Programming Languages and Compiler Construction
Department of Computer Science
Christian-Albrechts-University of Kiel

**Master Thesis**

# An LLVM Backend for Accelerate

Timo von Holtz

April 15, 2014

Advised By:
Priv.-Doz. Dr. Frank Huch
Assoc. Prof. Dr. Manuel M T Chakravarty

# Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

April 15, 2014

Timo von Holtz

# Todo list

# Contents

# List of Figures

# List of Tables

# Listings

# 1 Introduction

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Write Introduction

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

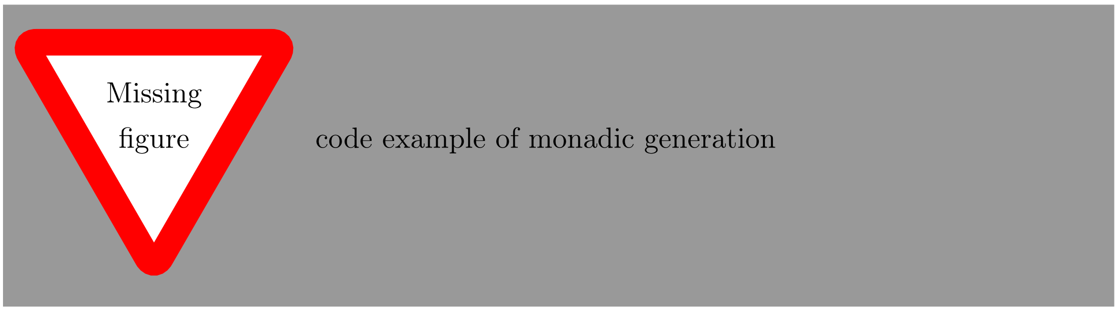
Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really?

Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

# 2 Contributions

## 2.1 llvm-general-quote

When writing a compiler using LLVM in Haskell there is a good tutorial on how to do it at `http://www.stephendiehl.com/llvm/`. It uses *llvm-general* to interface with LLVM. The general idea is to use a monadic generator to produce the AST on the fly. This has some obvious drawbacks, as the code can get unreadable pretty quickly.



code example of monadic generation

A solution is to use quasiquotation[5]. That way, one can write the llvm-ir directly, without having to manipulate the AST by hand. This could also be done with a simple parser though. The main advantage of quasiquotation is, the use of antiquotation. They allow you to reference arbitrary Haskell values inside the quotation.

I implemented *language-llvm-quote*, a quasiquotation library for LLVM. The design is inspired by *language-c-quote*, which is also used in the cuda implementation of Accelerate. I use "Happy" and *Alex*.

writeup of llvm-general-quote

```
test = putStrLn "bla"
```

# Bibliography

[1] Stephen Diehl. *Implementing a JIT Compiled Language with Haskell and LLVM.* Jan. 2014. URL: http://www.stephendiehl.com/llvm/.

[2] Chris Dornan and Simon Marlow. *Alex: A lexical analyser generator for Haskell.* URL: http://www.haskell.org/alex/.

[3] Andy Gill and Simon Marlow. “Happy: the parser generator for Haskell”. In: *University of Glasgow* (1995).

[4] Timo von Holtz. *language-llvm-quote.* URL: https://github.com/tvh/language-llvm-quote.

[5] Geoffrey Mainland. “Why it’s nice to be quoted: quasiquoting for haskell”. In: *Proceedings of the ACM SIGPLAN workshop on Haskell workshop.* ACM. 2007, pp. 73–82.

[6] Geoffrey Mainland and Manuel MT Chakravarty. *language-c-quote.* URL: http://hackage.haskell.org/package/language-c-quote.

[7] Benjamin Scarlet. *llvm-general.* 2013. URL: http://hackage.haskell.org/package/llvm-general.