

λ -poetry

Niklas Bühler

2020

Contents

| | |
|-------------------|----|
| Contents | 3 |
| 1 Introduction | 6 |
| 2 Booleans | 8 |
| 3 Natural Numbers | 10 |
| 4 Pairs | 12 |
| 5 Lists | 14 |
| 6 Binary Trees | 16 |
| 7 Recursion | 18 |

Preface

The lambda calculus is a formal system that is used to express computations. Like the Turing machine, it's a universal model of computation, however it's much simpler and more elegant than those bulky machines.

The following pages are filled with fundamental datastructures and the most important functions operating on them, in the untyped lambda calculus.

The datastructures presented differ from ordinary, imperative datastructures, as they are purely functional. That means they don't describe where and how the data is stored, but rather how functions are applied to that data.

Variable names are often chosen as a hint on the value they're holding, but aren't elaborated on. Their exact purpose and meaning is left open for exploration.

Have fun! – August 2020

Chapter 1

Introduction

There are three main constructs in the untyped lambda calculus:

1. **Variables** x ,
2. **Abstractions** $\lambda x. t$ and
3. **Applications** $f\ x$.

$$\mathbf{id} = \lambda x. x$$

Chapter 2

Booleans

True = $\lambda t. \lambda f. t$

False = $\lambda t. \lambda f. f$

Not = $\lambda b. b \text{ False } \text{True}$

And = $\lambda a. \lambda b. a \ b \text{ False}$

Or = $\lambda a. \lambda b. a \text{ True } b$

$$\begin{aligned}
\text{Not True} &= (\lambda b. b \text{ False True}) \text{ True} \\
&\Rightarrow \text{True False True} \\
&= (\lambda t. \lambda f. t) \text{ False True} \\
&\Rightarrow \text{False}
\end{aligned}$$

$$\begin{aligned}
\text{And True True} &= (\lambda a. \lambda b. a \ b \text{ False}) \text{ True True} \\
&\Rightarrow \text{True True False} \\
&= (\lambda t. \lambda f. t) \text{ True False} \\
&\Rightarrow \text{True}
\end{aligned}$$

$$\begin{aligned}
\text{Or False True} &= (\lambda a. \lambda b. a \text{ True } b) \text{ False True} \\
&\Rightarrow \text{False True True} \\
&= (\lambda t. \lambda f. f) \text{ True True} \\
&\Rightarrow \text{True}
\end{aligned}$$

Chapter 3

Natural Numbers

$$\mathbf{Zero} = \lambda s. \lambda z. z$$

$$\mathbf{Succ} = \lambda n. \lambda s. \lambda z. s (n s z)$$

$$\mathbf{Pred} = \lambda n. \lambda s. \lambda z. n (\lambda f. \lambda g. g (f s)) (\lambda x. z) \mathbf{id}$$

$$\mathbf{isZero} = \lambda n. n (\lambda x. \mathbf{False}) \mathbf{True}$$

$$\mathbf{Plus} = \lambda m. \lambda n. m \mathbf{Succ} n$$

$$\mathbf{Minus} = \lambda m. \lambda n. m \mathbf{Pred} n$$

$$\mathbf{Mult} = \lambda m. \lambda n. m (\mathbf{Plus} n) \mathbf{Zero}$$

$$\mathbf{Power} = \lambda b. \lambda e. e b$$

$$\begin{aligned}
\mathbf{Succ\ Zero} &= (\lambda n. \lambda s. \lambda z. s\ (n\ s\ z))\ \mathbf{Zero} \\
&\Rightarrow \lambda s. \lambda z. s\ (\mathbf{Zero}\ s\ z) \\
&= \lambda s. \lambda z. s\ ((\lambda s. \lambda z. z)\ s\ z) \\
&\Rightarrow \lambda s. \lambda z. s\ z \\
&= \mathbf{One}
\end{aligned}$$

$$\begin{aligned}
\mathbf{Plus\ One\ One} &= (\lambda m. \lambda n. m\ \mathbf{Succ}\ n)\ \mathbf{One}\ \mathbf{One} \\
&\Rightarrow \mathbf{One}\ \mathbf{Succ}\ \mathbf{One} \\
&= (\lambda s. \lambda z. s\ z)\ \mathbf{Succ}\ \mathbf{One} \\
&\Rightarrow \mathbf{Succ}\ \mathbf{One} \\
&= (\lambda n. \lambda s. \lambda z. s\ (n\ s\ z))\ \mathbf{One} \\
&\Rightarrow \lambda s. \lambda z. s\ (\mathbf{One}\ s\ z) \\
&= \lambda s. \lambda z. s\ ((\lambda s. \lambda z. s\ z)\ s\ z) \\
&\Rightarrow \lambda s. \lambda z. s\ (s\ z) \\
&= \mathbf{Two}
\end{aligned}$$

$$\begin{aligned}
\mathbf{Pred\ One} &= (\lambda n. \lambda s. \lambda z. n\ (\lambda f. \lambda g. g\ (f\ s))\ (\lambda x. z)\ \mathbf{id})\ \mathbf{One} \\
&\Rightarrow \lambda s. \lambda z. \mathbf{One}\ (\lambda f. \lambda g. g\ (f\ s))\ (\lambda x. z)\ \mathbf{id} \\
&= \lambda s. \lambda z. (\lambda s. \lambda z. s\ z)\ (\lambda f. \lambda g. g\ (f\ s))\ (\lambda x. z)\ \mathbf{id} \\
&\Rightarrow \lambda s. \lambda z. (\lambda f. \lambda g. g\ (f\ s))\ (\lambda x. z)\ \mathbf{id} \\
&\Rightarrow \lambda s. \lambda z. \mathbf{id}\ ((\lambda x. z)\ s) \\
&\Rightarrow \lambda s. \lambda z. (\lambda x. z)\ s \\
&\Rightarrow \lambda s. \lambda z. z \\
&= \mathbf{Zero}
\end{aligned}$$

Chapter 4

Pairs

$$\mathbf{Pair} = \lambda a. \lambda b. \lambda p. p \ a \ b$$

$$\mathbf{First} = \lambda p. p \ (\lambda a. \lambda b. a)$$

$$\mathbf{Second} = \lambda p. p \ (\lambda a. \lambda b. b)$$

$$\mathbf{Swap} = \lambda p. \mathbf{Pair} \ (\mathbf{Second} \ p) \ (\mathbf{First} \ p)$$

Nothing.

Chapter 5

Lists

Nil = $\lambda c . \lambda n . n$

Cons = $\lambda h . \lambda t . \lambda c . \lambda n . c \ h \ (t \ c \ n)$

Append = $\lambda k . \lambda l . k \ \mathbf{Cons} \ l$

ListSum = $\lambda l . l \ \mathbf{Plus} \ \mathbf{Zero}$

Nothing.

Chapter 6

Binary Trees

Leaf = $\lambda v . \lambda n . \lambda l . l \ v$

Node = $\lambda s . \lambda t . \lambda n . \lambda l . n \ (s \ n \ l) \ (t \ n \ l)$

TreeSum = $\lambda t . t \ \mathbf{Plus} \ \mathbf{id}$

Nothing.

Chapter 7

Recursion

$$\mathbf{Y} = \lambda f.(\lambda x. f \ (x \ x)) \ (\lambda x. f \ (x \ x))$$

$$\begin{aligned}
\mathbf{Y} \ f &= (\lambda f. (\lambda x. f \ (x \ x)) \ (\lambda x. f \ (x \ x))) \ f \\
&\Rightarrow (\lambda x. f \ (x \ x)) \ (\lambda x. f \ (x \ x)) \\
&\Rightarrow f \ ((\lambda x. f \ (x \ x)) \ (\lambda x. f \ (x \ x))) \\
&= f \ (\mathbf{Y} \ f) \\
&\Rightarrow \dots
\end{aligned}$$

$$\begin{aligned}
\mathbf{fact} &= \lambda f. \lambda n. (\mathbf{isZero} \ n) \ \mathbf{One} \ (\mathbf{Mult} \ n \ (f \ (\mathbf{Pred} \ n))) \\
\mathbf{Fact} &= \mathbf{Y} \ \mathbf{fact}
\end{aligned}$$

$$\begin{aligned}
\mathbf{Fact} \ \mathbf{Two} &= \mathbf{Y} \ \mathbf{fact} \ \mathbf{Two} \\
&\Rightarrow \mathbf{fact} \ (\mathbf{Y} \ \mathbf{fact}) \ \mathbf{Two} \\
&= (\lambda f. \lambda n. (\mathbf{isZero} \ n) \ \mathbf{One} \ (\mathbf{Mult} \ n \ (f \ (\mathbf{Pred} \ n)))) \ (\mathbf{Y} \ \mathbf{fact}) \ \mathbf{Two} \\
&\Rightarrow (\mathbf{isZero} \ \mathbf{Two}) \ \mathbf{One} \ (\mathbf{Mult} \ \mathbf{Two} \ ((\mathbf{Y} \ \mathbf{fact}) \ (\mathbf{Pred} \ \mathbf{Two}))) \\
&\Rightarrow \mathbf{False} \ \mathbf{One} \ (\mathbf{Mult} \ \mathbf{Two} \ ((\mathbf{Y} \ \mathbf{fact}) \ \mathbf{One})) \\
&\Rightarrow \mathbf{Mult} \ \mathbf{Two} \ ((\mathbf{Y} \ \mathbf{fact}) \ \mathbf{One}) \\
&\Rightarrow \mathbf{Mult} \ \mathbf{Two} \ ((\mathbf{fact} \ (\mathbf{Y} \ \mathbf{fact})) \ \mathbf{One}) \\
&= \mathbf{Mult} \ \mathbf{Two} \ ((\lambda f. \lambda n. (\mathbf{isZero} \ n) \ \mathbf{One} \ (\mathbf{Mult} \ n \ (f \ (\mathbf{Pred} \ n)))) \ (\mathbf{Y} \ \mathbf{fact})) \ \mathbf{One} \\
&\Rightarrow \mathbf{Mult} \ \mathbf{Two} \ (((\mathbf{isZero} \ \mathbf{One}) \ \mathbf{One} \ (\mathbf{Mult} \ \mathbf{One} \ ((\mathbf{Y} \ \mathbf{fact}) \ (\mathbf{Pred} \ \mathbf{One})))) \) \)
\end{aligned}$$