



# Team 41 – DAMAN

Nirmit Agrawal

Akshat Bakliwal

Drashti Patel

Manasi Anantpurkar

# TABLE OF CONTENTS

01

Features

03

Lexical  
Analyser

05

Evaluator

02

Grammar

04

Parser

06

Demo



01

FEATURES

# FEATURES

## Data Types

- ❑ Integer (int)
- ❑ Boolean (bool)
- ❑ String (st)

## Control Statements

- ❑ If then else fi
- ❑ Traditional While
- ❑ Traditional For
- ❑ For i in range
- ❑ print()

## Operators

- ❑ Assignment (=)
- ❑ Addition (+)
- ❑ Subtraction (-)
- ❑ Multiplication (\*)
- ❑ Division (/)
- ❑ Greater Than ( > )
- ❑ Less Than ( < )
- ❑ Less Than Equal To (<=)
- ❑ Greater Than Equal To (>=)
- ❑ Comparison (==)
- ❑ Ternary (?:)
- ❑ AND (and)
- ❑ OR(or)
- ❑ NOT(not)
- ❑ NOT EQUAL (!=)



02

GRAMMAR

# GRAMMAR

## Grammar:

P → Program

K → Block

DECL → Declaration

DECL\_VARIABLE → declaring variables

ASS\_VARIABLE → Assignment Declaration Variables

CMD → Command

NC → New Command

AE → Arithmetic Expression

BE → Boolean Expression

EXP → Print Expression

STRING → String Function

TEMP → Temporary Term for String

BOOL\_VAL -> True or False T -> Term

I → Identifier

N → Integer

# GRAMMAR

$P ::= K .$   
 $K ::= \text{start DECL; CMD finish}$

$\text{DECL} ::= \text{ASS\_VARIABLE ; DECL} \mid \text{DECL\_VARIABLE ; DECL} \mid \text{ASS\_VARIABLE} \mid \text{DECL\_VARIABLE}$

$\text{ASS\_VARIABLE} ::= \text{int } I = N \mid \text{bool } I = \text{BOOL\_VAL} \mid \text{st } I = \text{STRING}$

$\text{DECL\_VARIABLE} ::= \text{int } I \mid \text{bool } I \mid \text{st } I$

$\text{CMD} ::= \text{NC; CMD} \mid \text{NC}$

$\text{NC} ::= I = \text{AE} \mid \text{if BE then CMD else CMD fi} \mid \text{while BE begin CMD end} \mid \text{for( int } I = \text{AE; BE ; AE) begin CMD end} \mid \text{for } I \text{ in range(N,N) begin CMD end} \mid \text{BE? CMD : CMD} \mid \text{print (EXP)} \mid K$

$\text{EXP} ::= \text{AE;EXP} \mid \text{BE;EXP} \mid \text{STRING; EXP} \mid \text{AE} \mid \text{BE} \mid \text{STRING}$

$\text{STRING} ::= \text{"TEMP"}$   
 $\text{TEMP} ::= I \text{ TEMP} \mid N \text{ TEMP} \mid I \mid N$

$\text{BE} ::= \text{SUB and BE} \mid \text{SUB or BE} \mid \text{not SUB} \mid \text{SUB}$

$\text{SUB} ::= \text{AE} == \text{AE} \mid \text{AE} > \text{AE} \mid \text{AE} < \text{AE} \mid \text{AE} >= \text{AE} \mid \text{AE} <= \text{AE} \mid \text{AE} != \text{AE} \mid \text{BOOL\_VAL}$

$\text{BOOL\_VAL} ::= \text{true} \mid \text{false}$

$\text{AE} ::= I := T \mid T$   
 $T ::= T + T_2 \mid T - T_2 \mid T_2$

$T_2 ::= T_2 * T_3 \mid T_2 / T_3 \mid T_3$

$T_3 ::= (\text{AE}) \mid I \mid N$

$I ::= [a, z]$

$N ::= [0, 9]$

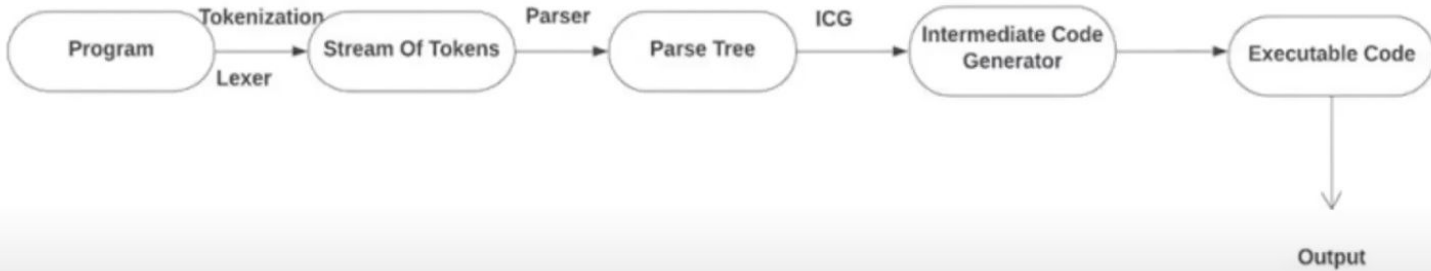


03

# LEXICAL ANALYSER



# FLOWCHART



# TOOLS

- DCG - Grammar
- SWI Prolog
- Pyswip - To connect python and prolog



# LEXICAL ANALYSER

```
from tokenize import tokenize, untokenize
from io import BytesIO
```

```
KEYWORDS = ["start", "finish", "int", "bool", "st", "if", "then", "else", "fi", "while", "begin", "end", "for", "in",
"range", "print", "and", "or", "true", "false", "not"]
OPERATORS = ["+", "-", "*", "/", "=", ">", "<", "!", "?", "."]
ARITHMETIC_ASSIGNMENT = ["==", "!=", "<=", ">="]
SEPARATORS = ["(", ")", " ", ","]
```

```
def get_tokens(file):
    ext = file.split('.')
    print(ext)
    if ext[1] != "daman":
        print("Unsupported file extension")
        return

    final_op = "["

    with open(file, 'r') as f:
        value = tokenize(BytesIO(f.read().encode('utf-8')).readline)

    for tokenNum, val, _, _, _ in value:
        if len(val) == 0:
            continue
        elif val == "\n" or val == " " or val == "\t":
            continue
```

```
        elif val in KEYWORDS or val in OPERATORS or val == ";":
            final_op += f'{{{val}}}', "
        elif val in SEPARATORS:
            final_op += f'{{{val}}}', "
        elif val.startswith("(") or val.startswith("("):
            temp = val[1:-1]
            final_op += f'{{{temp}}}', "
        elif val in ARITHMETIC_ASSIGNMENT:
            final_op += f'{{{val}}}', "
        elif val == ".":
            final_op += " ".
        elif val.isdigit():
            final_op += f'{{{val}}}', "
        elif val.isalpha():
            final_op += f'{{{val}}}', "
```

```
    final_op = final_op[:-1]
    final_op += "]"
    print(final_op)
    return final_op
```

```
if __name__ == "__main__":
    x = input("Enter the name of the file: ")
    get_tokens(x)
```



04

PARSER

# PARSER

```
program(t_p(Tb)) --> block(Tb),['.'].

block(t_b(Td,Tc)) --> ['start'],declaration(Td),[';'],command(Tc),[';'],['finish'].
block(t_b(Td)) --> ['start'],declaration(Td),[';'],['finish'].
block(t_b(Tc)) --> ['start'],command(Tc),[';'],['finish'].
declaration(t_ass_decl(A,D)) --> ass_variable(A),[';'],declaration(D).
declaration(t_decl_decl(A,D)) --> decl_variable(A),[';'],declaration(D).
declaration(t_ass_decl(A)) --> ass_variable(A).
declaration(t_decl_decl(A)) --> decl_variable(A).

% ASS_VARIABLE ::= int I = N | bool I = BOOL_VAL | st I = STRING
```

```
% BE ::= SUB and BE | SUB or BE | not BE | SUB
be(t_be_and(Sub,BE))--> sub(Sub),['and'],be(BE).
be(t_be_or(Sub,BE))--> sub(Sub),['or'],be(BE).
be(t_sub_not(Sub))--> ['not'],be(Sub).
be(Sub)--> sub(Sub).
```

```
% SUB ::= AE==AE | AE>AE | AE<AE | AE>= AE | AE<=AE | AE!= AE | BOOL_VAL
sub(t_sub_eq(AE1,AE2))--> ae(AE1),['=='],ae(AE2).
sub(t_sub_greaterthan(AE1,AE2))--> ae(AE1),['>'],ae(AE2).
sub(t_sub_lessthan(AE1,AE2))--> ae(AE1),['<'],ae(AE2).
sub(t_sub_gteqto(AE1,AE2))--> ae(AE1),['>='],ae(AE2).
sub(t_sub_lteqto(AE1,AE2))--> ae(AE1),['<='],ae(AE2).
```

# PARSER

```
% T ::= T + T2 | T - T2 | T2
t(t_term_plus(T,T2))--> t(T),['+'],t2(T2).
t(t_term_min(T,T2))--> t(T),['-'],t2(T2).
t(T2)--> t2(T2).

% T2 ::= T2 * T3 | T2 / T3 | T3
t2(t_t2_prod(T2,T3))--> t2(T2),['*'],t3(T3).
t2(t_t2_div(T2,T3))--> t2(T2),['/'],t3(T3).
t2(T3)--> t3(T3).

% T3 ::= (AE) | I | N
t3(t_t3_par(AE))--> ['(',')',ae(AE),['(',')']].
t3(t_identifier(ID))--> identifier(ID).
t3(t_num(Num))--> num(Num).
```

```
% TEMP ::= I TEMP | N TEMP | I | N
temp(t_temp(CH))--> identifier(CH).
temp(t_temp(Num))--> num(Num).
temp(t_temp(CH,Temp))--> identifier(CH),temp(Temp).
temp(t_temp(Num,Temp))--> num(Num),temp(Temp).
% atom_num(AN) --> atom(A), num(N), {atom_number(A, N), AN =.. [A, N]}.
```

```
% I ::= x | y | z | u | v
identifier(I) --> [I], {atom(I)}.
```

```
% N ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
num(N) --> [N], {integer(N)}.
```



05

EVALUATOR



06

DEMO



# DEMO

```
C:\Users\DELL\Desktop\Nirmit\ASU\Classes\SER 502\Projects\SER502-Spring2023-Team-41->python Token.py
Enter the name of the file data/condition.daman
['start', 'int', 'x','=', 8,;', 'if', 'x','==', 9,'then', 'x','=', 7,'else', 'x','=', 3,'fi', ';', 'print', '(', 'x',')', ';', 'finish', '.']
+ [31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:116:
Warning: Singleton variables: [I]
+ [0m+ [31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:118:
Warning: Singleton variables: [H]
+ [0m+ [31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:128:
Warning: Singleton variables: [EnvOut]
+ [0m
3
Program Compiled Successfully
```

```
C:\Users\DELL\Desktop\Nirmit\ASU\Classes\SER 502\Projects\SER502-Spring2023-Team-41->python Token.py
Enter the name of the file data/factorial.daman
['start', 'int', 'x','=', 5,;', 'int', 'a','=', 1,;', 'while', 'x','>', 1,'begin', 'a','=', 'a','*', 'x',;', 'x','=', 'x','-1', 'end', ';', 'print', '(', 'a',')', ';', 'finish', '.']
+ [31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:116:
Warning: Singleton variables: [I]
+ [0m+ [31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:118:
Warning: Singleton variables: [H]
+ [0m+ [31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:128:
Warning: Singleton variables: [EnvOut]
+ [0m

120
Program Compiled Successfully
```

# DEMO

```
C:\Users\DELL\Desktop\Nirmit\ASU\Classes\SER 502\Projects\SER502-Spring2023-Team-41->python Token.py
Enter the name of the file data/tern.daman
['start', 'int', 'x','=', 3,';', 'x','<', 3,'?', 'print', '(', 'x',')', ':', 'print', '(', 'x','+', 2,')', ';', 'finish', '.']
←[31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:116:
Warning: Singleton variables: [I]
←[0m←[31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:118:
Warning: Singleton variables: [H]
←[0m←[31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:128:
Warning: Singleton variables: [EnvOut]
←[0m
5
Program Compiled Successfully
```

```
C:\Users\DELL\Desktop\Nirmit\ASU\Classes\SER 502\Projects\SER502-Spring2023-Team-41->python Token.py
Enter the name of the file data/sums.daman
['start', 'int', 'ans','=', 0,';', 'for', 'i','in', 'range', '(', 1,',', 10,')', 'begin', 'ans','=', 'ans','+', 'i','end', ';', 'print', '(', 'ans',')', ';', 'finish', '.']
←[31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:116:
Warning: Singleton variables: [I]
←[0m←[31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:118:
Warning: Singleton variables: [H]
←[0m←[31mWarning: c:/users/dell/desktop/nirmit/asu/classes/ser 502/projects/ser502-spring2023-team-41-/src/compiler.pl:128:
Warning: Singleton variables: [EnvOut]
←[0m
```

```
45
Program Compiled Successfully
```



Thank you!