

Contributing to Futhark for your Bachelors Project

Troels Henriksen (athas@sigkill.dk)

Computer Science
University of Copenhagen

September 2022



Contribute to the best data-parallel GPU-targeting ML-like functional language developed at DIKU!

Looks a bit like a simplified combination of SML and Haskell.

```
def vector_add [n] (a: [n]i32) (b: [n]b): [n]i32 =  
  map2 (+) a b
```

```
def product [n] (a: [n]i32): i32 =  
  reduce (*) 1 a
```

```
def dot_product [n] (a: [n]i32) (b: [n]i32): i32  
  product (vector_add a b)
```

Futhark runs *very fast* on parallel hardware. Significantly faster than most hand-written C or similar.

Who I am

I am Troels Henriksen, and I conduct research into compiler optimisations and programming language design alongside Cosmin Oancea, Martin Elsmann, Philip Munksgaard, Robert Schenck, and anyone else who wants to play along.

Full disclosure

These are difficult projects

These are all programming-heavy projects. Most require knowledge of a language (Haskell or Futhark) that you have probably not been taught.

The upsides

- These projects are directly relevant to my research, so you will receive as much supervision as necessary.
- These are *real projects*, and your work will actually be used if it is functional.
- Futhark is run as an open source project, so your work will be visible to the entire world!
- They provide a good foundation for doing more involved compiler hacking projects later on.

Project idea: Implementing Automatic Differentiation in Futhark interpreter

Modern machine learning relies on Automatic Differentiation (AD); a technique for computing derivatives of arbitrary programs. We are implementing an advanced highly optimised AD transformation in the Futhark *compiler*, but we would also like to support AD in the Futhark *interpreter*. This is partly because the interpreter is much more useful for interactively debugging Futhark programs, and partly because a “known correct” implementation would be useful for testing the AD implementation in the compiler.

More information:

<https://github.com/diku-dk/futhark/issues/1556>

Project idea: Improve profiling

Currently it can be hard for users to figure out why their Futhark program is running slow. While Futhark does collect profiling information while it runs, there is currently no easy way for users to access it. This project is about wiring up the necessary parts to produce this information in a more digestible way, e.g. in reports or graphs. If time permits, the project could be extended to extend what kind of information is collected (currently Futhark does not track well the relation between source program and generated program), and perhaps visually show how the program is actually compiled.

More information:

`https://github.com/diku-dk/futhark/
issues/1280`

Project idea: Construct an automatic code formatter

Tools such as `go fmt`, `rustfmt`, and `black` have become popular for automatically (re)formatting source code. We would like to also have such a formatter for Futhark. Probably this will involve modifying the existing lexer and parser to provide the necessary information, and studying formatters for other languages to understand how they work. This is a project with potentially high impact, since most Futhark programmers would be using this tool very frequently.

More information:

`https://github.com/diku-dk/futhark/issues/787`

Contact

If you think any of this is interesting, come talk to Troels...

- ...in my office: 772-01-0-S14 at HCØ.
- ...via email: `athas@sigkill.dk`
- ...or on IRC: `#diku` on Freenode

Or talk to Martin Elsman or Cosmin Oancea.

Also check out the website at `https://futhark-lang.org`