

# Implementing UNIX with Effects Handlers

Ramsay Carlaw

October 6, 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Algebraic Effects and Effect Handlers . . . . .	3
2.2	UNIX . . . . .	4
<b>3</b>	<b>Methods</b>	<b>5</b>
<b>4</b>	<b>Results</b>	<b>6</b>
<b>5</b>	<b>Conclusion</b>	<b>7</b>

# Chapter 1

## Introduction

## Chapter 2

# Background

### 2.1 Algebraic Effects and Effect Handlers

Algebraic effects and their corresponding handlers [1] [2] are a programming paradigm that when paired together offers a novel way to compose programs. When programs are written that are ‘black boxes’, that is to say their outputs are defined entirely by their inputs and all functions are pure computation [3], it is safe to make assumptions about the inputs. Assumptions like an age will always be given as an integer or all strings will not exceed the length allocated for them. When programs interact with the real world it is no longer safe to make these assumptions. Consider the following example of heads or tails written conventionally.

```
1 function choose() -> bool {  
2   if (rand() > 0.5) {  
3     return true  
4   }  
5   return false  
6 }  
7  
8 function flip() {  
9   if (choose() == true) {  
10    print 'Heads'  
11  } else {  
12    print 'Tails'  
13  }  
14 }
```

The function `choose` returns `true` or `false` randomly, with equal probability and `flip` uses the result of `choose` to print heads on `true` and tails on `false`. This works fine but if we wanted multiple definitions of `choose` where we alter the probability or add an option for failure we would have to rewrite each function for each implementation.

We can use effects handlers as described by Pretnar et al [2] and with the syntax outlined by Kammar et al [4]

## 2.2 UNIX

UNIX [5] is an operating system designed and implemented by Dennis M. Ritchie and Ken Thompson at AT&T's Bell Labs in 1974. It provides a file system (directories, file protection etc.), a shell, processes (pipe, fork etc) and a userspace. Since it's first release it has been reimplemented for a variety of systems.

## Chapter 3

# Methods

## Chapter 4

# Results

## Chapter 5

## Conclusion



# Bibliography

- [1] Gordon Plotkin and Matija Pretnar. Handlers of algebraic effects. In *European Symposium on Programming*, pages 80–94. Springer, 2009.
- [2] Matija Pretnar. An introduction to algebraic effects and handlers. invited tutorial paper. *Electronic notes in theoretical computer science*, 319:19–35, 2015.
- [3] John Hughes. Why functional programming matters. *The computer journal*, 32(2):98–107, 1989.
- [4] Ohad Kammar, Sam Lindley, and Nicolas Oury. Handlers in action. *ACM SIGPLAN Notices*, 48(9):145–158, 2013.
- [5] Dennis M Ritchie and Ken Thompson. The unix time-sharing system. *Bell System Technical Journal*, 57(6):1905–1929, 1978.