**Compilers’ Theory**

**Milestone -2-**

**Note: *Terminal\_tokens* are written in bold and italic. non\_Terminal\_tokens are written in Bold.**

1. **Program → UserFunc MainFunc.**
2. **UserFunc → Function | *ε***
3. **MainFunc → Datetype *main* *()* Body**
4. **Function → Fun\_dec Body UserFunc**
5. **Fun\_dec → Datatype *identifier* ArgList**
6. **Datatype 🡪 *int* | *float* |*string***
7. **ArgList → *(*Arguments*)* | *()***
8. **Arguments → Arguments*,* *identifier* | *identifier*** *(left recursive)*
   * + **Arguments → Datatype *identifier* Arg**
     + **Arg →, Datatype *identifier* Arg | *ε***
9. **Body → *{*Stat\_seq return-stmt *}***
10. **Stat\_Seq → Stat\_Seq ; Statement | Statement** *(left recursive)* 
    * **Stat\_Seq → Statement State**
    * **State →*;* Statement State | *ε***
11. **Statement 🡪 if-stmt | repeat-stmt | assign-or-funcallstmt**

**| read-stmt | write-stmt| Decl-stmt| return-stmt**

1. **if-stmt → *if* Condition *then* Stat\_Seq ElseClosure**
2. **elseif-stmt → *elseif* Condition *then* Stat\_Seq ElseClosure**
3. **ElseClosure → *else* Stat\_Seq *end* | elseif-stmt**
4. **Condition → Expression RelOp Expression ConditionClosure**
5. **ConditionClosure → ConditionOps Condition | ε**
6. **assign-or-funcallstmt → assign-stmt| fun-call *(left factoring)***

* **assign-or-funcallstmt → *identifier* A**
* **A → fun-call|assign-stmt**

1. **Equation → Equation AddOp Term | Term** *(left recursive)*
   * + **Equation → Term Equ**
     + **Equ → AddOp Term Equ | ε**
2. **Term → Term MultOp Factor | Factor** *(left recursive)*
   * + **Term → Factor Ter**
     + **Ter → MultOp Factor Ter | ε**
3. **Factor → *constant* |*identifier* |FunCall *(left factoring)***

* **Factor → *constant* |*identifier* A| *(*Expression*)***
* **A → fun-call | ε**

1. **RelOp → *<*|*>*|*=*|*<>***
2. **CondationOps → *“||”* | *&&***
3. **AddOp → *+* | *-***
4. **MultOp → *\** | */***
5. **Expression** **→ *String* | Term | Equation** ***(left factoring)***

* **Expression** **→ *String* |exp**
* **exp *→* TermE**
* **E → Equ|ε**

1. **repeat-stmt → *repeat* Stat\_Seq *until* Expression**
2. **assign-stmt → *identifier:=* Expression**
3. **read-stmt → *read* *identifier***
4. **write-stmt → *write* Expression**
5. **Decl-stmt → DataType Id**
6. **Id → *identifier* | assign-stmt IdClosure *(left factoring)***

* **Id → *identifier* B IdClause**
* **B → assign-stmt | ε**

1. **IdClause → , Id | ε**
2. **fun-call → callArgList**
3. **CallArgList → *(*ArgumentsCall*)* | *()***
4. **ArgumentsCall → ArgumentsCall, *identifier* | *identifier*** ***(left recursive)***
   * + **ArgumentsCall → *identifier* ArgCall**
     + **ArgCall →, *identifier* ArgCall | ε**
5. **return-stmt → *return* Expression**