Types, inference, and F# data types

The INFDEV@HR Team

Hogeschool Rotterdam
Rotterdam, Netherlands

Introduction

Introduction

Lecture topics

- The typed lambda calculus
- Type checking
- Type inference
- F# basic types
- F# custom advanced types

The typed lambda calculus

The typed lambda calculus

Idea

We can add types to the lambda calculus!

The typed lambda calculus

Idea

This makes the lambda calculus safer to compose, as we run no more the risk of passing a parameter of the wrong shape (like an integer where a function is expected)

The typed lambda calculus

Types in the lambda calculus follow the same basic syntactic shape of the lambda calculus itself.

Consider any type τ; it will be made up, recursively, out of any of the following shapes:

- a type variable (like generic type parameters, such as `T`): α
- a generic type definition (like the definition of `List<T>`: $\forall\alpha \Rightarrow \tau$
- a function type: $\tau \rightarrow \tau$
- a type application (like `List<Customer>`): $\tau\tau$

The typed lambda calculus

Lambda calculus programs are now extended to include type declarations:

Consider any term `t`; it will be made up, recursively, out of any of the following shapes:

- a free variable: `x`
- a lambda definition: λ`x`$\rightarrow$`t`
- a generic parameter introduction: $\Lambda\alpha \Rightarrow$`t`
- a value application: `t t`
- a type application: `t` τ

The typed lambda calculus

True and false now become generic with respect to the inputs `t` and `f`:

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha)\ \ (\texttt{f}:\alpha) \rightarrow \texttt{t})$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha)\ \ (\texttt{f}:\alpha) \rightarrow \texttt{f})$$

The typed lambda calculus

Numbers now become generic with respect to the data manipulated through `z` and `s`:

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha))\ \ (\texttt{z} : \alpha) \to \texttt{z})$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha))\ \ (\texttt{z} : \alpha) \to (\texttt{s}\ \ (\texttt{s}\ \ \texttt{z})))$$

The typed lambda calculus

Typed pairs

- A pair takes as input the two values, x and y
- It then expects a function, g, which will consume x and y and produce some final result
- x and y have generic types, α and β
- g also has generic type, it takes as input an α and a β, and returns a γ

The typed lambda calculus

Pairs become generic with respect to the types of the input, but also the generated output:

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\texttt{x}:\alpha)\ \ (\texttt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow\ \ (\lambda(\texttt{f}:(\alpha \rightarrow \beta \rightarrow \gamma)) \rightarrow ((\texttt{f}\ \ \texttt{x})\ \ \texttt{y})))$$

The typed lambda calculus

Pairs become generic with respect to the types of the input, but also the generated output:

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((\mathtt{p}\ \ \alpha)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{x})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((\mathtt{p}\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

The typed lambda calculus

Typed union

- A union is built from one of two possible values, `x` and `y`
- It then expects two functions, `f` and `g`, which will consume `x` or `y` and produce some final result
- `x` and `y` have generic types, α and β
- `f` and `g` also have generic types, they takes as input an α or a β, but both return a γ

The typed lambda calculus

Unions become generic with respect to the types of the inputs, but also the generated output:

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{x} : \alpha) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f} : (\alpha \rightarrow \gamma)) \ (\mathtt{g} : (\beta \rightarrow \gamma)) \rightarrow (\mathtt{f}\ \mathtt{x})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f} : (\alpha \rightarrow \gamma)) \ (\mathtt{g} : (\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \mathtt{y})))$$

The typed lambda calculus

Unions become generic with respect to the types of the inputs, but also the generated output:

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{u}:(\alpha+\beta)) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha\rightarrow\gamma)) \ (\mathtt{g}:(\beta\rightarrow\gamma)) \rightarrow (((\mathtt{u}\ \gamma)\ \mathtt{f})\ \mathtt{g})))$$

Type checking

Type checking

Thanks to types, we can now verify that terms are well-composed: this can prevent various forms of bugs

- Passing an argument to a variable that is not a function
- Adding an integer and a list
- Performing the logical or between lists and trees
- Performing the logical and between functions
- ...

Type checking

- This verification is called type-checking
- We perform a fake execution of a term, making sure that all function applications are matching
- As we go, we replace parts of the term with its type
- When the whole term has been replaced with its type, we know the return type of the original program
- If the process gets stuck somewhere, we stop and give back an error[a]

[a] usually a compiler error

Type checking

Is a term well-typed

Type checking uses a series of typing rules that look a bit like execution rules.

Type checking

When we encounter a variable, we simply return its type from Γ, which contains all variable declarations found so far

$$\frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma}$$

Type checking

When we encounter a function declaration, we simply add the type of the parameter to Γ, and type check the function body

$$\frac{\Gamma, x : \sigma \vdash t : \tau}{\Gamma \vdash (\lambda x : \sigma \to t) : (\sigma \to \tau)}$$

Type checking

When we encounter a function application, we simply make sure that the input of the function has the same type as the parameter

$$\frac{\Gamma \vdash t : \sigma \to \tau \quad \Gamma \vdash u : \sigma}{\Gamma \vdash (t\ u) : \tau}$$

Type checking

Is a term well-typed

Thanks to these simple rules, which fully define how the type checker of a compiler works, we can find the types of a series of terms.

Type checking

Is a term well-typed

Let's begin with the type of `True`:

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ \ (\texttt{f} : \alpha) \rightarrow \texttt{t})$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t}:\alpha)\ \ (\mathtt{f}:\alpha) \rightarrow \mathtt{t})$$

$$\Lambda\alpha \Rightarrow \boxed{(\lambda(\mathtt{t}:\alpha)\ \ (\mathtt{f}:\alpha) \rightarrow \mathtt{t})}$$

Type checking

$\Lambda\alpha \Rightarrow$ `(λ(t:α) (f:α)→t)`

Type checking

$\Lambda\alpha \Rightarrow$ `(λ(t:α) (f:α)→t)`

$\Lambda\alpha \Rightarrow$ `(λ(t:α) (f:α)→` α `)`

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ \ (\texttt{f} : \alpha) \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha)\ \ (\texttt{f}:\alpha) \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha)\ (\texttt{f}:\alpha) \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \alpha\ )$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \alpha\ )$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ \ (\texttt{f} : \alpha) \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t} : \alpha)\ \ (\mathtt{f} : \alpha) \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ \ (\texttt{f} : \alpha) \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha) \rightarrow (\alpha \rightarrow \alpha))$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha) \rightarrow (\alpha \rightarrow \alpha))$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha) \rightarrow (\alpha \rightarrow \alpha))$$

$$\Lambda\alpha \Rightarrow \boxed{(\lambda(\texttt{t} : \alpha) \rightarrow (\alpha \rightarrow \alpha))}$$

Type checking

$\Lambda\alpha \Rightarrow$ `(λ(t:α)→(α→α))`

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha) \rightarrow (\alpha \rightarrow \alpha))$$

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

$$(\forall\alpha \ \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))$$

Type checking

$$(\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))$$

Type checking

$$(\forall\alpha \Rightarrow (\alpha \to \alpha \to \alpha))$$

$$(\forall\alpha \Rightarrow (\alpha \to \alpha \to \alpha))$$

Type checking

$$(\forall\alpha \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))$$

Type checking

$(\forall\alpha \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))$

`Boolean`

Type checking

Is a term well-typed

Let's then see the type of boolean `And`:

Type checking


```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) q) p)
   )
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) q) p)
    )
```

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) q) p))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) q) p))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) q) p))
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    q) Boolean))
```

Type checking


```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    q) Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    q) Boolean))
```

```
(λ(p:Boolean) (q:Boolean) →
    (((Boolean Boolean) q) Boolean) )
```

Type checking

```
(λ(p:Boolean) (q:Boolean) →
    (((Boolean Boolean) q) Boolean) )
```

Type checking

```
(λ(p:Boolean) (q:Boolean) →
    (((Boolean Boolean) q) Boolean) )
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean ) Boolean))
```

Type checking


```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

Type checking


```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
   Boolean) Boolean) Boolean))
```

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow ((((\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))\ \texttt{Boolean})\ \texttt{Boolean})\ \texttt{Boolean}))$$

Type checking

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    Boolean) Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    Boolean) Boolean) Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    Boolean) Boolean) Boolean))
```

Type checking


```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    Boolean) Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    (∀α ⇒(α→α→α)) ) Boolean) Boolean))
```

Type checking


```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    (∀α ⇒(α→α→α))) Boolean) Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    (∀α ⇒(α→α→α))) Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α)) (∀α ⇒(α→α→α))) Boolean)
    Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α)) (∀α ⇒(α→α→α))) Boolean)
    Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α)) (∀α ⇒(α→α→α))) Boolean)
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)))
    Boolean) Boolean))
```

Type checking


```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean)
    Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean)
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean )
    Boolean))
```

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow ((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}) \texttt{Boolean}))$$

Type checking

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean )
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)))
    (∀α ⇒(α→α→α)) ) Boolean))
```

Type checking


```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) (∀α ⇒(α→
    α→α))) Boolean))
```

Type checking


```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) (∀α ⇒(α→
    α→α))) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α
    Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→(
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)
    Boolean))
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→(
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(
    ((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean))
```

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}))$$

Type checking

```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(
    ∀α ⇒(α→α→α))) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(
    ∀α ⇒(α→α→α))) Boolean ))
```

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}))$$

Type checking


```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(
    ∀α ⇒(α→α→α))) Boolean ))
```

```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(
    ∀α ⇒(α→α→α))) (∀α ⇒(α→α→α)) ))
```

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))))$$

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow(((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))))$$

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow (((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha$$

Type checking

```
(λ(p:Boolean) (q:Boolean)→
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) (∀α ⇒(α→α→α
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) (∀α ⇒(α→α→α
```

```
(λ(p:Boolean) (q:Boolean)→ (∀α ⇒(α→α→α)) )
```

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (\forall\alpha\ \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)))$$

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))$$

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\ )$$

Type checking

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)))$$

Type checking

```
(λ(p:Boolean) (q:Boolean)→ (∀α ⇒(α→α→α)) )
```

```
(λ(p:Boolean) (q:Boolean)→ Boolean )
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→Boolean)
```

Type checking

```
(λ(p:Boolean) (q:Boolean)→Boolean)
```

```
(λ(p:Boolean) (q:Boolean) → Boolean )
```

Type checking


```
(λ(p:Boolean) (q:Boolean) → Boolean )
```

Type checking

```
(λ(p:Boolean) (q:Boolean) → Boolean )
```

```
(λ(p:Boolean) → (Boolean→Boolean) )
```

Type checking

```
(λ(p:Boolean)→(Boolean→Boolean))
```

Type checking

```
(λ(p:Boolean)→(Boolean→Boolean))
```

```
(λ(p:Boolean)→(Boolean→Boolean))
```

Type checking

```
(λ(p:Boolean)→(Boolean→Boolean))
```

Type checking

$$(\lambda(\texttt{p:Boolean}) \rightarrow (\texttt{Boolean} \rightarrow \texttt{Boolean}))$$

$$(\texttt{Boolean} \rightarrow \texttt{Boolean} \rightarrow \texttt{Boolean})$$

Type checking

Is a term well-typed

Similarly we can type check numbers:

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s} : (\alpha \to \alpha))\ \ (\mathtt{z} : \alpha) \to \mathtt{z})$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \ (\texttt{z}:\alpha) \to \texttt{z})$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \ (\texttt{z}:\alpha) \to \texttt{z})$$

Type checking

$\Lambda\alpha \Rightarrow$ `(λ(s:(α→α)) (z:α)→z)`

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha))\ (\mathtt{z}:\alpha) \to \mathtt{z})$$

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha))\ (\mathtt{z}:\alpha) \to \mathtt{z})$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha))\ \ (\mathtt{z}:\alpha) \to \mathtt{z})$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s} : (\alpha \to \alpha))\ \ (\mathtt{z} : \alpha) \to \mathtt{z})$$

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s} : (\alpha \to \alpha))\ (\mathtt{z} : \alpha) \to \mathtt{z})$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \texttt{(z:}\alpha\texttt{)} \to \texttt{z})$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to \texttt{z})$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \ (\texttt{z}:\alpha) \to \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha))\ \ (\texttt{z} : \alpha) \to \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \ (\texttt{z}:\alpha) \to \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha)) \to (\alpha \to \alpha))$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \rightarrow \alpha)) \rightarrow (\alpha \rightarrow \alpha))$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha)) \to (\alpha \to \alpha))$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha)) \to (\alpha \to \alpha))$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha)) \to (\alpha \to \alpha))$$

Type checking

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha)) \to (\alpha \to \alpha))$$

$$\Lambda\alpha \Rightarrow ((\alpha \to \alpha) \to \alpha \to \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

Type checking

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

$$(\forall\alpha \ \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha))$$

Type checking

$$(\forall\alpha \Rightarrow ((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

Type checking

$$(\forall\alpha \Rightarrow ((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

$$(\forall\alpha \Rightarrow ((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

Type checking

$$(\forall\alpha \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

Type checking

$$(\forall\alpha \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

`Nat`

Type inference

Type inference

Are type annotations necessary?

- Type annotations are not strictly necessary
- This means that, even if there are types, we might not need to write them all down
- Modern functional programming languages, like F#, Haskell, and Scala, are capable of guessing the types autonomously
- This is called type inference

Type inference

Are type annotations pointless?

- Type annotations are still useful at times
- This means that, even if there is type inference, sometimes we write the types explicitly
- Type inference might sometimes fail, so type annotations might solve the issue
- Type annotations act as documentation, and make complex code clearer

Type inference

Type annotations in practice

In practice you will see some type annotations, especially on global symbols and functions, and for smaller local values and anonymous functions we let the type inference do its job.

F# basic types

F# basic types

Types in F# are exactly the same as those in the lambda calculus:

- the empty type `Unit`, with its only value `()`
- primitive machine types, such as `int`, `bool`, `string`, etc.
- type variable, which instead of greek letters such as α have an apostrophe as prefix: `'a`
- function types: `t` $\rightarrow$ `t`
- tuples, in the form: `'a * 'b`
- discriminated unions, in the form: `Choice<'a,'b>`

F# basic types

Type declarations in F# are the same as those in the lambda calculus.

An identifier, a colon, and a type:

```
(fun (x:int) -> (x + 1))
```

F# basic types

The same applies to `let` bindings:

```
let (x:int) = 0
(x + 1)
```

F# basic types

Pairs (and tuples of size greater than two[a]) are similarly handled:

[a] Those can be matched, or extracted with `let (x,y,z,...) = t`

```
(fun (p:(int*bool)) -> ((fst p) + 1))
```

F# basic types

Discriminated unions are also no surprise:

```
(fun (p:Choice<int,bool>) ->
  match p with
  | Choice1Of2 x ->
     (x + 1)
  | Choice2Of2 y ->
     0
)
```

F# basic types

Of course nothing stops us from using function types:

```
let (f:(int -> int)) =
  fun (x:int) -> (x + 1)
(f 2)
```

F# basic types

F# type declarations

We can mix and match these types as much as we want: functions from tuples into function from discriminated unions to tuples of functions to ...

F# basic types

Demo

A short demo might be in order

Named types

Named types

We can use the `type` keyword to give a name to a type, for ease of later use:

```
type MyInt = int
let (f:(MyInt -> MyInt)) =
  fun (x:MyInt) -> (x + 1)
(f 2)
```

Named types

This makes the most sense when the name of the type is a tad longer[a]:

[a] Remember that a type might contain as many arrows, tuples, etc. as we might want!

```
type Int2 = (int -> int)
let (f:Int2) =
  fun (x:int) -> (x + 1)
(f 2)
```

Named types

Demo

Another short demo might be in order

Named types

Custom named types

- Sometimes existing type definitions are not clear or readable enough.
- For example, it is hard to guess that a long tuple like `string * string * int * int` actually represents a person
- Similarly, it is hard to guess that a long union like `Choice<Choice<int,int>, Choice<int, int>>` actually represents card and suit (hearts, diamonds, clubs, spades)

Named types

Custom named types

- The `type` keyword can define brand new types.
- These new types can have additional structure and internal names.
- The new types are substantially identical to (nested) tuples and (nested) unions.
- The only difference is that the elements of the tuple and the cases of the union can be augmented with a mnemonic name.

Named types

Records

- Tuples with named elements are called **records**.
- They are declared between curly brackets.
- Each field of the record has a name, followed by a colon and the type of the field.

Named types

An example record declaration for a person could look like:

```
type Person = { Name:string; Surname:string }
()
```

Named types

We can initialise a record with the values of the fields between curly brackets, each bound to the proper field name

```
type Person = { Name:string; Surname:string }
let (p:Person) = { Name = "Haskell"; Surname =
    "Curry" }
()
```

Named types

We can access a record fields with the usual dot notation coming from C-style languages:

```
type Person = { Name:string; Surname:string }
let (p:Person) = { Name = "Haskell"; Surname =
    "Curry" }
p.Name
```

Named types

We can modify large parts of a record fields with the `with` operator, which copies over all fields besides those explicitly specified:

```
type Person = { Name:string; Surname:string }
let (p:Person) = { Name = "Haskell"; Surname =
    "Curry" }
{ p with Name = "F\#" }
```

Named types

Demo

Yet another short demo might be in order

Named types

Records

- Unions with named cases are called **discriminated unions**.
- They are declared as a series of constructors, each with its own parameters.
- Matching is the only way to access the data inside the instance of the discriminated union.

Named types

An example record declaration for a widget size could look like:

```
type Size =
  | Small
  | Large
  | Custom of Nat
()
```

Named types

We can initialise a union with the values of a constructor between brackets, of no argument at all for parameterless constructors

```
type Size =
  | Small
  | Large
  | Custom of Nat
let (s1:Size) = Small
let (s2:Size) = (Custom 10)
()
```

Named types

We can access a union values with an extended pattern matching syntax:

```
type Size =
  | Small
  | Large
  | Custom of Nat
let (s1:Size) = Small
let (w:int) =
  match s1 with
  | Small -> 0
  | Large -> 10
  | Custom(c) -> c
()
```

Named types

Demo

Yet another short demo might be in order

Named types

A very important data structure in functional languages is the list.

In F# it is predefined[a] as union of the empty list and the non-empty constructor:

[a] So you do not need to define it yourself!

```
type List<'a> =
  | []
  | (::) of α*List<'a>
let (l1:List<int>) = []
let (l2:List<int>) =
  1 :: 2 :: 3 :: []
()
```

Named types

We could recursively perform all sorts of operations on lists with simple matching.

Fortunately, we can make use of many of the existing available functions on lists:

- `map`, which transforms all elements of a list according to a specific transformation function
- `filter`, which removes some elements of a list according to a specific predicate function
- `reduce`, which collapses a list onto a single value according to a specific aggregation function
- `fold`, which collapses a list onto a single value according to a specific aggregation function and a starting aggregate value

Named types

Demo

Yet another short demo might be in order

Named types

Lists are so important that they even get their own special syntax, called **list comprehensions**, to simplify their creation and manipulation.

Named types

Lists are so important that they even get their own special syntax, called **list comprehensions**, to simplify their creation and manipulation.

List comprehensions are substantially a way to map, filter, and cross lists easily.

Fold is excluded, and must be used for aggregate operations.

Named types

Demo

Yet another short demo might be in order

Conclusion

Conclusion

Recap

- The lambda calculus can be translated, type to type, into F#
- F# then adds handy primitive types, records, lists, discriminated unions, etc. to make it easier to build actual programs in practice

Appendix

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t} : \alpha)\ \ (\mathtt{f} : \alpha) \rightarrow \mathtt{f})$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t}:\alpha)\ \ (\mathtt{f}:\alpha) \rightarrow \mathtt{f})$$

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t}:\alpha)\ \ (\mathtt{f}:\alpha) \rightarrow \mathtt{f})$$

Appendix

$\Lambda\alpha \Rightarrow$ `(λ(t:α) (f:α)→f)`

Appendix

$\Lambda\alpha \Rightarrow$ `(λ(t:α) (f:α)→f)`

$\Lambda\alpha \Rightarrow$ `(λ(t:α) (f:α)→f)`

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t} : \alpha)\ \ (\mathtt{f} : \alpha) \rightarrow \mathtt{f})$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ \ (\texttt{f} : \alpha) \rightarrow \texttt{f})$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \texttt{f})$$

Appendix

$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \texttt{f})$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha)\ (\texttt{f}:\alpha) \rightarrow \texttt{f}\ )$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha)\ \ (\texttt{f}:\alpha) \rightarrow \alpha\ )$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t} : \alpha)\ \ (\mathtt{f} : \alpha) \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t} : \alpha)\ \ (\mathtt{f} : \alpha) \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{t} : \alpha)\ (\mathtt{f} : \alpha) \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha)\ (\texttt{f} : \alpha) \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha) \rightarrow (\alpha \rightarrow \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha) \rightarrow (\alpha \rightarrow \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha) \rightarrow (\alpha \rightarrow \alpha))$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t} : \alpha) \rightarrow (\alpha \rightarrow \alpha))$$

Appendix

$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha) \rightarrow (\alpha \rightarrow \alpha))$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{t}:\alpha) \rightarrow (\alpha \rightarrow \alpha))$$

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)$$

$$(\forall\alpha \ \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))$$

Appendix

$$(\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))$$

Appendix

$$(\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))$$

$$(\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))$$

Appendix

$$(\forall\alpha \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))$$

Appendix

$$(\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))$$

```
Boolean
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) p) q)
   )
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) p) q)
    )
```

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) p) q))
```

Appendix


```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) p) q))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((p Boolean) p) q))
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) q))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) q))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) q))
```

```
(λ(p:Boolean) (q:Boolean) →
    (((Boolean Boolean) Boolean) q) )
```

Appendix

```
(λ(p:Boolean) (q:Boolean) →
    (((Boolean Boolean) Boolean) q) )
```

Appendix

```
(λ(p:Boolean) (q:Boolean) →
    (((Boolean Boolean) Boolean) q) )
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean ))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((Boolean Boolean)
    Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))
   Boolean) Boolean) Boolean))
```

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow ((((\forall\alpha\ \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))\ \texttt{Boolean})\ \texttt{Boolean})\ \texttt{Boolean}))$$

Appendix

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    Boolean) Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    Boolean) Boolean) Boolean))
```

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow ((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\ \texttt{Boolean})\ \texttt{Boolean})\ \texttt{Boolean}))$$

Appendix

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    Boolean) Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    (∀α ⇒(α→α→α))) Boolean) Boolean))
```

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\ (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean})\ \texttt{Boolean}))$$

Appendix

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))
    (∀α ⇒(α→α→α))) Boolean) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α)) (∀α ⇒(α→α→α))) Boolean)
    Boolean))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α)) (∀α ⇒(α→α→α))) Boolean)
    Boolean))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α)) (∀α ⇒(α→α→α))) Boolean)
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((
    ((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)
    Boolean) Boolean))
```

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow ((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}) \texttt{Boolean}))$$

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean})\ \texttt{Boolean}))$$

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean})\ \texttt{Boolean}))$$

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}) \texttt{Boolean}))$$

Appendix

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean )
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)))
    (∀α ⇒(α→α→α)) ) Boolean))
```

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow ((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}))$$

Appendix

```
(λ(p:Boolean) (q:Boolean)→((((∀α ⇒(α→α→α))→
    (∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) (∀α ⇒(α→
    α→α))) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α
    Boolean))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α
    Boolean))
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→(
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))→(∀α ⇒(α→α→α)
    Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(
    ((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) Boolean))
```

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow(((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}))$$

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(
    ∀α ⇒(α→α→α))) Boolean))
```

```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(
    ∀α ⇒(α→α→α))) Boolean))
```

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow(((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}))$$

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow(((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ \texttt{Boolean}))$$

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow(((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))$$

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)) \rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))\ (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))))$$

Appendix

```
(λ(p:Boolean) (q:Boolean)→(((∀α ⇒(α→α→α))→(
    ∀α ⇒(α→α→α))) (∀α ⇒(α→α→α))))
```

```
(λ(p:Boolean) (q:Boolean)→
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) (∀α ⇒(α→α→α)→α
```

Appendix


```
(λ(p:Boolean) (q:Boolean)→
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) (∀α ⇒(α→α→α
```

Appendix

```
(λ(p:Boolean) (q:Boolean)→
    (((∀α ⇒(α→α→α))→(∀α ⇒(α→α→α))) (∀α ⇒(α→α→α
```

```
(λ(p:Boolean) (q:Boolean)→ (∀α ⇒(α→α→α)) )
```

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (\forall\alpha\ \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)))$$

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow(\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha)))$$

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean})\rightarrow (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\ )$$

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)))$$

Appendix

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow (\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha)))$$

$$(\lambda(\texttt{p:Boolean})\ (\texttt{q:Boolean}) \rightarrow \texttt{Boolean})$$

Appendix

```
(λ(p:Boolean) (q:Boolean)→Boolean)
```

Appendix


```
(λ(p:Boolean) (q:Boolean)→Boolean)
```

```
(λ(p:Boolean) (q:Boolean) → Boolean )
```

Appendix


```
(λ(p:Boolean) (q:Boolean) → Boolean )
```

Appendix

```
(λ(p:Boolean) (q:Boolean) → Boolean )
```

```
(λ(p:Boolean) → (Boolean→Boolean) )
```

Appendix

```
(λ(p:Boolean)→(Boolean→Boolean))
```

Appendix

```
(λ(p:Boolean)→(Boolean→Boolean))
```

```
(λ(p:Boolean)→(Boolean→Boolean))
```

Appendix


```
(λ(p:Boolean)→(Boolean→Boolean))
```

Appendix

$(\lambda(\texttt{p:Boolean}) \rightarrow (\texttt{Boolean} \rightarrow \texttt{Boolean}))$

$(\texttt{Boolean} \rightarrow \texttt{Boolean} \rightarrow \texttt{Boolean})$

Appendix

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β)→(((p β) th) el
    ))
```

Appendix

$$\Lambda\beta\Rightarrow(\lambda(\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta)\rightarrow(((\texttt{p}\ \beta)\ \texttt{th})\ \texttt{el}\ ))$$

$$\Lambda\beta\Rightarrow(\lambda(\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta)\rightarrow(((\texttt{p}\ \beta)\ \texttt{th})\ \texttt{el}))$$

Appendix

$\Lambda\beta\Rightarrow$ `(λ(p:Boolean) (th:β) (el:β)→(((p β) th) el))`

Appendix

$\Lambda\beta\Rightarrow$ `(λ(p:Boolean) (th:β) (el:β)→(((p β) th) el))`

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β)→(((Boolean β)
    th) el))
```

Appendix


```
Λβ⇒(λ(p:Boolean) (th:β) (el:β)→(((Boolean β)
    th) el))
```

Appendix

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β)→(((Boolean β)
    th) el))
```

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β) →
    (((Boolean β) th) el) )
```

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta) \rightarrow \texttt{(((Boolean } \beta\texttt{) th) el)})$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta) \rightarrow (((\texttt{Boolean}\ \beta)\ \texttt{th})\ \texttt{el}))$$

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta) \rightarrow (((\texttt{Boolean}\ \beta)\ \beta)\ \texttt{el}))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow (((\texttt{Boolean}\ \beta)\ \beta)\ \texttt{el}))$$

Appendix

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β)→(((Boolean β)
    β) el))
```

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β) →
    (((Boolean β) β) el) )
```

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta) \rightarrow (((\texttt{Boolean}\ \beta)\ \beta)\ \texttt{el}))$$

Appendix

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β) →
    (((Boolean β) β) el) )
```

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β)→(((Boolean β)
    β) β ))
```

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow (((\texttt{Boolean}\ \beta)\ \beta)\ \beta))$$

Appendix

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β)→(((Boolean β)
    β) β))
```

```
Λβ⇒(λ(p:Boolean) (th:β) (el:β)→(((Boolean β)
    β) β))
```

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow (((\texttt{Boolean}\ \beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta\Rightarrow(\lambda(\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta)\rightarrow(((\texttt{Boolean}\ \beta)\ \beta)\ \beta))$$

$$\Lambda\beta\Rightarrow(\lambda(\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta)\rightarrow(((\ (\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\ \beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p:Boolean})\ (\texttt{th}:\beta)\ (\texttt{el}:\beta) \rightarrow ((((\forall\alpha\ \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))\ \beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p:Boolean})\ (\texttt{th}:\beta)\ (\texttt{el}:\beta) \rightarrow ((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha \rightarrow\alpha))\ \beta)\ \beta)\ \beta))$$

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p:Boolean})\ (\texttt{th}:\beta)\ (\texttt{el}:\beta) \rightarrow ((((\forall\alpha\ \Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\ \beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow ((((\forall\alpha \Rightarrow (\alpha \rightarrow \alpha \rightarrow \alpha))\ \beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta\Rightarrow(\lambda(\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta)\rightarrow((((\forall\alpha\Rightarrow(\alpha\rightarrow\alpha\rightarrow\alpha))\ \beta)\ \beta)\ \beta))$$

$$\Lambda\beta\Rightarrow(\lambda(\texttt{p:Boolean})\ (\texttt{th:}\beta)\ (\texttt{el:}\beta)\rightarrow((\ (\beta\rightarrow\beta\rightarrow\beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p}:\texttt{Boolean})\ (\texttt{th}:\beta)\ (\texttt{el}:\beta) \rightarrow (((\beta \rightarrow \beta \rightarrow \beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow (((\beta \rightarrow \beta \rightarrow \beta)\ \beta)\ \beta))$$

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow (((\beta \rightarrow \beta \rightarrow \beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow (((\beta \rightarrow \beta \rightarrow \beta)\ \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow (((\beta \rightarrow \beta \rightarrow \beta)\ \beta)\ \beta))$$

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow ((\beta \rightarrow \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow ((\beta \rightarrow \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow ((\beta \rightarrow \beta)\ \beta))$$

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow ((\beta \rightarrow \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow ((\beta \rightarrow \beta)\ \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow ((\beta \rightarrow \beta)\ \beta))$$

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow \beta)$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow \beta)$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow \beta)$$

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow \beta)$$

Appendix

$\Lambda\beta \Rightarrow$ (λ(p : Boolean) (th : β) (el : β) → β)

Appendix

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta)\ (\texttt{el} : \beta) \rightarrow \beta\ )$$

$$\Lambda\beta \Rightarrow (\lambda (\texttt{p} : \texttt{Boolean})\ (\texttt{th} : \beta) \rightarrow (\beta \rightarrow \beta)\ )$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p}:\texttt{Boolean})\ \ (\texttt{th}:\beta) \rightarrow (\beta \rightarrow \beta))$$

Appendix

$$\Lambda\beta\Rightarrow(\lambda(\texttt{p:Boolean})\ \ (\texttt{th:}\beta)\rightarrow(\beta\rightarrow\beta))$$

$$\Lambda\beta\Rightarrow(\lambda(\texttt{p:Boolean})\ (\texttt{th:}\beta)\rightarrow(\beta\rightarrow\beta))$$

Appendix

$\Lambda\beta \Rightarrow$ (λ(p : Boolean) (th : β) → ($\beta \to \beta$))

Appendix

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p}:\texttt{Boolean})\ (\texttt{th}:\beta) \rightarrow (\beta \rightarrow \beta))$$

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p}:\texttt{Boolean}) \rightarrow (\beta \rightarrow \beta \rightarrow \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p}:\texttt{Boolean}) \rightarrow (\beta \rightarrow \beta \rightarrow \beta))$$

Appendix

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p}:\texttt{Boolean}) \rightarrow (\beta \rightarrow \beta \rightarrow \beta))$$

$$\Lambda\beta \Rightarrow (\lambda(\texttt{p}:\texttt{Boolean}) \rightarrow (\beta \rightarrow \beta \rightarrow \beta))$$

Appendix

$\Lambda\beta\Rightarrow$ `(λ(p:Boolean)→(β→β→β))`

Appendix

$\Lambda\beta \Rightarrow$ `(λ(p:Boolean)→(β→β→β))`

$\Lambda\beta \Rightarrow$ `(Boolean→β→β→β)`

Appendix

$$\Lambda\beta \Rightarrow (\texttt{Boolean} \rightarrow \beta \rightarrow \beta \rightarrow \beta)$$

Appendix

$$\Lambda\beta \Rightarrow (\texttt{Boolean} \rightarrow \beta \rightarrow \beta \rightarrow \beta)$$

$$\Lambda\beta \Rightarrow (\texttt{Boolean} \rightarrow \beta \rightarrow \beta \rightarrow \beta)$$

Appendix

$$\Lambda\beta \Rightarrow (\texttt{Boolean} \rightarrow \beta \rightarrow \beta \rightarrow \beta)$$

Appendix

$$\Lambda\beta \Rightarrow (\texttt{Boolean} \rightarrow \beta \rightarrow \beta \rightarrow \beta)$$

$$(\forall\beta \Rightarrow (\texttt{Boolean} \rightarrow \beta \rightarrow \beta \rightarrow \beta))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \ (\texttt{z}:\alpha) \to (\texttt{s}\ \ (\texttt{s}\ \ (\texttt{s}\ \ \texttt{z}))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to (\texttt{s}\ (\texttt{s}\ (\texttt{s}\ \texttt{z}))))$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to (\texttt{s}\ (\texttt{s}\ (\texttt{s}\ \texttt{z}))))$$

Appendix

$\Lambda\alpha \Rightarrow$ `(λ(s:(α→α)) (z:α)→(s (s (s z))))`

Appendix

$\Lambda\alpha \Rightarrow$ `(λ(s:(α→α)) (z:α)→(s (s (s z))))`

$\Lambda\alpha \Rightarrow$ `(λ(s:(α→α)) (z:α)→(` $(\alpha \to \alpha)$ `(` $(\alpha \to \alpha)$ `(` $(\alpha \to \alpha)$ `z))))`

Appendix

```
Λα⇒(λ(s:(α→α)) (z:α)→((α→α) ((α→α) ((α→α)
    z))))
```

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \texttt{z}))))$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \texttt{z}))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha\to\alpha)\ ((\alpha\to\alpha)\ ((\alpha\to\alpha)\ \texttt{z})))\ )$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \texttt{z})))\ )$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \alpha))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \alpha))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha))\ (\mathtt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \alpha))))$$

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha))\ (\mathtt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \alpha))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \alpha))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \alpha))))$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ ((\alpha \to \alpha)\ \alpha)))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha))\ \ (\texttt{z} : \alpha) \to ((\alpha \to \alpha)\ \ ((\alpha \to \alpha)\ \ \alpha)))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s} : (\alpha \to \alpha)) \ (\mathtt{z} : \alpha) \to ((\alpha \to \alpha) \ ((\alpha \to \alpha) \ \alpha)))$$

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s} : (\alpha \to \alpha)) \ (\mathtt{z} : \alpha) \to ((\alpha \to \alpha) \ \color{orange}{((\alpha \to \alpha) \ \alpha)}))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s} : (\alpha \to \alpha))\ \ (\mathtt{z} : \alpha) \to ((\alpha \to \alpha)\ \ ((\alpha \to \alpha)\ \ \alpha)))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s} : (\alpha \to \alpha)) \ (\mathtt{z} : \alpha) \to ((\alpha \to \alpha) \ ((\alpha \to \alpha) \ \alpha)))$$

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s} : (\alpha \to \alpha)) \ (\mathtt{z} : \alpha) \to ((\alpha \to \alpha) \ \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ \ \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ \ \alpha))$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ \ (\texttt{z}:\alpha) \to ((\alpha \to \alpha)\ \ \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha))\ \ (\texttt{z} : \alpha) \to ((\alpha \to \alpha)\ \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha))\ \ (\texttt{z} : \alpha) \to ((\alpha \to \alpha)\ \alpha))$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha))\ \ (\texttt{z} : \alpha) \to \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha))\ \ (\texttt{z} : \alpha) \to \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha))\ \ (\mathtt{z}:\alpha) \to \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha))\ (\mathtt{z}:\alpha) \to \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha))\ (\texttt{z}:\alpha) \to \alpha)$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s}:(\alpha \to \alpha)) \to (\alpha \to \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \rightarrow \alpha)) \rightarrow (\alpha \rightarrow \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha)) \to (\alpha \to \alpha))$$

$$\Lambda\alpha \Rightarrow (\lambda(\texttt{s} : (\alpha \to \alpha)) \to (\alpha \to \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha)) \to (\alpha \to \alpha))$$

Appendix

$$\Lambda\alpha \Rightarrow (\lambda(\mathtt{s}:(\alpha \to \alpha)) \to (\alpha \to \alpha))$$

$$\Lambda\alpha \Rightarrow ((\alpha \to \alpha) \to \alpha \to \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

Appendix

$$\Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)$$

$$(\forall\alpha \ \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha))$$

Appendix

$$(\forall\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha))$$

Appendix

$$(\forall\alpha \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

$$(\forall\alpha \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

Appendix

$$(\forall\alpha \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

Appendix

$$(\forall\alpha \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

`Nat`

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    m α) s) (((n α) s) z))))
```

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    m α) s) (((n α) s) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((m α) s) (
```

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((m α) s) (
```

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((m α) s) (
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) s) (((n α) s) z))))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) s) (((n α) s) z))))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) s) (((n α) s) z))))
```

```
(λ(m:Nat) (n:Nat) →
    Λα⇒(λ(s:(α→α)) (z:α)→(((Nat α) s) (((n α) s) z)))
```

Appendix


```
(λ(m:Nat) (n:Nat) →
    Λα⇒(λ(s:(α→α)) (z:α)→(((Nat α) s) (((n α) s) z)))
```

Appendix

```
(λ(m:Nat) (n:Nat) →
    Λα⇒(λ(s:(α→α)) (z:α)→(((Nat α) s) (((n α) s) z)))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) s) (((Nat α) s) z))))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) s) (((Nat α) s) z))))
```

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) s) (((Nat α) s) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒
    (λ(s:(α→α)) (z:α)→(((Nat α) s) (((Nat α) s) z))))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒
    (λ(s:(α→α)) (z:α)→(((Nat α) s) (((Nat α) s) z))) )
```

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒
    (λ(s:(α→α)) (z:α)→(((Nat α) s) (((Nat α) s) z))) )
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) (α→α)) (((Nat α) (α→α)) z))))
```

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\to\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\to\alpha))\ (\texttt{z}:\alpha)\to(((\texttt{Nat}\ \alpha)\ (\alpha\to\alpha))\ (((\texttt{Nat}\ \alpha)\ (\alpha\to\alpha))\ \texttt{z}))))$$

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) (α→α)) (((Nat α) (α→α)) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α) →
    (((Nat α) (α→α)) (((Nat α) (α→α)) z)) ))
```

Appendix

$$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat}) \to \Lambda\alpha \Rightarrow\ (\lambda(\texttt{s}:(\alpha\to\alpha))\ (\texttt{z}:\alpha) \to \\ (((\texttt{Nat}\ \alpha)\ (\alpha\to\alpha))\ (((\texttt{Nat}\ \alpha)\ (\alpha\to\alpha))\ \texttt{z}))\ ))$$

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α) →
    (((Nat α) (α→α)) (((Nat α) (α→α)) z)) ))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) (α→α)) (((Nat α) (α→α)) α))))
```

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow(((\texttt{Nat}\ \alpha)\ (\alpha\rightarrow\alpha))\ (((\texttt{Nat}\ \alpha)\ (\alpha\rightarrow\alpha))\ \alpha))))$$

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) (α→α)) (((Nat α) (α→α)) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) (α→α)) (((Nat α) (α→α)) α))))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) (α→α)) (((Nat α) (α→α)) α))))
```

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    Nat α) (α→α)) (((Nat α) (α→α)) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((
    (∀α ⇒((α→α)→α→α)) α) (α→α)) (((Nat α) (α
    →α)) α))))
```

Appendix

$$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat}) \to \Lambda\alpha \Rightarrow\ (\lambda(\texttt{s:}(\alpha \to \alpha))\ (\texttt{z:}\alpha) \to (((((\forall\alpha\ \Rightarrow((\alpha \to \alpha) \to \alpha \to \alpha))\ \alpha)\ (\alpha \to \alpha))\ (((\texttt{Nat}\ \alpha)\ (\alpha \to \alpha))\ \alpha))))$$

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→
    ((((∀α ⇒((α→α)→α→α)) α) (α→α)) (((Nat α
    ) (α→α)) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((
    ((∀α ⇒((α→α)→α→α)) α) (α→α)) (((Nat α) (
    α→α)) α))))
```

Appendix

$$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat}) \to \Lambda\alpha \Rightarrow\ (\lambda(\texttt{s}:(\alpha\to\alpha))\ (\texttt{z}:\alpha) \to ((((\forall\alpha \Rightarrow((\alpha\to\alpha)\to\alpha\to\alpha))\ \alpha)\ (\alpha\to\alpha))\ (((\texttt{Nat}\ \alpha)\ (\alpha\to\alpha))\ \alpha))))$$

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((
    ((∀α ⇒((α→α)→α→α)) α) (α→α)) (((Nat α) (
    α→α)) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((
    ((α→α)→α→α) (α→α)) (((Nat α) (α→α)) α))
    ))
```

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\to\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\to\alpha))\ (\texttt{z}:\alpha)\to (((((\alpha\to\alpha)\to\alpha\to\alpha)\ (\alpha\to\alpha))\ (((\texttt{Nat}\ \alpha)\ (\alpha\to\alpha))\ \alpha))))$$

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→
    ((((α→α)→α→α) (α→α)) (((Nat α) (α→α)) α
    ))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(
    (((α→α)→α→α) (α→α)) (((Nat α) (α→α)) α))
    ))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(
    (((α→α)→α→α) (α→α)) (((Nat α) (α→α)) α))
    ))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(
    (((α→α)→α→α) (α→α)) (((Nat α) (α→α)) α))
    ))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(
    (α→α) (((Nat α) (α→α)) α))))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) (((Nat α) (α→α)) α))))
```

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) (((Nat α) (α→α)) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) (((Nat α) (α→α)) α))))
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) (((Nat α) (α→α)) α))))
```

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) (((Nat α) (α→α)) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) (((∀α ⇒((α→α)→α→α)) α) (α→α)) α)))
    )
```

Appendix


```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) ((((∀α ⇒((α→α)→α→α)) α) (α→α)) α)))
    )
```

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\to\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\to\alpha))\ (\texttt{z}:\alpha)\to((\alpha\to\alpha)\ (((((\forall\alpha\ \Rightarrow((\alpha\to\alpha)\to\alpha\to\alpha))\ \alpha)\ (\alpha\to\alpha))\ \alpha)))$$

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\to\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\to\alpha))\ (\texttt{z}:\alpha)\to((\alpha\to\alpha)\ ((((\forall\alpha\ \Rightarrow((\alpha\to\alpha)\to\alpha\to\alpha))\ \alpha)\ (\alpha\to\alpha))\ \alpha))))$$

Appendix

$$(\lambda(\mathtt{m}:\mathtt{Nat})\ (\mathtt{n}:\mathtt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\mathtt{s}:(\alpha\rightarrow\alpha))\ (\mathtt{z}:\alpha)\rightarrow((\alpha \rightarrow\alpha)\ ((((\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))\ \alpha)\ (\alpha\rightarrow\alpha))\ \alpha))))$$

Appendix

$$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ ((((\forall\alpha\Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))\ \alpha)\ (\alpha\rightarrow\alpha))\ \alpha))))$$

$$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ ((((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)\ (\alpha\rightarrow\alpha))\ \alpha))))$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ (((((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)\ (\alpha\rightarrow\alpha))\ \alpha))))$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ (((((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)\ (\alpha\rightarrow\alpha))\ \alpha))))$$

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ (((((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)\ (\alpha\rightarrow\alpha))\ \alpha))))$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\to\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\to\alpha))\ (\texttt{z}:\alpha)\to((\alpha\to\alpha)\ (((((\alpha\to\alpha)\to\alpha\to\alpha)\ (\alpha\to\alpha))\ \alpha))))$$

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) ((((α→α)→α→α) (α→α)) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) ((α→α) α))))
```

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ ((\alpha\rightarrow\alpha)\ \alpha))))$$

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) ((α→α) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) ((α→α) α))))
```

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ ((\alpha\rightarrow\alpha)\ \alpha))))$$

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) ((α→α) α)))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α
    →α) α)))
```

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ \alpha)))$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow((\alpha\rightarrow\alpha)\ \alpha)))$$

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ (\texttt{z}:\alpha)\rightarrow ((\alpha\rightarrow\alpha)\ \alpha)))$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ (\texttt{n}:\texttt{Nat}) \rightarrow \Lambda\alpha \Rightarrow\ (\lambda(\texttt{s}:(\alpha \rightarrow \alpha))\ (\texttt{z}:\alpha) \rightarrow ((\alpha \rightarrow \alpha)\ \alpha)))$$

Appendix

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→
    ((α→α) α) ))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→ α )
    )
```

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ \ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ \ (\texttt{z}:\alpha)\rightarrow\alpha))$$

Appendix

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ \ (\texttt{z}:\alpha)\rightarrow\alpha))$$

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\ \ (\texttt{z}:\alpha)\rightarrow\alpha))$$

Appendix

$$(\lambda(\mathtt{m}:\mathtt{Nat})\ \ (\mathtt{n}:\mathtt{Nat}) \rightarrow \Lambda\alpha \Rightarrow\ \ (\lambda(\mathtt{s}:(\alpha \rightarrow \alpha))\ (\mathtt{z}:\alpha) \rightarrow \alpha))$$

Appendix

$$(\lambda(\mathtt{m}:\mathtt{Nat})\ \ (\mathtt{n}:\mathtt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\mathtt{s}:(\alpha\rightarrow\alpha))\ (\mathtt{z}:\alpha)\rightarrow\alpha))$$

$$(\lambda(\mathtt{m}:\mathtt{Nat})\ \ (\mathtt{n}:\mathtt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\mathtt{s}:(\alpha\rightarrow\alpha))\rightarrow(\alpha\rightarrow\alpha)))$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ \ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\rightarrow(\alpha\rightarrow\alpha)))$$

Appendix

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\rightarrow(\alpha\rightarrow\alpha)))$$

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\rightarrow(\alpha\rightarrow\alpha))\ )$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ \ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ (\lambda(\texttt{s}:(\alpha\rightarrow\alpha))\rightarrow(\alpha\rightarrow\alpha))\ )$$

Appendix

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat}) \rightarrow \Lambda\alpha \Rightarrow\ \ (\lambda(\texttt{s:}(\alpha \rightarrow \alpha)) \rightarrow (\alpha \rightarrow \alpha))\ )$$

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat}) \rightarrow \Lambda\alpha \Rightarrow\ \ ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)\ )$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ \ (\texttt{n}:\texttt{Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ ((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

Appendix

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ \ ((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))$$

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat})\rightarrow \Lambda\alpha\Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)\ )$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ \ (\texttt{n}:\texttt{Nat}) \rightarrow \Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)\ )$$

Appendix

$$(\lambda(\mathtt{m}:\mathtt{Nat})\ \ (\mathtt{n}:\mathtt{Nat}) \rightarrow \Lambda\alpha \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha))$$

$$(\lambda(\mathtt{m}:\mathtt{Nat})\ \ (\mathtt{n}:\mathtt{Nat}) \rightarrow (\forall\alpha\ \Rightarrow ((\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha)))$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\ \ (\texttt{n}:\texttt{Nat})\rightarrow(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)))$$

Appendix

$$(\lambda(\texttt{m:Nat})\ \ (\texttt{n:Nat}) \rightarrow (\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)))$$

$$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat}) \rightarrow (\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))\ )$$

Appendix

$$(\lambda(\texttt{m:Nat})\ \texttt{(n:Nat)} \rightarrow\ (\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))\ )$$

Appendix

$$(\lambda(\texttt{m:Nat})\ \texttt{(n:Nat)} \rightarrow\ (\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))\ )$$

$$(\lambda(\texttt{m:Nat})\rightarrow\ (\texttt{Nat}\rightarrow(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)))\ )$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\rightarrow(\texttt{Nat}\rightarrow(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))))$$

Appendix

$$(\lambda(\texttt{m}:\texttt{Nat})\rightarrow(\texttt{Nat}\rightarrow(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))))$$

$$(\lambda(\texttt{m}:\texttt{Nat})\rightarrow(\texttt{Nat}\rightarrow(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))))$$

Appendix

$$(\lambda(\texttt{m:Nat})\rightarrow(\texttt{Nat}\rightarrow(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))))$$

Appendix

$$(\lambda(\texttt{m:Nat})\rightarrow(\texttt{Nat}\rightarrow(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))))$$

$$(\texttt{Nat}\rightarrow\texttt{Nat}\rightarrow(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow ((\mathtt{p}\ \ \beta)\ \ (\lambda(\mathtt{x} : \alpha)\ \ (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((\mathtt{p}\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((\mathtt{p}\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow$ `(λ(p:(α×β))→((p β) (λ(x:α) (y:β)→y)))`

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((\mathtt{p}\ \beta)\ (\lambda(\mathtt{x}:\alpha)\ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((\ (\alpha \times \beta)\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \times \beta)\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y}))))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \times \beta)\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y}))))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((((\times\ \ \alpha)\ \ \beta)\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y}))))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow ((((\times\ \alpha)\ \beta)\ \beta)\ (\lambda(\mathtt{x} : \alpha)\ (\mathtt{y} : \beta) \rightarrow \mathtt{y}))))$$

Appendix

```
Λα⇒Λβ⇒(λ(p:(α×β))→((((( × α) β) β) (λ(x:α) (
    y:β)→y)))
```

```
Λα⇒Λβ⇒(λ(p:(α×β))→((((
    (∀α β γ ⇒((α→β→γ)→γ)))) α) β) β) (λ(x:α)
    (y:β)→y)))
```

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((((((\forall\alpha\ \ \beta\ \ \gamma\ \ \Rightarrow((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)))\ \ \alpha)\ \ \beta)\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow ((((((\forall\alpha\ \beta\ \gamma\ \Rightarrow((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)))) \alpha)\ \beta)\ \beta)\ (\lambda(\mathtt{x}:\alpha)\ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((((\forall\alpha\ \beta\ \gamma\ \Rightarrow((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)))) \alpha)\ \beta)\ \beta)\ (\lambda(\mathtt{x}:\alpha)\ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((((\forall\alpha\ \beta\ \gamma \Rightarrow ((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)))\ \alpha)\ \beta)\ \beta)\ (\lambda(\mathtt{x}:\alpha)(\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

```
Λα⇒Λβ⇒(λ(p:(α×β))→(((
    ((∀α β γ ⇒((α→β→γ)→γ)))) α) β) β) (λ(x:α)
    (y:β)→y)))
```

```
Λα⇒Λβ⇒(λ(p:(α×β))→(((∀β γ ⇒((α→β→γ)→γ)))
    β) β) (λ(x:α) (y:β)→y)))
```

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((((\forall\beta\ \ \gamma\ \ \Rightarrow((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)))\ \ \beta)\ \ \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

```
Λα⇒Λβ⇒(λ(p:(α×β))→((((∀β γ ⇒((α→β→γ)→γ)))
    β) β) (λ(x:α) (y:β)→y)))
```

```
Λα⇒Λβ⇒(λ(p:(α×β))→((
    ((∀β γ ⇒((α→β→γ)→γ))) β) β) (λ(x:α) (y:β)
    →y)))
```

Appendix

```
Λα⇒Λβ⇒(λ(p:(α×β))→((
    ((∀β γ ⇒((α→β→γ)→γ))) β) β) (λ(x:α) (y:β)
    →y)))
```

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow ((((\forall\beta\ \gamma \Rightarrow ((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)))\ \beta)\ \beta)\ (\lambda(\mathtt{x} : \alpha)\ (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow ((\forall\gamma \Rightarrow ((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma))\ \beta)\ (\lambda(\mathtt{x} : \alpha)\ (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\forall\gamma \Rightarrow ((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)) \ \beta) \ (\lambda(\mathtt{x}:\alpha) \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\forall\gamma \Rightarrow ((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)) \ \beta) \ (\lambda(\mathtt{x} : \alpha) \ (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\forall\gamma \Rightarrow ((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)) \ \beta) \ (\lambda(\mathtt{x} : \alpha) \ (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\forall\gamma \Rightarrow ((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)) \ \beta) \ (\lambda(\mathtt{x} : \alpha) \ (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\forall\gamma \Rightarrow ((\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow \gamma)) \ \beta) \ (\lambda(\mathtt{x} : \alpha) \ (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \ (\lambda(\mathtt{x} : \alpha) \ (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x} : \alpha)\ \ (\mathtt{y} : \beta) \rightarrow \mathtt{y}))))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)$$
$$(\lambda(\mathtt{x}:\alpha)\ (\mathtt{y}:\beta) \rightarrow \mathtt{y}))$$

Appendix


```
Λα⇒Λβ⇒(λ(p:(α×β))→(((α→β→β)→β)
      (λ(x:α) (y:β)→y)))
```

```
Λα⇒Λβ⇒(λ(p:(α×β))→(((α→β→β)→β) (λ(x:α) (y:
    β)→y)))
```

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y}))))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \quad (\lambda(\mathtt{x}:\alpha) \; (\mathtt{y}:\beta) \rightarrow \mathtt{y}\,)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \quad (\lambda(\mathtt{x} : \alpha) \quad (\mathtt{y} : \beta) \rightarrow \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \quad (\lambda(\mathtt{x} : \alpha) \quad (\mathtt{y} : \beta) \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ \ (\mathtt{y}:\beta) \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \quad (\lambda(\mathtt{x}:\alpha) \quad (\mathtt{y}:\beta) \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha)\ (\mathtt{y}:\beta) \rightarrow \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha) \rightarrow (\beta \rightarrow \beta))))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\texttt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\texttt{x} : \alpha) \rightarrow (\beta \rightarrow \beta))))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\lambda(\mathtt{x}:\alpha) \rightarrow (\beta \rightarrow \beta))))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ (\lambda(\mathtt{x}:\alpha) \rightarrow (\beta \rightarrow \beta))))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \; (\lambda(\mathtt{x}:\alpha) \rightarrow (\beta \rightarrow \beta))))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \; (\lambda(\mathtt{x}:\alpha) \rightarrow (\beta \rightarrow \beta))))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \; (\alpha \rightarrow \beta \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta) \quad (\alpha \rightarrow \beta \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\alpha \rightarrow \beta \rightarrow \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\alpha \rightarrow \beta \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\alpha \rightarrow \beta \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow (((\alpha \rightarrow \beta \rightarrow \beta) \rightarrow \beta)\ \ (\alpha \rightarrow \beta \rightarrow \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow \beta)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p} : (\alpha \times \beta)) \rightarrow \beta)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow \beta)$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow \boxed{(\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow \beta)}$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{p}:(\alpha \times \beta)) \rightarrow \beta)$$

Appendix

$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow$ `(λ(p:(α×β))→β)`

$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow$ `((α×β)→β)`

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow ((\alpha \times \beta) \rightarrow \beta)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow ((\alpha \times \beta) \rightarrow \beta)$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow ((\alpha \times \beta) \rightarrow \beta)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow ((\alpha \times \beta) \rightarrow \beta)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow ((\alpha \times \beta) \rightarrow \beta)$$

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow ((\alpha \times \beta) \rightarrow \beta))$$

Appendix

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow ((\alpha \times \beta) \rightarrow \beta))$$

Appendix

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow ((\alpha \times \beta) \rightarrow \beta))$$

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow ((\alpha \times \beta) \rightarrow \beta))$$

Appendix

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow ((\alpha \times \beta) \rightarrow \beta))$$

Appendix

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow ((\alpha \times \beta) \rightarrow \beta))$$

$$(\forall\alpha \ \beta \ \Rightarrow ((\alpha \times \beta) \rightarrow \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f} : (\alpha \rightarrow \gamma)) \quad (\mathtt{g} : (\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \mathtt{y})))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma))\ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \ \mathtt{y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow$$
$$(\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma))\ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \ \mathtt{y})))$$

Appendix

$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow$

$(\lambda(\texttt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow\ (\lambda(\texttt{f}:(\alpha \rightarrow \gamma))\ (\texttt{g}:(\beta \rightarrow \gamma)) \rightarrow (\texttt{g}\ \texttt{y})))$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g\ y})))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g} \ \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g} \ \beta)) \ )$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow$$
$$(\lambda(\mathtt{f}:(\alpha \rightarrow \gamma))\ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \beta))\ )$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma))\ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \beta))\ )$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow\ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma))\ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g}\ \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \quad (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\mathtt{g} \quad \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \; (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \quad (\mathtt{g} \;\; \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \; (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \; (\mathtt{g} \;\; \beta)\,))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \; (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \quad (\mathtt{g} \;\; \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \quad (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow (\quad (\beta \rightarrow \gamma) \quad \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \ \ \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \ \ \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \ \ \beta))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \ \ (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \ \ \beta)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \quad (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \quad ((\beta \rightarrow \gamma) \quad \beta)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \quad (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \quad \gamma))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow \ (\lambda(\mathtt{f} : (\alpha \rightarrow \gamma)) \ \ (\mathtt{g} : (\beta \rightarrow \gamma)) \rightarrow \gamma)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \quad (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \gamma)$$
$$)$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \; (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \gamma)$$
$$)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \; (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \gamma) \quad)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \; (\mathtt{g}:(\beta \rightarrow \gamma)) \rightarrow \gamma) \quad)$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f} : (\alpha \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \rightarrow \gamma)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f}:(\alpha \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \rightarrow \gamma)) \;)$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow \quad (\lambda(\mathtt{f} : (\alpha \rightarrow \gamma)) \rightarrow ((\beta \rightarrow \gamma) \rightarrow \gamma)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow \quad ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow \quad ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow \quad ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow \Lambda\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y} : \beta) \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\lambda(\mathtt{y}:\beta) \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\beta \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\beta \rightarrow (\forall\gamma \ \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\beta \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\beta \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\beta \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

Appendix

$$\Lambda\alpha \Rightarrow \Lambda\beta \Rightarrow (\beta \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma)))$$

$$\Lambda\alpha \Rightarrow (\forall\beta \Rightarrow (\beta \rightarrow (\forall\gamma \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow (\beta \rightarrow (\forall\gamma \ \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow (\beta \rightarrow (\forall\gamma \ \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))))$$

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow (\beta \rightarrow (\forall\gamma \ \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow (\beta \rightarrow (\forall\gamma \ \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))))$$

Appendix

$$\Lambda\alpha \Rightarrow (\forall\beta \ \Rightarrow (\beta \rightarrow (\forall\gamma \ \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))))$$

$$(\forall\alpha \ \beta \ \Rightarrow (\beta \rightarrow (\forall\gamma \ \Rightarrow ((\alpha \rightarrow \gamma) \rightarrow (\beta \rightarrow \gamma) \rightarrow \gamma))))$$

This is it!

The best of luck, and thanks for the attention!