

Helium Technical report

Table of Contents

1 Overall structure.....	1
1.1 Techniques & libraries.....	1
1.1.1 Attribute Grammars.....	1
1.1.2 Extreme programming.....	1
1.1.3 Parsec.....	1
1.1.4 Lazy Virtual Machine.....	1
1.1.5 Constraint-based type inference.....	1
1.2 Intermediate data types.....	1
1.2.1 Unified Haskell Architecture.....	1
1.2.2 Core.....	1
1.3 Modules.....	1
1.3.1 Main.....	1
1.3.2 Compile.....	1
2 Front-end.....	2
2.1 Lexer.....	2
2.1.1 Layout-rule.....	2
2.2 Parser.....	2
2.2.1 Operators & priorities.....	2
2.2.2 Putting multiple function bindings together.....	2
3 Static Analysis.....	3
3.1 Type inference.....	3
3.2 Other static checks.....	3
4 Back-end.....	4
5 Case studies: enhancing the compiler.....	5
5.1 Adding a static check.....	5
5.2 Using another backend.....	5
5.3 Providing a new syntax.....	5
5.4 Inserting a better pattern-matching compiler.....	5
6 Appendices.....	6
6.1 AG by example.....	6

1 Overall structure

1.1 Techniques &libraries

1.1.1 Attribute Grammars

1.1.2 Extreme programming

1.1.3 Parsec

1.1.4 Lazy Virtual Machine

1.1.5 Constraint-based type inference

1.2 Intermediate data types

1.2.1 Unified Haskell Architecture

1.2.2 Core

1.3 Modules

1.3.1 Main

1.3.2 Compile

2 Front-end

2.1 Lexer

2.1.1 Layout-rule

2.2 Parser

2.2.1 Operators & priorities

2.2.2 Putting multiple functionbindings together

3 Static Analysis

3.1 Typeinference

3.2 Other static checks

4 Back-end

5 Case studies: enhancing the compiler

5.1 Adding a static check

5.2 Using another backend

5.3 Providing a new syntax

5.4 Inserting a better pattern-matching compiler

6 Appendices

6.1 AG by example