**• Matching identifiers**

Letter = [a-z]|[A-Z]

Digit = [0-9]

Identifier = letter (letter|\_|digit)\* , identifier | letter (letter|\_|digit)\*

**• Matching reserved words**

Yesif-Otherwise-------- > Condition

RepeatWhen/Reiterate-------- > Loop

**regular expressions**

Yesif-Otherwise -------- > Condition

RepeatWhen/Reiterate -------- > Loop

(+,-,\*,/)-------- >Arithmetic Operation

(&&, ||, ~)-------- >Logic operators

(==, , !=, <=, >=) -------- >relational operators

(=)-------- >Assignment operator

(->)-------- >Access Operator

(,),[,]{,} -------- >Braces

“,’ -------- >Quotation Mark

/@-------- >Comment

@/-------- >Comment

/^ -------- >Comment

$-------- >Token Delimiter

. -------- >Line Delimiter

Omw -------- > Integer

SIMww -------- > SInteger

Chji -------- > Character

Seriestl -------- > String

IMwf -------- > Float

SIMwf -------- > SFloat

NOReturn-------- >Void

GetBack-------- > Return

OutLoop-------- >Break

Loli -------- > Struct

Include -------- > Inclusion

Start -------- > Start

Last -------- > End

All ----🡪**∑**{+,-,\*,/, &&, ||, ~,==, , !=, <=, >=,=,->, ( , ) , [ , ] { , } , -> , “,’ , /@ ,@/ ,/^ , $, Include ,Start ,Last ,Omw , SIMww , Chji , Seriestl , IMwf , SIMwf , Yesif-Otherwise, RepeatWhen, Reiterate}

Identifier = letter (letter|\_|digit)\* , identifier | letter (letter|\_|digit)\*

Type ----🡪 Omw , SIMww , Chji , Seriestl , IMwf , SIMwf

Integers: [+]?[0-9]+

Floats: [+]?(([0-9]+ (.[0-9]⇤)?|.[0-9]+)([eE][+]?[0-9]+)?)

String constants: “([a-zA-Z0-9]|\[a-zA-Z])⇤”

Expr = ()

Statement = identifier

L(a) = identifier

**regular expression of Start**

R = Start

**regular expression of define var**

R = (ε |Type) (identifier | ,)+ (ε | = (Constant| identifier)).

**regular expression of Condition**

R = Yesif (identifier relational operators (identifier | Constant) )

{

(Statement (. | Last))\*

}

Otherwise

{

(Statement (. | Last))\*

}

**regular expression of Function**

R = (Type | void) identifier (ε | (Type identifier)\*)

{

(Statement (. | Last))\*

(ε | GetBack (. | Last))

}

**regular expression of Loop**

Reiterate

{

(Statement (. | Last))\*

(ε | OutLoop (. | Last))

} RepeatWhen (identifier relational operators (identifier | Constant))

**regular expression of Include**

Include identifier (. | Last)

**regular expression of Struct**

Loli identifier

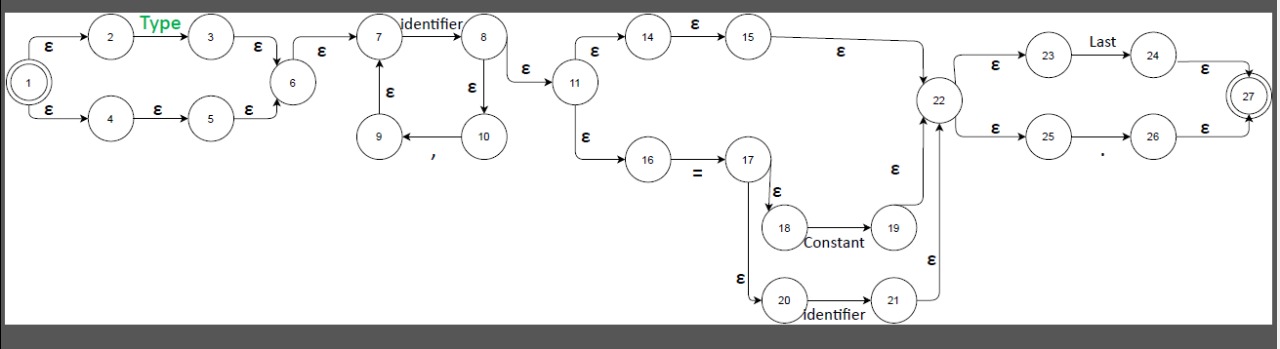
{

(Statement (. | Last))\*

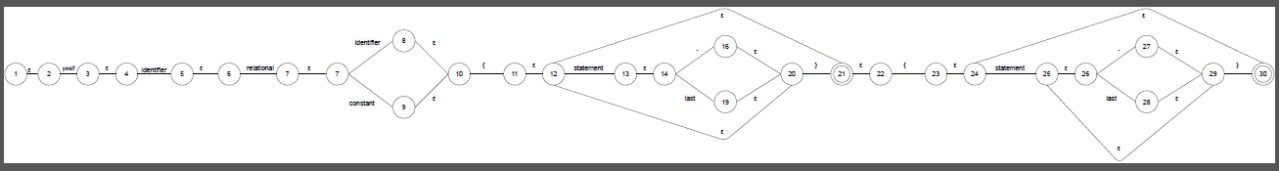
}

**Non-deterministic Finite Automata (NFA)**

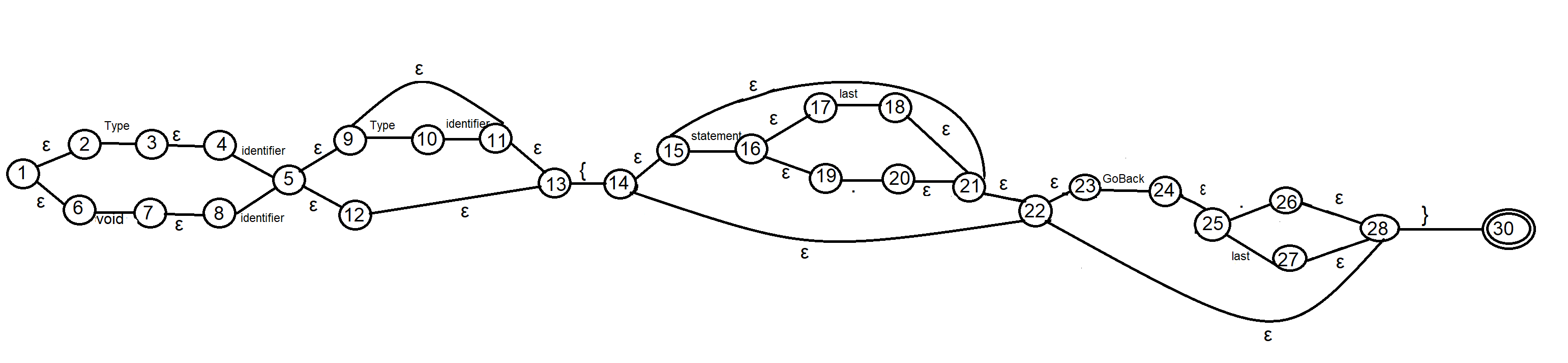
**(NFA) of define var**

****

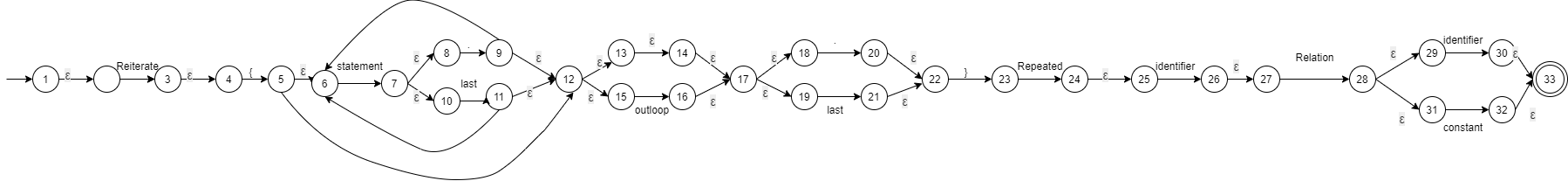
**(NFA) of Condition**

****

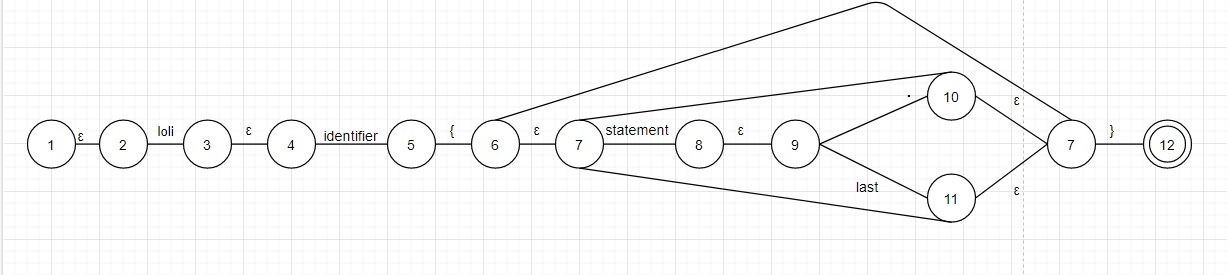
**(NFA) of Function**

****

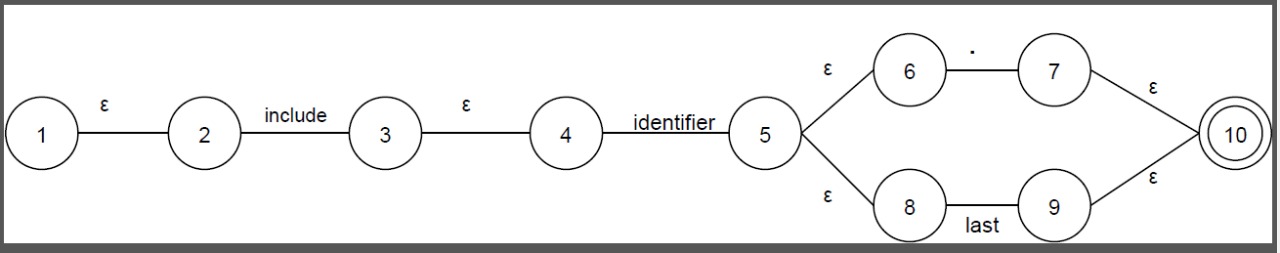
**(NFA) of Loop**

****

**(NFA) of Struct**

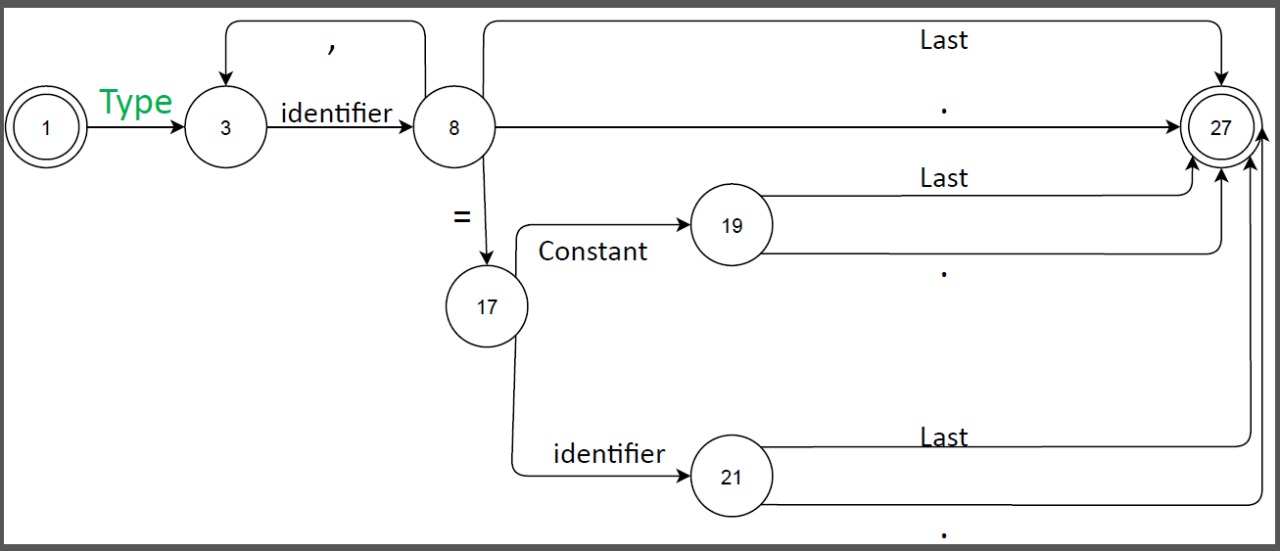
****

**(NFA) of Include**

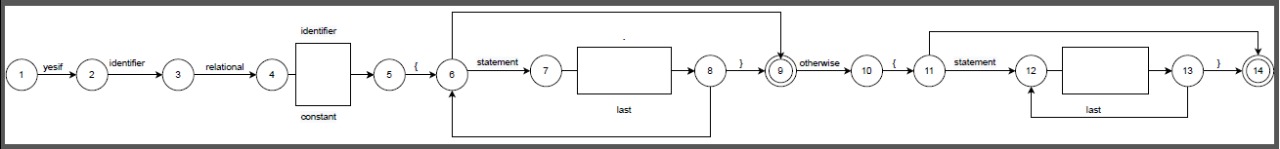
******

**Deterministic Finite Automata (DFA)**

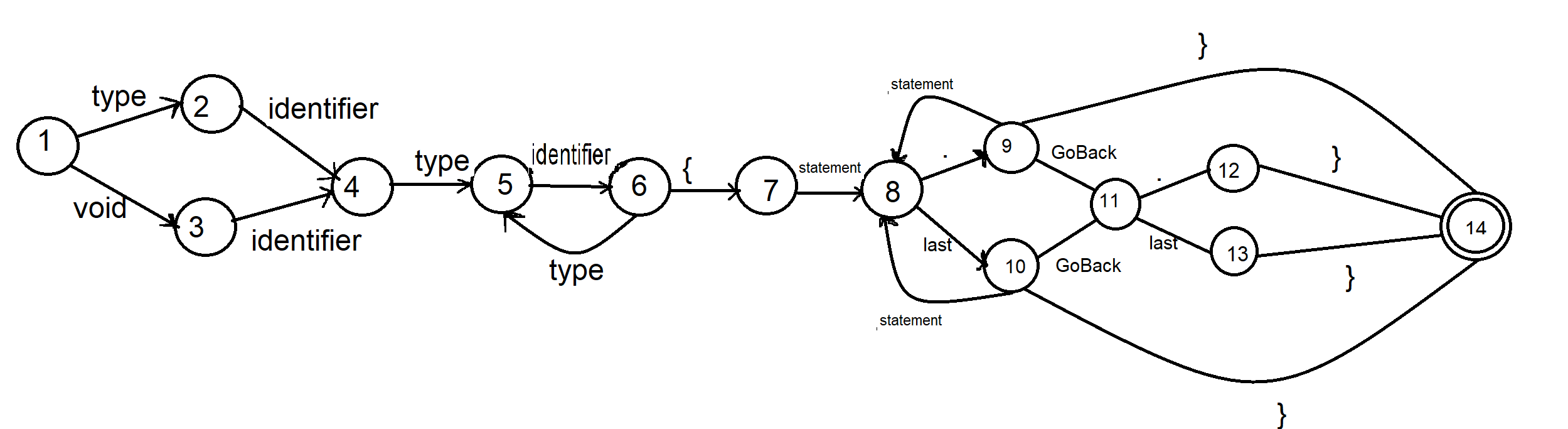
**(DFA) of define var**

****

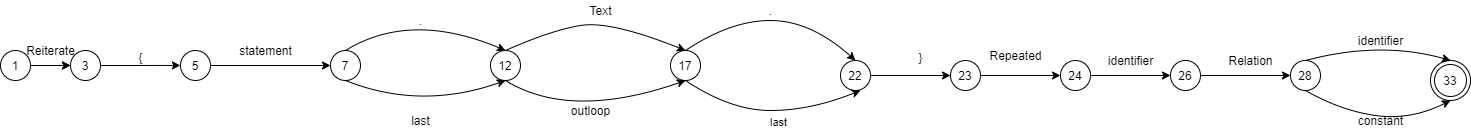
**(DFA) of Condition**



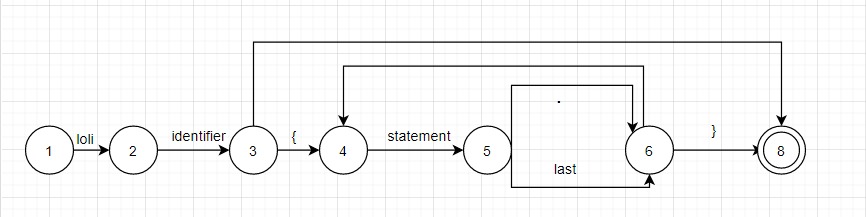
**(DFA) of Function**

****

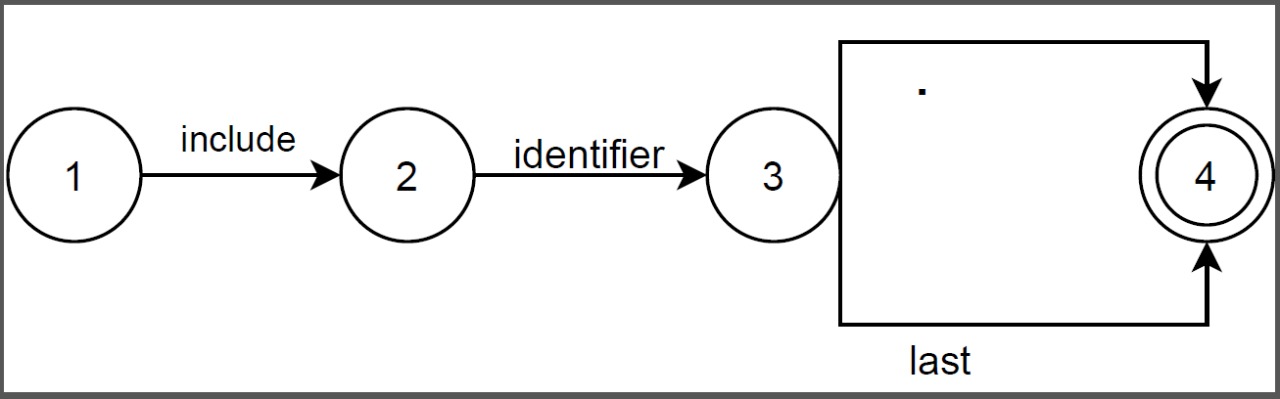
**(DFA) of Loop**

****

**(DFA) of Struct**

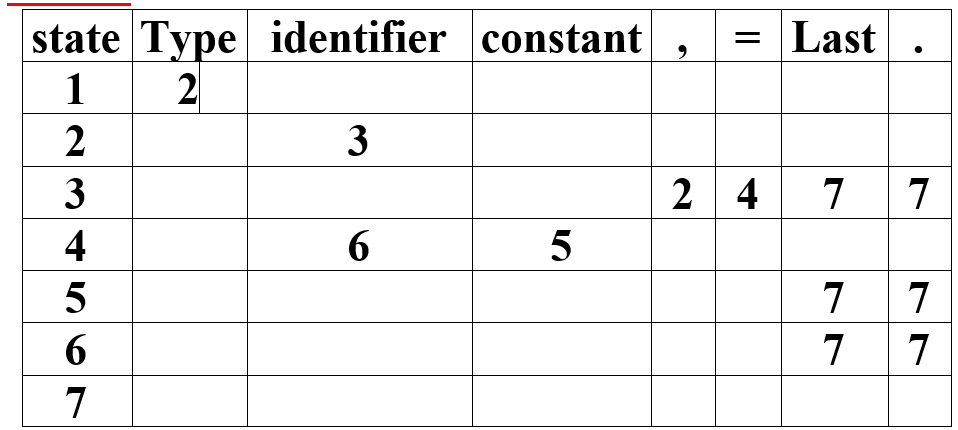
****

**(DFA) of Include**

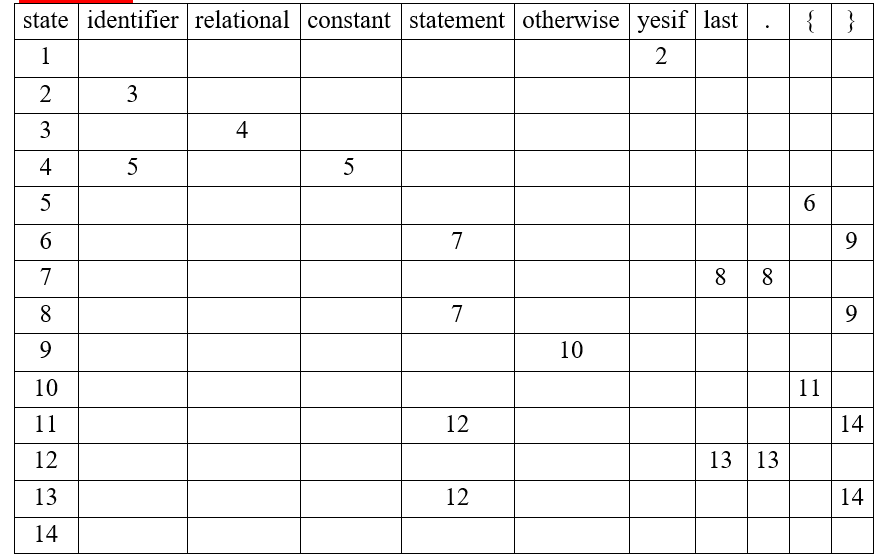
****

**Transition Table**

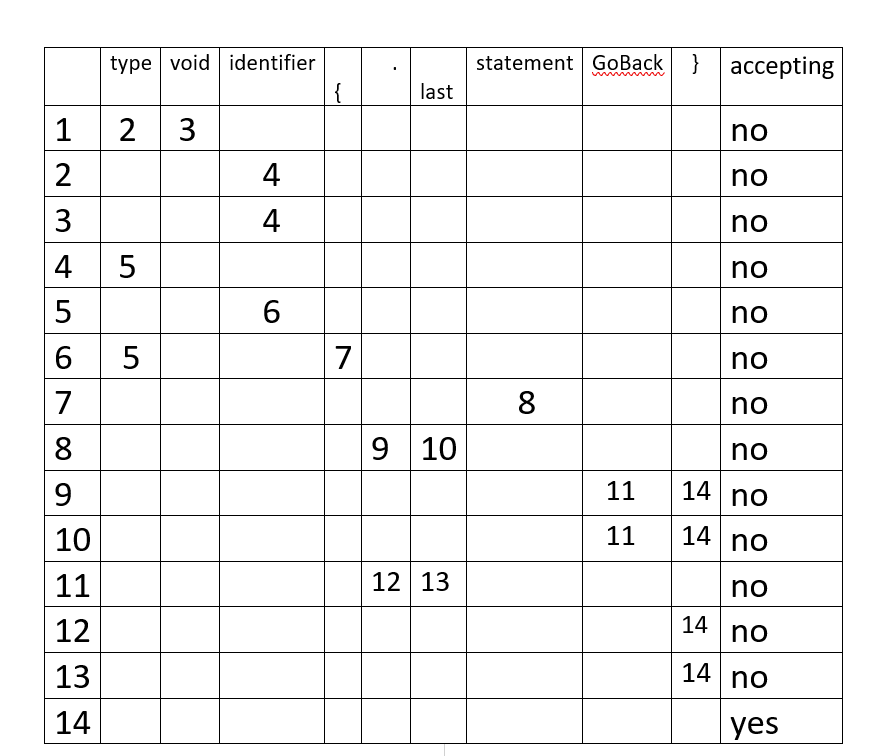
**Transition Table of Define Variable**

****

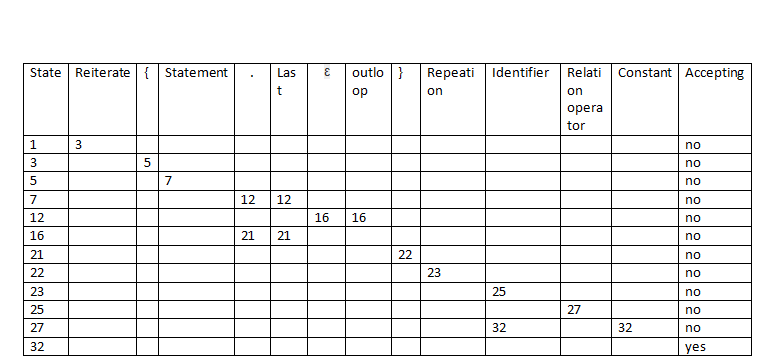
**Transition Table of Condition**

****

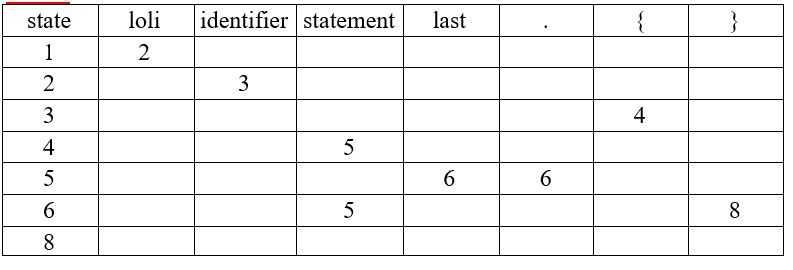
**Transition Table of Function**

****

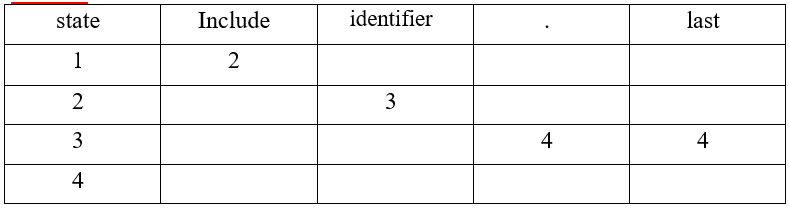
**Transition Table of Loop**

****

**Transition Table of Struct**

****

**Transition Table of Include**

****