Actividad 3

Para este caso se realizó la actividad 3 la que se pedía que se modificara la gramática de Triangle, donde se busca realizar varios cambios como el formato EBNF, factorizando por la izquierda todos las reglas y además eliminando la recursión por la izquierda. En este caso se muestra la gramática ya corregida, donde en varios casos se aplicó una de las especificaciones, en otros casos 2 casos o en otros casos los 3. También ocurrió que algunos casos no se podía realizar mayor cambio y se quedó igual a como ya estaba. En varios casos se le añadió el signo de "*" que significa según la presentación "significa 0 o más ocurrencias de", entonces en varios casos se utilizó.
Entonces a continuación se ve cada regla numerada con los cambios respectivos.

1. Command ::= single-Command (; single-Command)*

2. single-Command ::= V-name := Expression
   | Identifier (Actual-Param-Seq)
   | let Declaration in single-Command
   | begin Declaration in single-Command
   | if Expression then single-Command else single-Command
   | while Expression do single-Command

3. Expression :: = second-Expression
   | let Declaration in Expression
   | if Expression then Expression else Expression

4. second-Expression ::= primary-Expression (Operator primary-Expression)*

5. primary-Expression ::= Integer-Literal
   | Character-Literal
   | V-name
   | Identifier (Actual-Param-Seq)
   | Operator primary-Expression
   | (Expression)
   | { Record-Aggregate }
   | [Array-Aggregate]

6. record-Aggregate ::= Identifier ~ Expression (, Identifier ~ Expression)*

7. array-Aggregate ::= Expression (, Expression)*

8. V-name ::= Identifier (. Identifier | [Expression])*

9. Declaration ::= single-Declaration (; single-Declaration)

10. single-Declaration ::= (const Identifier ~ Expression
| var Identifier : Type-denoter
| proc Identifier (Formal-Parameter-Sequence) ~ single-Command
| func Identifier (Formal-Parameter-Sequence) : Type-denoter ~ Expression
| type Identifier ~ Type-denoter)

11. Formal-Param-Seq ::= proper-FP-Sequence?

12. proper-FP-Sequence ::= Formal-Parameter (, Formal-Parameter)*

13. Formal-Parameter ::= Identifier : Type-denoter
| var Identifier : Type-denoter
| proc Identifier (Formal-Param-Seq)
| func Identifier (Formal-Param-Seq) : Type-denoter

14. Actual-Param-Seq ::= proper-AP-Sequence?

15. proper-AP-Sequence ::= Actual-Parameter (, Actual-Parameter)*

16. Actual-Parameter ::= Expression
| var V-name
| proc Identifier
| func Identifier

17. Type-denoter ::= Identifier
| array Integer-Literal of Type-denoter
| record Record-Type-denoter end

18. Record-Type-denoter ::= Identifier : Type-denoter (, Identifier : Type-denoter)*