parser generators?

9) Explain recursive descent parser with appropriate examples.

10) Compare SLR, LALR and LR parser.

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