Shift Reducer Parser

Bachelor of Technology
Computer Science and Engineering

Submitted By

Snikdhendu Pramanik
13000122061
CSE-A , 5th SEM

September 2024

Techno Main
EM-4/1, Sector-V, Salt Lake
Kolkata- 700091
West Bengal
India

TABLE OF CONTENTS

1. Introduction :

Shift Reduce Parser is a type of **Bottom-Up Parser**. It generates the Parse Tree from Leaves to the Root. In Shift Reduce Parser, the input string will be reduced to the starting symbol. This reduction can be produced by handling the rightmost derivation in reverse, i.e., from starting symbol to the input string.

This parser requires some data structures i.e.

- An input buffer for storing the input string.
- A stack for storing and accessing the production rules.

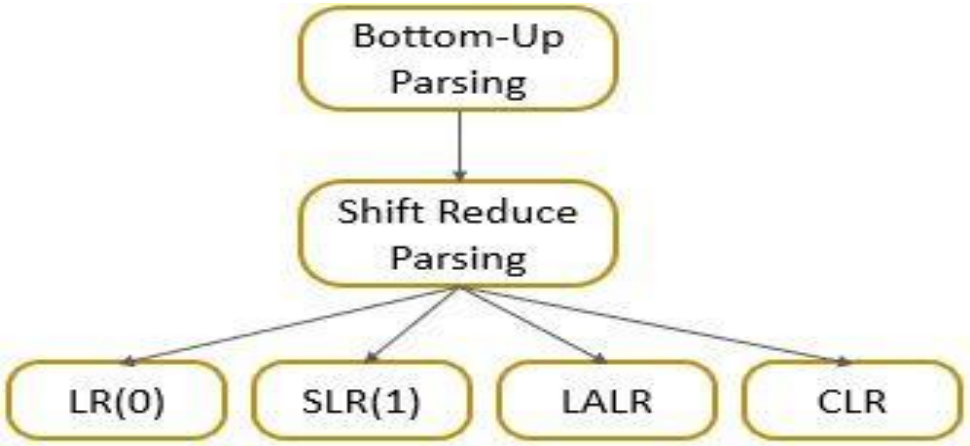
2. Basic Operations :

- **Shift:** This involves moving symbols from the input buffer onto the stack.
- **Reduce:** If the handle appears on top of the stack then, its reduction by using appropriate production rule is done i.e. RHS of a production rule is popped out of a stack and LHS of a production rule is pushed onto the stack.
- **Accept:** If only the start symbol is present in the stack and the input buffer is empty then, the parsing action is called accept. When accepted action is obtained, it is means successful parsing is done.
- **Error:** This is the situation in which the parser can neither perform shift action nor reduce action and not even accept action.

3. Types of Shift-Reduce Parsers

1. **SLR Parser (Simple LR)**: A basic shift-reduce parser with simple look-ahead to resolve conflicts.
2. **LR Parser**: Uses look-ahead symbols and parsing tables to handle more complex grammars.

3. **LALR Parser (Look-Ahead LR)**: Combines the simplicity of SLR with the power of LR to handle more grammars with fewer states.



4. Working of Shift-Reduce Parsing :

- **Initialization:** The parser begins with an empty stack and reads tokens from the input buffer.
- **Shifting:** The parser shifts symbols from the input buffer to the stack until a reducible configuration is identified.
- **Reducing:** When the top of the stack contains a sequence of symbols that match the right-hand side of a production rule, the parser performs a reduction by replacing those symbols with the corresponding non-terminal from the grammar.
- **Acceptance:** The parser continues performing shift and reduce operations until the entire input is consumed, and the stack contains the start symbol of the grammar. If so, the input string is successfully parsed.
- **Error Handling:** If no shift or reduce operation is possible, and the input buffer or stack is in an unexpected configuration, the parser may report a syntax error.

5. Example :

Grammar :

1. S → S+S
2. S → S-S
3. S → (S)
4. S → a

Input string :

1. a1-(a2+a3)

Parsing table :

Stack contents	Input string	Actions
$	a1-(a2+a3)$	shift a1
$a1	-(a2+a3)$	reduce by S → a
$S	-(a2+a3)$	shift -
$S-	(a2+a3)$	shift (
$S-(	a2+a3)$	shift a2
$S-(a2	+a3)$	reduce by S → a
$S-(S	+a3) $	shift +
$S-(S+	a3) $	shift a3
$S-(S+a3	) $	reduce by S → a
$S-(S+S	) $	shift)
$S-(S+S)	$	reduce by S → S+S
$S-(S)	$	reduce by S → (S)
$S-S	$	reduce by S → S-S
$S	$	Accept

6. **Advantages of Shift-Reduce Parsing :**

- **Efficiency**: Shift-reduce parsers, especially LR parsers, are efficient and can handle a large class of grammars.
- **Wide Applicability**: It works well with most context-free grammars used in programming languages.

7. **Disadvantages of Shift-Reduce Parsing :**

- **Conflicts**: There can be conflicts such as shift/reduce conflicts or reduce/reduce conflicts when the grammar is ambiguous or complex, which requires additional handling (e.g., LR parsing tables).

8. **Conclusion :**

Shift-Reduce Parsing is a powerful technique for syntax analysis in compilers, providing a systematic way to handle bottom-up parsing. Its application in LR, SLR, and LALR parsers makes it a key part of modern compiler design, enabling efficient parsing of programming languages.

9. **References :**

- https://www.geeksforgeeks.org/shift-reduce-parser-compiler/
- https://en.wikipedia.org/wiki/Shift-reduce_parser
- https://www.tutorialspoint.com/what-is-shift-reduce-parser