Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals of DAAAM International

PROGRAM FLOW CONTROL IN WOLFRAM MATHEMATICA

LACKOVIC, A[ndrej]

Abstract: *Program flow control in Mathematica is usually done using two mechanisms. The first mechanism is a conditional statement based on the If command, while the second mechanism is a loop mechanism. There are two basic loop structures in Mathematica: the DO structure and the WHILE structure. Conditional expressions within these structures are examined using relational and logical operators.*
Key words: *flow control, operators, conditional expression, loop*

1. INTRODUCTION

When solving certain problems, it is necessary to repeat a series of commands or switch to a specific location in the program depending on the given parameters, in other words, to have the necessary control over the program execution flow. Basic mechanisms controlling the program execution flow, which can be used with almost all programming languages, are loops, branches and jumps. Apart from these, Mathematica includes other management structures to facilitate writing of diverse simple and effective programs.

2. RELATIONAL AND LOGICAL OPERATORS

Relational operators (comparison operators) compare two numbers and determine whether the comparing statement is "true" ("correct") or "false" ("incorrect"). Logical operators examine expressions whose values may be true or false. Relational and logical operators can be used in mathematical expressions, and are often used in combination with other operators when making management related decisions during the execution of the program. (Wolfram, 2008.)

Relational operator	description
<	Less than
>	More than
<=	Equal to or less than
>=	Equal to or more than
==	Equal to
!=	Different from

Tab. 1. Relational operators

logical operator	description
&&	If both have the value true, the result is true; otherwise the result is false
||	If one or both have the value true, the result is true; otherwise the result is false
!	Gives a value opposite of the operand value; the result is true if the operand is false, and if the operand is true, the result is false

Tab. 2. Logical operators

3. FLOW CONTROL COMMANDS

3.1 Conditional statements

Conditional statement is a command that allows the program to decide whether to perform a group of commands following the statement for conditional execution, or to skip these commands. (Wolfram, 2008.)

IF – END structure*:*

If[conditional expression, command group]

If the conditional expression has the value true, the program executes commands following the comma. If the conditional expression is false, the program skips the group of commands after the comma, and continues to perform the commands after].

IF – ELSE – END structure:

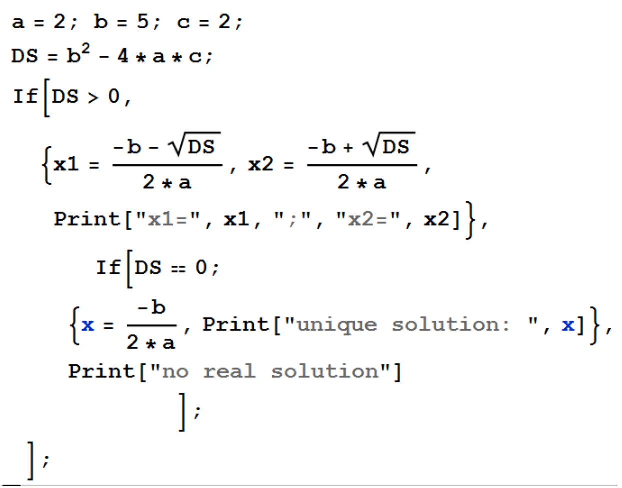
If[conditional expression, command group (1),
 command group (2)]

If the value of the conditional expression is true, the program executes the first group of commands after the comma, and if the value of the conditional expression is false, the program executes the second group of commands (after the second comma).

ELSEIF structure:

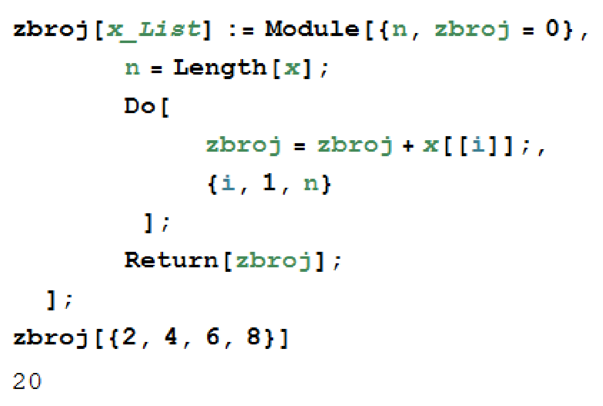
```
If[condition(1) ,  commands (1),
       If[condition(2), commands (2), ...
                ]
         ]
```

If the conditional expression has the value true, the program executes the first group of commands. If the conditional expression of the statement has the value false, the programmoves to a new statement If, etc.



```
a = 2; b = 5; c = 2;
DS = b^2 - 4 * a * c;
If[DS > 0,
   {x1 = (-b - Sqrt[DS])/(2 * a), x2 = (-b + Sqrt[DS])/(2 * a),
    Print["x1=", x1, ";", "x2=", x2]},
       If[DS == 0;
        {x = -b/(2 * a), Print["unique solution: ", x]},
        Print["no real solution"]
              ];
  ];
```

Fig. 1. Source code of Example 1



```
zbroj[x_List] := Module[{n, zbroj = 0},
        n = Length[x];
        Do[
              zbroj = zbroj + x[[i]];,
              {i, 1, n}
         ];
        Return[zbroj];
   ];
zbroj[{2, 4, 6, 8}]
20
```

Fig. 2. Source code of Example 2

Example 1: Along with default parameters a,b and c of the quadratic equation

$$ax^2 + bx + c = 0 \quad (1)$$

the program displays two real solutions, one double solution or no real solutions.

3.2. Loops

Mathematica supports two loop structures: DO and WHILE (Maeder, 1997.)

DO structure*:*

Do[loop body, {s, s_{min}, s_{max}, step}]

The body of the loop is executed until s takes integer values from smin to smax with the default step.

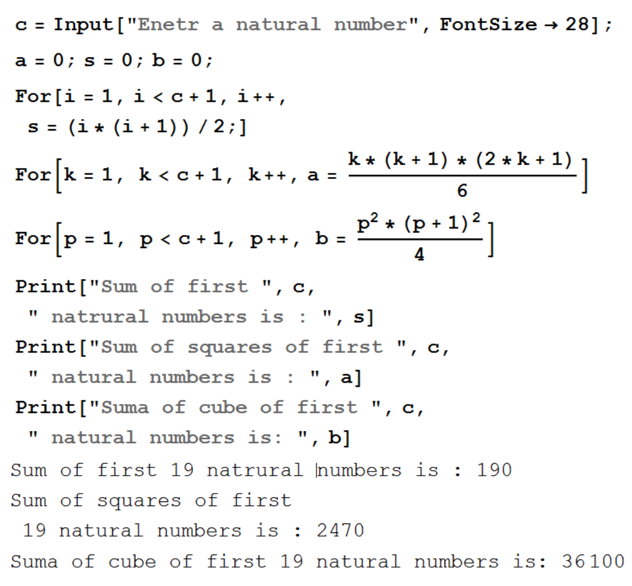
Example 2: The program takes vector (list) as argument and returns the sum of its elements.

FOR structure:

For[start, conditional expression, step, loop body]

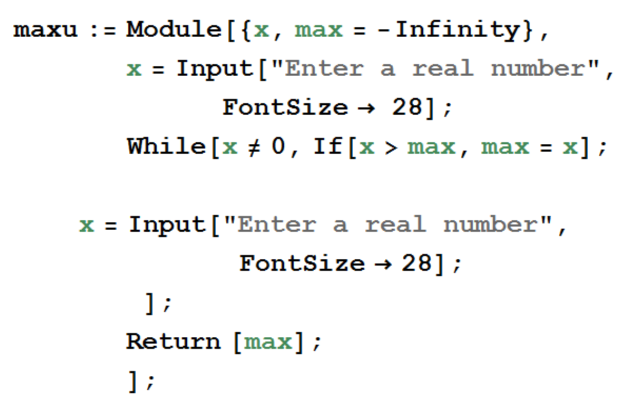
The body of the loop is executed until the value of the conditional expression is true.

Example 3: The program takes one real number as argument an returns the sum of first n natural numbers, sum of squares n natural numbers and sum of cube n natural numbers.



```
c = Input["Enetr a natural number", FontSize → 28];
a = 0; s = 0; b = 0;
For[i = 1, i < c + 1, i++,
 s = (i * (i + 1)) / 2;]
For[k = 1, k < c + 1, k++, a = k * (k + 1) * (2 * k + 1) / 6]
For[p = 1, p < c + 1, p++, b = p^2 * (p + 1)^2 / 4]
Print["Sum of first ", c,
 " natrural numbers is : ", s]
Print["Sum of squares of first ", c,
 " natural numbers is : ", a]
Print["Suma of cube of first ", c,
 " natural numbers is: ", b]
Sum of first 19 natrural numbers is : 190
Sum of squares of first
 19 natural numbers is : 2470
Suma of cube of first 19 natural numbers is: 36100
```

Fig. 3. Source code of Example 3



```
maxu := Module[{x, max = -Infinity},
        x = Input["Enter a real number",
              FontSize → 28];
        While[x ≠ 0, If[x > max, max = x];

      x = Input["Enter a real number",
                FontSize → 28];
         ];
        Return [max];
        ];
```

Fig. 4. Source code of Example 4

WHILE structure:

While[conditional expression, loop body]

Conditional expression in the While command must contain at least one variable, whose value should be known when Mathematica executes the While command for the first time. Within the loop body there must be at least one command that assigns a new value for at least one of these variables. Otherwise, the execution of the loop would never stop, because the conditional expression would always have a value of true. (Wolfram, 2008.)

Note that in an example like:

i=0; While [i<0 , tot+=f[i] ; i++]

the roles of ; and , are reversed relative to C-like programming languages.

Example 4: The program loads the numbers one by one as arguments using the *Input* command, and specifies the maximum number loaded. The loading process is stopped when number 0 is loaded.

4. CONCLUSION

Only the programming in Mathematica software package is called rule-based programming. As regarding the program execution flow control, there is a standard algorithm that we could program in any procedural language like Fortran or C. However, Mathematica has a very simplified syntax for using commands. In addition, the advantage is that it enables us to use software package functions (completed functions within Mathematica) within a standard algorithm, which in most other programming languages would need to be written separately, especially in the case of mathematical functions. In future research plan is to study additional options of commands for flow control of the program, and make detailed comparisons with other programming languages (like C, C++, C#).

5. REFERENCES

Maeder E., Roman (1997). Programing in Mathematica (3rd Edition), Addison-Wesley Professionals, ISBN-978-0201854497, States of America

Maeder E., Roman (1996). The Mathematica programmer II, Academic Press, ISBN 978-0124649927, States of America

Wolfram, Stephen (2008). *The Mathematica Book, Fourth Edition*, Wolfram Media, ISBN 1-57955-004-5, United States of America

Wolfram, Stephan (1994). Mathemathica – The Student book, Addison-Wesley Pub, ISBN 978-0201554793, USA

*** (2010) http://reference.wolfram.com – Wolfram Mathematica Documentation Center, Accesed on : 2010-08-13