

## 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/2005-03/0134.html>

---

*From:* John Creighton ([JohnCreighton\\_\\_at\\_hotmail.com](mailto:JohnCreighton__at_hotmail.com))

*Date:* 03/14/05

Date: 13 Mar 2005 16:42:20 -0800

Jens Axel Sjøgaard <[usenet@soegaard.net](mailto:usenet@soegaard.net)> wrote in message  
news:<4231b555\$0\$246\$edfadb0f@dread12.news.tele.dk>...

> *John Creighton wrote:*

>

> > *Lisp has since been*

> > *expanded to include many programming paradigms and is now known as*

> > *Common Lisp (CLisp). Although, Common Lisp is no longer a pure*

> > *functional language some functional programming language lisp dialects*

> > *have been created (e.g. Scheme).*

>

> *Scheme support the paradigm of functional programming very well,*

> *but is also impure.*

I think I will put Scheme on my list of things to read about. I have  
been reading a tutorial on Haskell

<http://www.isi.edu/~hdaume/htut/>

and a primer on Lisp.

<http://mypage.iu.edu/~colallen/lp/node11.html>

I have not seen a lot of the concepts in the Lisp tutorial that I  
have been reading about in Haskell tutorial. Perhaps this is just  
because the Haskell tutorial is much more thorough.

> > *In a pure functional (e.g. Haskell) languages such as Haskell [39]*

> > *assignment is not allowed*

After some reading I guess I should use the word reassignment. In  
Haskell, you can define symbols to represent functions and parts of  
expressions using the Let statement. It is called something different  
when something does not have a consistent value like reading from the  
keyboard. I am not yet clear on these concepts.

>> *in order to eliminate side effects.*

> > *Alternatively monads are used to solve problems such as exception*

> > *handling which is difficult to do without assignment.=20*

>

> *Are you saying that it is difficult for the user of Haskell or the*

ath.symbolic: Re: EE Student, Edit, Proposal Masters, Help (concepts of functional programming, symbolic programming a

- > *implementor of haskell to implement exceptions? Or perhaps you*
- > *are talking about exceptional situations rather than "traditional"*
- > *exceptions.*

I think I concluded to much from reading the first of the paper at:

<http://homepages.inf.ed.ac.uk/wadler/papers/marktoberdorf/marktoberdorf.pdf>

"Say I write an evaluator in a pure functional language.

{ To add error handling to it, I need to modify each recursive call to check

for and handle errors appropriately. Had I used an impure language with

exceptions, no such restructuring would be needed.

{ To add a count of operations performed to it, I need to modify each recursive

call to pass around such counts appropriately. Had I used an impure language

with a global variable that could be incremented, no such restructuring would

be needed.

{ To add an execution trace to it, I need to modify each recursive call to

pass around such traces appropriately. Had I used an impure language that

performed output as a side effect, no such restructuring would be needed."

I will wait until I finish the tutorial on Haskell before I read the paper with the above quote.

- > > *This is not*
- > > *necessary in Maple because Maple is not a pure functional language. In*
- > > *Maple assignment can be done as in a procedural style:*
- > > *x:=3D2*
- > > *x:=3Dx+1*
- >

> *Perhaps you mean "imperative style" rather than "procedural style"?*

> *(Since you don't define any procedures nor call them in the example)*

Maybe this is right but for some reason people seem to separate languages as object oriented, functional and procedural. Is imperative a relatively new way of describing languages. I wonder when it was first introduced.

- > > *Variables can represent many kinds of data structures.=20*
- > > *A data structure that can be evaluated is known as an expression.=20*
- >

> *Variables are bound to values. How do you define data structure?*

I'll take a crack this. A data structure definition describes the operations by which data can be accessed for all instances of the data structure. A group of data that can be accessed in this fashion is considered an instance of the data structure. For instance an array can be accessed though indices and the data of a tree can be accessed though the operations get root, get left node, get right node. Thus

Re: EE Student, Edit, Proposal Masters, Help (concepts of functional programming, symbolic programming a

from this perspective an assignment chain could be considered a data structure that could be accessed through the operations  $\text{eval}(x,n)$ .

Where  $x$  is a member of the assignment chain and  $n$  is the level of evaluation.

> > *The symbolic package for MATLAB performs symbolic operations by storing symbolic expressions as objects. MATLAB manipulates expressions by: passing the expression to the Maple kernel and then storing the resulting string in a field of a symbolic object [7].* =20

>

> *What is a symbolic object?*

In MATLAB a symbolic object is an object with one field which contains a symbolic expression in a string representation. To make this behave similarly to a symbolic variable in Maple many operations are overridden including  $+$ ,  $-$ ,  $/$   $*$ . Many overridden functions and all operations are performed via the Maple Kernel.

> > *However, there is a size limitation on the size of a symbolic*

> > *variables that can be used within the symbolic package.*

>

> *What is the size of a variable?*

General Size is thought of as a memory requirement. But it could also include measures like the number of elements. Since the memory space of a MATLAB symbolic object depends on the amount of characters needed to make up the symbolic expression, the number of characters in the symbolic object is a reasonable measure of its size.

> --=20

> *Jens Axel S=F8gaard*