Name: Fahad Ali
Reg ID: L1F22BSCS0421

Course: CC

Section: G5

Project Phase 2 Document: Jugr Language.

---

1. Context-Free Grammar (CFG) for Jugr Language

1. Program Structure

<Program> -> <ImportLibraries> <FunctionList> <mainFunc>

<ImportLibraries> -> shamil_karo string_literal <ImportLibraries> | ε

2. Data Types

<DType> -> pura_number | thora_number | jumla | harf

3. Expressions

<Expression> -> <DType> Identifier ;

| <DType> Identifier = <E> ;

| Identifier = <E> ;

| <E> ;

<E> -> <E> + <F> | <E> - <F> | <F>

<F> -> <F> * <G> | <F> / <G> | <F> % <G> | <G>

<G> -> Identifier | Number | <U> | (<E>)

<U> -> Identifier++ | ++Identifier | Identifier-- | --Identifier

4. Arrays

<ArrayStatement> -> <DType> Identifier <Array> = <list> ;

| <DType> Identifier <Array> ;

| Identifier <Array> = <E> ;

<Array> -> [<E>] <Array> | [<E>]

<list> -> { <Element> }

<Element> -> <Value> , <Element> | <Value>

<Value> -> number | <list>

5. Conditional Statements

<condStatement> -> maan_lo_ye (<condition>) { <code_block> } <condElse>

<condElse> -> warna_maan_lo_ye (<condition>) { <code_block> } <condElse>

| bas_maan_lo_bhai { <code_block> }

| ε

<condition> -> <E> <RelOp> <E> | <E> <LogOp> <E> | <E>

<RelOp> -> ke_barabar_ha | ke_barabar_nahi_ha | se_bada_ha | se_chota_ha

| se_bada_ya_barabar_ha | se_chota_ya_barabar_ha

<LogOp> -> aur_bhi | ya_phir

6. Loops

<LoopType> -> <ForLoop> | <WhileLoop> | <DoWhile>

<ForLoop> -> ghuma_de (<ForInit> <condition> ; <UpdateLoop>) { <code_block> }

<WhileLoop> -> ghumata_reh (<condition>) { <code_block> }

<DoWhile> -> pehle_kar { <code_block> } ghumata_reh (<condition>) ;

<UpdateLoop> -> Identifier = <E> | <U>

7. Input and Output

<Input> -> sunle (string_literal , <Id_List>) ;

<Id_List> -> Identifier , <Id_List> | Identifier

<Output> -> likhle (<Output_Content>) ;

<Output_Content> -> <Output_Content> + <Output_Value> | <Output_Value>

<Output_Value> -> string_literal | Identifier | Number | <E>

8. Functions

<FunctionList> -> <function> <FunctionList> | ε

<function> -> <DType> Identifier (<func_Parameter>) { <code_block> }

<mainFunc> -> khali_kaam_kar shuru_hoja ( ) { <code_block> }

<func_Parameter> -> <DType> Identifier <ParamRest> | ε

<ParamRest> -> , <DType> Identifier <ParamRest> | ε

<ReturnStatement> -> nikal_lo <E> ; | nikal_lo ;

9. Comments

<Comment> -> <SingleLine> | <MultiLine>

<SingleLine> -> // {Any character} \n

<MultiLine> -> /* {Any character} */

10. Code Block and Statements

<code_block> -> <Statement> <code_block> | ε

<Statement> -> <Expression> | <ArrayStatement> | <condStatement>

| <LoopType> | <Input> | <Output> | <FunctionCall> ;

| <ReturnStatement> | tham_ja ; | wapas_aja ; | ;

---

2. First And Follow sets

Non-Terminal	Production Rules	First( )	Follow( )
<DType>	pura_number | thora_number | jumla | harf	{pura_number, thora_number, jumla, harf}	{Identifier, shuru_hoja }
<RelOp>	ke_barabar_ha | ke_barabar_nahi_ha | se_bada_ha | se_chota_ha | se_bada_ya_barabar_ha | se_chota_ya_barabar_ha	{ke_barabar_ha, ke_barabar_nahi_ha, se_bada_ha, se_chota_ha, se_bada_ya_barabar_ha, se_chota_ya_barabar_ha}	{Identifier, Number, ++, --, (}

---

3. Program:

```
pura_number shuru_hoja() {
   ghuma_de (pura_number i = 1; i se_chota_ya_barabar_ha 10; i++) {
      likhle("5 * " + i + " = " + 5*i);
   }
   nikal_lo 0;
}
```

---

4. Parse Tree of Program: