

What are the common conflicts that can be encountered in shift-reduce parser ?

1. Shift-Reduce Conflict:

- What is it?: This happens when the parser isn't sure whether to shift (move to the next input token) or reduce (apply a grammar rule) based on the current state and input.
- Why does it happen?: Often occurs in recursive grammar where the parser can't determine when one rule ends and another begins, leading to ambiguity.
- Example: Imagine a rule like "if-else". When the parser sees "if", it might be unsure whether to wait for the "else" or to proceed with what's next. This indecision causes the conflict.

2. Reduce-Reduce Conflict:

- What is it?: This occurs when the parser encounters a point where it can apply two or more grammar rules, but it's uncertain which one to choose.
- Why does it happen?: It usually arises when different rules lead to the same state, causing ambiguity about which rule to follow.
- Example: Consider a grammar where both "if-else" and "if-elseif" are allowed. If the parser encounters an "if" followed by conditions for both "else" and "elseif", it might not know which rule to apply, leading to the conflict.

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.

What are the common conflicts that can be encountered in shift-reduce parser ?

8. Explain the term token, lexeme and pattern.
9. What are the various LEX actions that are used in LEX programming ?
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. Differentiate between top-down and bottom-up parser. Under which conditions predictive parsing can be constructed for a grammar ?
16. Differentiate between recursive descent parsing and predictive parsing.
17. What is the difference between S-attributed and L-attributed definitions ?
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 ?