

# Coalgebraic effects and their cohandlers in programming languages

(Efekty koalgebraiczne oraz ich kohandlery  
w językach programowania)

Mateusz Urbańczyk

Praca inżynierska

**Promotor:** dr Maciej Piróg

Uniwersytet Wrocławski  
Wydział Matematyki i Informatyki  
Instytut Informatyki

30 czerwca 2020



## Abstract

...

...



# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Problem Analysis . . . . .	7
1.2	Problem Statement . . . . .	7
1.3	Thesis Outline . . . . .	7
<b>2</b>	<b>Background</b>	<b>9</b>
2.1	Algebraic Effects . . . . .	9
2.2	Categorical Setting for Universal Algebra . . . . .	9
2.3	Comodels and Duality . . . . .	9
<b>3</b>	<b>Proposed Solutions</b>	<b>11</b>
3.1	Dynamic Constraints Checking . . . . .	11
3.2	Linear Types . . . . .	11
3.3	Data-Flow Analysis . . . . .	11
3.4	Cohandlers as a Separate Construct . . . . .	11
3.5	Related Work . . . . .	11
<b>4</b>	<b>Coeffectful Programming with Cohandlers</b>	<b>13</b>
4.1	Examples . . . . .	13
4.2	Usage guide . . . . .	13
<b>5</b>	<b>Calculus of Freak language</b>	<b>15</b>
5.1	Syntax . . . . .	15
5.2	Typing Rules . . . . .	15
5.3	Operational Semantics . . . . .	15

5.4	Continuation Passing Style Transformation . . . . .	15
<b>6</b>	<b>Implementation</b>	<b>17</b>
6.1	Abstract Syntax Trees . . . . .	17
6.2	Curried Translation . . . . .	17
6.3	Uncurried Translation . . . . .	17
6.4	Cohandlers . . . . .	17
6.5	Type System . . . . .	17
6.6	Source Code Structure . . . . .	17
<b>7</b>	<b>Conclusion</b>	<b>19</b>
7.1	Summary . . . . .	19
7.2	Future work . . . . .	19

# Chapter 1

## Introduction

### 1.1 Problem Analysis

### 1.2 Problem Statement

### 1.3 Thesis Outline

... [1]





## Chapter 2

# Background

### 2.1 Algebraic Effects

### 2.2 Categorical Setting for Universal Algebra

### 2.3 Comodels and Duality



## Chapter 3

# Proposed Solutions

3.1 Dynamic Constraints Checking

3.2 Linear Types

3.3 Data-Flow Analysis

3.4 Cohandlers as a Separate Construct

3.5 Related Work



## Chapter 4

# Coeffectful Programming with Cohandlers

### 4.1 Examples

### 4.2 Usage guide



## Chapter 5

# Calculus of Freak language

### 5.1 Syntax

### 5.2 Typing Rules

### 5.3 Operational Semantics

### 5.4 Continuation Passing Style Transformation





## Chapter 6

# Implementation

6.1 Abstract Syntax Trees

6.2 Curried Translation

6.3 Uncurried Translation

6.4 Cohandlers

6.5 Type System

6.6 Source Code Structure



## Chapter 7

# Conclusion

### 7.1 Summary

### 7.2 Future work



# Bibliography

- [1] Sam Lindley, Conor McBride, and Craig McLaughlin. “Do be do be do”. In: *CoRR* abs/1611.09259 (2016). arXiv: 1611.09259. URL: <http://arxiv.org/abs/1611.09259>.