PAPER ON BOOLEAN EXPRESSION IN DIFFERENT PROGRAMMING LANGUAGES

Jeevan Chandrashekar
University Of Alabama In Hunstville
Huntsville,AL
jcc0034@uah.edu

I. ABSTRACT

This paper talks about different programming language paradigm features. Paper mainly focuses on contrasting and comparing and the four different programming languages based on the File Handling features. The programming languages which are discussed in this paper are from different paradigm which includes imperative (Python), object-oriented (Java), functional (Lisp) and logical (Prolog).

This paper discusses about a common feature called Boolean expressions used in different programming languages example: Java, Python, Prolog, Lisp. The paper does this by considering some basic concepts like Syntax, Semantics, Readability, writability, reliability, its Implementation and its Evolution in different programming languages and how they are similar or different. In addition it describes about the merits and demerits of the way they are using in different languages. Finally, it concludes with author opinion on each language by the way it uses this feature.

Keywords
Boolean expressions; Programming language; Syntax; Semantics; Readability; writeability; reliability;

II. DEFINITIONS

There are some important aspects used in this paper before understanding what' s written in the paper first we must try to understand few basic definitions.

Imperative Paradigm: It is the most dominant paradigm of all the others. An imperative language is said to be Turing Complete if it provides an effective basis for implementing any algorithm that can be designed. It is closest to the structure of actual computers.

Object-Oriented Paradigm: It is type of programming in which programmers define not only the data type of a data structure, but also the types of operations that can be applied to the data structure. The object-oriented paradigm shifts the focus of attention from code to data.

Functional Paradigm: A functional programming language is a language with higher order functions which enables programming in a functional style with higher order functions.

Logical Paradigm : A Logical programming language is a language written in the form of logic based on mathematical features.

III. INTRODUCTION

This paper talks about a common feature, what is consider as common feature? Feature which is defined in most of the well-recognized programing language. There are lots of common features since it is difficult to brief all, it chose Boolean expressions. It validate the common feature in terms of its Syntax, Semantics, Readability, writeability, reliability, its Implementation and its Evolution in some popular programming languages Java, Python, Prolog, Lisp. In order to understand this paper, it has a definition section in which definition of technical terms used in this paper are defined.

A. Definition section

- Boolean expression:

  "A Boolean expression is a logical statement that is either TRUE or FALSE. Boolean expressions can compare data of any type as long as both parts of the expression have the same basic data type. You can test data to see if it is equal to, greater than, or less than other data."[1]

"A Boolean expression consists of Boolean data:

BOOLEAN values (YES and NO, and their synonyms, ON and OFF, TRUE and FALSE)

BOOLEAN variables or formulas

Functions that yield BOOLEAN results

BOOLEAN values calculated by comparing the operators"

- Programming language:

  "A programming language is a computer programming language used to create a standard form of commands, the commands which are created are interpreted into a code which is understood by a machine. Programs are created through programming languages to control the output which are similar to the human communication process."[2]

- Syntax:

  "The syntax of a programming language describes all its grammatically correct programs."[3]
  "Definition: A language's lexical syntax defines the rules for basic symbols including literals, operators, identifiers, and punctuation."
  "Definition: A language's concrete syntax refers to the actual representation of its programs using lexical symbols as its alphabet."
  "Definition: A abstract syntax language carries the essential program data, without affecting syntactic idiosyncrasies parentheses."

- Semantics:

  "Semantics of a programming language is a precise meaning of various elements in the program that is syntactically and type-wise correct."

- Readability:

  "The ease with which programs can be read and understood by other readers"

- Writability:

  "The programs can be drawn and built for a particular problem domain."[4]

- Reliability:

  "The programs in which the degrees are written in the language perform according to their identification under all conditions"

- Relational Operators:

  "The relational operators compares values to each other. The comparison operators are <, >, <=, =, <>, and >=. All of the relational operators result in a Boolean value if it satisfies the condition."[5]

- Conditional Operators:

  "Conditional operators are used to examine a condition that's used for one or two boolean expressions. The result of the evaluation will be either true or false.

  There are three conditional operators: && the logical AND operator. || the logical OR operator. ?: the ternary operator."[6]

- Boolean variable:

  "Holds True or False values. The True and False value relate to the two states of Boolean variables."[7]

- Abstract syntax:

  "A abstract syntax language's carries the data which is essential to the program , without concern for syntactic idiosyncrasies like parentheses, punctuation, or word order"[7]

- Short Circuit Boolean evaluation:

  "Short-circuit evaluation examines a Boolean expression from left to right and stops as soon as the value of the expression can be determined as true."

- Membership Operator:

  "If the variable sequence specified is found it evaluates to True or else specifies false. Evaluates to False if it does not find a variable in a sequence, True otherwise."[8]

- Identity operators:

  “Evaluates to True if the variables of the operator point to the same object on either side and Evaluates False if its not evaluated to the same object. If the variables on either side of the operator point to the same object and True otherwise”.

- And Operation

  “If both the values are true the result will be true or else result will be false.”[9]

- OR Operations:

  “If One of the value is true, then the result is true. Or else result will be false”[9]

- Not Operations:

  “The NOT operation is a unary operation, which means that it works on only one operand. The true or false value are reversed of its operand.”[9]

IV. Brief Description

This section provides the overview of basic concepts like Syntax, Semantics, Readability, writeability, reliability, its Implementation and its Evolution in different programming languages.

A. Basic concepts

- Syntax:
  Abstract syntax of Boolean expression

  ```
  boolean_var=exp relation_opt/condition_opt exp
  ```

  syntax used in different programming languages

  - Java

    ```
    boolean_var=exp relation_opt/condition_opt exp;
    ```

  - Python

    ```
    boolean_var=exp relation_opt/condition_opt  exp
    ```

  - Prolog

    ```
    boolean_var= relation_opt/condition_opt(exp,exp)
    ```

  - Lisp

    ```
    Boolean_var=exp relation_opt/condition_opt exp
    ```

By considering the syntax in different languages, it can be observed that there is no major difference. It almost resembles the abstract syntax.

- Semantics:

  Consider semantics of Boolean expression in different programming languages, what output will syntactically and type-wise correct statement yield. The main tasks of boolean expression is to either applying relational operator for two variable of same type or expression which results same type value and gives the output either true or false. Or by applying logical operator between two boolean variable and return its boolean value.

- Readability and Writability:

  To discuss readability or writeability of programming language , lets us consider some complex examples for boolean expressions in different programming language.
  Let’s assume u, v, w, x, y, z are variables of int data type. Boolean expression involving these variables in different programming language,

  - Java:

    ```
    If((u>=v)&&(v<=w)||(u<=w)&&(y!=z)||
    (y!=x)){
    //do something…..
    }
    ```

  - Python:

    ```
    If u>=v and v<=w or u<=w and y!=z or
    y!=x
    //do something
    ```

  - Prolog:

    ```
    u>=v, (v<=w ; u<=w ),%do something.
    ```

  - Lisp:

    ```
    If u>=v AndAlso v<=w OrElse u<=w
    AndAlso y!=z OrElse y!=x Then
    //do something
    End If
    ```

- Implementation:

  Boolean expressions results with Boolean data, different implementation methods used to store and retrieve this in different programming languages.

  - Java:

    Boolean values are of boolean data type. Boolean type variables have two pre-defined possible values True and False, it uses 1 byte or 2 byte of memory space to store the value. It stores all bits by value 1 to represent true and 0 to represent false.

  - Python:

    "In Python, we have two Boolean literals - True and False. But Python will consider as False - the reserved word as value zero or None or an empty string. And all other values are considered as True."[10]

  - Prolog:

    "Boolean type integer variable might have two pre-defined possible values one is True and other is False.

    Free Prolog also supports the ByteBool, WordBool and LongBool types. These are of type Byte, Word or Longint, respectively.

    The nonzero value is considered True and value False is equivalent to 0. when converting to a Boolean value. A Boolean value of True is converted to -1 in case it is assigned to a variable of type LongBool."[11]

  - Lisp:

    Lisp consumes 2 bytes of memory to store the value of boolean variable like in java.

B. *Comparison*

- Readability and Writability

  By considering Readability and Writability sub-section of Basic concepts in Brief description Section, it can be stated that readability of python and prolog are almost same whereas Lisp has highest and prolog has lowest readability. In terms of Writiability prolog has most followed by java, python and prolog, least in Lisp.

- Short Circuit Boolean evaluation support

  All the programming language considered in paper support short circuit boolean evaluation, whereas Lisp has separate operator especially for short circuit boolean evaluation i.e. AndAlso, OrElse. If And, OR operator used in Lisp then it will not exhibit short circuit boolean evaluation.

- Operators in different languages

  - Python supports Membership and Identity operators in boolean expression. Whereas java, prolog, Lisp doesn't support.

  - The comparison operator to check equality has different notation in different programming languages, in java and python it is used as "==" whereas in prolog and Lisp "=" whereas other relational operators are same.

  - Condition operators in different programming languages.

Java	Python	prolog	Lisp
&&	and	,	AndAlso
||	or	;	OrElse
?:	a if test else b		
!	Not	not	Not

V. CONCLUSION

This section will brief about which programming language out of the above considered 5 languages has used the Boolean expression in best way in author view, i.e. it has simple syntax, common semantics, better readability and writeability, complete reliability, cost efficient implementation.

From the considered boolean expression example in different programming language Writeability of Python is better than other considered languages. Readability is best in Java.

There is very little difference in the syntax of these languages, semantics of these languages are same.

VI. BIBLIOGRAPHY

[1]https://docs.oracle.com. Oracle® OLAP DML Reference 10g Release 1 (10.1)
Part Number B10339-02.

[2]https://www.techopedia.com. Technopedia, August 24 2015

[3] Noonan Tucker, “Programming_Languages_Principles_and_Paradigms” 2nd_Edition

[4] https://msdn.microsoft.com. Microsoft Virtual Academy
Section 11.14 Relational Operators, October 2013.

[5]http://java.about.com. By Paul Leahy.

[6]http://faculty.ccp.edu. By Cher Bert, chapter 05 Boolean Logic in Programming.

[7] https://msdn.microsoft.com. Microsoft Virtual Academy
Section Boolean Data types, October 2013 .

[8]http://www.thomascokelaer.info.
© Copyright Thomas Cokelaer.
Last updated on Aug 14, 2014.
Created using Sphinx 1.2.2.

[9] http://www.cs.utexas.edu. By Mitra.

[10]http://www.tutorialspoint.com. Tutorials Point, By Omer August 17 2015.

[11] www.codecademy.com. Code Academy