

SymTEX: HackCU Demo

Omkar H. Ramachandran,Anthony Tracy,Aidan Bohenick

February 25, 2018

1 Abstract

Here we report on the development of a new \LaTeX package SymTex. The intention behind this package is reduce the amount of work it takes to add mathematics to a \LaTeX project. Our package takes advantage of the built in command 'write18' which gives us access to python scripts within a \LaTeX document. This package will allow users the ability to write a document and ask \LaTeX to solve a mathematical equation, such as an integral or derivative, from which the package will return a formatted line of the user's equation set equal to the symbolic solution. This paper reports on the current progress of this package, points out the current assumptions and where it plans to go in the future.

2 Methods

2.1 Resources

Inorder to accomplish this task we have relied heavily on python and the SymPy's libraries, this allows us to focus less on writing our own solvers and instead allow us to focus on writing a python parser to communicate between the \LaTeX user's and python's mathematical solvers.

2.2 Procedure

The current procedure is as follows:

1. Parse \LaTeX string and convert to algebraic expression.
2. Build symbolic Talbe from expressions.
3. Run expressions through SymPy.
4. Return expressions to \LaTeX .

2.3 Assumptions

There have been a few assumptions made as this is just the start of this project and this is the first report to be made. As we continue to complete this project these assumptions will be addressed.

1. $x*y == x\ y$ (not xy . Notice the space)

2. Subscripts can only be one character long (No Greek Letters!)
3. Only two commands right now: `intop` and `fracddx`
4. Basic Syntax: `commandexpression`
 - (a) For integral: `intopexpression dvars`
 - (b) For derivative: `fracddxexpression`
 - (c) For General: no slashes within the expression. i.e: `fracddxsin(x)` would not be a `sin(x)`
 - (d) For General: No greek charecters, while some work at this time not all do.

,

3 Results

At this time there are plenty of boundary conditions that still need to be assessed, which is to be expected for any parser. However we have shown a proof of concept with the current implimentation as we can show that it works with integrals and derivatives of functions. An example would be the following function:

$$\int 2dx = 2x$$

Which utilized the following code written in L^AT_EX:

```

1. \write18{python3 symtex.py '\intop{2 dx}' > intermediate.txt}
2. $$\intop{2 dx} = \input{intermediate.txt}$$

```

4 Conclusion

In conclusion there is plenty of work that still can be done, however the main proof of concept is done and works great. We have shown in this report that we can have latex, using python's SymPy library, solve integrals and derivatives, then format it all for the user. The next steps are to continue developing the parser inorder to allow users to use more syntax. Then to start implimenting more operations, binary and unary, to allow for more ways in which the user can use this package to have L^AT_EXsolve then format mathematical equations quickly and easily.