COMPILER DESIGN PROJECT

GROUP6

AP19110010518 Anishka Chauhan
AP19110010446 Alugubilli Navya Sri
AP19110010456 Garapati Vishnu Srinija
AP1911001040 Anuraag Tsunduru
AP19110010318 P.Sai Suhas
AP19110010331 Kalash sailesh

ACKNOWLEDGEMENT

We would like to express our sincerest gratitude and appreciation to our project supervisor/mentor, Ms.Jaya Lakshmi for her guidance and support throughout the project period. This project gave us a lot of exposure and helped us to learn many new things.

We would also like to extend my gratitude to the Management, CR/CS department, and all involved faculty from SRM University, AP, for enabling us to have this experience and helping us achieve it.

We are grateful for these wonderful teammates who helped each other and made sure that this project was successful.

CONTENTS

OBJECTIVE:

• Conceptualization and Design:

o Data types available in the language

o Syntax of variable declaration if your language needs pre declaration before its use. Otherwise, the assumptions.

o One decision making statement

o At least two iterative statements

o CFG for at least five constructs in your language

o Design of Parser

o Semantic Actions

o Syntax of target language

o Following features are optional

▪ Derived Data types like Array/Structure etc.

▪ Function/Procedure/Module

• Implementation:

o Lexical Analyzer

o Parser

LEARNINGS :

- how to use lex and yacc tools

- how to implement compiler design concepts like lexical analyser and LALR.

- deeper understanding of concepts like error handling and shift reduce parser.

THEORY:

Data types available in the source language :

Data type	keyword	Description
Number	int,double,num,float	Numbers in c are used to represent numeric literal
String	Char	Strings represent a sequence of characters

Data types available in the target language :

Data type	Keyword	Description
Number	int,double,num	Numbers in Dart are used to represent numeric literal
String	string	Strings represent a sequence of characters

Syntax of variable declaration of source language if needs pre declaration before its use. Otherwise, the assumptions.

A variable in simple terms is a storage place which has some memory allocated to it. Basically, a variable used to store some form of data. Different types of variables require different amounts of memory, and have some specific set of operations which can be applied on them.

```
variable_name;
 or for multiple variables:
    variable1_name, variable2_name, variable3_name;
```

A variable name can consist of alphabets (both upper and lower case), numbers and the underscore '_' character. However, the name must not start with a number.

Syntax of variable declaration of target language if needs pre declaration before its use. Otherwise, the assumptions.

Variable is an identifier used to refer to a memory location in computer memory that holds a value for that variable, this value can be changed during the execution of the program. When you create a variable in Dart, this means you are allocating some space in the memory for that variable. The size of the memory block allocated and the type of the value it holds are completely dependent upon the type of variable.

Syntax:-

```
var <variable_name>;
```

Or

```
var <name> = <expression>;
```

One decision-making statement

A conditional/decision-making construct evaluates a condition before the instructions are executed.

<u>If statement:</u>-

The if…else construct evaluates a condition before a block of code is executed.

Syntax:-

```
if(condition)
{
   // Statements to execute if
   // condition is true
}
```

If-else statement:-

An if can be followed by an optional else block. The else block will execute if the Boolean expression tested by the if block evaluates to false.

Syntax:-

```
if (condition)
{
    // Executes this block if
    // condition is true
}
else
{
    // Executes this block if
    // condition is false
}
```

Two iterative statements :

At times, certain instructions require repeated execution. Loops are an ideal way to do the same. A loop represents a set of instructions that must be repeated. In a loop's context, a repetition is termed as an iteration.

For loop:-

The for loop executes the code block a specified number of times.

Syntax:

```
for ( init; condition; increment ) {
   statement(s);
}
```

While Loop:

The while loop executes the instructions each time the condition specified evaluates to true.
Syntax:

```
while(condition) {
   statement(s);
}
```

CFG with parse tree for at least five constructs in your language

While Loop:

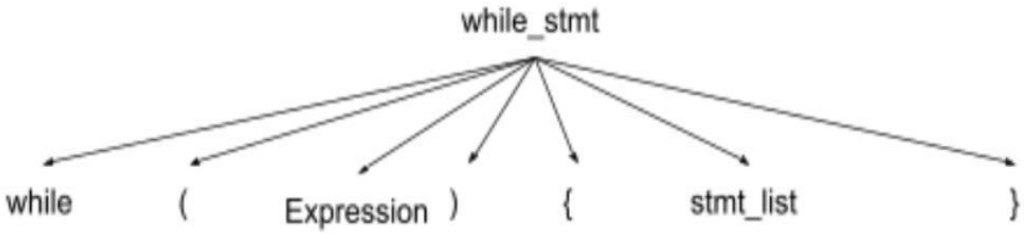
stmt → id | expression | assign_stmt; | if_stmt | while_stmt | dowhile_stmt | print_stmt; | scan_stmt; |...
stmt_list → body | (stmt)*
expression→ id | expression | expression operator expression |expression operator operator expression ε |..
operator → + | - | * | / | % | ** | // | == | =! | < | <= | > | >=
datatype → int | float
print_stmt → printf (“”)
scan_stmt → scanf (id)

while_stmt → while (<expression>)
{
<stmt_list>
}

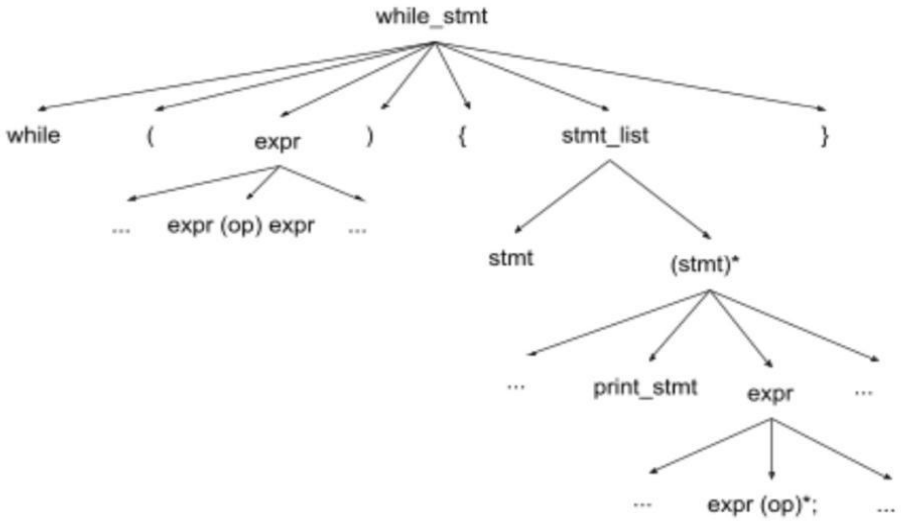
Example Code:

```
while (i<6) {
        printf (“%d \n”,i);
        i++;
}
```

Parse Tree:



Parse Tree for the Example Code:



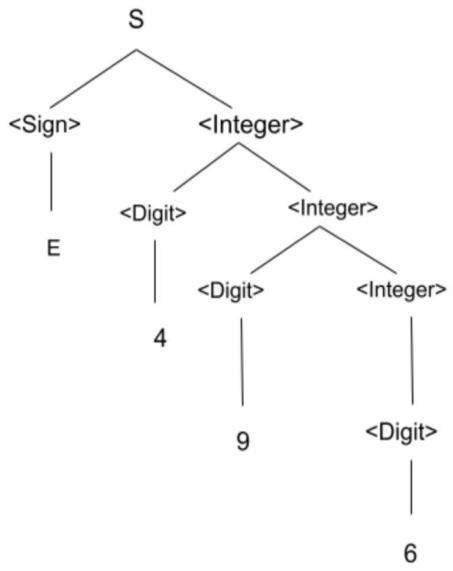
Integer:

P= { S → <sign><Integer>

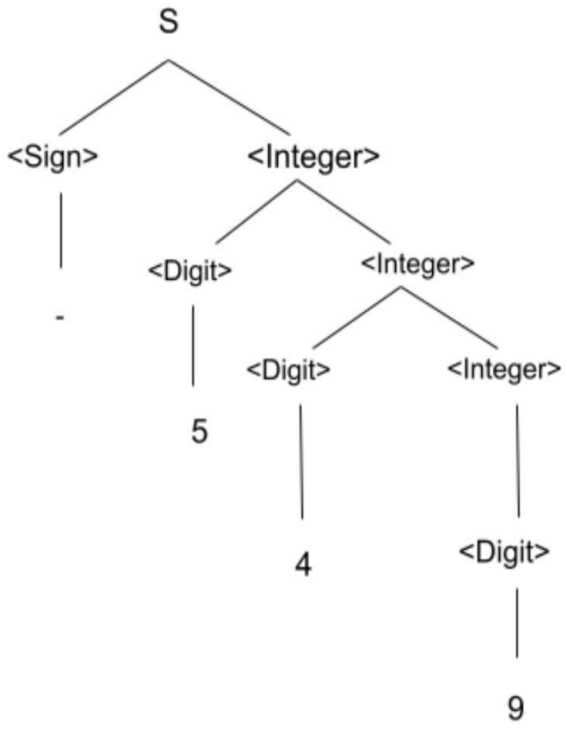
<sign → +|-|ε
<Integer> → <digit> |<digit><Integer>
<digit> → 0|1|2|3……|9
}

G= (V,T,P,S)
NT=V={S, <Sign>, <Integer>, <digit>}
ε= T={+,-,0,1,2,3,....9}

1. Input 496



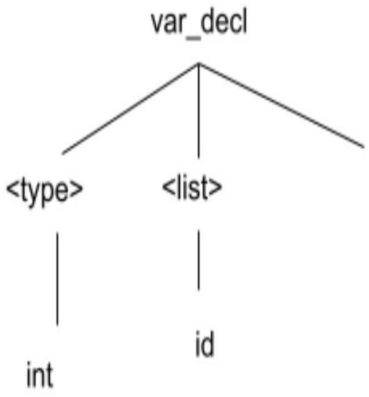
2. -549



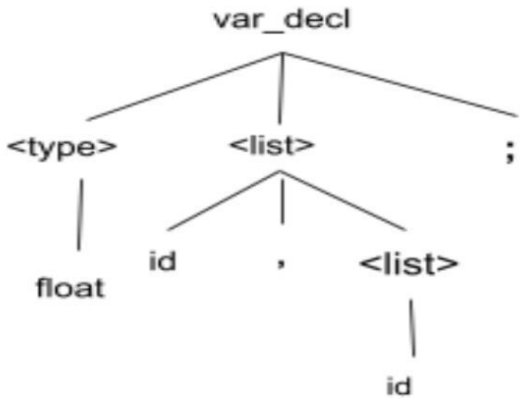
Variable:

<var_ declare> → <type><list>;
<type> → int | char | float
<list> → id | id, <list>

G= (V,T,P,S)
V={<var_declare>, <type>, <list>}
S= <var_declare>
T={; , int,char,float,id, ,}



float a,b;



If-else:

(if_statement) → (logic_expression) then (operation code)

| (logic_expression) then (operation code) else (operation code)

(operation code) →(if_statement) | (any non-if statement)

logic_expression → id | logic_expression (op)*; | logic_expression (op)
logic_expression | logic_expression (op)* logic_expression; | ε |..
op → + | - | * | / | % | ** | // | == | =! | < | <= | > | >= | ...

➔ As a single string exist more than one parse tree it is considered as Ambiguous Grammar

(if_statement) →(logic_expression)then if (logic_expression) then

(operation code) else (operation code)

Parse Tree:

1. *if(condition) {*

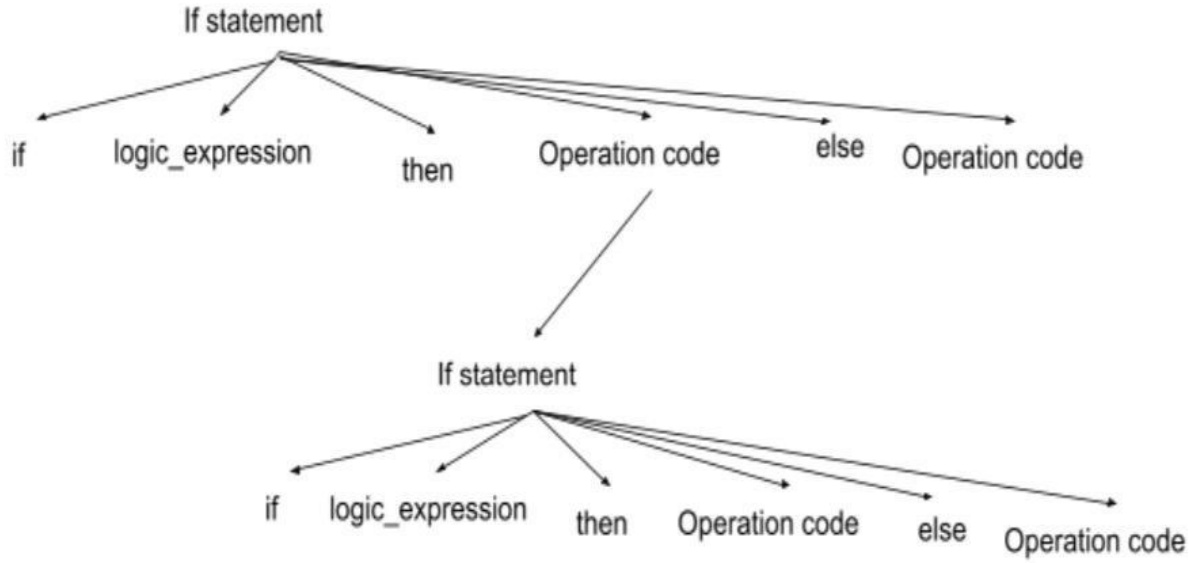
   if (condition) statement; else statement;

   }

   Else{

   Statement

   }



2. if(condition) {

   if (condition) statement; else statement;

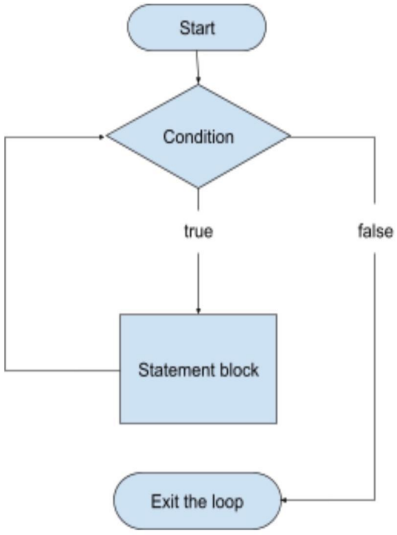
   }

Semantic Actions

While:

- When the control reaches the while loop, it evaluates the condition to determine whether the loop should be run.
- The statements inside the loop are performed if the condition is met.
- After running the statements once, the control is returned to the while condition, which is checked to see if the condition is satisfied before proceeding.
- The statements (loop body) are skipped if the condition is false, and control is passed to the statement after the loop.

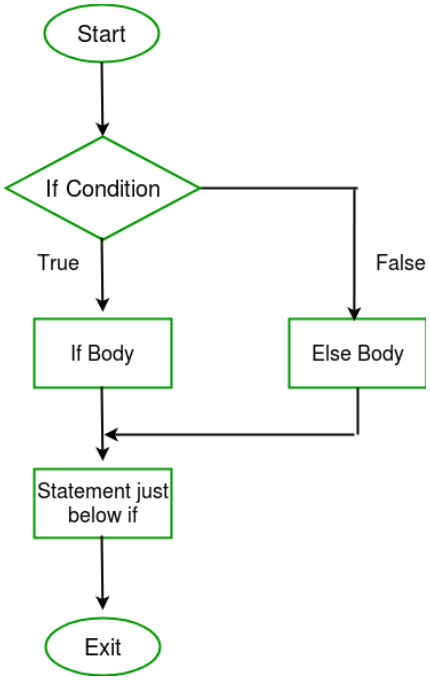
Flowchart:



If-else:

The if-else statement is an addition to the if statement that allows us to conduct two different operations, one for the condition's correctness and the other for its incorrectness. The if and else blocks cannot be run at the same time in this case. It is usually preferable to use an if-else statement because every if condition always results in an otherwise scenario.

Flowchart:



Syntax of the target language

```
void main() {

  // declare and initialize a
variable
  int A = 2;
  int B = 3;

  // displaying the output
  print( A + B );
}
```

- Main() is a function that is a predefined method in Dart. It acts as an entry point to the program
- Void is a special type that is used when a method doesn't explicitly return a value
- “//declare and initialize a variable” is a single-line comment. Multi-line comments start with “/*.....*/”

- Int is a predefined data type that is used to initialize variables and assign them to integer values.
- print() is another function that is a predefined method in Dart. It is used to display the output
- '+' is an operator in Dart that is used to add values be it either of variables or predefined ones
- "{.....}" These parentheses define the content of the program

```
void main() {

  // declare and initialized a variable
  int A = 20;
  double B = 30.20;

  // displaying the output
  print( A + B );
  string name = 'Bob';
  var name = 'Bob';
  object name = 'Bob';
}
```

- Double is another predefined data type in Dart that supports fractional numeric values
- When type int and type double undergo operations with each other, the end result is always in double type.
- Strings are used to assign character types to the variables
- Var is used to assign values to the variables whose type can later be changed as per the need.
- An object is used to assign values to the variables that aren't just restricted to one data type

late String description;

```
void main() {
  description = 'GreatEvening!';
  print(description);
  final name = 'Bob'; // Without a type annotation
  final String nickname = 'Bobby';
}
```

- Late is a keyword that is used when a variable is not found defined in the program. It initializes the variable
- Final is a keyword that assigns a fixed value to a variable. These variables will have one variable throughout the program.

Procedure & Result:

https://github.com/Anishkaa/AP19110010518-CSE306L.git