1. Program -> Declaration-list $

2. Declaration-list -> Declaration Declaration-list | EPSILON

3. Declaration -> **%def\_addr %def\_scope** Declaration-initial Declaration-prime

4. Declaration-initial -> **%def\_type** Type-specifier **%def\_lex** ID

5. Declaration-prime -> Fun-declaration-prime | Var-declaration-prime

6. Var-declaration-prime -> **%update\_addr\_1** **%args\_1 %var\_spec** ; | [ **%update\_addr\_2** **%args\_2** NUM ] **%arr\_spec** ;

7. Fun-declaration-prime -> **%func\_spec** **%return\_addr\_val** (Params ) **%code\_addr** Compound-stmt

8. Type-specifier -> int | void

9. Params -> **%def\_type %def\_scope** **%def\_addr** int **%def\_lex** ID Param-prime **%update\_addr\_1** Param-list | void

10. Param-list -> , Param **%update\_addr\_1** Param-list | EPSILON

11. Param -> **%def\_scope** **%def\_addr** Declaration-initial Param-prime

12. Param-prime -> [ ] **%arr\_pointer\_spec** | EPSILON **%var\_spec**

13. Compound-stmt -> { **%update\_scope\_inc** Declaration-list Statement-list } **%update\_scope\_dec**

14. Statement-list -> Statement Statement-list | EPSILON

15. Statement -> Expression-stmt | Compound-stmt | Selection-stmt | Iteration-stmt | Return-stmt

16. Expression-stmt -> Expression ; | break ; **#JumpToEnd** | ;

17. Selection-stmt -> if ( Expression ) **#SAVE\_IF** Statement **#JIF** Else-stmt

18. Else-stmt -> endif | else **#SAVE\_IF\_2** Statement endif **#JPF**

19. Iteration-stmt -> repeat **#LI** Statement until ( Expression ) **#JI** **#EOI**

20. Return-stmt -> return Return-stmt-prime **#jR**

21. Return-stmt-prime -> ; | Expression **#assR** ;

22. Expression -> Simple-expression-zegond | **#pidAddr** ID B

23. B -> **#readAddr** = Expression **#assign** | [ Expression ] **#ArrIdx #readAddr** H | Simple-expression-prime

24. H -> = Expression **#assign** | G D C

25. Simple-expression-zegond -> Additive-expression-zegond C

26. Simple-expression-prime -> Additive-expression-prime C

27. C -> Relop Additive-expression **#compare** | EPSILON

28. Relop -> < **#push\_<** | == **#push\_==**

29. Additive-expression -> Term D

30. Additive-expression-prime -> Term-prime D

31. Additive-expression-zegond -> Term-zegond D

32. D -> Addop Term **#add\_minus** D | EPSILON

33. Addop -> + **#push\_+**| - **#push\_-**

34. Term -> Factor G

35. Term-prime -> Factor-prime G

36. Term-zegond -> Factor-zegond G

37. G -> \* Factor **#mult** G | EPSILON

38. Factor -> ( Expression ) | **#pidAddr** ID Var-call-prime | NUM **#push\_NUM**

39. Var-call-prime -> (Args ) **#jf** **#push\_return**| Var-prime

40. Var-prime -> [ Expression ] **#ArrIdx #readAddr** | EPSILON **#readAddr**

41. Factor-prime -> ( Args ) **#jf** **#push\_return** | EPSILON

42. Factor-zegond -> ( Expression ) | NUM **#push\_NUM**

43. Args -> Arg-list | EPSILON

44. Arg-list -> **#spec\_data\_addr\_func #push\_again** Expression **#assign #update\_next\_argAddr** Arg-list-prime

45. Arg-list-prime -> , Expression **#assign** **#update\_next\_argAddr** Arg-list-prime | EPSILON **#pop\_again**