

1.

→ N-program.node-sym = make-node ("N-program",
make-node ("N-moduleDeclarations", NmoduleDeclarations.list-head-sym),
make-node ("MODDEF", NotherModule1.list-head-sym),
N-driverModule.node-sym,
make-node ("MODDEF", N-otherModule2.list-head-sym))
)

→ free (N-moduleDeclarations)
→ free (N-otherModule1)
→ free (N-driverModule)
→ free (N-otherModule2)

2. → N-moduleDeclarations.list-head-sym = insert-at-head (

N-moduleDeclarations1.list-head-sym, N-moduleDeclaration.node-sym)

→ free (N-moduleDeclaration)
→ free (N-moduleDeclarations1).

3. → N-moduleDeclarations.list-head-sym = NULL

4. → N-moduleDeclaration.node-sym = id

→ free (T-DECLARE)
→ free (T-MODULE-ID)
→ free (T-SEMICOL)

5. N-otherModules.list-head-sym = insert-at-head (N-otherModules1.list-head-sym,
N-module.node-sym)
→ free (N-module)

→ free(N_otherModule)

6. N_otherModule . list_head . sym = NULL

7. → N_driverModule . node_sym = make_node(" DRIVER ", N_moduleDef .
node_sym)

→ free(T_DRIVERDEF)

→ free(T_DRIVER)

→ free(T_PROGRAM)

→ free(T_DRIVERENDDEF)

→ free(N_moduleDef)

8. → N_module . node_sym = make_node(" N-MODULE ", FID ,
make_node(" PARAMLIST ", N_input_list . list_head . sym),
N_ret . node_sym ,
N_moduleDef . node_sym)

→ free(T_PEF)

→ free(T_MODULE)

→ free(T_ENDDDEF)

→ free(T_TAKES)

→ free(T_INPUT)

→ free(T_SQLBO)

→ free(N_input_list)

→ free(SQBC)

→ free(T_SEMICOL)

→ free(N_ret)

→ free(N_moduleDef)

2
9. $\rightarrow N_ret_node_syn = make_node("PARAMS LIST", N_output_plist, list_head_syn)$
 $\rightarrow free(T_RETURNS)$
 $\rightarrow free(T_SQB0)$
 $\rightarrow free(N_output_plist)$
 $\rightarrow free(T_SQBC)$
 $\rightarrow free(T_SEMICOL)$

10. $N_ret_node_syn = make_node("PARAMS LIST", NULL)$

11. $\rightarrow N_nl_list_head_inh = make_list(N_dataType.node_syn)$
 $\rightarrow N_nl_list_head_inh = insert_at_end(N_nl_list_head_inh, T_ID)$
 $\rightarrow N_input_plist.list_head_syn = N_nl_list_head_syn$
 $\rightarrow free(T_COLON)$
 $\rightarrow free(N_dataType)$
 $\rightarrow free(N_nl)$

12. $\rightarrow N_nl_list_head_inh = insert_at_end(N_nl_list_head_inh,$
 $N_dataType.node_syn)$
 $\rightarrow N_nl1.list_head_inh = insert_at_end(N_nl.list_head_inh, T_ID)$
 $\rightarrow N_nl.list_head_syn = N_nl1.list_head_syn$
 $\rightarrow free(T_COMMA)$
 $\rightarrow free(T_COLON)$
 $\rightarrow free(N_dataType)$
 $\rightarrow free(N_nl1)$

13. $\rightarrow N-n1.\text{list_head_sym} = N-n1.\text{list_head_inh}$

14. $\rightarrow N-n2.\text{list_head_inh} = \text{make_list}(N\text{-type}.\text{node_sym})$
 $\rightarrow N-n2.\text{list_head_inh} = \text{insert_at_end}(N-n2.\text{list_head_inh}, T-1D)$
 $\rightarrow N\text{-datatype_plist}.\text{list_head_sym} = N-n2.\text{list_head_sym}$
 $\rightarrow \text{free}(T\text{-COLON})$
 $\rightarrow \text{free}(N\text{-type})$
 $\rightarrow \text{free}(N-n2)$

15. $\rightarrow N-n21.\text{list_head_inh} = \text{insert_at_end}(N-n2.\text{list_head_inh},$
 $N\text{-type}.\text{node_sym})$

- $\rightarrow N-n21.\text{list_head_inh} = \text{insert_at_end}(N-n2.\text{list_head_inh}, T-1D)$
 $\rightarrow N-n2.\text{list_head_sym} = N-n21.\text{list_head_sym}$
 $\rightarrow \text{free}(T\text{-COMMA})$
 $\rightarrow \text{free}(T\text{-COLON})$
 $\rightarrow \text{free}(N\text{-type})$
 $\rightarrow \text{free}(N-n21)$

16. $N-n2.\text{list_head_sym} = N-n2.\text{list_head_inh}$

17. $N\text{-dataType}.\text{node_sym} = T\text{-INTEGER}$

18. $N\text{-dataType}.\text{node_sym} = T\text{-REAL}$

19. $N\text{-dataType}.\text{node_sym} = T\text{-BOOLEAN}$

$\text{free}(T\text{-END})$

20. $\rightarrow N\text{-dataType}.\text{node_sym} = \text{make_node}(\text{"ARR"}, N\text{-type}.\text{node_sym},$
 $N\text{-range_arrays}.\text{node_sym})$
 - $\rightarrow \text{free}(T\text{-ARRAY})$
 - $\rightarrow \text{free}(T\text{-SQB0})$
 - $\rightarrow \text{free}(N\text{-range_arrays})$
 - $\rightarrow \text{free}(T\text{-SQBc})$
 - $\rightarrow \text{free}(T\text{-OF})$
 - $\rightarrow \text{free}(N\text{-type})$
21. $\rightarrow N\text{-range_arrays}.\text{node_sym} = \text{make_node}(\text{"INDEX"}, N\text{-index_arr}.\text{node_sym},$
 $N\text{-index_arr}.\text{node_sym})$
 - $\rightarrow \text{free}(N\text{-index_arr})$
 - $\rightarrow \text{free}(T\text{-RANGEOP})$
 - $\rightarrow \text{free}(N\text{-index_arr1})$
 - $\rightarrow \text{free}(N\text{-index_arr2})$
22. $\rightarrow N\text{-type}.\text{node_sym} = T\text{-INTEGER}$
23. $\rightarrow N\text{-type}.\text{node_sym} = T\text{-REAL}$
24. $\rightarrow N\text{-type}.\text{node_sym} = T\text{-BOOLEAN}$
25. $\rightarrow N\text{-moduleref}.\text{node_sym} = \text{make_node}(\text{"STMTS"}, N\text{-statement}.\text{list_head_sym})$
 - $\rightarrow \text{free}(T\text{-START})$
 - $\rightarrow \text{free}(N\text{-statement})$
 - $\rightarrow \text{free}(T\text{-END})$(3)

26. $N\text{-statements}.\text{list_head_sym} = \text{insert_at_head}(N\text{-statements1}.\text{list_head_sym},$
 $N\text{-statement}.\text{node_sym})$

27. $N\text{-statements}.\text{list_head_sym} = \text{NULL}$

28. $\rightarrow N\text{-statements}.\text{node_sym} = N\text{-io Stmt}.\text{node_sym}$
 $\rightarrow \text{free}(N\text{-io Stmt})$

29. $\rightarrow N\text{-statements}.\text{node_sym} = N\text{-simple Stmt}.\text{node_sym}$
 $\rightarrow \text{free}(N\text{-simple Stmt})$

30. $\rightarrow N\text{-statements}.\text{node_sym} = N\text{-declare Stmt}.\text{node_sym}$
 $\rightarrow \text{free}(N\text{-declare Stmt})$

31. $\rightarrow N\text{-statements}.\text{node_sym} = N\text{-conditional Stmt}.\text{node_sym}$
 $\rightarrow \text{free}(N\text{-conditional Stmt})$

32. $\rightarrow N\text{-statements}.\text{node_sym} = N\text{-iterative Stmt}.\text{node_sym}$
 $\rightarrow \text{free}(N\text{-iterative Stmt})$

33. $\rightarrow N\text{-io Stmt}.\text{node_sym} = \text{make_node}("GETVAL", T\text{-ID})$
 $\rightarrow \text{free}(T\text{-GET-VALUE})$
 $\rightarrow \text{free}(T\text{-BO})$
 $\rightarrow \text{free}(T\text{-BC})$
 $\rightarrow \text{free}(T\text{-SEMICOL})$

34. $\rightarrow N\text{-io Stmt}.\text{node_sym} = \text{make_node}("PRINT", N\text{-var_print}.\text{node_sym})$
 $\rightarrow \text{free}(T\text{-PRINT})$

- free(T-ID)
- free(N-var-point)
- free(T-BC)
- free(T-SEMICOL)

35. → N-pl.node-inh = T-ID

→ N-var-point.node-sym = N-pl.node-sym

→ free(N-pl)

36. → N-var-point.node-sym = T-NUM

37. → N-var-point.node-sym = T-RNUM

38. → N-var-point.node-sym = T-BOOLCONST

39. → N-pl.node-sym = make-node("ACCESS", N-pl.node-inh,
N-new-index.node-sym)

40. → N-pl.node-sym = N-pl.node-inh

41. N-boolconst.node-sym = T-TRUE

42. N-boolconst.node-sym = T-FALSE

43. N-id-num-rnum.node-sym = T-ID

44. N-id-num-rnum.node-sym = T-NUM

45. N-id-num-rnum.node-sym = T-RNUM

46. $\rightarrow N\text{-index_arr_node_inh} = T_IP$

$\rightarrow N\text{-array_element_for_print_node_sym} = N\text{-index_arr_node_sym}$

$\rightarrow \text{free}(T\text{-SQBO})$

$\rightarrow \text{free}(N\text{-index_arr})$

$\rightarrow \text{free}(T\text{-SQBC})$

47. $\rightarrow N\text{-simple_stmt_node_sym} = N\text{-assignment_stmt_node_sym}$

$\rightarrow \text{free}(N\text{-assignment_stmt})$

48. $\rightarrow N\text{-simple_stmt_node_sym} = N\text{-moduleReuse_stmt_node_sym}$

$\rightarrow \text{free}(N\text{-moduleReuse_stmt})$

49. $\rightarrow N\text{-which_stmt_node_inh} = T_IP$

$\rightarrow N\text{-assignment_stmt_node_sym} = N\text{-which_stmt_node_sym}$

$\rightarrow \text{free}(N\text{-which_stmt})$

50. $\rightarrow N\text{-lvalue_ID_stmt_node_inh} = N\text{-which_stmt_node_inh}$

$\rightarrow N\text{-which_stmt_node_sym} = N\text{-lvalue_ID_stmt_node_sym}$

$\rightarrow \text{free}(N\text{-lvalue_ID_stmt})$

51. $\rightarrow N\text{-lvalue_ARR_stmt_node_inh} = N\text{-which_stmt_node_inh}$

$\rightarrow N\text{-which_stmt_node_sym} = N\text{-lvalue_ARR_stmt_node_sym}$

$\rightarrow \text{free}(N\text{-lvalue_ARR_stmt})$

52. $\rightarrow N\text{-lvalue_ID_stmt_node_sym} = \text{make_node}("TASSIGNVOP",$

~~make_node("APRACC")~~, $N\text{-lvalue_ARR_stmt_node_inh}$, $N\text{-expression_node_sym}$)

→ free(T-ASSIGNOP)

→ free(N-expression)

→ free(T-SEMICOL)

- 53) → N-walkARRstmt. node-sym = make-node("T-ASSIGNOP",
make-node("ARR ACC", N-walkARRstmt. node-inh,
N-element-index-with-expression. node-sm), N-expression. node-sym)
→ free(T-SQB)
- free(N-element-index-with-expression)
- free(T-SQB)
- free(T-ASSIGNOP)
- free(N-expression)
- free(T-SEMICOL)

54) → N-new-index. node-inh = N-sign. node-sym

→ N-index-arr = N-new-index. node-sym

→ free(N-sign)

→ free(N-new-index)

55) N-new-index. node-sym = T-num

56) N-new-index. node-sym = T-ID

57) N-sign. node-sym = T-PLUS

58) N-sign. node-sym = T-MINUS

59) N-sign. lit-head-sym = NULL

60) \rightarrow $N_module\ Resestmt$, $node_syn = make_node("FNCCALL", T_ID,$
 $make_node(" PARA MSUSt", N_optional . list_head - syn),$
 $make_node (" PARAMSList", N_idList . list_head - syn))$
 $\rightarrow free (N_optional)$
 $\rightarrow free (T_USE)$
 $\rightarrow free (T_MODULE)$
 $\rightarrow free (T_WITH)$
 $\rightarrow free (T_PARAMETERS)$
 $\rightarrow free (N_rdList)$
 $\rightarrow free (T_SEMICOR)$

61) $N_actual_para_list . node_syn = T_SEMICOR$

62) $N_append_para_list . node_inh = make_node(N_sign . node_syn$
 $\rightarrow N_actual_para_list . node_inh, N_K . node_syn)$
 $\rightarrow N_actual_para_list . node_syn = N_append_para_list . node_syn$
 $\rightarrow free (N_sign)$
 $\rightarrow free (N_K)$
 $\rightarrow free (N_append_para_list)$

63) $N_optional_list_head_syn = N_idList . list_head_sign$

$free (T_SQBO)$
 $free (N_idList)$
 $free (T_SQBC)$
 $free (T_ASSIGNOP)$

64. $N \cdot \text{optional} \cdot \text{list_head_sym} = \text{NULL}$

65. $\rightarrow N \cdot n3 \cdot \text{list_head_inh} = \text{make_list}(T\text{-ID})$

$\rightarrow N \cdot \text{idlist} \cdot \text{list_head_sym} = N \cdot n3 \cdot \text{list_head_sym}$

$\rightarrow \text{free}(N \cdot n3)$

66. $\rightarrow N \cdot n3l \cdot \text{list_head_inh} = \text{insert_at_end}(N \cdot n3 \cdot \text{list_head_inh}, T\text{-ID})$
 $\rightarrow N \cdot n3 \cdot \text{list_head_sym} = N \cdot n3l \cdot \text{list_head_sym}$
 $\rightarrow \text{free}(T\text{-COMMA})$
 $\rightarrow \text{free}(N \cdot n3l)$

67. $N \cdot n3 \cdot \text{list_head_sym} = N \cdot n3 \cdot \text{list_head_inh}$

68. $\rightarrow N \cdot \text{expression} \cdot \text{node_sym} = N \cdot \text{arithmetc Or Boolean Expr} \cdot \text{node_sym}$
 $\rightarrow \text{free}(N \cdot \text{arithmetc or Boolean Expr})$

69. $\rightarrow N \cdot \text{expression} \cdot \text{node_sym} = N \cdot u \cdot \text{node_sym}$
 $\rightarrow \text{free}(N \cdot u)$

70. $\rightarrow N \cdot \text{new-NT} \cdot \text{node_inh} = N \cdot \text{unary_op} \cdot \text{node_sym}$
 $\rightarrow N \cdot u \cdot \text{node_sym} = N \cdot \text{new-NT} \cdot \text{node_sym}$
 $\rightarrow \text{free}(N \cdot \text{unary_op})$
 $\rightarrow \text{free}(N \cdot \text{new-NT})$

71. $\rightarrow N \cdot \text{new-NT} \cdot \text{node_sym} = \text{make_node}("N\text{-NEW-NT}", N \cdot \text{new-NT} \cdot \text{node_inh},$
 $\quad N \cdot \text{arithmetc Expr} \cdot \text{node_sym})$
 $\rightarrow \text{free}(T\text{-BO})$
 $\rightarrow \text{free}(T\text{-BC})$

(6)

72. $\rightarrow N\text{-new-NT}.\text{node-sym} = N\text{-var-id-num}.\text{node-sym}$
 $\rightarrow \text{free}(N\text{-var-id-num})$

73. $N\text{-var-id-num}.\text{node-sym} = T\text{-ID}$

74. $N\text{-var-id-num}.\text{node-sym} = T\text{-NUM}$

75. $N\text{-var-id-num}.\text{node-sym} = T\text{-RNUM}$

76. $N\text{-unary-op}.\text{node-sym} = T\text{-PLUS}$

77. $N\text{-unary-op}.\text{node-sym} = T\text{-MINUS}$

78. $\rightarrow N\text{-n7}.\text{node-inh} = N\text{-anyTerm}.\text{node-sym}$

$\rightarrow N\text{-arithmeticOrBooleanExpr}.\text{node-sym} = N\text{-n7}.\text{node-sym}$

$\rightarrow \text{free}(N\text{-anyTerm})$

$\rightarrow \text{free}(N\text{-n7})$

79. $\rightarrow N\text{-n71}.\text{node-inh} = \text{make-node}(N\text{-logicalOp}.\text{node-sym},$
 $N\text{-n7}.\text{node-inh}, N\text{-anyTerm}.\text{node-sym})$

$\rightarrow N\text{-n7}.\text{node-hm} = N\text{-n7-1}.\text{node-sym}$

$\rightarrow \text{free}(N\text{-logicalOp})$

$\rightarrow \text{free}(N\text{-anyTerm})$

$\rightarrow \text{free}(N\text{-n71})$

80. $N\text{-n7}.\text{list-head-sym} = \text{NULL}$

81. $\rightarrow N\text{-} \cancel{\text{arithmeticExpr}}^{\text{n8}}.\text{node-inh} = N\text{-arithmeticExpr}.\text{node-sym}$
 $\rightarrow N\text{-anyTerm}.\text{node-sym} = N\text{-n8}.\text{node-sym}$
 $\rightarrow \text{free}(N\text{-arithmetic Expr})$

→ free(N-n8)

82. → N-anyTerm.node-sym = N-boolConst.node-sym
→ free(N-boolConst)

83. → N-arithmeticExpr.node-inh = N-relationalOp.node-sym
→ N-n8.node-sym = N-arithmeticExpr.node-sym
→ free(N-relationalOp)
→ free(N-arithmeticOp)

84. → N-n8.list-head-sym = NULL

85. → N-n4.node-inh = N-term.node-sym
→ N-arithmeticExp.node-sym = N-term.node-sym
→ free(N-term)
→ free(N-n4)

86. → N-n4-1.node-inh = make-node(N-op1.node-sym,
N-n4.node-inh, N-term.node-sym)
→ N-n4.node-sym = N-n4-1.node-sym
→ free(N-op1)
→ free(N-term)
→ free(N-n4-1)

87. N-n4.list-head-sym = NULL

88. → N-n5.node-inh = N-factor.node-inh
→ N-term.node-sym = N-n5.node-sym

(7)

17
→ free (N-factor)
→ free (N-ns)

89. → N-ns-1.node-link = make-node (N-op 2.node-sym,
 N-ns.node-link, N-factor.node-hm)
→ N-ns.node-sym = N-ns-1.node-sym
→ free (N-op2)
→ free (N-factor)
→ free (N-ns-1)

90. N-ns.list-head.sym = NULL

91. → N-factor.node-sym = make-node ("N-factor", N-factor-node-link,
 N-arithmeticOrBooleanExpr.node-sym)
→ free (T-Bo)
→ free (T-BC)
→ free (N-arithmeticOrBooleanExpr)

92. N-factor.node-sym = T-NUM

93. N-factor.node-sym = T-RNUM

94. N-factor.node-sym = N-boolConstt.node-sym
free (N-boolConstt)

95. → N-n-11.node-link = T-ID
→ N-factor.node-sym = N-n-11.node-sym
→ free (N-n-11)

12. $96 \rightarrow N-n-11$. node - syn = make-node("N-n-11", $N-n-11$. node - inh,
 N -element-index-with-expressions. node - syn)
 \rightarrow free(T-SQ-B0)
 \rightarrow free(T-SQ-BC)
 \rightarrow free(N-element-index-with-expressions)
97. $N-n11$. list-head - syn = NULL
98. $N-all-n4$. node - inh = $N-allTerm$. node - syn
 N -all Expr. node - syn = $N-n4$. node - syn
free($N-allTerm$)
free($N-all-n4$)
99. $N-n4-1$. node - inh = make-node($N-opt1$. node - syn, $N-n4$. node - inh,
 N -all Term. node - syn)
 \rightarrow $N-n4$. node - syn = $N-n4-1$. node - syn
 \rightarrow free($N-opt1$)
 \rightarrow free($N-allTerm$)
 \rightarrow free($N-n4-1$)
100. $N-all-n4$. list-head - syn = NULL
101. $N-all-n5$. node - inh = $N-allFactor$. node - syn
 N -all Term. node - syn = $N-all-n5$. node - syn
free($N-allFactor$)
free($N-all-n5$)

102. $N_ns_1.\text{node_inh} = \text{make_node}(N_op2.\text{node_sym}, N_ns.\text{node_inh},$
 $N_arrFactor.\text{node_sym})$
- $N_ns.\text{node_sym} = N_ns_1.\text{node_sym}$
- $\rightarrow \text{free}(N_op2)$
- $\rightarrow \text{free}(N_arrFactor)$
- $\rightarrow \text{free}(N_ns_1)$
103. $N_arr_ns, \text{list_head_sym} = \text{NULL}$
104. $N_arrFactor.\text{node_sym} = T_{\text{ID}}$
105. $N_arrFactor.\text{node_sym} = T_NUM$
106. $\rightarrow N_arrFactor.\text{node_sym} = N_boolConst.\text{node_sym}$
 $\rightarrow \text{free}(N_boolConst)$
107. $\rightarrow N_arrFactor.\text{node_sym} = \text{make_node}("N_arrFactor", N_arrFactor_inh,$
 $N_arrExpr.\text{node_sym})$
- $\rightarrow \text{free}(T_BO)$
- $\rightarrow \text{free}(T_BX)$
- $\rightarrow \text{free}(N_arrExpr)$
108. $N_n_10.\text{node_inh} = N_sign.\text{node_sym}$
- $\rightarrow N_element_index_with_expression.\text{node_sym} = N_n_10.\text{node_sym}$
- $\rightarrow \text{free}(N_sign)$
- $\rightarrow \text{free}(N_n_10)$

109. $\rightarrow N\text{-element-index}\cdot\text{with-expressions}\cdot\text{node_sym} = N\text{-arrExpr}\cdot\text{node_sym}$
 $\rightarrow \text{free}(N\text{-arrExpr})$

110. $\rightarrow N\text{-n-10}\cdot\text{node_sym} = N\text{-new-index}\cdot\text{node_sym}$
 $\rightarrow \text{free}(N\text{-new-index})$

111. $\rightarrow N\text{-n-10}\cdot\text{node_sym} = \text{make-node}("N\text{-n-10}", N\text{-n-10}\cdot\text{node_inh},$
 $N\text{-arrExpr}\cdot\text{node_sym})$
 $\rightarrow \text{free}(T\text{-BO})$
 $\rightarrow \text{free}(T\text{-BC})$
 $\rightarrow \text{free}(N\text{-arr Expr})$

112. $N\text{-op1}\cdot\text{node_sym} = T\text{-PLUS}$

113. $N\text{-op1}\cdot\text{node_sym} = T\text{-MINUS}$

114. $N\text{-op1}\cdot\text{node_sym} = T\text{-MUL}$

115. $N\text{-op1}\cdot\text{node_sym} = T\text{-DIV}$

116. $N\text{-logicalOp}\cdot\text{node_sym} = T\text{-AND}$

117. $N\text{-logicalOp}\cdot\text{node_sym} = T\text{-OR}$

118. $N\text{-logicalOp}\cdot\text{node_sym} = T\text{-LT}$

119. $N\text{-logicalOp}\cdot\text{node_sym} = T\text{-LE}$

120. $N\text{-logicalOp}\cdot\text{node_sym} = T\text{-GTE}$

121. $N_logicalOp.\text{node_sym} = T_CRE$

122. $N_logicalOp.\text{node_sym} = T_EQ$

123. $N_logicalOp.\text{node_sym} = T_NE$

124. $N_declare \text{stmt}.\text{node_sym} = \text{make_node}("DECLARE", N_dataType.\text{node_sym})$
→ $N_idList.\text{list_head_sym})$

→ free(T-DECLARE)

→ free(N-idList)

→ free(T-COLON)

→ free(N-dataType)

→ free(T-SEMICOL)

125. $N_conditional \text{stmt}.\text{node_sym} = \text{make_node}("SWITCH", T_ID,$

→ $\text{make_node}("CASES", N_caseStmts.\text{list_head_sym}),$
 $N_default.\text{node_sym})$

→ free(T-SWITCH)

→ free(T-B0)

→ free(T-BC)

→ free(T-START)

→ free(N-caseStmts)

→ free(N-default)

→ free(T-END)

126.

→ N-CASE stmts. list-head-syn = insert-at-head(N-nq, list-head-syn,
make-node("CASE", N-value-node-syn, make-node("STATEMENTS",
N-statements · list-head-syn)))

→ free(T-CASE)

→ free(N-value)

→ free(T-COLON)

→ free(N-statements)

→ free(T-BREAK)

→ free(T-SEMICOL)

→ free(N-nq)

127. N-nq.list-head-syn = insert-at-head(N-nq, list-head-syn, make-node
"CASE", N-value-node-syn, make-node("STATEMENTS",
N-statements · list-head-syn))

→ free(T-CASE)

→ free(N-value)

→ free(T-COLON)

→ free(N-statements)

→ free(T-BREAK)

→ free(T-SEMICOL)

→ free(N-nq)

→ free(N-nq).list-head-syn = NULL

129. N-value-node-syn = T-NUM

130. N-value-node-syn = T-TRUE

131. N-value-node-syn = T-FALSE

132.

- $N_default.\text{node_sym} = \text{make_node}(\text{"DEFAULT"},$
 $\quad \quad \quad \text{make_node}(\text{"STATEMENTS"}, N_statements.\text{list_head_sym}))$
- free(T-DEFAULT)
- free(T-COLON)
- free(N-statements)
- free(T-BREAK)
- free(T-SEMI(COL))

133. $N_default.\text{list_head_sym} = \text{NULL}$

134. → $N_iterativestmt.\text{node_sym} = \text{make_node}(\text{"FORLOOP"}, T-ID, T-IN1,$
 $\quad \quad \quad T-IN2, \text{make_node}(\text{"STATEMENTS"}, N_statements, list_head_sym))$
- free(T-FOR)
 - free(T-BO)
 - free(T-IN)
 - free(T-BC)
 - free(N-statements)
 - free(T-END)
 - free(T-START)

135. → $N_iterativestmt.\text{node_sym} = \text{make_node}(\text{"WHILELOOP"}, N_arithmeticOrBooleanExpr,$
 $\quad \quad \quad \text{node_sym}, \text{make_node}(\text{"STATEMENTS"}, N_statements.\text{list_head_sym}))$

- free(T-WHILE)
- free(T-BO)
- free($T^N_{\text{arithmeticOrBooleanExpr}}$)
→ free(BC)

→ free(T-START)

→ free(N-statements)

→ free(T-END)

136. N-range-for-loop. node-sym = make-node("RANGE", index-for-loop,
node-sym,

→ index-for-loop2. node-sym);

→ free(N-index-for-loop1)

→ free(T-RANGEOP)

→ free(N-index-for-loop2)

137. N-new-index-for-loop1. node-sym = N-sign-for-loop. node-sym

→ N-index-for-loop. node-sym = N-new-index-for-loop1. node-sym

→ free(N-sign-for-loop)

→ free(N-new-index-for-loop)

138. N-new-index-for-loop . node - sym = T-NUM

139. N-sign-for-loop. node-sym = T-PLUS

140. N-sign-for-loop . node - sym = T-MINUS

141. N-sign-for-loop. list-head - sym = NULL

142. → N-append-para-list1. node - inh = make-node(

N-sign. node-sym, N-append-para-list. node - inh,
N-K . node - sym)

→ N-append-para-list . node - sym = N-append-paralist1. node - sym

(11)

→ free(T-COMMA)

→ free(N-sign)

→ free(N-k)

→ free(cN-append-para-list)

143. N-append-para-list . list-head . sym = NULL

144.

→ N-m-l1 . node-linh = T-ID

→ N-k . node-sym = N-n-l1 . node-sym

→ free(N-n-l1)

145. N-k . node-sym = NUM

146. N-k . node-sym = RNUM

147. N-k . node-sym = N-boolConst . node-sym

free(N-boolConst)