Tecnológico de Monterrey

Project Proposal

Francisco Salgado A01365047

April 23, 2023

Diseño de compiladores

Elda G. Quiroga, M.C.

Dr. Héctor Ceballos, PhD

Language main objective

The language's main objective will be to provide the tools for doing meaningful statistical analysis. It will have many of the objects and classes that other similar languages have with also the capacity to generate statistical models. Dataframes can be created and special functions are already implemented in the language to work with the dataframes.

Language requirements

Basic Elements

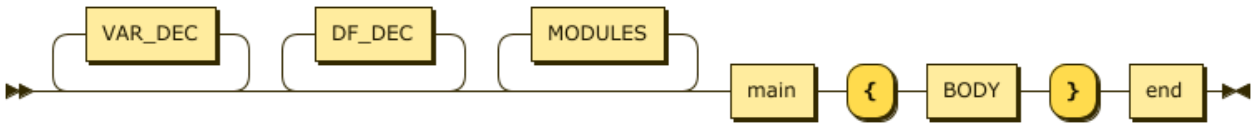
Token name	Description	Regular Expression
program	symbol that indicates the program and execution	\bprogram\b
id	indicates that the name of the id cannot be declared again	[a-zA-Z_][a-zA-Z0-9_]*
var	indicates the start of the variable declaration process	\bvar\b
null	represents a null value	\bnull\b
func	represents the declaration of a function	\bfunc\b
void	a type of function that doesn't return any value	\bvoid\b
int	indicates data is of type integer	\d+
double	indicates data is of type double	[+-]?([0-9]*[.])?[0-9]+
char	indicates data that consists of one character	[^"]? | [^']?
string	indicates data that consists of multiple characters	[^"]* | [^']*
bool	indicates data is of type boolean	(\btrue\b | \bfalse\b) | (1 | 0)
array	indicates an array data	\barray\b

	structure of two dimensions at max	
dataframe	indicates a data structure that contains a series of vectors of the same type	\bdataframe\b
if	reserved word for conditions	\bif\b
else	reserved word for conditions	\belse\b
while	reserved word for cycles	\bwhile\b
for	reserved word for cycle	\bfor\b
read	reserved word for reading input from the user	\bread\b
write	reserved word for writing the value of a variable or expression	\bwrite\b
[ ]	symbols to declare and access the elements of an array	'[' ']'
( )	symbols to declare and indicate the parameters of a function	'(' ')'
{ }	symbols that indicate a block of code	'{' '}'
;	symbol that indicate the end of a statement	';'
,	symbol that separates statements or id’s	','
+	plus symbol for arithmetic expressions	'+'
-	minus symbol for arithmetic expressions	'-'
*	times symbol for arithmetic expressions	'*'
=	equal symbol for assigning value of a variable	'='

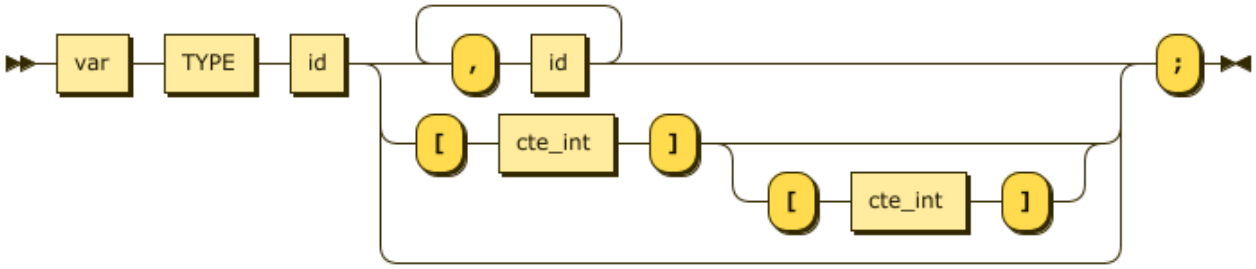
<	less than symbol for logical evaluations	'<'
>	more than symbol for logical evaluations	'>'
<=	less than or equal than symbol for logical evaluations	'<='
>=	more than or equal than symbol for logical evaluations	'>='
==	symbol to use on statements that evaluates whether two data are the same	'=='
!=	symbol to use on statements that evaluates whether two data are different	'!='
&&	symbol that indicates an AND logic gate for the evaluation	'&&'
| |	symbol that indicates an OR logic gate for the evaluation	'| |'

Syntax Diagrams

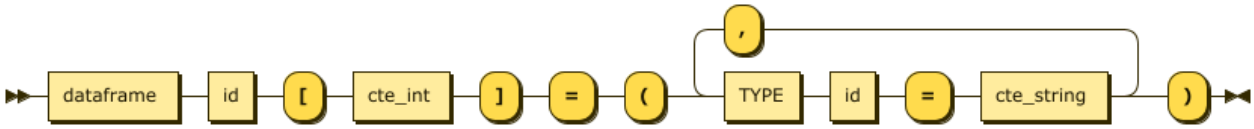
PROGRAM



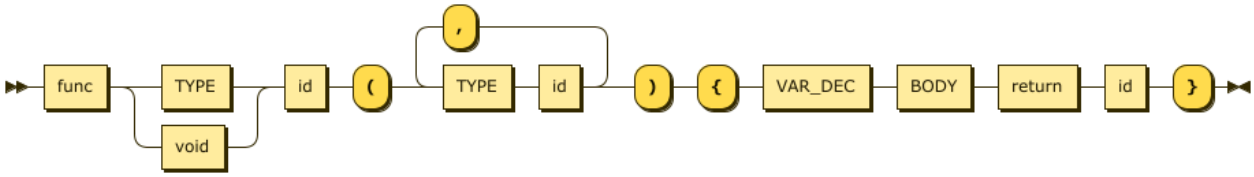
VAR_DEC



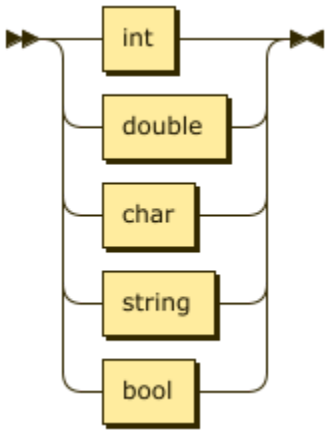
DF_DEC



MODULE

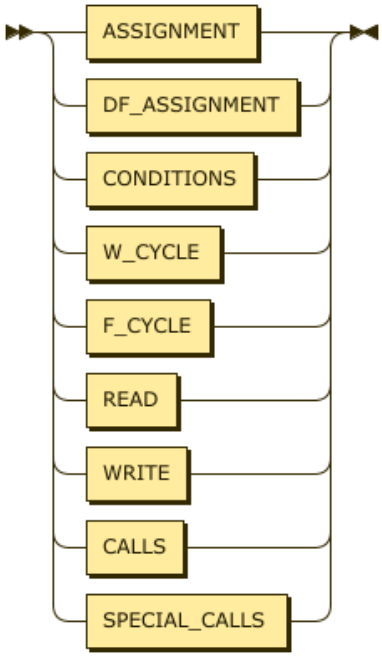


TYPE

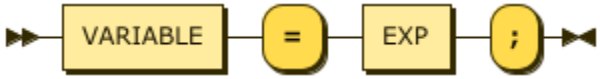


BODY

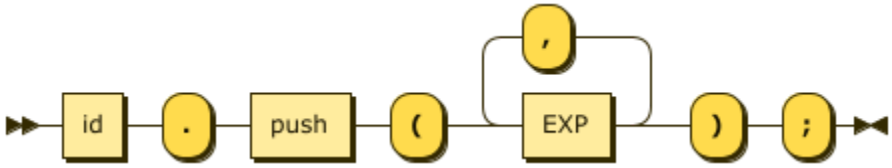
STATEMENT



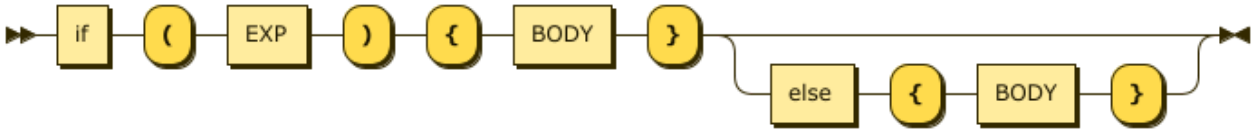
ASSIGNMENT



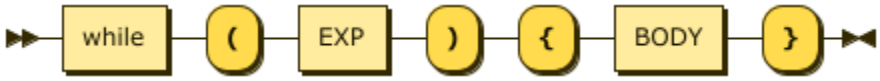
DF_ASSIGNMENT



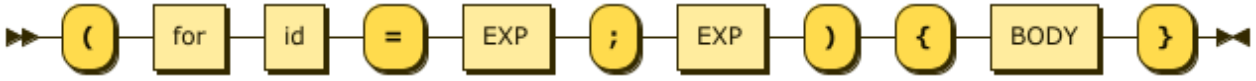
CONDITION



W_CYCLE

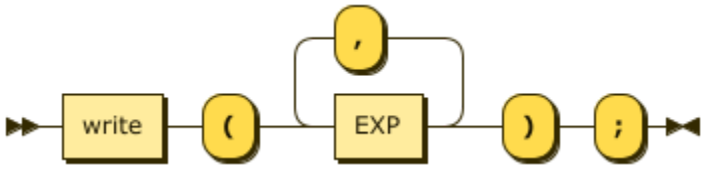


F_CYCLE

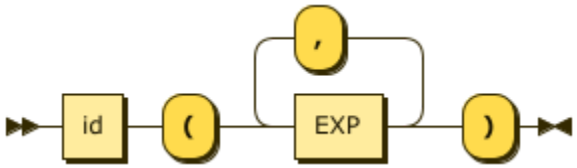


READ

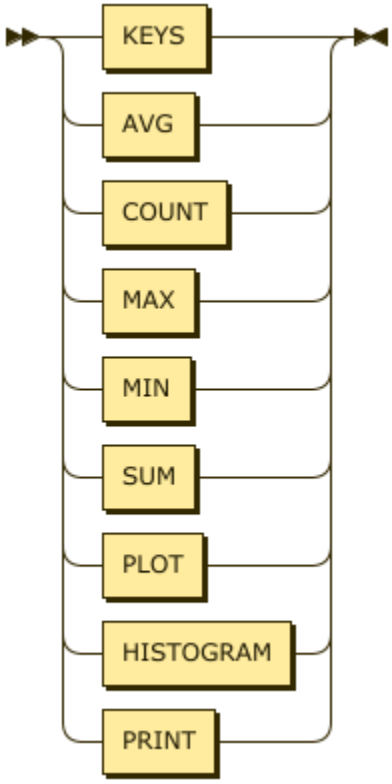
WRITE



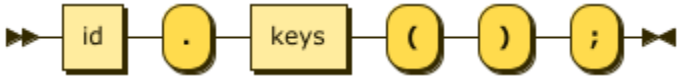
CALLS



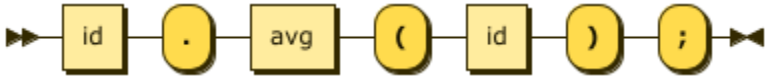
SPECIAL_CALLS



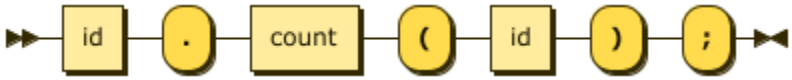
KEYS



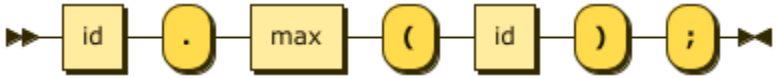
AVG



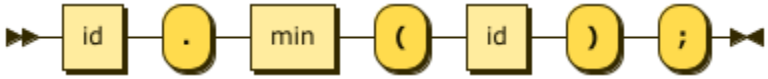
COUNT



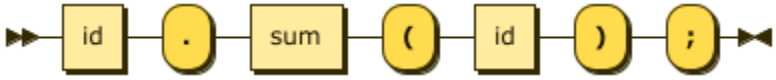
MAX



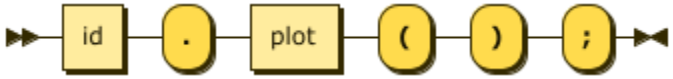
MIN



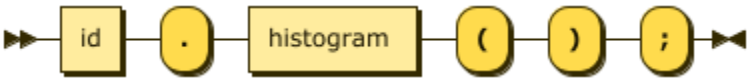
SUM



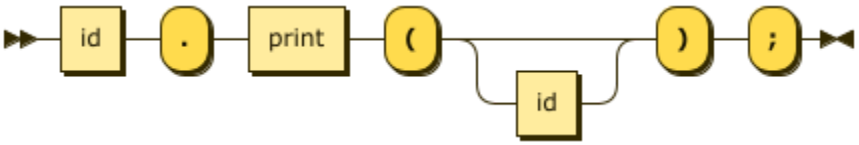
PLOT



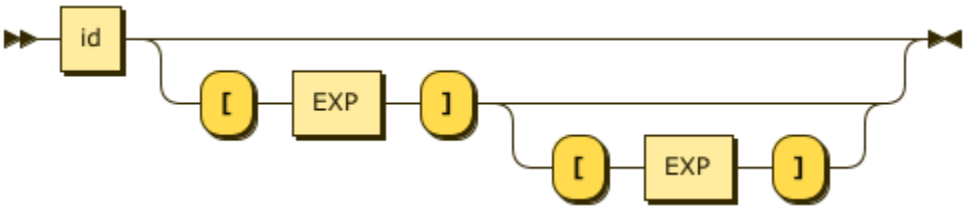
HISTOGRAM



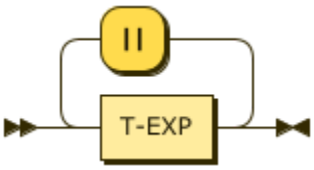
PRINT



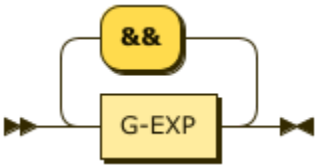
VARIABLE



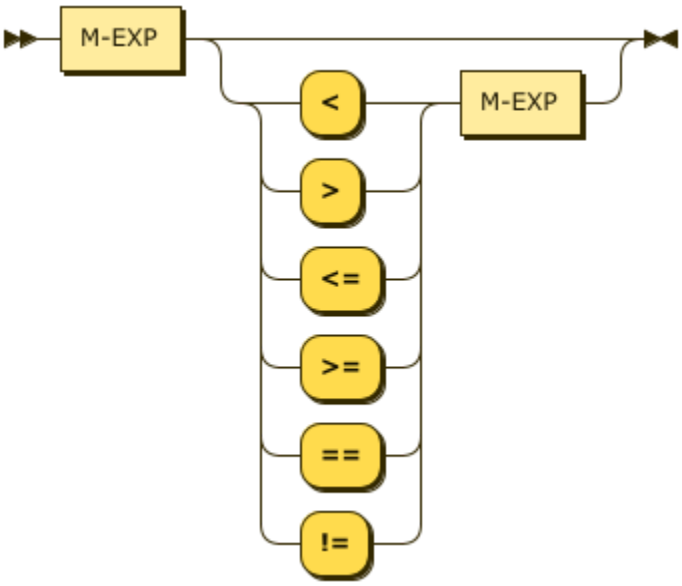
EXP



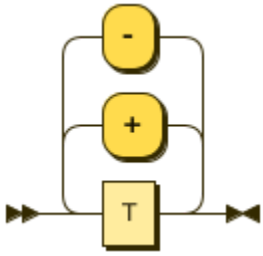
T-EXP



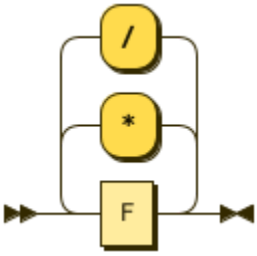
G-EXP



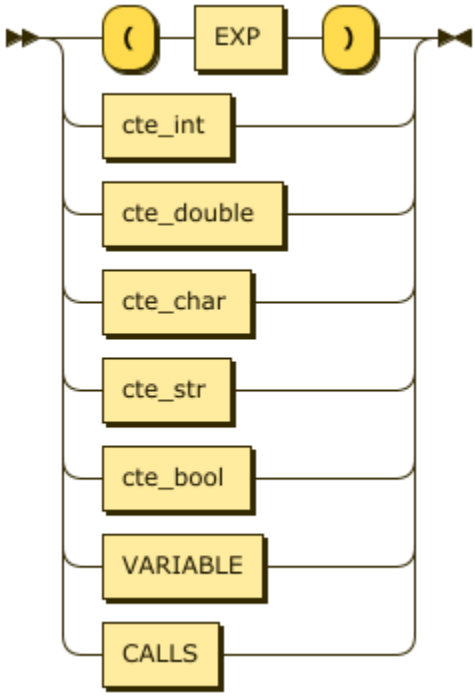
M-EXP


-
+
T

T


/
*
F

F


(
EXP
)
cte_int
cte_double
cte_char
cte_str
cte_bool
VARIABLE
CALLS

Semantic rules

The rules with which the program will base its semantic analysis can be summed up as follows:

1. When evaluating an expression, the variables that will be evaluated have to be the same type. This rule also applies to assignments in which the result of the assignment has to be the same as the variable that is going to be assigned. “int with int, double with double, etc.”
2. Variables, functions and data frames have to be declared before they can be worked with. The number of parameters for the functions has to be the same as when they are declared and functions are defined before the main flow of the program.
3. Functions that are not of the void type must return a value of the same type as they were declared before the main flow of the program.
4. Dataframes have to be declared with their columns defined by ids which are also defined by the user, as well as the type of data that the column will contain.
5. Rows can be added to dataframes with a push function which need to have as many parameters as there are columns on the data frame. These parameters are expressions, similar to those of functions and these parameters’ types must match those of the columns in the dataframe (no inserting integers in a column of doubles).

Special functions

Dataframes are a special type of data type that has functions specified by default in the language. These special functions are:

1. Keys: This function will return a list of strings that represent the name of the columns in the dataframe.
   dataframeid.keys();
2. Average: This function will return the average of a list of numbers contained in a column of the data frame, whether these numbers are integers or doubles. The result will be returned as a double.
   dataframeid.average(columnid);
3. Count: This function returns the number of rows or elements of a dataframe.
   dataframeid.count(columnid);
4. Max: This function returns the number which is the maximum value in a column in the data frame, whether this number is an integer or a double.
   dataframeid.max(columnid);
5. Min: This function returns the number which is the minimum value in a column in the data frame, whether this number is an integer or a double.
   dataframeid.min(columnid);
6. Sum: This function returns the sum of all numbers in a column in the data frame as the type specified by the column.
   dataframeid.sum(columnid);
7. Plot: This function creates an image of a plot diagram of the data contained in the data frame and saves it.
   dataframeid.plot();

8. Histogram: This function creates an image of a histogram of the data contained in the data frame and saves it.
   dataframeid.histogram();
9. Print: This function prints the data frame in an orderly manner in the console.
   dataframeid.histogram();