# *Compiler construction project*

## Compiler name : C3

## Team members :

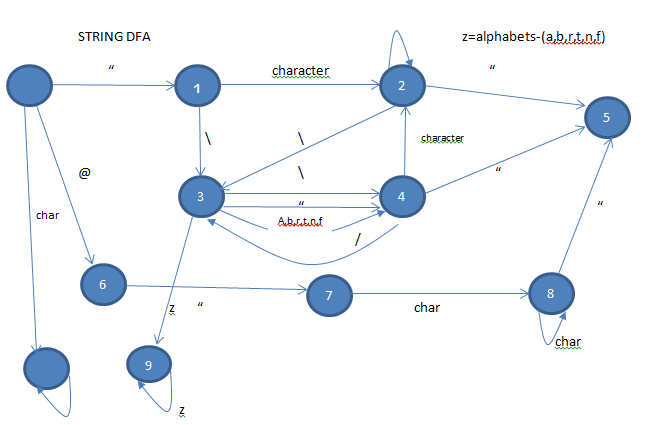
## Saqib abbas (EP-1349112)

## Arsalan shaikh (EP-1349057)

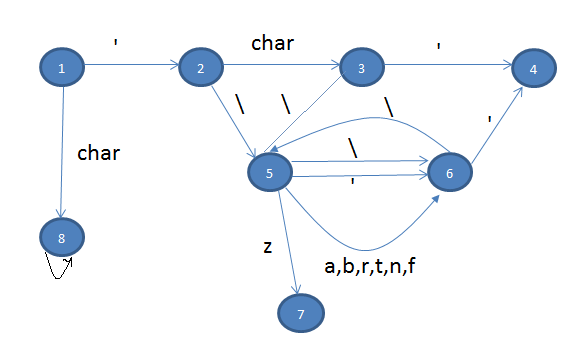
## Abdul basit rana (EP-1349001)

## Project incharge : Miss amna

STRING DFA



CHARACER DFA:



INT DFA: (D=0,1,2,3,4,5,6,7,8,9)

D NULL

Z=CHAR-D .

.

Z

FLOAT DFA:

D D

D . D

.

## KEYWORDS:

|  |  |
| --- | --- |
| KEYWORDS | CLASSPART |
| Class | class |
| Line | Data type |
| Num | Data type |
| Char | Data type |
| Bginum | Data type |
| Bool | Data type |
| Repeat | Repeat |
| Either | Either |
| Or | Or |
| Repeatwhile | Repeatwhile |
| Bound | Bound |
| Default | Default |
| Shift | Shift |
| Void | Void |
| Main | Main |
| Define | Define |
| Retun | Return |

## Operators

|  |  |
| --- | --- |
| KeyWord | ClassPart |
| = | = |
| += | Asg-Operator |
| -= | Asg-Operator |
| /= | Asg-Operator |
| %= | Asg-Operator |
| \*= | Asg-Operator |
| < | Relation-Operator |
| > | Relation-Operator |
| <= | Relation-Operator |
| >= | Relation-Operator |
| != | Relation-Operator |
| == | Relation-Operator |
| ++ | Inc-Dec |
| -- | Inc-Dec |
| + | P-M |
| - | P-M |
| \* | \* |
| / | D-M |
| % | D-M |
| ++ | Inc-Dec |
| -- | Inc-Dec |
| && | && |
| || | || |
| & | & |
| ! | ! |
| | | | |
| ~ | ~ |
| << | Insert |
| >> | Insert |

## LOOKUPS:

* Dlookup(N,S) for looking up identifiers

If (Dookup(N,S)==false)

Insert(N,T,S)

Else

“redeclaration error”

* Type Lookup(N,S) for type checking

If (Dookup(N,S)==false)

Insert(N,T,S)

Else

“redeclaration error”

* Return Lookup(N,PL)for return type checking

## INSERT:

* Insert (N,T,S) for declaring identifiers
* Insert (N,TE) for checking type expression

## COMPATIBILITY CHECK:

* Compatibility (LT,RT,Opr) for operands compatibility
* Compatibility (OPtype)
* Return Type Compatibility (AL)

## Main Method

<Main> [create scope]Main() { [create scope]<Main-Body> [destroy scope]} [destroy scope] <Main-Foll>

<Main-Body> <M-st> | ϵ

< Main-Foll > < class> <Main-Foll> | ϵ

## Class

<Class> Class ID(dlookup){ [create scope]< class body > [destroy scope]}

<class body> ϵ| < Datatype\_identifier ><ClassBody>|<identifier> <ClassBody>|< Fn-Def> <ClassBody>

# Variables

<Datatype\_identifier> DT ID(dlookup) <options>

<Identifer> ID(dlookup) <options>

< options > <Dec-Assgn>|< Array >

# Declaration and assignment

<Dec-Assgn> < Init > < List >

<Init> ϵ | = < Init1 >

<Init1> ID (dlookup < Init > | < Const >

<List> ;| , < ID > < Init > < List >

# Array

< Array > [ < size > ] <detail> ;

< size > <OE> | ϵ

< Detail > = {<F> <ID-const-More> } | ϵ

< ID-const-More > ϵ |,<F> <ID-const-More>

# Expression

<OE> < AE > < OE’ >(Compatibility (LT,RT,Opr))

<OE’> || < AE > < OE’ > | ϵ

<AE> < RE > < AE’ >(Compatibility (LT,RT,Opr))

<AE’> && < RE > < AE > | ϵ

<RE> < E > < RE’ >

<RE’> < Ro > < E > < RE’ > | ϵ

<E> < T > < E’ >(Compatibility (LT,RT,Opr))

<E’> < P-M > < T > < E’ > (Compatibility (LT,RT,Opr))| ϵ

<T> < F > < T’ >(Compatibility (LT,RT,Opr))

<T’> < M-D-M > < F > < T’ > (Compatibility (LT,RT,Opr))| ϵ

<F> ID <Additions> | < Const >

<Additions> < Fn-Call> | < Array > | <inc\_dec> | . <F>

# Repeat

<Repeat> Repeat ( < Datatype\_identiifer > ; ID <RO>< OE> ; < F1 > ) { [create scope]< M-ST>[destroy scope] }

<F1> ID (dlookup )<F2> | <INC\_DEC> ID | ϵ

<F2> <assg\_opper> <OE>| <INC\_DEC>

# Repeat While

<Repeat-While> Repeat-While ( <OE> ) { [create scope]< M-ST>[destroy scope] }

# Either Or

<Either-Or> Either ( < OE> ) { [create scope]< M-St >[destroy scope] } < O-Or>

< O-Or> ϵ| Or { [create scope]< M-ST >[destroy scope] }

# Function Definition

< Fn-Def>(T,Pl) Define <DT(T=DT.VP)> ID (dlookup) (pl=”” < FN\_ Def \_parameter(pl) > ) { [create scope]<Fn\_body > }

< FN\_ Def \_parameter>(pl) DT(pl+=DT.VP) ID (dlookup )< Fn\_INIT> < FN\_ Def \_parameter 1(pl)>

< FN\_ Def \_parameter1>(pl) ,DT (pl+=DT.Vp)ID(dlookup) < Fn\_INIT> < FN\_ Def \_parameter 1(pl)>

< FN\_ Def \_parameter> ϵ

< FN\_ Def \_parameter1>(AL) ϵ [AL=PL]

< Fn\_INIT> =<OE>\*return type| ϵ

< Fn\_body> < M-ST><Fn-Return> | ϵ

< Fn-return >(T) return < OE(T)> Compatibility(T,AL);| ϵ

# Function Calling

< Fn-Call>(N) (< FN\_call\_parameter >(return type lookup(N,PL)) ) ;

< FN\_call\_parameter>(PL) <DT > ID(dlookup(N,S)) < Fn\_INIT> < FN\_call\_parameter 1>

< FN\_call\_parameter>(PL) ID(dlookup) < Fn\_INIT> < FN\_call\_parameter1 >

< FN\_call\_parameter>( PL+=AL) ϵ

< FN\_call\_parameter1>(PL) , < FN\_call\_parameter2>

< FN\_call\_parameter1> (PL+=AL) ϵ

< FN\_call\_parameter2> <DT> ID(dlookup) < Fn\_INIT> < FN\_call\_parameter1 >

< FN\_call\_parameter2> ID(dlookup) < Fn\_INIT> < FN\_call\_parameter 1>

<Fn-Object> . <ID> <Fn-Call>

# Shift

< Shift> shift ( ID (dlookup) ) { < Bound> < Default >}

< Bound> bound < Const > : < < Bound-Body> > < Bound > | ϵ

< Bound-Body> <M-St> | ϵ

<Default> Default : < M-ST > | ϵ

# Others

<M-ST> ϵ| < S-ST > < M-ST >

<S-ST> < Datatype\_identiifer >| <Identifier> | < Repeat > | < Repeat-While > | < Either-Or > | < Shift > | < Define > |< Fn-Def > | < Fn-Call > | <Break>

<Const> String-Const | Integer-Const | double-Const | Char-Const

<Inc-Dec> ID (dlookup)<Inc-Dec>