No Left Recursion Grammar

-----------------------------------

1. program → **program id (** identifier\_list **);** declarations subprogram\_declarations compound\_statement **.**

**| program id (** identifier\_list **);** subprogram\_declarations compound\_statement **.**

**| program id (** identifier\_list **);** declarations compound\_statement **.**

**| program id (** identifier\_list **);** compound\_statement **.**

2. identifier\_list → **id** identifier\_list1

2.1 identifier\_list1 → **, id** identifier\_list1 **| epsilon**

3. declarations → **var id :** type declarations1

3.1 declarations1 → **var id :** type declarations1 | **epsilon**

4. type → standard\_type | **array [ num .. num ] of** standard\_type

5. standard\_type → **integer | real**

6. subprogram\_declarations → subprogram\_declaration **;** subprogram\_declarations1

6.1 subprogram\_declarations1 → subprogram\_declaration **;** subprogram\_declarations1 | **epsilon**

7. subprogram\_declaration → subprogram\_head declarations subprogram\_declarations compound\_statement | subprogram\_head subprogram\_declarations compound\_statement

| subprogram\_head declarations compound\_statement

| subprogram\_head compound\_statement

8. subprogram\_head → **function id** **(** parameter\_list **)** **:** standard\_type **;**

**| function id** **:** standard\_type **;**

9. parameter\_list → **id** : type parameter\_list1

9.1 parameter\_list1 → **; id** **:** type parameter\_list1 | **epsilon**

10. compound\_statement → **begin** statement\_list **end** | **begin** **end**

11. statement\_list → statement statement\_list1

11.1 statement\_list1 = **;** statement statement\_list1 | **epsilon**

12. statement → variable **assignop** expression

| compound\_statement

| **if** expression **then** statement

| **if** expression **then** statement **else** statement

| **while** expression **do** statement

13. variable → **id** | **id [** expression **]**

14. expression\_list →expression expression\_list1

14.1 expression\_list1 → **,** expression | **epsilon**

15. expression → simple\_expression | simple\_expression **relop** simple\_expression

16. simple\_expression → term simple\_expression1 | sign term simple\_expression1

16.1 simple\_expression1 → **addop** term simple\_expression1 | **epsilon**

17. term → factor term1

17.1 term1 → **multop** factor term1 | **epsilon**

18. factor → **id (** expression\_list **) | num** | **(** expression **) |** variable| **not** factor

19. sign → **+** | **-**