

GROUP NO: 39

VIJAYAPAREDDY
BHAVANA

Diagram illustrating the execution flow for the expression `return (TK_OR, OR)` and `return (TK_AND, AND)` in a parser.


The flow starts from a common entry point (labeled "exit" with an arrow pointing to it) and branches into two paths:

- Path 1 (Left):** Labeled `return (TK_OR, OR)`. It consists of a sequence of four states connected by arrows labeled `OR`. The final state leads to an "exit" point.
- Path 2 (Right):** Labeled `return (TK_AND, AND)`. It consists of a sequence of four states connected by arrows labeled `AND`. The final state leads to an "exit" point.

Additional labels and symbols include:

- `[2-3]` and `[1-3]` near the bottom of the paths.
- A vertical list of tokens on the right: `return (TK_OR, OR)`, `return (TK_AND, AND)`, `return (TK_COMMA, ,)`, `return (TK_SEM, ;)`, `return (TK_COLON, :)`, `return (TK_DOT, .)`, `return (TK_OP, (`, and `return (TK_CL,)`.

Line TX-Not, 2)

*  $\text{return } (T_k - L_T, <)$

1000

---) 11/11/11

return

卷之七

yes, yes, yes

return (x)

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100.

17) *unmarked*

(2) For

(C 85-242)

178-PLUS, +

between (two minus -)

return

return
FTE-MUL, 0

return
(Tk-Div,)

5 June - 1968

$$\text{delim} \rightarrow \text{blank} / \text{tab} / \text{newline}$$

getToken() : Examines
Symbol table for lexeme
found and returns
the token
installID() : It places
lexeme in the symbol
table if not already
there and returns
the lexeme