SER 502 – Team 11 | Spring19

Team Members:

Bharat Goel
Madhukar Raj
Palak Chugh
Yuti Desai

Language Design

- Name of the Language– BUMPY
- Operators and Constructs:

- Operators: +,-,*,/,%,<,>,<=,>=, ~=, :=:, = ,and, or,
- Arithmetic Operator: +,-,*,/,%

  Addition : +

  Subtraction: -

  Multiplication: *

  Division: /

  Mod: %
- Assignment Operator: =
- Comparison Operator: <,>,<=,>=, ~=, :=:

  Less than: <

  Greater than: >

  Less than equal to: <=

  Greater than equal to: >=

  Not equals to: ~=

  Equals to: :=:

- Boolean Operator : and, or, not
- Primitive types: bool, var

  Bool : takes Boolean value

  Var: takes integer value
- Decision Constructs: incase do otherwise endcase

  Incase (condition) do (process) otherwise (process) endcase
- Iterative Constructs: when repeat endrepeat

  When (condition) repeat (process) endrepeat

Grammar

- Parser → Program
- Program → Comment Block | Block
- Comment → @ Words @
- Block → start Declaration Process stop
- Words → Identifier Words | Numb Words| Identifier| Numb.
- Declaration → Datatype Identifier ; Declaration | Datatype Identifier ;
- Process → Assignvalue ; Process | Control Process | Iterate Process | Print Process |ReadValue Process | Assignvalue ; | Control | Iterate | Print | ReadValue
- Datatype → var | bool
- Assignvalue → Identifier = Expression| Identifier is Boolexp
- Control → incase Condition do Process otherwise Process endcase
- Iterate → when Condition repeat Process endrepeat
- Print → show Expression ; | show * value * ;
- ReadValue → input Identifier ;
- Condition → Boolexp and Boolexp| Boolexp or Boolexp | ~ Boolexp| Boolexp

- Boolexp → Expression :=: Expression |Expression ~= Expression| Expression <= Expression| Expression >= Expression|Expression < Expression| Expression > Expression |Expression :=: Boolexp|Expression ~= Boolexp| yes | no
- Expression → Term + Expression |Term – Expression | Term
- Term → Identifier * Term| Numb * Term| Numbneg * Term | Identifier / Term|Numb /Term|Numbneg /Term | Identifier mod Term| Numb mod Term|Numbneg mod Term| Identifier|Numb|Numbneg
- Identifier → _[^a-z]alphanumeric | [^a-z]alphanumeric
- Numb → number
- Numbneg → - Numb

Features

- Parsing technique: We are using Top- down parsing technique, our parser constructs the parse tree from the start and then tries to convert it the start symbol into input.

- Data structures used by the parser and interpreter: List

- Interpreter: Our interpreter is based on Reduction machine.

- Programing language used for implementation: Prolog