International Journal of Mathematical Sciences and Applications
Vol. 1 No. 2 (May, 2011)

www.ijmsa.yolasite.com

Convert Comparison Expression to Arithmetic Expression

Pramod S. Jadhav,
Student of B.P.H.E Society's IMSCD&R,
Ahmednagar, Maharashtra, India
jadhav.009@gmail.com

Abstract: *In day to day Life there are many problems bases on logical statement such as (If - Then). We know that arithmetic Expression gives exact result of problem. In another world we solve many problems in day to day life using arithmetic expression, but solving Comparison expressions we use programming language such as c, java, or some language which are work for development Artificer Intelligent devices In the low leval language it is very difficult job to evaluate comparison expression as camper to arithmetic Expression. In this paper I give some arithmetic expression instead of some comparison expression. With the help of this Arithmetic expression we solve some conditional expression in mathematics also with the help of this we use direct arithmetic expression in low level language.*

Keywords: Axioms for Real Number System, Floor and ceiling functions, Relation operators, Mathematical Expression

.

1. INTRODUCTION

In Mathematic world it is difficult to convert to logical expression in to the arithmetic Expression. In low level language it is very difficult job to write conditional statement if we use arithmetic expression directly instated of comparison expression it is easily we write code for arithmetic expression. With the help of axioms for real number and Floor ceiling function I create equivalent arithmetic Expression for the comparison expression.

2. AXIOMS FOR REAL NUMBER SYSTEM

The Real Number System Consist of Four Parts

1. A set (R). We will call the elements of this set real numbers.
2. A relation < on R. This is the order relation.
3. A function + : R × R ! R. This is the addition operation.
4. A function · : R × R ! R. This is the multiplication operation.

We will state 12 axioms that describe how the real number system behaves. The first eleven will say that the real number system forms an ordered field.
The final axiom will require a little discussion.

2.1 OPERATION AXIOMS

For all x, y, and z

1. Associative laws:
8x8y8z [(x + y) + z = x + (y + z) and (x · y) · z = x · (y · z)]
2. Commutative laws:
8x8y [x + y = y + x and x · y = y · x]
3. Distributive law:
8x8y8z [x · (y + z) = x · y + x · z]

2.2 IDENTITY AND INVERSE AXIOMS

4. Additive identity:
There is an element (called 0) such that 8x [0 + x = x].
[Uniqueness can be proved.]
5. Additive inverse:
8x9y [x + y = 0]. [We write y = −x; uniqueness can be proved.]
6. Multiplicative identity:
There is an element (called 1) such that 0 6= 1 and 8x 1 · x = x.
[Uniqueness can be proved.]
7. Multiplicative inverse:
8x [x 6= 0 =) 9y (x · y = 1)]. [We write y = 1 x.
Uniqueness can be proved.

2.3 ORDER AXIOMS

8. Translation invariance of order:
8x8y [x < y =) x + z < y + z].
9. Transitivity of order:
8x8y [(x < y and y < z) =) x < z].
10. Tracheotomy:
8x8y exactly one of the following is true: x < y, y < x, or x = y.
11. Scaling and order:
8x8y8z [(x < y and z > 0) =) xy < yz]
Any number system that satisfies Axioms 1–11 is called an ordered field.
Examples: Q and R are both ordered fields.

2.4 THE COMPLETENESS AXIOM

12. Every non-empty subset that is bounded above has a least upper bound.

3. FLOOR AND CEILING FUNCTIONS

Gauss introduced the square bracket notation [*x*] for the floor function in his third proof of quadratic reciprocity (1808). This remained the standard in mathematics until Iverson introduced the names "floor" and "ceiling" and the corresponding notations $\lfloor x \rfloor$ and $\lceil x \rceil$ in his 1962 book *A Programming Language*. Both notations are now used in mathematics; this article follows Iverson.

The floor function is also called the greatest integer or entire (French for "integer") function, and its value at *x* is called the integral part or integer part of *x*; for negative values of *x* the latter terms are sometimes instead taken to be the value of the *ceiling* function, i.e., the value of *x* rounded to an integer towards 0.

The language APL (programming language) uses `⌊x`; other computer languages commonly use notations like `entier(x)` (Algol), `INT(x)` (BASIC), or `floor(x)` (C and C++). In mathematics, it can also be written with boldface or double brackets. $[\![x]\!]$

The ceiling function is usually denoted by `ceil(x)` or `ceiling(x)` in non-APL computer languages that have a notation for this function. In mathematics, there is another notation with reversed boldface or double brackets $]\!]x[\![$ or just using normal reversed brackets]*x*[.

The fractional part saw tooth function, denoted by $\{x\}$ for real *x*, is defined by the formula

$$\{x\} = x - \lfloor x \rfloor .$$

For all *x*,

$$0 \leq \{x\} < 1.$$

Sample value	**Floor** $\lfloor\ \rfloor$	**Ceiling** $\lceil\ \rceil$	**Fractional part** $\{\ \}$
12/5 = 2.4	2	3	2/5 = 0.4
2.7	2	3	0.7
−2.7	−3	−2	0.3
−2	−2	−2	0

Table 3.1

3.1 DEFINITION AND PROPERTIES

In the following formulas, x and y are real numbers, k, m, and n are integers, and $\mathbb{Z}$ is the set of integers (positive, negative, and zero).

Floor and ceiling may be defined by the set equations

$$\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \le x\},$$

$$\lceil x \rceil = \min\{n \in \mathbb{Z} \mid n \ge x\}.$$

Since there is exactly one integer in a half-open interval of length one, for any real x there are unique integers m and n satisfying

$$x - 1 < m \le x \le n < x + 1.$$

Then $\lfloor x \rfloor = m$ and $\lceil x \rceil = n$ may also be taken as the definition of floor and ceiling.

4. RELATIONAL OPERATORS

In computer science a relational operator is a programming language construct or operator that tests some kind of relation between two entities. These include numerical equality (e.g., 5 = 5) and inequalities (e.g., 4 ≥ 3). In programming languages that include a distinct Boolean data type in their type system, like Java, these operators return true or false, depending on whether the conditional relationship between the two operands holds or not. In other languages such as C, relational operators return the integers 0 or 1.

An expression created using a relational operator forms what is known as a *relational expression* or a *condition*. Relational operators are also used in technical literature instead of words. Relational operators are usually written in infix notation, if supported by the programming language, which means that they appear between their operands (the two expressions being related). For example, an expression in C will print the message if the x is less than y:

5. MATHEMATICAL EXPRESSION

In mathematics, an expression is a finite combination of symbols that are well-formed according to the rules applicable in the context at hand. Symbols can designate values (constants), variables, operations, relations, or can constitute punctuation or other syntactic entities. The use of expressions can range from simple arithmetic operations like

$$3 + 5 \times \left((-2)^7 - \frac{3}{2} \right)$$

to more complicated constructs that can include variables, functions, factorials, summations, derivatives and integrals, like

$$f(a) + \sum_{k=1}^{n} \frac{1}{k!} \frac{d^k}{dt^k} \bigg|_{t=0} f(u(t)) + \int_0^1 \frac{(1-t)^n}{n!} \frac{d^{n+1}}{dt^{n+1}} f(u(t))\, dt.$$

However a construction that violates the syntactic rules like

$$\times 4)x+,/y[\int \partial$$

is not well-formed, and therefore not an expression.

In algebra an expression may be used to designate a value, which value might depend on values assigned to variables occurring in the expression; the determination of this value depends on the semantics attached to the symbols of the expression. These semantic rules may declare that certain expressions do not designate any value; such expressions are said to have an undefined value, but they are well-formed expressions nonetheless. In general the meaning of expressions is not limited to designating values; for instance, an expression might designate a condition, or an equation that is to be solved, or it can be viewed as an object in its own right that can be manipulated according to certain rules. Certain expressions that designate a value simultaneously express a condition that is assumed to hold, for instance those involving the operator to designate an internal direct sum.

Being an expression is a syntactic concept; although different mathematical fields have different notions of valid expressions, the values associated to variables do not play a role. See formal language for general considerations on how expressions are constructed, and formal semantics for questions concerning attaching meaning (values) to expressions

6. METADOLOGY TO CONVERT COMPARISON EXPRESSION TO ARITHMETIC EXPRESSION

In Comparison Expression we define such as such as
Comparison operators Set is C= {<, >, >=, <=} **α** be any element in C and a, b are Real Number then Comparison Expression we write as follows

(a **α** b)
IF True then do Condition 1
IF False then do Condition 2

Here Condition 1 and Condition 2 are Statement or Arithmetic Expression.
For Conversion of Comparison Expression in Arithmetic Expression the Comparison expression satisfied following conditions

1. a, b are Natural Number
2. Condition1 and Condition 2 are Valid Number or Condition 1 and Condition 2 are Arithmetic Expressions which give valid result in number.

If These 2 Condition are satisfied by conditional expression then following way we convert conditional expression in to Arithmetic expression.

Case 1. when α= '<=' then equivalent Arithmetic expression is

a<=b
IF True do Condition 1
IF False do Condition 2

Is Equivalent to

$$\boxed{A = \lfloor(\lfloor b/a \rfloor / b) + \partial \rfloor \text{ Where } \partial \rightarrow 1 \text{ such as } \partial = 0.9999999999}$$

Here A gives value 0 or 1 we use multiplicative and additive property of Real number
Arithmetic Expression is
(A)*(Condition 1) + (1-A)*(Condition 2)

Case 2. when α= '>=' then equivalent Arithmetic expression is..

Expression is
a >= b
IF True do Condition 1
IF False do Condition 2

Is Equivalent to

$\boxed{A = \lfloor(\lfloor a/b\rfloor/a) + \partial\rfloor \text{ Where } \partial\to 1 \text{ such as } \partial = 0.9999999999}$

Here A gives value 0 or 1 we use multiplicative and additive property of
Real number
Arithmetic Expression is

(A)*(Condition 1) + (1-A)*(Condition 2)

Case 3. when α ' < ' then equivalent Arithmetic expression is..

Expression is
a < b
IF True do Condition 1
IF False do Condition 2

Is Equivalent to

$\boxed{A = \lfloor(\lfloor(b-1)/a\rfloor/b) + \partial\rfloor \text{ Where } \partial\to 1 \text{ such as } \partial = 0.9999999999}$

Where $\partial\to 1$ such as Here A gives value 0 or 1 we use multiplicative and additive property of
Real number
Arithmetic Expression is
(A)*(Condition 1) + (1-A)*(Condition 2)

Case 4. when α '>' then equivalent Arithmetic expression is..

Expression is
a >b
IF True do Condition 1
IF False do Condition 2

Is Equivalent to

$$\boxed{A = \lfloor(\lfloor(a-1)/b\rfloor/a) + \partial\rfloor \text{ Where } \partial \rightarrow 1 \text{ such as } \partial = 0.9999999999}$$

Here A gives value 0 or 1 we use multiplicative and additive property of Real number Arithmetic Expression is

(A)*(Condition 1) + (1-A)*(Condition 2)

7. APPLACATION

Problems:
1. Suppose we calculate salary of employee with the base of their expression of work in industry work experience is more than 5 years salary is increase by 5000 Rs
Solution
Let x be an employee work experience sand S in basic salary .

If x >= 5
If True Then S=S+5000
IF False Then S=S

Arithmetic Expression of these solution is
Camper with case 1;

$$A = \lfloor(\lfloor a/b\rfloor/a) + \partial\rfloor \text{ Where } \partial \rightarrow 1 \text{ such as } \partial = 0.9999999999$$

Hear we put b=5 constant value, a=x changing value

If condition stratified A=1
Else Condition not Satisfied A=0

Total Salary of Employee is=basic Salary + (A*5000)

8. CONCLUSION

In This paper we understand how to solve how to convert comparison expression in to arithmetic expression. With the help of this methodology we calculate many problems in day-to-day life using arithmetic calculation. Also we solve some proof in mathematic using this methodology. In other words we convert some logical statement in mathematics format. With the help of this methodology we solve many problems in day to day life and which is very help full for future research in mathematics. In this methodology we use in Artificial intelligence.

REFRRENCES

[1]Real Number Axioms from http://www.calvin.edu/~rpruim/courses/m361/F03/overheads/real-axioms-print-pp4.pdf

[2] Ceiling Function from http://en.wikipedia.org/wiki/Ceiling_function

[3] Comparison Operator from http://en.wikipedia.org/wiki/Comparison_operator

[4] Kenneth H.Rosen "*Discrete Mathematics*" 5th ed.