Planning report

Sebastian Selander
gusselase@student.gu.se

Colubridae

Colubridae[1] is a compiler for a programming language inspired by the simplicity of Python. However, Colubridae will also include a nominal type system, some type inference, as well as being compiled to LLVM IR.
Some notable features (if time allows) of Colubridae will be:

- higher-order functions.
- algebraic data types
- arrays

A large focus of the project will be to develop the compiler using good techniques for the future, e.g trees that grow, bidirectional type checking

To be considered a useful language I would argue that Colubridae needs a garbage collector, but unfortunately that is out of the scope of this project.

Experiences

I'm just finished my fourth year studying computer science at Gothenburg university. No previous programming experience prior to starting my university studies. This is not the first compiler I make, however, all previous attempts have resulted in compilers that I am not confident in whatsoever, thus this is something I would like to avoid this time.

Techniques

I will be writing the compiler in Haskell, using some standard helpful libraries such as: `lens`, `mtl`, `megaparsec`. I will use the LLVM IR library as well if it still is maintained.

Time plan

Details

Followng is a list of that need to be implemented that I know of in advance:

- Parser
- Alpha renamer
- Typechecker
- Desugaring
- Internal intermediate language
- LLVM IR generation phase

More things will probably arise during development, but I can't think of anything else at the moment.

Time plan

Total time available roughly: `8 * 40h`

The first month will be spent implementing the basics of the language. That is, parsing, renaming, type checking, and code generating basic expressions and statements. Unfortunately at this point I don't know exactly how long this will take.

[1] *Colubridae was chosen as it is a family of mostly non-poisonous (safe) snakes*

I reckon roughly that the following is how my time will be spent:

- `3 * 40` on implementing the basics of the compiler
- `2 * 40` on implementing higher-order functions
- `2 * 40` on implementing algebraic data types and pattern matching
- `1 * 40` on implementing arrays

Time will also be spent refactoring as well as I will make mistakes that need to be corrected for future additions to work.

Please reach out to me if anything is unclear!

[1] *Colubridae was chosen as it is a family of mostly non-poisonous (safe) snakes*