

What is the difference between S-attributed and L-attributed definitions ?

Aspect	S-attributed Definitions	L-attributed Definitions
Order of attribute evaluation	Attributes are calculated as we traverse the parse tree from leaves to root, allowing attributes to depend on children and siblings.	Attributes are evaluated in a single pass from root to leaves, with restrictions on which attributes can depend on each other.
Dependency restrictions	No strict rules on attribute dependencies.	Some rules limit where attributes can get their values from, especially for inherited attributes.
Evaluation strategy	Typically uses bottom-up evaluation, computing synthesized attributes as we move up the tree.	Typically uses top-down evaluation, passing inherited attributes down the tree.
Example	In arithmetic expressions, the value of an expression depends on the values of its parts, so it's S-attributed.	In type checking, the type of an expression might depend on its context, so it's often L-attributed.

## Related Posts:

1. What are the types of passes in compiler ?
2. Discuss the role of compiler writing tools. Describe various compiler writing tools.
3. What do you mean by regular expression ? Write the formal recursive definition of a regular expression.
4. How does finite automata useful for lexical analysis ?
5. Explain the implementation of lexical analyzer.
6. Write short notes on lexical analyzer generator.
7. Explain the automatic generation of lexical analyzer.
8. Explain the term token, lexeme and pattern.
9. What are the various LEX actions that are used in LEX programming ?

What is the difference between S-attributed and L-attributed definitions ?

10. Describe grammar.
11. Explain formal grammar and its application to syntax analyzer.
12. Define parse tree. What are the conditions for constructing a parse tree from a CFG ?
13. Describe the capabilities of CFG.
14. What is parser ? Write the role of parser. What are the most popular parsing techniques ? OR Explain about basic parsing techniques. What is top-down parsing ? Explain in detail.
15. What are the common conflicts that can be encountered in shift-reduce parser ?
16. Differentiate between top-down and bottom-up parser. Under which conditions predictive parsing can be constructed for a grammar ?
17. Differentiate between recursive descent parsing and predictive parsing.
18. What is intermediate code generation and discuss benefits of intermediate code ?
19. Define parse tree. Why parse tree construction is only possible for CFG ?
20. Discuss symbol table with its capabilities ?
21. What are the symbol table requirements ? What are the demerits in the uniform structure of symbol table ?