

Re: EE Student, Edit, Proposal Masters, Help (concepts of functional programming, symbolic programming and MATLAB)

Source: <http://sci.tech-archive.net/Archive/sci.math.symbolic/2005-03/0092.html>

From: Robert Israel (israel_at_math.ubc.ca)

Date: 03/08/05

Date: 8 Mar 2005 22:27:56 GMT

In article <c5e803c2.0503072121.39295dd1@posting.google.com>, John Creighton <JohnCreighton_@hotmail.com> wrote:

>Perhaps from an abstract mathematical perspective the low level
>interpretation is not important except when the low level behavior has
>a direct consequence on the abstract high level behavior. In the case
>of MAPLE the pointers are not completely encapsulated. Consider the
>example given in

>Rofer Kraft, *Programming In Maple*

><http://adept.maplesoft.com/powertools/programming/html/2.01and2.02.html>

>"Now make x a name for y , y a name for z , and give z a numeric
>value, in that order.

>> $x := y;$

>> $y := z;$

>> $z := 3;$

> $x:=y$

> $y:=z$

> $z:=3$

>What does x evaluate to now?

>> $x;$

> 3

>Change the value of z .

>> $z := 5;$

> $X:=5$

>Check the value of x .

>

>> $x;$

> 5

>In this example, x points to y , y points to z , and at first z

>pointed to 3, so x (and y) evaluated to 3. Then z was pointed at

>5, so x (and y) evaluated to 5. This is full evaluation. Maple keeps

>following the trail of assignments until it gets to a "dead end",

>either a numeric value or an unassigned name. "

Yes, but this should be thought of as a chain of assigned values rather than a chain of pointers. A rather subtle difference, to be sure...

>>Expressions describe how data is evaluated mathematically in terms of
>>data (e.g. variables) and functions (e.g plus). Expressions can be
>>represented with strings but are usually represented as a list.

>"No, expressions are not lists. A list is a particular type of
>expression."

>Do you have a reference for this? I wonder if the definition of a list
>is agreed upon in the context of computer science.

I think we have a terminology problem here. Maple has a specific data type called a list, which is not the same as what computer science calls a list. In Maple, a list (at the user level) is an ordered sequence of expressions enclosed in square brackets.

>By the very sound of its name I would of took the word expression to
>be something which can be evaluated. Continuing with my reasoning I
>would think that a list would be no more a type of expression then a
>numeric would be an expression and a number a set.

Basically every Maple object is an expression, unless it's a sequence of expressions.

>It would seem to me that since expressions are stored in an operation
>tree that to get to the innermost argument you must start from the
>outside and work inward. Perhaps expressions are by default
>automatically evaluated by maple before being passed to a function.
>And perhaps there are ways to prevent this. Given a simply command
>like collect or expand it would seem like a waste of computer cycles
>to reevaluate the expression before the function which simplifies the
>expression is carried out. I am interested to here more on this do you
>have any good references?

You might start with the Maple 9.5 "Introductory Programming Guide".

Robert Israel israel@math.ubc.ca
Department of Mathematics <http://www.math.ubc.ca/~israel>
University of British Columbia Vancouver, BC, Canada