Emerging Languages and Programming Paradigm Project: Group - 41

Language Name: Daman Why Daman ?

First Initial of each team member name i.e

D - Drashti
A - Alok
M - Manasi
A - Akshat
N - Nirmit

Interpreter : SWI - Prolog

Why SWI - Prolog ?

As prolog is a declarative language, it allows for concise and elegant specification of a language behavior, making it easier to develop interpreters for other programming languages as well as its pattern matching and backtracking capabilities, it is well-suited for defining and implementing complex language constructs with ease.

Tools: SWI Prolog

Parsing Technique: Definite Clause Grammar (DCG)

Why DCG ?

It allows for the definition of context-free grammars in a concise and readable manner which helps understand and maintain the code. It also helps generate parse trees which can be used by an interpreter or compiler to execute the code.

Design:

1. **Primitive Data Types Used:**

- Integer (int)

- boolean (bool)

- String (st)

2. **Operators Used:**

   a. Assignment (=)

   b. Addition (+)

   c. Subtraction (-)

   d. Multiplication (*)

   e. Division (/)

   f. Greater Than (>)

   g. Less Than (<)

   h. Less Than Equal To (<=)

   i. Greater Than Equal To (>=)

   j. Comparison (==)

k. Ternary (?:)

l. AND(and)

m. OR(or)

n. NOT(not)

o. NOT EQUAL(!=)

3. **Control Statements**

   a. If-then-else

   b. Traditional While

   c. Traditional For

   d. For i in range

4. **Print Display**
   a. Print()

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
CH → Character
SUB_VAR → Identifier or Integer

Keywords: start,finish, int, bool, st, if, then, else, fi, while, begin,end,for, in range, print, not

P::= K .
K ::= start DECL; CMD finish

DECL ::= ASS_VARIABLE ; DECL | DECL_VARIABLE ; DECL | ASS_VARIABLE | DECL_VARIABLE

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

DECL_VARIABLE ::= int I | bool I | st I

CMD ::= NC; CMD | NC

NC ::= I = AE | if BE then CMD else CMD fi | while BE begin CMD end | for(int I = AE; BE ; AE) begin CMD end | for I in range(N,N) begin CMD end | BE? CMD : CMD | print (EXP) | K

EXP ::= AE;EXP | BE;EXP | STRING; EXP | AE | BE | STRING

STRING ::= "TEMP"
TEMP ::= I TEMP | N TEMP | I | N

BE ::= SUB and BE | SUB or BE | not SUB | SUB

SUB ::= AE==AE | AE>AE | AE<AE | AE>= AE | AE<=AE | AE!= AE | BOOL_VAL

BOOL_VAL ::= true|false

AE ::= I:=T|T
T::=T + T2 | T – T2 | T2

T2::= T2 * T3 | T2 / T3 | T3

T3 ::= (AE)| I |N

I ::= [a,z]

N := [0,9]