Curry-Howard/$\lambda 2$

Adrián Enríquez Ballester

January 9, 2022

The Curry-Howard isomorphism

Exercise 1

Provide a λ-term equivalent to a proof in NJ of the following:

$$p \to (q \to r) \to ((p \to q) \to (p \to r))$$

Answer

The term can be

$$\lambda x.\lambda f.\lambda g.\lambda y.f\ (g\ x)$$

and the proof tree for its proposed type is the following:

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$f : p \to r \vdash f : p \to r$}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$g : p \to q \vdash g : p \to q$}
\AxiomC{}
\RightLabel{VAR}
\UnaryInfC{$x : p \vdash x : p$}
\RightLabel{APP}
\BinaryInfC{$g : p \to q, x : p \vdash g\ x : p$}
\RightLabel{APP}
\BinaryInfC{$x : p, f : q \to r, g : p \to q, y : p \vdash f\ (g\ x) : r$}
\RightLabel{ABS}
\UnaryInfC{$x : p, f : q \to r, g : p \to q \vdash \lambda y.f\ (g\ x) : p \to r$}
\RightLabel{ABS}
\UnaryInfC{$x : p, f : q \to r \vdash \lambda g.\lambda y.f\ (g\ x) : (p \to q) \to (p \to r)$}
\RightLabel{ABS}
\UnaryInfC{$x : p \vdash \lambda f.\lambda g.\lambda y.f\ (g\ x) : (q \to r) \to ((p \to q) \to (p \to r))$}
\RightLabel{ABS}
\UnaryInfC{$\vdash \lambda x.\lambda f.\lambda g.\lambda y.f\ (g\ x) : p \to (q \to r) \to ((p \to q) \to (p \to r))$}
\end{prooftree}
$$

Here is also its proof in NJ, in order to show their analogy:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[q \to r]^b \qquad \dfrac{[p \to q]^c \qquad [p]^a}{q}\to e}{r}\to e}{p \to r}\to i^d}{(p \to q) \to (p \to r)}\to i^c}{(q \to r) \to ((p \to q) \to (p \to r))}\to i^b}{p \to (q \to r) \to ((p \to q) \to (p \to r))}\to i^a
$$

This can be checked in the Haskell REPL, for example, by declaring the term with the type specified and observing that it is accepted without no complain:

```
Prelude> :{
  Prelude| term :: p -> (q -> r) -> ((p -> q) -> (p -> r))
  Prelude| term = \x f g y -> f (g x)
  Prelude| :}
Prelude>
```

Exercise 2

Extend the Curry-Howard isomorphism to consider the conjunction operator ($\wedge$). Define a translation of proofs in the subset of NJ with implication and conjunction into extended λ-terms and provide rules for β-reduction. Justify your decisions.

Answer

On the one hand, we add the following to the syntax of lambda terms:

$$
\begin{aligned}
&\vdots \\
&\mid \langle T, T' \rangle \\
&\mid \pi^1\, T \\
&\mid \pi^2\, T
\end{aligned}
$$

which correspond to a pair term, and the first and second component projections. The following β-reduction rules are also added so we can effectively use these projections:

$$
\begin{aligned}
\pi^1\, \langle T, T' \rangle &\rightsquigarrow_\beta T \\
\pi^2\, \langle T, T' \rangle &\rightsquigarrow_\beta T'
\end{aligned}
$$

On the other hand, we add the following to the syntax of types:

$$\begin{array}{l} \vdots \\ |\tau \times \tau' \end{array}$$

which corresponds to a product type and is intended to be the type for pair terms. We define the following typing rules for these new terms:

$$\frac{\Gamma \vdash T : \tau \qquad \Gamma' \vdash T' : \tau'}{\Gamma \cup \Gamma' \vdash \langle T, T' \rangle : \tau \times \tau'} \text{ PROD}$$

$$\frac{\Gamma \vdash T : \tau \times \tau'}{\Gamma \vdash \pi^1\, T : \tau} \text{ PROJ1} \qquad\qquad \frac{\Gamma \vdash T : \tau \times \tau'}{\Gamma \vdash \pi^2\, T : \tau'} \text{ PROJ2}$$

Finally, the traduction of proofs in the subset of NJ to terms taking into account the conjunction operator adds the following to the traduction that we already have:

$$\frac{p \qquad q}{p \wedge q} \wedge i \qquad \Longrightarrow \qquad \langle P, Q \rangle$$

$$\frac{p \wedge q}{p} \wedge e_1 \qquad \Longrightarrow \qquad \pi^1\, T$$

$$\frac{p \wedge q}{q} \wedge e_2 \qquad \Longrightarrow \qquad \pi^2\, T$$

where P is the term corresponding to the proof of p, Q is the term corresponding to the proof of q and T is the term corresponding to the proof of $p \wedge q$.

Exercise 3

Provide a type λ-term equivalent to a proof in NJ of the following:

$$(p \wedge q \wedge r) \rightarrow (p \wedge r)$$

Answer

The required term can be

$$\lambda x.\langle \pi^1\ x, \pi^2\ (\pi^2\ x)\rangle$$

and the proof tree for its proposed type is the following:

$$\dfrac{\dfrac{\dfrac{}{x : p \times q \times r \vdash x : p \times q \times r}\ \text{VAR}}{x : p \times q \times r \vdash \pi^1\ x : p}\ \text{PROJ1} \qquad \dfrac{\dfrac{\dfrac{}{x : p \times q \times r \vdash x : p \times q \times r}\ \text{VAR}}{x : p \times q \times r \vdash \pi^2\ x : q \times r}\ \text{PROJ2}}{x : p \times q \times r \vdash \pi^2\ (\pi^2\ x) : r}\ \text{PROJ2}}{\dfrac{x : p \times q \times r \vdash \langle \pi^1\ x, \pi^2\ (\pi^2\ x)\rangle : p \times r}{\vdash \lambda x.\langle \pi^1\ x, \pi^2\ (\pi^2\ x)\rangle : (p \times q \times r) \to (p \times r)}\ \text{ABS}}\ \text{PROD}$$

As in the Exercise 1 , we can check this with the Haskell REPL by declaring the term with this specific type signature and observing that it is accepted with no complain:

```
Prelude> :{
Prelude| term :: (p, (q, r)) -> (p, r)
Prelude| term = \x -> (fst x, snd (snd x))
Prelude| :}
Prelude>
```

Exercise 4

Extend the Curry-Howard isomorphism to support disjunction ($\vee$) analogously to the Exercise 2 .

Answer

On the one hand, we add the following to the syntax of lambda terms:

$$\begin{aligned} &\vdots \\ &\mid \iota^1\ T \\ &\mid \iota^2\ T \\ &\mid \delta\ T_1\ T_2\ T \end{aligned}$$

which correspond to a left and right sum term, and its eliminator. The following β-reduction rules are also added so we can effectively use the sum eliminator:

$$\delta\ T_1\ T_2\ (\iota^1\ T) \rightsquigarrow_\beta T_1\ T$$
$$\delta\ T_1\ T_2\ (\iota^2\ T) \rightsquigarrow_\beta T_2\ T$$

On the other hand, we add the following to the syntax of types:

$$\begin{array}{l} \vdots \\ |\tau + \tau' \end{array}$$

which corresponds to a sum type and is intended to be the type for sum terms. We define the following typing rules for these new terms:

$$\frac{\Gamma_1 \vdash T_1 : \tau \to \alpha \qquad \Gamma_2 \vdash T_2 : \tau' \to \alpha \qquad \Gamma \vdash T : \tau + \tau'}{\Gamma_1 \cup \Gamma_2 \cup \Gamma \vdash \delta\ T_1\ T_2\ T : \alpha}\ \text{UNSUM}$$

$$\frac{\Gamma \vdash T : \tau}{\Gamma \vdash \iota^1\ T : \tau + \tau'}\ \text{SUML} \qquad\qquad \frac{\Gamma \vdash T : \tau'}{\Gamma \vdash \iota^2\ T : \tau + \tau'}\ \text{SUMR}$$

Finally, the traduction of proofs to terms in the subset of NJ taking into account the disjunction operator adds the following to the traduction that we already have:

$$\frac{p \vee q \qquad p \to e \qquad q \to e}{r}\ \vee e \quad \Longrightarrow \quad \delta\ E_p\ E_q\ T$$

$$\frac{p}{p \vee q}\ \vee i_1 \quad \Longrightarrow \quad \iota^1\ P$$

$$\frac{q}{p \vee q}\ \vee i_2 \quad \Longrightarrow \quad \iota^2\ Q$$

where P is the term corresponding to the proof of p, Q is the term corresponding to the proof of q, T is the term corresponding to the proof of $p \vee q$, E_p is the proof of $p \to e$ and E_q is the proof of $q \to e$.

Exercise 5

Provide a λ-term equivalent to a proof in NJ of:

$$p \to (p \land (p \lor q))$$

Answer

The term can be

$$\lambda x.\langle x, \iota^1\ x\rangle$$

and the proof tree for its proposed type is the following:

$$\dfrac{\dfrac{}{x : p \vdash x : p}\ \text{VAR} \qquad \dfrac{\dfrac{}{x : p \vdash x : p}\ \text{VAR}}{\iota^1\ x : p + q}\ \text{SUML}}{\dfrac{x : p \vdash \langle x, \iota^1\ x\rangle : p \times (p + q)}{\vdash \lambda x.\langle x, \iota^1\ x\rangle : p \to (p \times (p + q))}\ \text{ABS}}\ \text{PROD}$$

Again, we can check this in the Haskell REPL by declaring this therm together with its type signature, and observing that it is accepted:

```
Prelude> :{
Prelude| term :: p -> (p, Either p q)
Prelude| term = \x -> (x, Left x)
Prelude| :}
Prelude>
```

The polymorphic λ-calculus

Exercise 6

Provide a polymorphic definition of function composition (Exercise 10 in the previous exercise sheet) in $\lambda 2$. Use it to provide an alternate *cheap* definition for the Church numerals $\mathsf{2}, \mathsf{3}$, and $\mathsf{4}$. Verify your answer via reduction.

Answer

The term is

$$\mathsf{COMP} \triangleq \Lambda a.\Lambda b.\Lambda c.\lambda f^{b \to c}.\lambda g^{a \to b}.\lambda x^a.f\ (g\ x)$$

and its proof tree is as follows:

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$f : b \to c \vdash f : b \to c$}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$g : a \to b \vdash g : a \to b$}
\AxiomC{}
\RightLabel{VAR}
\UnaryInfC{$x : a \vdash x : a$}
\RightLabel{APP}
\BinaryInfC{$g : a \to b, x : a \vdash g\ x : b$}
\RightLabel{APP}
\BinaryInfC{$f : b \to c, g : a \to b, x : a \vdash f\ (g\ x) : c$}
\RightLabel{ABS}
\UnaryInfC{$f : b \to c, g : a \to b \vdash \lambda x^a.f\ (g\ x) : a \to c$}
\RightLabel{ABS}
\UnaryInfC{$f : b \to c \vdash \lambda g^{a \to b}.\lambda x^a.f\ (g\ x) : (a \to b) \to (a \to c)$}
\RightLabel{ABS}
\UnaryInfC{$\vdash \lambda f^{b \to c}.\lambda g^{a \to b}.\lambda x^a.f\ (g\ x) : (b \to c) \to (a \to b) \to (a \to c)$}
\RightLabel{$\forall i$}
\UnaryInfC{$\vdash \Lambda c.\lambda f^{b \to c}.\lambda g^{a \to b}.\lambda x^a.f\ (g\ x) : \forall c.(b \to c) \to (a \to b) \to (a \to c)$}
\RightLabel{$\forall i$}
\UnaryInfC{$\vdash \Lambda b.\Lambda c.\lambda f^{b \to c}.\lambda g^{a \to b}.\lambda x^a.f\ (g\ x) : \forall b.\forall c.(b \to c) \to (a \to b) \to (a \to c)$}
\RightLabel{$\forall i$}
\UnaryInfC{$\vdash \Lambda a.\Lambda b.\Lambda c.\lambda f^{b \to c}.\lambda g^{a \to b}.\lambda x^a.f\ (g\ x) : \forall a.\forall b.\forall c.(b \to c) \to (a \to b) \to (a \to c)$}
\end{prooftree}
$$

By having $\mathsf{1}$ defined as $\Lambda t.\lambda f^{t \to t}.\lambda x^t.f\ x$, we can, for example, define the following numbers as

$$
\begin{aligned}
\mathsf{2} &\triangleq \Lambda t.\lambda f^{t \to t}.\mathsf{COMP}[t][t][t]\ (\mathsf{1}[t]\ f)\ (\mathsf{1}[t]\ f) \\
\mathsf{3} &\triangleq \Lambda t.\lambda f^{t \to t}.\mathsf{COMP}[t][t][t]\ (\mathsf{1}[t]\ f)\ (\mathsf{2}[t]\ f) \\
\mathsf{4} &\triangleq \Lambda t.\lambda f^{t \to t}.\mathsf{COMP}[t][t][t]\ (\mathsf{2}[t]\ f)\ (\mathsf{2}[t]\ f)
\end{aligned}
$$

First, this shows how the defined term $\mathsf{2}$ reduces as expected:

$$
\begin{aligned}
\mathsf{2} \equiv\ &\Lambda t.\lambda f^{t \to t}.\mathsf{COMP}[t][t][t]\ (\mathsf{1}[t]\ f)\ (\mathsf{1}[t]\ f) \\
\equiv\ &\Lambda t.\lambda f^{t \to t}. \\
&\qquad (\Lambda a.\Lambda b.\Lambda c.\lambda f^{b \to c}.\lambda g^{a \to b}.\lambda x^a.f\ (g\ x))[t][t][t] \\
&\qquad ((\Lambda t.\lambda f^{t \to t}.\lambda x^t.f\ x)[t]\ f) \\
&\qquad ((\Lambda t.\lambda f^{t \to t}.\lambda x^t.f\ x)[t]\ f) \\
\rightsquigarrow^5\ &\Lambda t.\lambda f^{t \to t}. \\
&\qquad (\lambda f^{t \to t}.\lambda g^{t \to t}.\lambda x^t.f\ (g\ x)) \\
&\qquad ((\lambda f^{t \to t}.\lambda x^t.f\ x)\ f) \\
&\qquad ((\lambda f^{t \to t}.\lambda x^t.f\ x)\ f) \\
\rightsquigarrow^2\ &\Lambda t.\lambda f^{t \to t}.(\lambda f^{t \to t}.\lambda g^{t \to t}.\lambda x^t.f\ (g\ x))\ \ (\lambda x^t.f\ x)\ (\lambda x^t.f\ x) \\
\rightsquigarrow^2\ &\Lambda t.\lambda f^{t \to t}.\lambda x^t.(\lambda x^t.f\ x)\ ((\lambda x^t.f\ x)\ x) \\
\rightsquigarrow\ &\Lambda t.\lambda f^{t \to t}.\lambda x^t.(\lambda x^t.f\ x)\ (f\ x) \\
\rightsquigarrow\ &\Lambda t.\lambda f^{t \to t}.\lambda x^t.f\ (f\ x)
\end{aligned}
$$

Second, this shows how the defined $\mathsf{3}$ reduces to the expected term by also using the previous reduction for $\mathsf{2}$:

$$
\begin{aligned}
\mathsf{3} \equiv\, & \Lambda t.\lambda f^{t\to t}.\mathsf{COMP}[t][t][t]\ (\mathsf{1}[t]\ f)\ (\mathsf{2}[t]\ f) \\
\equiv\, & \Lambda t.\lambda f^{t\to t}. \\
& \qquad (\Lambda a.\Lambda b.\Lambda c.\lambda f^{b\to c}.\lambda g^{a\to b}.\lambda x^{a}.f\ (g\ x))[t][t][t] \\
& \qquad ((\Lambda t.\lambda f^{t\to t}.\lambda x^{t}.f\ x)[t]\ f) \\
& \qquad (\mathsf{2}[t]\ f) \\
\rightsquigarrow^{*} & \Lambda t.\lambda f^{t\to t}. \\
& \qquad (\Lambda a.\Lambda b.\Lambda c.\lambda f^{b\to c}.\lambda g^{a\to b}.\lambda x^{a}.f\ (g\ x))[t][t][t] \\
& \qquad ((\Lambda t.\lambda f^{t\to t}.\lambda x^{t}.f\ x)[t]\ f) \\
& \qquad ((\Lambda t.\lambda f^{t\to t}.\lambda x^{t}.f\ (f\ x))[t]\ f) \\
\rightsquigarrow^{5} & \Lambda t.\lambda f^{t\to t}. \\
& \qquad (\lambda f^{t\to t}.\lambda g^{t\to t}.\lambda x^{t}.f\ (g\ x)) \\
& \qquad ((\lambda f^{t\to t}.\lambda x^{t}.f\ x)\ f) \\
& \qquad ((\lambda f^{t\to t}.\lambda x^{t}.f\ (f\ x))\ f) \\
\rightsquigarrow^{2} & \Lambda t.\lambda f^{t\to t}.(\lambda f^{t\to t}.\lambda g^{t\to t}.\lambda x^{t}.f\ (g\ x))\ \ (\lambda x^{t}.f\ x)\ (\lambda x^{t}.f\ (f\ x)) \\
\rightsquigarrow^{2} & \Lambda t.\lambda f^{t\to t}.\lambda x^{t}.(\lambda x^{t}.f\ x)\ ((\lambda x^{t}.f\ (f\ x))\ x) \\
\rightsquigarrow\, & \Lambda t.\lambda f^{t\to t}.\lambda x^{t}.(\lambda x^{t}.f\ x)\ (f\ (f\ x)) \\
\rightsquigarrow\, & \Lambda t.\lambda f^{t\to t}.\lambda x^{t}.f\ (f\ (f\ x))
\end{aligned}
$$

Finally, this shows how the defined $\mathsf{4}$ also reduces to the expected term by also using the previous reduction for $\mathsf{2}$:

$$
\begin{aligned}
\mathsf{4} \equiv\, & \Lambda t.\lambda f^{t\to t}.\mathsf{COMP}[t][t][t]\ (\mathsf{2}[t]\ f)\ (\mathsf{2}[t]\ f) \\
\equiv\, & \Lambda t.\lambda f^{t\to t}. \\
& \qquad (\Lambda a.\Lambda b.\Lambda c.\lambda f^{b\to c}.\lambda g^{a\to b}.\lambda x^{a}.f\ (g\ x))[t][t][t] \\
& \qquad (\mathsf{2}[t]\ f) \\
& \qquad (\mathsf{2}[t]\ f) \\
\rightsquigarrow^{*} & \Lambda t.\lambda f^{t\to t}. \\
& \qquad (\Lambda a.\Lambda b.\Lambda c.\lambda f^{b\to c}.\lambda g^{a\to b}.\lambda x^{a}.f\ (g\ x))[t][t][t] \\
& \qquad ((\Lambda t.\lambda f^{t\to t}.\lambda x^{t}.f\ (f\ x))[t]\ f) \\
& \qquad ((\Lambda t.\lambda f^{t\to t}.\lambda x^{t}.f\ (f\ x))[t]\ f) \\
\rightsquigarrow^{5} & \Lambda t.\lambda f^{t\to t}. \\
& \qquad (\lambda f^{t\to t}.\lambda g^{t\to t}.\lambda x^{t}.f\ (g\ x))
\end{aligned}
$$

$$
\begin{aligned}
&((\lambda f^{t\to t}.\lambda x^t.f\ (f\ x))\ f)\\
&((\lambda f^{t\to t}.\lambda x^t.f\ (f\ x))\ f)\\
\rightsquigarrow^2&\Lambda t.\lambda f^{t\to t}.(\lambda f^{t\to t}.\lambda g^{t\to t}.\lambda x^t.f\ (g\ x))\ \ (\lambda x^t.f\ (f\ x))\ (\lambda x^t.f\ (f\ x))\\
\rightsquigarrow^2&\Lambda t.\lambda f^{t\to t}.\lambda x^t.(\lambda x^t.f\ (f\ x))\ ((\lambda x^t.f\ (f\ x))\ x)\\
\rightsquigarrow&\Lambda t.\lambda f^{t\to t}.\lambda x^t.(\lambda x^t.f\ (f\ x))\ (f\ (f\ x))\\
\rightsquigarrow&\Lambda t.\lambda f^{t\to t}.\lambda x^t.f\ (f\ (f\ (f\ x)))
\end{aligned}
$$

Exercise 7

Revisit Church booleans in $\lambda 2$ (Exercise 5 of the previous sheet). Use reduction to check the correctness of your definitions (i.e. of the type *bool* and the terms $\mathsf{TRUE}, \mathsf{FALSE}, \mathsf{NEG}, \mathsf{CONJ}$ and DISJ).

Answer

The type *bool* is defined as

$$bool = \forall t.t \to t \to t$$

and the terms TRUE and FALSE as

$$
\begin{aligned}
\mathsf{TRUE} &\triangleq \Lambda t.\lambda x^t.\lambda y^t.x\\
\mathsf{FALSE} &\triangleq \Lambda t.\lambda x^t.\lambda y^t.y
\end{aligned}
$$

whose proof for its *bool* type assignment is the following, from left to right:

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{x : t, y : t \vdash x : t}\ \text{VAR}}{x : t \vdash \lambda y^t.x : t \to t}\ \text{ABS}}{\vdash \lambda x^t.\lambda y^t.x : t \to t \to t}\ \text{ABS}}{\vdash \Lambda t.\lambda x^t.\lambda y^t.x : \forall t.t \to t \to t}\ \forall i
\qquad\qquad
\dfrac{\dfrac{\dfrac{\dfrac{}{x : t, y : t \vdash y : t}\ \text{VAR}}{x : t \vdash \lambda y^t.y : t \to t}\ \text{ABS}}{\vdash \lambda x^t.\lambda y^t.y : t \to t \to t}\ \text{ABS}}{\vdash \Lambda t.\lambda x^t.\lambda y^t.y : \forall t.t \to t \to t}\ \forall i
$$

The term NOT can be defined as

$$\lambda b^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ y\ x$$

On the one hand, its proof to be assigned type $bool \to bool$ is as follows:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{b : \forall t.t \to t \to t \vdash b : \forall t.t \to t \to t}\,\text{VAR}}{b : \forall t.t \to t \to t \vdash b[t] : t \to t \to t}\,\forall e \qquad \dfrac{}{y : t \vdash y : t}\,\text{VAR}}{b : \forall t.t \to t \to t, y : t \vdash b[t]\ y : t \to t}\,\text{APP} \qquad \dfrac{}{x : t \vdash x : t}\,\text{VAR}}{b : \forall t.t \to t \to t, x : t, y : t \vdash b[t]\ y\ x : t}\,\text{APP}}{b : \forall t.t \to t \to t, x : t \vdash \lambda y^t.b[t]\ y\ x : t \to t}\,\text{ABS}}{b : \forall t.t \to t \to t \vdash \lambda x^t.\lambda y^t.b[t]\ y\ x : t \to t \to t}\,\text{ABS}}{b : \forall t.t \to t \to t \vdash \Lambda t.\lambda x^t.\lambda y^t.b[t]\ y\ x : \forall t.t \to t \to t}\,\forall i}{\vdash \lambda b^{\forall t.t \to t \to t}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ y\ x : (\forall t.t \to t \to t) \to (\forall t.t \to t \to t)}\,\text{ABS}
$$

On the other hand, its behavior is the following:

$$
\begin{aligned}
&\mathsf{NEG\ TRUE} \\
&\qquad \equiv (\lambda b^{\forall t.t \to t \to t}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ y\ x)\ (\Lambda t.\lambda x^t.\lambda y^t.x) \\
&\qquad \rightsquigarrow \Lambda t.\lambda x^t.\lambda y^t.(\Lambda t.\lambda x^t.\lambda y^t.x)[t]\ y\ x \\
&\qquad \rightsquigarrow \Lambda t.\lambda x^t.\lambda y^t.(\lambda x^t.\lambda y^t.x)\ y\ x \\
&\qquad \rightsquigarrow \Lambda t.\lambda x^t.\lambda y^t.y \\
&\qquad \equiv \mathsf{FALSE}
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{NEG\ FALSE} \\
&\qquad \equiv (\lambda b^{\forall t.t \to t \to t}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ y\ x)\ (\Lambda t.\lambda x^t.\lambda y^t.y) \\
&\qquad \rightsquigarrow \Lambda t.\lambda x^t.\lambda y^t.(\Lambda t.\lambda x^t.\lambda y^t.y)[t]\ y\ x \\
&\qquad \rightsquigarrow \Lambda t.\lambda x^t.\lambda y^t.(\lambda x^t.\lambda y^t.y)\ y\ x \\
&\qquad \rightsquigarrow \Lambda t.\lambda x^t.\lambda y^t.x \\
&\qquad \equiv \mathsf{TRUE}
\end{aligned}
$$

The term CONJ can be defined as

$$
\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ (c[t]\ x\ y)\ y
$$

On the one hand, its proof to be assigned type $bool \to bool \to bool$ is as follows:

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$b : \mathit{bool} \vdash b : \mathit{bool}$}
\LeftLabel{$\forall e$}
\UnaryInfC{$b : \mathit{bool} \vdash b[t] : t \to t \to t$}
\AxiomC{Δ_1}
\LeftLabel{APP}
\BinaryInfC{$b : \mathit{bool}, c : \mathit{bool}, x : t, y : t \vdash b\ (c\ x\ y) : t \to t$}
\AxiomC{}
\RightLabel{VAR}
\UnaryInfC{$y : t \vdash y : t$}
\RightLabel{APP}
\BinaryInfC{$b : \mathit{bool}, c : \mathit{bool}, x : t, y : t \vdash b[t]\ (c[t]\ x\ y)\ y : t$}
\RightLabel{ABS}
\UnaryInfC{$b : \mathit{bool}, c : \mathit{bool}, x : t \vdash \lambda y^t.b[t]\ (c[t]\ x\ y)\ y : t \to t$}
\RightLabel{ABS}
\UnaryInfC{$b : \mathit{bool}, c : \mathit{bool} \vdash \lambda x^t.\lambda y^t.b[t]\ (c[t]\ x\ y)\ y : t \to t \to t$}
\RightLabel{$\forall i$}
\UnaryInfC{$b : \mathit{bool}, c : \mathit{bool} \vdash \Lambda t.\lambda x^t.\lambda y^t.b[t]\ (c[t]\ x\ y)\ y : \mathit{bool}$}
\RightLabel{ABS}
\UnaryInfC{$b : \mathit{bool} \vdash \lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ (c[t]\ x\ y)\ y : \mathit{bool} \to \mathit{bool}$}
\RightLabel{ABS}
\UnaryInfC{$\vdash \lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ (c[t]\ x\ y)\ y : \mathit{bool} \to \mathit{bool} \to \mathit{bool}$}
\end{prooftree}
$$

where the subtree Δ_1 is the following:

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$c : \mathit{bool} \vdash c : \mathit{bool}$}
\LeftLabel{$\forall e$}
\UnaryInfC{$c : \mathit{bool} \vdash c[t] : t \to t \to t$}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$x : t \vdash x : t$}
\LeftLabel{APP}
\BinaryInfC{$c : \mathit{bool}, x : t \vdash c[t]\ x : t \to t$}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$y : t \vdash y : t$}
\RightLabel{APP}
\BinaryInfC{$c : \mathit{bool}, x : t, y : t \vdash c[t]\ x\ y : t$}
\end{prooftree}
$$

On the other hand, its behavior is the following:

$$
\begin{aligned}
&\mathsf{CONJ\ FALSE}\\
&\equiv (\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ (c[t]\ x\ y)\ y)\ (\Lambda t.\lambda x^t.\lambda y^t.y)\\
&\rightsquigarrow \lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.(\Lambda t.\lambda x^t.\lambda y^t.y)[t]\ (c[t]\ x\ y)\ y\\
&\rightsquigarrow \lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.(\lambda x^t.\lambda y^t.y)\ (c[t]\ x\ y)\ y\\
&\rightsquigarrow^2 \lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.y\\
&\equiv \lambda c^{bool}.\mathsf{FALSE}
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{CONJ\ TRUE}\\
&\equiv (\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ (c[t]\ x\ y)\ y)\ (\Lambda t.\lambda x^t.\lambda y^t.x)\\
&\rightsquigarrow \lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.(\Lambda t.\lambda x^t.\lambda y^t.x)[t]\ (c[t]\ x\ y)\ y\\
&\rightsquigarrow \lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.(\lambda x^t.\lambda y^t.x)\ (c[t]\ x\ y)\ y\\
&\rightsquigarrow^2 \lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.c[t]\ x\ y\\
&\rightsquigarrow^2_\eta \lambda c^{bool}.\Lambda t.c[t]
\end{aligned}
$$

The reduction for $\mathsf{CONJ\ TRUE}$ shows that it will reduce to the same *bool* term that can be applied to it, although we cannot reduce it more unless we apply the remaining parameter. Maybe a similar reduction rule as the known η-reduction for terms can be defined for type abstractions and applications.

The term DISJ can be defined as

$$\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ x\ (c[t]\ x\ y)$$

On the one hand, its proof to be assigned type $bool \to bool \to bool$ is as follows:

$$
\begin{prooftree}
\AxiomC{Δ_2}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$c : bool \vdash c : bool$}
\LeftLabel{$\forall e$}
\UnaryInfC{$c : bool \vdash c[t] : t \to t \to t$}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$x : t \vdash x : t$}
\RightLabel{APP}
\BinaryInfC{$c : bool, x : t \vdash c[t]\ x : t \to t$}
\AxiomC{}
\RightLabel{VAR}
\UnaryInfC{$y : d \vdash y : d$}
\RightLabel{APP}
\BinaryInfC{$c : bool, x : t, y : t \vdash c[t]\ x\ y : t$}
\RightLabel{APP}
\BinaryInfC{$b : bool, c : bool, x : t, y : t \vdash b[t]\ x\ (c[t]\ x\ y) : t$}
\RightLabel{ABS}
\UnaryInfC{$b : bool, c : bool, x : t \vdash \lambda y^t.b[t]\ x\ (c[t]\ x\ y) : t \to t$}
\RightLabel{ABS}
\UnaryInfC{$b : bool, c : bool \vdash \lambda x^t.\lambda y^t.b[t]\ x\ (c[t]\ x\ y) : t \to t \to t$}
\RightLabel{$\forall i$}
\UnaryInfC{$b : bool, c : bool \vdash \Lambda t.\lambda x^t.\lambda y^t.b[t]\ x\ (c[t]\ x\ y) : bool$}
\RightLabel{ABS}
\UnaryInfC{$b : bool \vdash \lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ x\ (c[t]\ x\ y) : bool \to bool$}
\RightLabel{ABS}
\UnaryInfC{$\vdash \lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ x\ (c[t]\ x\ y) : bool \to bool \to bool$}
\end{prooftree}
$$

where the subtree Δ_2 is the following:

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$b : bool \vdash b : bool$}
\LeftLabel{$\forall e$}
\UnaryInfC{$b : bool \vdash b[t] : t \to t \to t$}
\AxiomC{}
\LeftLabel{VAR}
\UnaryInfC{$x : t \vdash x : t$}
\LeftLabel{APP}
\BinaryInfC{$b : bool, x : t \vdash b[t]\ x : t \to t$}
\end{prooftree}
$$

On the other hand, its behavior is the following:

$$
\begin{aligned}
&\mathsf{DISJ\ TRUE}\\
&\qquad \equiv (\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ x\ (c[t]\ x\ y)\ (\Lambda t.\lambda x^t.\lambda y^t.x)\\
&\qquad \rightsquigarrow \lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.(\Lambda t.\lambda x^t.\lambda y^t.x)[t]\ x\ (c[t]\ x\ y)\\
&\qquad \rightsquigarrow \lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.(\lambda x^t.\lambda y^t.x)\ x\ (c[t]\ x\ y)\\
&\qquad \rightsquigarrow^2 \lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.x\\
&\qquad \equiv \lambda c^{bool}.\mathsf{TRUE}
\end{aligned}
$$

DISJ FALSE

$$
\begin{aligned}
&\equiv(\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.b[t]\ x\ (c[t]\ x\ y)\ (\Lambda t.\lambda x^t.\lambda y^t.y)\\
&\rightsquigarrow\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.(\Lambda t.\lambda x^t.\lambda y^t.y)[t]\ x\ (c[t]\ x\ y)\\
&\rightsquigarrow\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.(\lambda x^t.\lambda y^t.y)\ x\ (c[t]\ x\ y)\\
&\rightsquigarrow^2\lambda b^{bool}.\lambda c^{bool}.\Lambda t.\lambda x^t.\lambda y^t.c[t]\ x\ y\\
&\rightsquigarrow^2_\eta\lambda c^{bool}.\Lambda t.c[t]
\end{aligned}
$$

As mentioned in the CONJ TRUE reduction, this last one will reduce to the same *bool* term that can be applied to it, but it could be seen more clearly if we had a new rule of η-reduction for type abstractions and applications.

Exercise 8

Implement the two previous exercises in Haskell using the extension for *rank n polymorphic types*.

Answer

The implementation is provided as a Haskell project managed with Cabal within the folder `church-encodings`.

Although it contains implementations for booleans, numerals, pairs and lists, the ones of interest for this exercise correspond to the modules `Encoding.BoolChurch` and `Encoding.NatChurch`.

A test suite is also available, which can be executed with the script `test.sh` as follows:

```
chmod u+x test.sh && ./test.sh
```