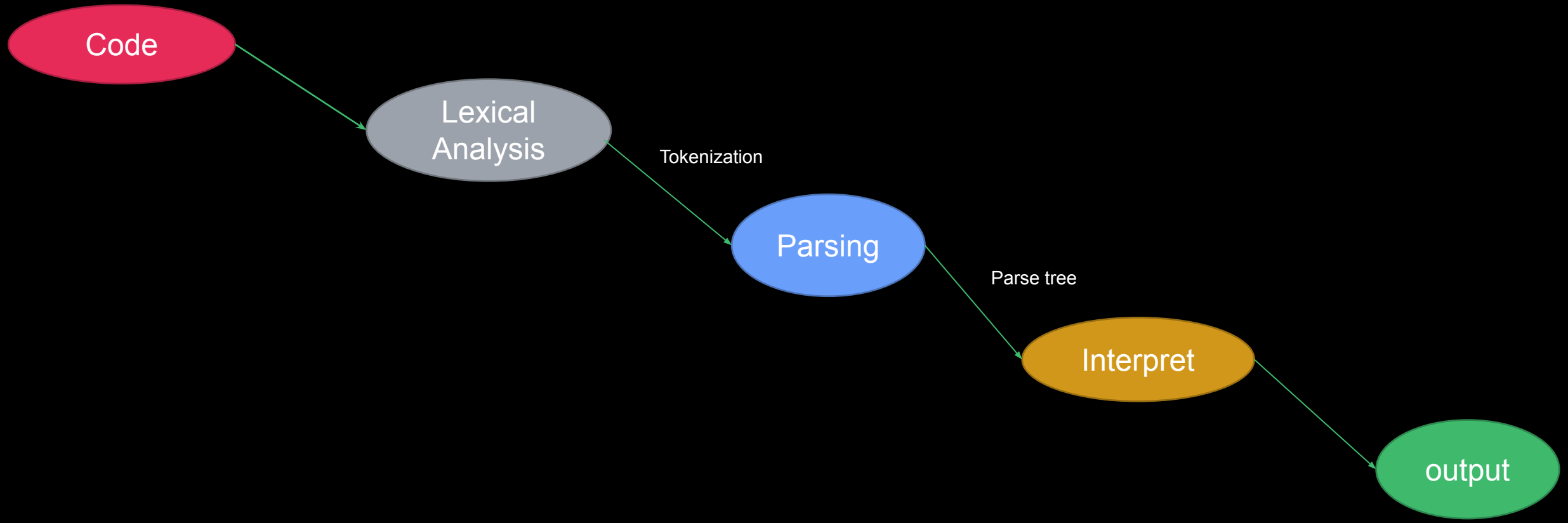
VENOM
SER 502 TEAM-12

- Achuth Reddy Rajula
- Varshik Sonti
- Rajiv Kashyap Jalakam
- Rahul Vuppula
- RajaLaxman Rao Thakkalapelli

DESIGN


Code
Lexical Analysis
Tokenization
Parsing
Parse tree
Interpret
output

Required Tools

1. SWI Prolog CLI
2. Python 3.8

Language Features

Data Types

1. Integer
2. Boolean
3. String

Arithmetic Operators

1. Addition
2. Subtraction
3. Multiplication
4. Division

Relational Operators

1. Equal
2. Not Equal
3. Greater/Lesser than
4. Lesser/Greater or equal to

Execution Steps

1. The first step would be to clone the repository and download the source files
2. Open a terminal and go the source directory
3. Run the command ‘pip install -e .’ to install the modules
4. Run the command ‘venom’
5. Then all the .vnm files are displayed
6. Select the file you want to execute.

IF Condition

```
start
{

        bool check = true;
        int num1 = 2;
        int num2 =20;

        if(num2/num1 == 10)
        {
                flash check;
        }
        else
        {
                flash not(check);
        }

}
end
```

```
start
{
int x =10;
int y=2;

int result =0 ;

if(true)
{
        result =  x+y;
}
else
{
        result = x-y;
}
}
```

For Loop

```
start
{

for(int i=1;i<20;i++)
{
flash i;
}


}
end
```

While Loop

```
start
{
        int x=0;
        while(x != 20)
        {
        x = x + 2;
        flash x;
        }
}
end
```

For Range

```
start
{

for i in range(0:30)
{
flash i;
}


}
end
```

Execution

```
varshik@varshik-Latitude-5510:~$ venom
Select the index of the file you want to execute:

[0] fibonacci.vnm
[1] swap.vnm
[2] print.vnm
[3] factor.vnm
[4] check.vnm
[5] sum.vnm
[6] power.vnm
```

```
Output:
0
1
1
2
3
5
8
13
21
```

Grammar

<program> → <Begin> <statement> <End>

<statement> → <declaration> <statement>
| <assign> <statement>
| <if> <statement>
| <if_else> <statement>
| <while> <statement>
| <for> <statement>
| <for_range> <statement>
| <statement>

<declaration> → <Datatype> <Space> <Identifier> <Space> <Assignment> <Space>
<declaration_helper>
| <Datatype> <Space> <Identifier>

<declaration_helper> → <expression> | <value>

<assign> → <Identifier> <Assignment> <expression>

<print> → <Print> <Space> <StartQuote> <String> <EndQuote>
| <Print> <Space> <Identifier>

<if> → <If> <OpenParenthesis> <condition> <CloseParenthesis> <StartBlock> <statement>
<EndBlock>