

SPPU-BE-COMP-CONTENT - KSKA Git

Q1> Compare Syntactic Analysis with Semantic Analysis.

ANS.
NO.

SYNTACTIC ANALYSIS
(Structure)

NO.

SEMANTIC ANALYSIS.
(Meaning)

- | | |
|---|--|
| 1. Checks grammatical structure of sentence. | 1. Determines meaning of sentence. |
| 2. Focuses on Syntax (Rules, Grammar) | 2. Focuses on Meaning (Interpretation) |
| 3. Inputs are tokens (words) | 3. Input is parse tree from syntax. |
| 4. Output: parse tree / syntax tree | 4. Output: Meaning Representation. |
| 5. Detects grammatical errors. | 5. Detects meaning inconsistencies. |
| 6. <u>Example:-</u>
"He go to School."
(Syntax error) | 6. <u>Example:-</u>
"Colorless but green ideas sleep"
(Syntax but Meaningless) |

Q2> Elaborate Syntactic Representation of Natural Language.

ANS.

Syntactic Representation shows how words are organized into phrases and sentences.

→ TYPES:-

1. Parse Tree (Constituency Tree)
 - Represents hierarchical structure.
 - Based on grammar rules.

For Eg:-

Sentence(s)

```

    |
    | NP (Noun Phrase)
    |
    | VP (Verb phrase)
    |
    |
    |
    
```

2. Dependency structure.

- Shows relationship between words.

SPPU-BE-COMP-CONTENT - KSKA Git

- Focus on head-dependent relation.

Example:-

"She eats apple"

eats \rightarrow root

she \rightarrow subject

apple \rightarrow object.

3. Context Free Grammar (CFG):-

• Uses Rules like:

$S \rightarrow NP VP$

$VP \rightarrow V NP$

• Syntactic representation helps machine understand sentence structure.

Q4.) Write short Note on:-

1. Probabilistic Context Free Grammar (PCFG)

\rightarrow • Extension of CFG with probabilities.

• Each rule has a probability.

For Eg:-

$S \rightarrow NP VP (0.9)$

$S \rightarrow VP (0.1)$

Used to:-

• Handle Ambiguity.

• Choose most likely parse.

2. Statistical Parsing

\rightarrow • Uses probability and data instead of fixed rules.

• Based on training corpus.

Advantages :-

• More Accurate

• Learns from Real Data.

SPPU-BE-COMP-CONTENT - KSKA Git

3. Lexical Semantics.

1. Study of Word meanings.

• Focus On:-

- Word Relationships.
- Meaning variations.

Example:-

→ "Bank" → river bank or Financial Bank.

4. Dictionary based Approach.

- Uses dictionary to find meaning of words.

Methods.

- Lookup Definitions.
- Compare meanings.

Limitation.

- Cannot handle context well.

Q3) Describe parsing algorithms in Detail.

ANS. Parsing is the process of analyzing a sequence of tokens to determine its grammatical structure with respect to a given grammar (usually Context Free Grammar (CFG))

→ It is performed after lexical analysis in a compiler.

TYPES:-

1. Top-Down Parsing.

- Starts from the start symbol and tries to derive the input string.
- Builds parse tree from root to leaves.

Key Idea: Try to expand the start symbol to match the input.

⇒ 1.1 Recursive Descent Parsing

- A set of recursive procedures for each non-terminal
- May require backtracking.

SPPU-BE-COMP-CONTENT - KSKA Git

Features:-

Simple to Implement, but not efficient due to backtracking

→ Example:- Grammar:

$E \rightarrow E + T \mid T$

$T \rightarrow id$

1.2 Predictive Parsing (LL(1))

- Uses a parsing table
- A non-backtracking version of recursive descent parsing

LL(1)

L - Left to right input scanning.

L - Leftmost derivation.

1 - One Lookahead Symbol

Features:-

- Efficient (no backtracking)
- Requires grammar to be Left recursion^{Free} and left factored.

Steps:-

1. Compute First and Follow.

2. Construct Parsing table.

3. parse input using stack.

2. Bottom-Up Parsing.

- start from input string and reduce it to start symbol.
- Builds parse tree from leaves to root.

2.1 shift Reduce Parsing.

- Uses a stack to shift input symbols and reduce them.

Operations:-

① Shift - Push input symbol into stack.

② Reduce - Replace symbols using grammar rule.

③ Accept - Successful parsing.

SPPU-BE-COMP-CONTENT - KSKA Git

④ Error - Invalid string.

2.2 LR Parsing (Left to right, Rightmost derivation in reverse)

A Powerful bottom-up parsing technique.

• TYPE OF LR PARSERS:-

1. LR(0)
2. SLR(Simple LR)
3. LALR(Look-Ahead LR)
4. CLR(Canonical LR)

→ LR Parsing Features:-

- No backtracking
- Handles a large class of Grammar.
- Used in real compilers. (like GCC)

→ LR Parsing Table Components:

- Action Table → shift, reduce, accept;
- Goto Table → transitions, for non-terminals.

Q5) Discuss relation among lexemes and their senses:-

→ (a) Homonymy

Homonymy are the words with same spelling or pronunciation but different meanings.

Example: "Bat" → Animal / Cricket Bat.

"Bank" → River bank(side) / financial Bank.

(b) Polysemy

- One word, multiple related meanings.

Example: Head → Head of organization / institution (leader)

→ Body part (organ)

→ Top of Table

(c) Synonym.

SPPU-BE-COMP-CONTENT - KSKA Git

• Different Word, same meanings.

• Example:- Big = Large.

(d) Hyponymy

• Hierarchical Relationship. (IS-A Relationship)

• It denotes the specific term in relation

• Example:-

Dog is a type of Animal.

↳ (Hyponymy)

• Opposite of it Hyponymy is Hypernymy.

(e) Wordnet

• Lexical database of Words.

• Group words in synsets (Synonym sets)

• Features:-

Synonyms, Antonyms, Hierarchical relationships.

(f) Word sense Disambiguation (WSD)

• Identify correct meaning of a word based on context. Resolve meaning dispute.

• Example:- "Bank" in:

• "He went to bank" → Financial institution

• "River bank" → land. (River side)

Techniques:-

• Knowledge based.

• Supervised Learning.

• Context Analysis.