Homework 1

Exercise 1.1

Define $l \coloneqq \lambda x \lambda y\,(x\ y)$, $r \coloneqq \lambda x \lambda y\,(y\ x)$. Then write a term *if-then-else* such that for all u, v

$$(\textit{if-then-else})\ l\ u\ v \mapsto u$$
$$(\textit{if-then-else})\ r\ u\ v \mapsto v$$

Solution: One possible definition for $(\textit{if-then-else})$ is

$$(\textit{if-then-else}) \coloneqq \lambda o \lambda u \lambda v\,(o\ False\ True\ True\ u\ v)$$

with $True \coloneqq \lambda x \lambda y\ x$ and $False \coloneqq \lambda x \lambda y\ y$ as defined in the lecture.
We can observe for arbitrary λ-terms u, v that

$$\begin{aligned}
&(\textit{if-then-else})\ l\ u\ v = (\lambda o \lambda u \lambda v\,(o\ False\ True\ True\ u\ v))\ l\ u\ v \\
&\mapsto^{*} l\ False\ True\ True\ u\ v = (\lambda x \lambda y\ (x\ y))\ False\ True\ True\ u\ v \\
&\mapsto^{*} False\ True\ True\ u\ v = (\lambda x \lambda y\ y)\ True\ True\ u\ v \\
&\mapsto^{*} True\ u\ v = (\lambda x \lambda y\ x)\ u\ v \\
&\mapsto^{*} u
\end{aligned}$$

and

$$\begin{aligned}
&(\textit{if-then-else})\ r\ u\ v = (\lambda o \lambda u \lambda v\,(o\ False\ True\ True\ u\ v))\ r\ u\ v \\
&\mapsto^{*} r\ False\ True\ True\ u\ v = (\lambda x \lambda y\ (y\ x))\ False\ True\ True\ u\ v \\
&\mapsto^{*} True\ False\ True\ u\ v = (\lambda x \lambda y\ x)\ False\ True\ u\ v \\
&\mapsto^{*} False\ u\ v = (\lambda x \lambda y\ y)\ u\ v \\
&\mapsto^{*} v
\end{aligned}$$

Hence, the term $(\textit{if-then-else})$ exhibits the desired behaviour.

Exercise 1.2

Write a λ-term *Smaller* such that

$$\begin{aligned}
&\textit{Smaller}\ \overline{n}\ \overline{m} \mapsto True && \textit{if } n < m \\
&\textit{Smaller}\ \overline{n}\ \overline{m} \mapsto False && \textit{otw.}
\end{aligned}$$

Solution: Firstly, housekeeping. It is assumed that, as presented in the lecture, $\overline{n}$ and $\overline{m}$ refer to Church-numerals. Moreover, it is assumed that "$<$" refers to the smaller relation in $\mathbb{N}$.

Secondly, the general idea of the presented approach will be discussed. We shall construct the smaller relation as follows.

$$not(isZero(\dot{-}(x,y)))$$

with *not* being defined as

$$not(x) := \begin{cases} 0 & \textit{if } x = 1 \\ 1 & \textit{if } x = 0 \\ undef. & \textit{otw.} \end{cases}$$

with *isZero* being defined as

$$isZero(x) := \begin{cases} 1 & \textit{if } x = 0 \\ 0 & \textit{otw.} \end{cases}$$

and with $\dot{-}$ being defined as

$$\dot{-}(x,y) := \begin{cases} x - y & \textit{if } x > y \\ 0 & \textit{otw.} \end{cases}$$

Hence, the following behaviour can be observed.
If $x \leq y$ the term evaluates to

$$not(isZero(\dot{-}(x,y))) = not(isZero(0)) = not(1) = 0$$

and if $x > y$ we have $x - y > 0$ resulting in

$$not(isZero(\dot{-}(x,y))) = not(isZero(x-y)) = not(0) = 1$$

Thirdly, we have to translate this intuition into λ-terms. Before we do so, however, a small remark regarding notation. That is, a λ-term of the shape $f^n\ x$ represents $\underbrace{f(f \cdots (f}_{\text{n-times}}\ x) \cdots)$.

We start the translation with $\dot{-}$. To capture the behaviour

$$\begin{aligned} \dot{-}\ \overline{x}\ \overline{y} &\mapsto \overline{x-y} && \textit{if } x > y \\ \dot{-}\ \overline{x}\ \overline{y} &\mapsto \overline{0} && \textit{otw} \end{aligned}$$

we define $\dot{-} := \lambda x \lambda y\,(y\ \ Pred\ x)$, with *Pred* as defined in the lecture and thus obtain for the arbitrary Church-numerals $\overline{n}$ and $\overline{m}$

$$\begin{aligned} \dot{-}\ \overline{n}\ \overline{m} &= (\lambda x \lambda y\,(y\ Pred\, x))\ \overline{n}\ \overline{m} \\ &\mapsto^* \overline{m}\ Pred\, \overline{n} = (\lambda f \lambda x\,(f^m\ x))\ Pred\, \overline{n} \\ &\mapsto^* Pred^m\ \overline{n} = Pred^m\ (\lambda f \lambda x\,(f^n\ x)) \end{aligned}$$

From there we have two cases to account for. if $n > m$, then

$$\mapsto^* \mathit{Pred}^{m-m}\,(\lambda f \lambda x\,(f^{n-m}\;x) = \lambda f \lambda x\,(f^{n-m}\;x) = \overline{n-m}$$

and if $n \leq m$, then

$$\mapsto^* \mathit{Pred}^{m-n}\,(\lambda f \lambda x\,(f^{n-n}\;x)) = \mathit{Pred}^{m-n}\,(\lambda f \lambda x\,x) = \mathit{Pred}^{m-n}\,\overline{0} \mapsto^* \overline{0}$$

because $Pred\,\overline{0} \mapsto^* \overline{0}$.
Now, moving on towards *isZero*. Here we simply use the definition provided in the lecture. That is,

$$\begin{aligned} \mathit{isZero}\,\overline{0} &\mapsto True \\ \mathit{isZero}\,\overline{n+1} &\mapsto False \end{aligned}$$

with

$$\mathit{isZero} := \lambda n\, n\;(\lambda z\, False)\; True$$

Moreover, for *not* we define

$$\begin{aligned} \mathit{not}\, True &\mapsto False \\ \mathit{not}\, False &\mapsto True \end{aligned}$$

with

$$\mathit{not} := \lambda o \lambda u \lambda v\,(o\;v\;u)$$

Allowing us to observe

$$\begin{aligned} \mathit{not}\, True &= (\lambda o \lambda u \lambda v\,(o\;v\;u))\; True \mapsto \lambda u \lambda v\,(True\;v\;u) \\ &= \lambda u \lambda v\,((\lambda x \lambda y\, x)\;v\;u) \mapsto^* \lambda u \lambda v\, v = False \end{aligned}$$

and

$$\begin{aligned} \mathit{not}\, False &= (\lambda o \lambda u \lambda v\,(o\;v\;u))\; False \mapsto \lambda u \lambda v\,(False\;v\;u) \\ &= \lambda u \lambda v\,((\lambda x \lambda y\, y)\;v\;u) \mapsto^* \lambda u \lambda v\, u = True \end{aligned}$$

Lastly, we have to combine these terms to create

$$\mathit{Smaller} := \lambda n \lambda m\,(\mathit{not}\,(\mathit{isZero}\,(\dot{-}\;m\;n)))$$

Hence, for any Church-numerals $\overline{n}$ and $\overline{m}$ we have

$$\begin{aligned} &\mathit{Smaller}\,\overline{n}\,\overline{m} = (\lambda n \lambda m\,(\mathit{not}\,(\mathit{isZero}\,(\dot{-}\;m\;n))))\,\overline{n}\,\overline{m} \\ &\mapsto^* \mathit{not}\,(\mathit{isZero}\,(\dot{-}\;\overline{m}\,\overline{n})) = \mathit{not}\,(\mathit{isZero}\,((\lambda x \lambda y\, y\;\mathit{Pred}\,x)\;\overline{m}\,\overline{n})) \\ &\mapsto^* \mathit{not}\,(\mathit{isZero}\,(\overline{n}\;\mathit{Pred}\,\overline{m})) = \mathit{not}\,(\mathit{isZero}\,((\lambda f \lambda x\,(f^n\;x))\;\mathit{Pred}\,\overline{m})) \\ &\mapsto^* \mathit{not}\,(\mathit{isZero}\,(\mathit{Pred}^n\;\overline{m})) \mapsto^* \mathit{not}\,(\mathit{isZero}\,\overline{m \dot{-} n}) \end{aligned}$$

Now we are again confronted with two cases. That is, in the case $m \leq n$ it follows that $m - n \leq 0$ and thus $\overline{m \dot{-} n} = \overline{0}$ resulting in

$$\begin{aligned}= \mathit{not}\,(\mathit{isZero}\,\overline{0}) \mapsto^* \mathit{not}\ \mathit{True} = (\lambda o \lambda u \lambda v\,(o\ v\ u))\ \mathit{True}\\ \mapsto \lambda u \lambda v\,(\mathit{True}\ v\ u) \mapsto \lambda u \lambda v\, v = \mathit{False}\end{aligned}$$

and in the case $n < m$ we have $\overline{m \dot{-} n} = \overline{m - n} \neq \overline{0}$ resulting in

$$\begin{aligned}= \mathit{not}\,(\mathit{isZero}\,\overline{m-n}) \mapsto^* \mathit{not}\ \mathit{False} = (\lambda o \lambda u \lambda v\,(o\ v\ u))\ \mathit{False}\\ \mapsto \lambda u \lambda v\,(\mathit{False}\ v\ u) \mapsto \lambda u \lambda v\, u = \mathit{True}\end{aligned}$$

which is exactly the desired behaviour of *Smaller*.

Exercise 1.3

Define for every $n \in \mathbb{N}$

$$\begin{aligned}\underline{0} &:= \lambda x \lambda f\, x\\ \underline{n+1} &:= \lambda x \lambda f\,(f\ \underline{n}\ (f\ \underline{n-1}\ (\cdots(f\ \underline{0}\ x)\cdots)))\end{aligned}$$

Examples:

$$\begin{aligned}\underline{1} &:= \lambda x \lambda f\,(f\ \underline{0}\ x)\\ \underline{2} &:= \lambda x \lambda f\,(f\ \underline{1}\ (f\ \underline{0}\ x))\\ \underline{3} &:= \lambda x \lambda f\,(f\ \underline{2}\ (f\ \underline{1}\ (f\ \underline{0}\ x)))\end{aligned}$$

1. Write a λ-term t such that for all $n \in \mathbb{N}$

$$\begin{aligned}t\ \underline{0} &\mapsto \underline{0}\\ t\ \underline{n+1} &\mapsto \underline{n}\end{aligned}$$

2. Write a λ-term t such that for all $n \in \mathbb{N}$

$$t\ \underline{n} \mapsto \underline{n+1}$$

3. Write a λ-term t such that for all $n \in \mathbb{N}$

$$t\ \underline{n}\ \underline{m} \mapsto \underline{n+m}$$

Solution:

Term 1

Let

$$t = P := \lambda n\, n\ \underline{0}\ \mathit{True}$$

For the case $\underline{n} = \underline{0}$ we have,

$$\begin{aligned} &P\ \underline{0} = (\lambda n\, n\ \underline{0}\ True)\ \underline{0} \\ &\mapsto^* \underline{0}\ \underline{0}\ True = (\lambda x \lambda f\, x)\ \underline{0}\ True \mapsto^* \underline{0} \end{aligned}$$

and otherwise for $\underline{n+1}$ we have

$$\begin{aligned} &P\ \underline{n+1} = (\lambda n\, n\ \underline{0}\ True)\ \underline{n+1} \mapsto^* \underline{n+1}\ \underline{0}\ True \\ &= (\lambda x \lambda f\, (f\ \underline{n}\ (\cdots x \cdots)))\ \underline{0}\ True \\ &\mapsto^* True\ \underline{n}\ (\cdots \underline{0} \cdots) \\ &= (\lambda x \lambda y\, x)\ True\ \underline{n}\ (\cdots \underline{0} \cdots) \mapsto^* \underline{n} \end{aligned}$$

Term 2

Let

$$t = S := \lambda n \lambda y \lambda g\, (g\ n\ (n\ y\ g))$$

We separate the proof into two cases.
For the case $\underline{n} = \underline{0}$ we have,

$$\begin{aligned} &S\ \underline{0} = (\lambda n \lambda y \lambda g\, (g\ n\ (n\ y\ g)))\ \underline{0} \mapsto \lambda y \lambda g\, (g\ \underline{0}\ (\underline{0}\ y\ g)) \\ &= \lambda y \lambda g\, (g\ \underline{0}\ ((\lambda x \lambda f\, x)\ y\ g)) \mapsto^* \lambda y \lambda g\, (g\ \underline{0}\ y) = \underline{0} \end{aligned}$$

and otherwise for $\underline{n+1}$ we have

$$\begin{aligned} &S\ \underline{n+1} = (\lambda n \lambda y \lambda g\, (g\ n\ (n\ y\ g)))\ \underline{n+1} \mapsto \lambda y \lambda g\, (g\ \underline{n+1}\ (\underline{n+1}\ y\ g)) \\ &= \lambda y \lambda g\, (g\ \underline{n+1}\ ((\lambda x \lambda f\, (f\ \underline{n}\ (\cdots x \cdots)))\ y\ g)) \\ &\mapsto^* \lambda y \lambda g\, (g\ \underline{n+1}\ (g\ \underline{n}\ (\cdots y \cdots))) = \underline{n+2} \end{aligned}$$

Term 3

Let

$$t = A := \lambda n \lambda m\, n\ m\ (\lambda x \lambda y\, S\ y)$$

Let $\underline{m}$ be an arbitrary numeral, then we have for $\underline{n} = \underline{0}$

$$\begin{aligned} A\ \underline{n}\ \underline{m} &= (\lambda n \lambda m\, n\ m\ (\lambda x \lambda y\, S\ y))\ \underline{n}\ \underline{m} \mapsto^* \underline{n}\ \underline{m}\ (\lambda x \lambda y\, S\ y) \\ &= (\lambda x \lambda f\, x)\ \underline{m}\ (\lambda x \lambda y\, S\ y) \mapsto^* \underline{m} \end{aligned}$$

and otherwise for $\underline{n+1}$ we have

$$\begin{aligned} A\ \underline{n+1}\ \underline{m} &= (\lambda n \lambda m\, n\ m\ (\lambda x \lambda y\, S\ y))\ \underline{n+1}\ \underline{m} \mapsto^* \underline{n+1}\ \underline{m}\ (\lambda x \lambda y\, S\ y) \\ &= (\lambda x \lambda f\, (f\ \underline{n+1}\ (\cdots x \cdots)))\ \underline{m}\ (\lambda x \lambda y\, S\ y) \\ &\mapsto^* (\lambda x \lambda y\, S\ y)\ \underline{n+1}\ (\cdots \underline{m} \cdots) \\ &\mapsto^* S\ (\cdots \underline{m} \cdots) \mapsto^* S^{n+1}\ \underline{m} \\ &\mapsto^* S^n\ \underline{m+1} \mapsto^* S\ \underline{m+n} \mapsto^* \underline{m+(n+1)} \end{aligned}$$

Homework 2

Exercise 2.1

Formally prove by induction on u that

$$u[v/y][t/x] = u[t/x][v[t/x]/y]$$

provided $x \neq y$ and y does not occur in t.

Solution: By induction on u

- For $u = y$ we have

$$\begin{aligned} u[v/y][t/x] &= y[v/y][t/x] = v[t/x] = y[v[t/x]/y] \\ &\overset{x\neq y}{=} y[t/x][v[t/x]/y] = u[t/x][v[t/x]/y] \end{aligned}$$

- For $u = x$ we have

$$\begin{aligned} u[v/y][t/x] &= x[v/y][t/x] \overset{x\neq y}{=} x[t/x] = t \\ &\overset{y\ n.\ in\ t}{=} t[v[t/x]/y] = x[t/x][v[t/x]/y] = u[t/x][v[t/x]/y] \end{aligned}$$

- For $u = z$ with $x \neq z$ and $y \neq z$ we have

$$\begin{aligned} u[v/y][t/x] &= z[v/y][t/x] \overset{y\neq z}{=} z[t/x] \overset{x\neq z}{=} z \\ &\overset{y\neq z}{=} z[v[t/x]/y] \overset{x\neq z}{=} z[t/x][v[t/x]/y] = u[t/x][v[t/x]/y] \end{aligned}$$

- For $u = \lambda z\, w$ we have

$$\begin{aligned} u[v/y][t/x] &= (\lambda z\, w)[v/y][t/x] \overset{Def.}{=} \lambda z\, w[v/y][t/x] \overset{IH}{=} \lambda z\, w[t/x][v[t/x]/y] \\ &\overset{Def.}{=} (\lambda z\, w)[t/x][v[t/x]/y] = u[t/x][v[t/x]/y] \end{aligned}$$

- For $u = w_1\ w_2$ we have

$$\begin{aligned} u[v/y][t/x] &= (w_1\ w_2)[v/y][t/x] \overset{Def.}{=} w_1[v/y][t/x]\ w_2[v/y][t/x] \\ &\overset{IH}{=} w_1[t/x][v[t/x]/y]\ w_2[t/x][v[t/x]/y] \\ &\overset{Def.}{=} (w_1\ w_2)[t/x][v[t/x]/y] = u[t/x][v[t/x]/y] \end{aligned}$$

Exercise 2.2

Prove or disprove:

> If w is elementary and $w \mapsto w'$, then w' is elementary.

Solution: We shall disprove this claim by constructing an elementary λ-term which does not reduce (within one step) to an elementary λ-term.
Firstly, we know that

- a λ-term $u\,t$ is elementary, if $t \in SN$, $u \in SN$ but $t\,u \notin SN$ and that
- a λ-term t is strongly normalisable, i.e. $t \in SN$, if there is no infinite reduction of t, i.e. $h(t)$ is finite.

Secondly, let w be the λ-term

$$w := t\,u = (\lambda x(z(x\,x)))(\lambda x\,x\,x)$$

Thirdly, we need to show that w is elementary.

1. We can observe that t and u contain no redex, thus can no longer be reduced and are therefore in normal form. Hence, allowing us to conclude $t \in SN$ and $u \in SN$.
2. To show that $w \notin SN$ we will reduce the term once. In this case the only possibility to do so is to apply t to u, i.e.

   $$w = t\,u = (\lambda x(z(x\,x)))(\lambda x\,x\,x) \mapsto z((\lambda x\,x\,x)\;(\lambda x\,x\,x)) = w'$$

   Given w' we can observe that $w' = t'u'$ with $h(u')$ being infinite. That is, as shown in the lecture the term $u' = ((\lambda x\,x\,x)\;(\lambda x\,x\,x))$ has an infinite reduction. Therefore, $h(w')$ is infinite. Moreover, since w' has not the form $(\lambda x\,v)\;t\,t_1 \ldots t_n$ with $u, t, t_1, \ldots, t_n \in SN$ it can not be elementary.

Hence, the claim is disproven.

Exercise 2.3

Prove or disprove:

> There are terms u and t such that x does not occur in u and $(\lambda x\,u)\;t$ is elementary.

Solution: Assume that the λ-terms u and t exists. That is, we let u and t be λ-terms, such that

$$w = v\;t = (\lambda x\,u)\;t$$

with x not in u, is elementary.
Since, w is elementary we know that $(\lambda x\,u) \in SN$, $t \in SN$ and $w \notin SN$. Moreover, by Prop. 12 we know that

$$(\lambda x\, u)\; t \text{ is elementary implies } u[t/x] \notin SN$$

However, since x not in u we obtain

$$u[t/x] = u \notin SN$$

Thus we obtain $u \notin SN$ and $(\lambda x\, u) \in SN$, which is a contradiction. That is, with $h(u)$ being infinite, it follows that $h((\lambda x\, u))$ is also infinite, as we can always find a redex within u to contract. To conclude, assuming the existence of such terms induces a contradiction.

Homework 3

Exercise 3.1

Let $t : A$ be a term that does not contain free variables and is in normal form. Prove that

- if $A = B \wedge C$ then $t = \langle u, v \rangle$ and
- if $A = B \to C$ then $t = \lambda x^B\, u$

Solution: Proof by induction on the type derivation $\mathcal{D}$ of t, with the IH:

For a λ-term $t : A$, which does not contain free variables and is in normal form, its type derivation $\mathcal{D}$ must be of the shape.

$$\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}'\\ u : C\end{array}}{\lambda x^B\, u : B \to C} \qquad \textit{if } A = B \to C$$

and

$$\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}'\\ u : B\end{array} \qquad \begin{array}{c}\mathcal{E}'\\ v : C\end{array}}{\langle u, v \rangle : B \wedge C} \qquad \textit{if } A = B \wedge C$$

Now we are going to distinguish by cases according to the last rule of the type derivation $\mathcal{D}$ of t.

1. If $\mathcal{D} = \dfrac{}{t : A}$ we know that t is a variable. Hence, t contains itself as a free variable. However, we know that t can not contain free variables. Therefore, this rule can not be the last rule of the type derivation $\mathcal{D}$.
2. If $\mathcal{D} = \dfrac{\begin{array}{c}\mathcal{D}'\\ u : C\end{array}}{\lambda x^B\, u : B \to C}$ we have our thesis. That is, $A = B \to C$ and t was derived by the desired rule.

3. If $\mathcal{D} = \dfrac{\begin{matrix}\mathcal{D}' & \mathcal{E}' \\ u : B & v : C\end{matrix}}{\langle u, v\rangle : B \wedge C}$ we have our thesis. That is, $A = B \wedge C$ and t was derived by the desired rule.

4. If $\mathcal{D} = \dfrac{\begin{matrix}\mathcal{D}' \\ u : E \wedge F\end{matrix}}{u\ \pi_0 : E}$ with $t = u\ \pi_0$. We can observe that if u contains free variables, then $u\ \pi_0$ must contain the same free variables. Because the applied rule does not bin free variables in u. Therefore, it follows from $t = u\ \pi_0$ being free variable free, that u does not contain free variables. Moreover, if u contains a redex or the subterm $\langle t_0, t_1\rangle\ \pi_i$ with $i \in \{0, 1\}$. Then the same must be contained in $u\ \pi_0$ as u was not modified by this rule. Therefore, if $t = u\ \pi_0$ is in normal form, then u must also be in normal form.

   Hence, we conclude that u must be in normal form and can not contain free variables, allowing us to apply the IH for $u : E \wedge F$. That is, we obtain

$$\mathcal{D} = \dfrac{\dfrac{\begin{matrix}\mathcal{D}'' & \mathcal{E}'' \\ r : E & o : F\end{matrix}}{\langle r, o\rangle : E \wedge F}}{\langle r, o\rangle \pi_0 : E}$$

   However, as $\langle t_0, t_1\rangle\ \pi_i \mapsto t_i$ with $i \in \{0, 1\}$ it follows that $t = \langle r, o\rangle\ \pi_0$ is not in normal form. Hence, the type derivation of t can not be of this shape.

5. If $\mathcal{D} = \dfrac{\begin{matrix}\mathcal{D}' \\ u : E \wedge F\end{matrix}}{u\ \pi_1 : F}$ we can discard this case in analogue to the case 4.

6. If $\mathcal{D} = \dfrac{\begin{matrix}\mathcal{D}' & \mathcal{E}' \\ u : E \to F & v : E\end{matrix}}{u\ v : F}$ with $t = u\ v$. Here we argue in a similar fashion as in case 4. That is, we will argue that u must be free variable free and in normal form.

   As this rule does not bind free variables in u, those variables must remain free in $u\ v$. Furthermore, if u contains a redex or $\langle t_0, t_1\rangle \pi_i$ with $i \in \{0, 1\}$, those must also be present in $u\ v$, as u is not modified by this rule. Hence, it follows that u must be in normal form and can not contain free variables. Furthermore, with $u : E \to F$ it follows by IH that

$$\mathcal{D} = \dfrac{\dfrac{\begin{matrix}\mathcal{D}'' \\ r : F\end{matrix}}{\lambda x^E\, r : E \to F} \qquad \begin{matrix}\mathcal{E}' \\ v : E\end{matrix}}{(\lambda x^E\, r)\ v : F}$$

that is $t = (\lambda x^E\, r)\ v$ and thus t contains a redex and is therefore not in normal form. Hence, the type derivation of t can not be of this shape.

Given this result our thesis follows immediately.

Homework 4

Exercise 4.1

Define $\neg A := A \to \bot$, where $\bot$ is a type variable. Write a natural deduction of

$$\neg\neg(A \land B) \to \neg\neg A$$

Solution: Firstly, we substitute

$$\begin{aligned} \neg\neg(A \land B) \to \neg\neg A &= \neg((A \land B) \to \bot) \to \neg(A \to \bot) \\ &= ((A \land B) \to \bot) \to \bot) \to ((A \to \bot) \to \bot) \end{aligned}$$

Secondly, the derivation.
Two derivations will be presented. The first one is merely the natural deduction, while the second one depicts the natural deduction together with its corresponding λ-terms.

$$\dfrac{\dfrac{\dfrac{[((A \land B) \to \bot) \to \bot]^{(1)} \qquad \dfrac{\dfrac{[A \to \bot]^{(2)} \qquad \dfrac{[(A \land B)]^{(3)}}{A}}{\bot}}{(A \land B) \to \bot}\,(3)}{\bot}}{(A \to \bot) \to \bot}\,(2)}{(((A \land B) \to \bot) \to \bot) \to ((A \to \bot) \to \bot)}\,(1)$$

And now with the corresponding λ-terms

$$\dfrac{\dfrac{\dfrac{z^{((A\land B)\to\bot)\to\bot} : [((A \land B) \to \bot) \to \bot]^{(1)} \qquad \dfrac{\dfrac{y^{A\to\bot} : [A \to \bot]^{(2)} \qquad \dfrac{x^{A\land B} : [(A \land B)]^{(3)}}{x^{A\land B}\,\pi_0 : A}}{y^{A\to\bot}\,(x^{A\land B}\,\pi_0) : \bot}}{\lambda x^{A\land B}(y^{A\to\bot}\,(x\,\pi_0)) : (A \land B) \to \bot}\,(3)}{z^{((A\land B)\to\bot)\to\bot}\lambda x^{A\land B}(y^{A\to\bot}\,(x\,\pi_0)) : \bot}}{\lambda y^{A\to\bot}(z^{((A\land B)\to\bot)\to\bot}\lambda x^{A\land B}(y\,(x\,\pi_0))) : (A \to \bot) \to \bot}\,(2)}{\lambda z^{((A\land B)\to\bot)\to\bot}\lambda y^{A\to\bot}(z\,\lambda x^{A\land B}(y\,(x\,\pi_0))) : (((A \land B) \to \bot) \to \bot) \to ((A \to \bot) \to \bot)}\,(1)$$

Homework 5

Exercise 5.1

Show that the λ-term

$$\lambda y\, y\, y\, (\lambda x\, x\, y\, x)$$

is typable in $D\Omega$ with a type that does not contain $\top$.

Solution: Given the bracketing convention we have

$$t := \lambda y\, y\, y\, (\lambda x\, x\, y\, x) = \lambda y\, ((y\, y)\, (\lambda x\, ((x\, y)\, x)))$$

for which we shall provide a type derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{y : b_0}{y : b_2} \quad \dfrac{y : b_0}{y : b_3}}{y\, y : c_3} \quad \dfrac{\dfrac{\dfrac{\dfrac{x : a_0}{x : a_1} \quad \dfrac{y : b_0}{y : b_1}}{x\, y : c_1} \quad \dfrac{x : a_0}{x : a_2}}{(x\, y)\, x : c_2}}{\lambda x\, ((x\, y)\, x) : c_4}}{(y\, y)\, (\lambda x\, ((x\, y)\, x)) : c_5}}{\lambda y\, ((y\, y)\, (\lambda x\, ((x\, y)\, x))) : c_6}$$

Based on this derivation we start by constructing the type schema.

$$\begin{aligned}
&a_0 = a_1 \wedge a_2 = (b_1 \to (a_2 \to c_2)) \wedge a_2 \\
&a_1 = b_1 \to c_1 = (b_1 \to (a_2 \to c_2)) \\
&a_2 = ? \\
&b_0 = b_1 \wedge b_2 \wedge b_3 = b_1 \wedge (b_3 \to ((((b_1 \to (a_2 \to c_2)) \wedge a_2) \to c_2) \to c_5)) \wedge b_3 \\
&b_1 = ? \\
&b_2 = b_3 \to c_3 = b_3 \to (a_0 \to c_2) \to c_5 = b_3 \to ((((b_1 \to (a_2 \to c_2)) \wedge a_2) \to c_2) \to c_5) \\
&b_3 = ? \\
&c_1 = a_2 \to c_2 \\
&c_2 = ? \\
&c_3 = c_4 \to c_5 = (((b_1 \to (a_2 \to c_2)) \wedge a_2) \to c_2) \to c_5 \\
&c_4 = a_0 \to c_2 = ((b_1 \to (a_2 \to c_2)) \wedge a_2) \to c_2 \\
&c_5 = ? \\
&c_6 = b_0 \to c_5 = (b_1 \wedge (b_3 \to ((((b_1 \to (a_2 \to c_2)) \wedge a_2) \to c_2) \to c_5)) \wedge b_3) \to c_5
\end{aligned}$$

This is the general schema from which we are going to build our types.
We start by setting the following types.

$$a_2 = A; \quad b_1 = B; \quad b_3 = B; \quad c_2 = C; \quad c_5 = D$$

and substitute to obtain

$$\begin{aligned}
a_0 &= (B \to (A \to C)) \land A \\
a_1 &= (B \to (A \to C)) \\
a_2 &= A \\
b_0 &= B \land (B \to ((((B \to (A \to C)) \land A) \to C) \to D)) \\
b_1 &= B \\
b_2 &= B \to ((((B \to (A \to C)) \land A) \to C) \to D) \\
b_3 &= B \\
c_1 &= A \to C \\
c_2 &= C \\
c_3 &= (((B \to (A \to C)) \land A) \to C) \to D \\
c_4 &= ((B \to (A \to C)) \land A) \to C \\
c_5 &= D \\
c_6 &= (B \land (B \to ((((B \to (A \to C)) \land A) \to C) \to D)) \land B) \to D
\end{aligned}$$

resulting in

$$
\dfrac{\dfrac{\dfrac{\dfrac{y : B \land (B \to ((((B \to (A \to C)) \land A) \to C) \to D))^{(*)}}{y : B \to ((((B \to (A \to C)) \land A) \to C) \to D)} \qquad \dfrac{y : (*)}{y : B}}{y\,y : (((B \to (A \to C)) \land A) \to C) \to D} \qquad \dfrac{\dfrac{\dfrac{\dfrac{x : (B \to (A \to C)) \land A}{x : B \to (A \to C)} \qquad \dfrac{y : (*)}{y : B}}{x\,y : A \to C} \qquad \dfrac{x : (B \to (A \to C)) \land A}{x : A}}{(x\,y)\,x : C}}{\lambda x\,((x\,y)\,x) : ((B \to (A \to C)) \land A) \to C}}{(y\,y)\,(\lambda x\,((x\,y)\,x)) : D}}{\lambda y\,((y\,y)\,(\lambda x\,((x\,y)\,x))) : (B \land (B \to ((((B \to (A \to C)) \land A) \to C) \to D))) \to D}
$$