

The background features a series of concentric circles in shades of dark blue and teal, centered on a black background. A wavy, horizontal line in a similar color palette cuts across the lower right portion of the image.

# CS 6315: Final Project

## Formal Methods in Exo

Youngjae Moon



# Contents

1. Reminder: About Exo
2. How I have verified the Exocompiler
3. Future project ideas for further improvements
4. GitHub repository of this project for further information




# About Exo

- Exo is an embedded DSL that helps low-level performance engineers transform very simple programs that specify what they want to compute into very complex programs that do the same thing as the specification, only much, much faster.
- Developed by Professor Gilbert Bernstein at University of Washington and other people
- Based on Python
- Compiles to C



# How I have verified the Exocompiler

1. Used Exocompiler to compile the sample codes given from the GitHub repository of Exo language into C.
2. Next, I have encoded the original Python and transformed C functions in Python.
3. Then, I have encoded them again in SMT.
4. Finally, I have used an SMT solver to prove that the original and transformed programs are equivalent.




## Future project ideas for further improvements

- Represent Python and C functions directly in SMT instead of first representing them in Python and then SMT.
  - I have initially tried this using PySMT expressions
    - such as ForAll, Implies, And, Equals, Plus, Times, etc.
  - For instance, ForAll and Implies can be used to represent the for loop structure in Python instead.
  - Nonetheless, this gave me endless errors for debugging.
  - Hence I have stopped approaching in this way and changed the approach.



## Future project ideas for further improvements

- Write more complex Exo functions for this project
  - I have noted that the Exo language lacks documentation.
  - Hence when I have tried out implementing a simple ReLU function and an artificial neural network, they weren't able to be compiled.
  - All Exo functions except conv2d (convolution for 2D) and gaussian blur in this file have been obtained from analyzing the GitHub repository of the Exo language and extracting specific functions mostly from their testing files, other remaining files, and from their web page.



## Future project ideas for further improvements

- Optimizing transformation
  - PySMT can help in determining the optimal set of transformations and optimizations for the Exocompiler to generate optimised code for the target hardware accelerators.
- Code synthesis
  - PySMT can be used to synthesize code fragments that meet specific requirements:
    - e.g., hardware compatibility or performance constraints.
  - This can then be integrated into the generated low-level code.

The background is black with several concentric circles in shades of blue and green. A wavy line in the same color palette starts from the bottom right and curves upwards towards the center.

Check out the GitHub  
repo:

[https://github.com/Pingumaniac/Verification\\_of\\_correctness\\_of\\_Exo\\_programs/tree/main](https://github.com/Pingumaniac/Verification_of_correctness_of_Exo_programs/tree/main)



Any questions?

Thank you!

