# "Assignment" 1

William Troiani

September 2022

## 1 Introduction

We define a category $\mathcal{L}$ whose objects are the types of simply-typed lambda calculus, and whose morphisms are the terms of that calculus. The natural desiderata for such a category are that the fundamental algebraic structure of lambda calculus, function application and lambda abstraction, should be realised by categorical algebra.

We assume familiarity with simply-typed lambda calculus; some details are recalled in Appendix A or one can consult [2].

Following Church's original presentation our lambda calculus only contains function types and $\Phi_{\to}$ denotes the set of simple types. We write $\Lambda_\sigma$ for the set of $\alpha$-equivalence classes of lambda terms of type $\sigma$, and we write $=_{\beta\eta}$ for the equivalence relation generated by $\beta\eta$ equivalence.

**Definition 1.1 (Category of lambda terms).** The category $\mathcal{L}$ has objects

$$\mathrm{ob}(\mathcal{L}) = \Phi_{\to} \cup \{\mathbf{1}\}$$

and morphisms given for types $\sigma, \tau \in \Phi_{\to}$ by

$$\begin{aligned}
\mathcal{L}(\sigma,\tau) &= \Lambda_{\sigma\to\tau}/=_{\beta\eta}\\
\mathcal{L}(\mathbf{1},\sigma) &= \Lambda_{\sigma}/=_{\beta\eta}\\
\mathcal{L}(\sigma,\mathbf{1}) &= \{\star\}\\
\mathcal{L}(\mathbf{1},\mathbf{1}) &= \{\star\}\,,
\end{aligned}$$

where $\star$ is a new symbol. For $\sigma, \tau, \rho \in \Phi_{\to}$ the composition rule is the function

$$\mathcal{L}(\tau,\rho)\times\mathcal{L}(\sigma,\tau) \longrightarrow \mathcal{L}(\sigma,\rho) \tag{1.1}$$

$$(N, M) \longmapsto \lambda x^\sigma\,.\,(N(Mx)) \tag{1.2}$$

where $x \notin \mathrm{FV}(N) \cup \mathrm{FV}(M)$. We write the composite as $N \circ M$. In the remaining special cases the composite is given by the rules

$$\mathcal{L}(\tau,\rho)\times\mathcal{L}(\mathbf{1},\tau) \longrightarrow \mathcal{L}(\mathbf{1},\rho)\,, \qquad N\circ M = (N\,M)\,, \tag{1.3}$$

$$\mathcal{L}(\mathbf{1},\rho)\times\mathcal{L}(\mathbf{1},\mathbf{1}) \longrightarrow \mathcal{L}(\mathbf{1},\rho)\,, \qquad N\circ\star = N\,, \tag{1.4}$$

$$\mathcal{L}(\mathbf{1},\rho)\times\mathcal{L}(\sigma,\mathbf{1}) \longrightarrow \mathcal{L}(\sigma,\rho)\,, \qquad N\circ\star = \lambda t^\sigma\,.\,N\,, \tag{1.5}$$

where in the final rule $t \notin \mathrm{FV}(N)$. Notice that these functions, although their rules depend on representatives of equivalence classes, are none-the-less well defined.

For terms $M, N$ the expression $M = N$ always means equality of terms (that is, up to $\alpha$-equivalence) and we write $M =_{\beta\eta} N$ if we want to indicate equality up to $\beta\eta$-equivalence (for example as morphisms in the category $\mathcal{L}$). Since the free variable set of a lambda term is not invariant under $\beta$-reduction, some care is necessary in defining the category $\mathcal{L}_Q$ below. Let $\twoheadrightarrow_\beta$ denote multi-step $\beta$-reduction [1, Definition 1.3.3].

**Lemma 1.2.** *If* $M \twoheadrightarrow_\beta N$ *then* $\mathrm{FV}(N) \subseteq \mathrm{FV}(M)$.

**Definition 1.3.** Given a term $M$ we define

$$\mathrm{FV}_\beta(M) = \bigcap_{N =_\beta M} \mathrm{FV}(N)$$

where the intersection is over all terms $N$ which are $\beta$-equivalent to $M$.

Clearly if $M =_\beta M'$ then $\mathrm{FV}_\beta(M) = \mathrm{FV}_\beta(M')$.

**Lemma 1.4.** *Given terms* $M : \sigma \to \rho$ *and* $N : \sigma$ *we have*

$$\mathrm{FV}_\beta((MN)) \subseteq \mathrm{FV}_\beta(M) \cup \mathrm{FV}_\beta(N)\,.$$

**Lemma 1.5.** *Given* $M : \sigma \to \rho$ *and* $N : \tau \to \sigma$ *we have*

$$\mathrm{FV}_\beta(M \circ N) \subseteq \mathrm{FV}_\beta(M) \cup \mathrm{FV}_\beta(N)\,. \tag{1.6}$$

Given a set $Q$ of variables we write $\Lambda^Q_\sigma$ for the set of lambda terms $M$ of type $\sigma$ with $\mathrm{FV}(M) \subseteq Q$. Let $=_{\beta\eta}$ denote the induced relation on this subset of $\Lambda_\sigma$.

**Lemma 1.6.** *For any type* $\sigma$ *and set* $Q$ *of variables the image of the injective map*

$$\Lambda^Q_p / =_{\beta\eta} \longrightarrow \Lambda_p / =_{\beta\eta} \tag{1.7}$$

*is the set of equivalence classes of terms* $M$ *with* $\mathrm{FV}_\beta(M) \subseteq Q$.

*Proof.* Since the simply-typed lambda calculus is strongly normalising [1, Theorem 3.5.1] and confluent [1, Theorem 3.6.3] there is a unique normal form $\widehat{M}$ in the $\beta$-equivalence class of $M$, and $\mathrm{FV}_\beta(M) = \mathrm{FV}(\widehat{M})$. Hence if $\mathrm{FV}_\beta(M) \subseteq Q$ then $\mathrm{FV}(\widehat{M}) \subseteq Q$ and so $M$ is in the image of (1.7). □

**Definition 1.7.** For a set of variables $Q$ we define a subcategory $\mathcal{L}_Q \subseteq \mathcal{L}$ by

$$\mathrm{ob}(\mathcal{L}_Q) = \mathrm{ob}(\mathcal{L}) = \Phi_\to \cup \{\mathbf{1}\}$$

and for types $\sigma, \rho$

$$\begin{aligned}
\mathcal{L}_Q(\sigma,\rho) &= \{M \in \mathcal{L}(\sigma,\rho) \mid \mathrm{FV}_\beta(M) \subseteq Q\}\,,\\
\mathcal{L}_Q(\mathbf{1},\sigma) &= \{M \in \mathcal{L}(\mathbf{1},\sigma) \mid \mathrm{FV}_\beta(M) \subseteq Q\}\,,\\
\mathcal{L}_Q(\sigma,\mathbf{1}) &= \mathcal{L}(\sigma,\mathbf{1}) = \{\star\}\,,\\
\mathcal{L}_Q(\mathbf{1},\mathbf{1}) &= \mathcal{L}(\mathbf{1},\mathbf{1}) = \{\star\}\,.
\end{aligned}$$

Note that the last two lines have the same form using the convention that $\mathrm{FV}_\beta(\star) = \emptyset$. We denote the inclusion functor by $I_Q : \mathcal{L}_Q \longrightarrow \mathcal{L}$. We write $\mathcal{L}_{cl}$ for $\mathcal{L}_Q$ when $Q = \emptyset$ and call this the category of **closed** lambda terms.

We claim that the inclusion $I_Q$ has a right adjoint, provided $Q$ is **cofinite**, by which we mean that $Q^c = Y \setminus Q$ is a finite set. Our convention is to use letters $\mathfrak{p}, \mathfrak{q}, \ldots$ for ordered sets of variables, with $\mathfrak{q}$ always denoting an ordering on the finite unordered set of variables $Q^c$. With this notation, we next define a functor

$$\Gamma_{\mathfrak{q}} : \mathcal{L} \longrightarrow \mathcal{L}_Q$$

which we will prove is right adjoint to $I_Q$, with counit a natural transformation

$$\mathscr{U}^{\mathfrak{q}} : I_Q \circ \Gamma_{\mathfrak{q}} \longrightarrow 1_{\mathcal{L}}\,.$$

For the rest of this section let $Q$ be a cofinite set of variables and $\mathfrak{q} = (q_1 : \tau_1, \ldots, t_k : q_k : \tau_k)$ an ordering of the complement. While the functor $\Gamma_{\mathfrak{q}}$ and natural transformation $\mathscr{U}^{\mathfrak{q}}$ depend on the choice of ordering, by the uniqueness of adjoints they are independent of the ordering up to unique natural isomorphism.

**Definition 1.8.** For a type $\rho$ we define

$$\Gamma_{\mathfrak{q}}(\rho) = \tau_1 \to \tau_2 \to \cdots \to \tau_k \to \rho$$

which is $\rho$ if $Q$ is empty. We set $\Gamma_{\mathfrak{q}}(\mathbf{1}) = \mathbf{1}$. For types $\sigma, \tau$ we define a function

$$\Gamma_{\mathfrak{q}} : \mathcal{L}(\sigma,\tau) \longrightarrow \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\sigma, \Gamma_{\mathfrak{q}}\tau) \tag{1.8}$$

on a term $M : \sigma \to \tau$ by

$$\Gamma_{\mathfrak{q}}(M) = \lambda U^{\tau_1 \to \cdots \to \tau_k \to \sigma} q_1^{\tau_1} \cdots q_k^{\tau_k} \,.\, \big(M(\cdots(U q_1)\cdots q_k)\big)\,. \tag{1.9}$$

Since it is clear by inspection that $\mathrm{FV}_\beta(\Gamma_{\mathfrak{q}} M) \subseteq \mathrm{FV}_\beta(M) \setminus Q^c$ we have $\Gamma_{\mathfrak{q}} M \in \mathcal{L}_Q$. In the special cases involving $\mathbf{1}$ we define $\Gamma_{\mathfrak{q}}$ by

$$\begin{aligned}
\mathcal{L}(\sigma,\mathbf{1}) \longrightarrow \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\sigma, \Gamma_{\mathfrak{q}}\mathbf{1}) &= \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\sigma, \mathbf{1})\,, && \star \mapsto \star\\
\mathcal{L}(\mathbf{1},\rho) \longrightarrow \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\mathbf{1}, \Gamma_{\mathfrak{q}}\rho) &= \mathcal{L}_Q(\mathbf{1}, \Gamma_{\mathfrak{q}}\rho)\,, && M \mapsto \lambda q_1^{\tau_1} \cdots q_k^{\tau_k} \,.\, M\\
\mathcal{L}(\mathbf{1},\mathbf{1}) \longrightarrow \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\mathbf{1}, \Gamma_{\mathfrak{q}}\mathbf{1}) &= \mathcal{L}_Q(\mathbf{1},\mathbf{1}) && \star \mapsto \star\,.
\end{aligned}$$

**Remark 1.9.** It is important in (1.9) that we lambda abstract over the particular variables $q_i$ that belong to $Q^c$. By $\alpha$-equivalence the result of a lambda abstraction is independent of the variable we use *if* the term being lambda abstracted does not contain that variable as a free variable. However we are certainly interested in the case where $M$ *does* contain the $q_i$ as free variables, and in these cases $\Gamma_{\mathfrak{q}}(M)$ defined using, say, a sequence of variables $v_1^{\tau_1}, \dots, v_k^{\tau_k}$ distinct from $\mathfrak{q}$ would be a different morphism in $\mathcal{L}$.

**Lemma 1.10.** *$\Gamma_{\mathfrak{q}}$ is a functor $\mathcal{L} \longrightarrow \mathcal{L}_Q$.*

With the same notation as in Definition 1.8:

**Definition 1.11.** For a type $\rho$ we define $\mathscr{U}_\rho^{\mathfrak{q}} \in \mathcal{L}(\Gamma_{\mathfrak{q}}\rho, \rho)$ by

$$\mathscr{U}_\rho^{\mathfrak{q}} = \lambda U^{\Gamma_{\mathfrak{q}}\rho} \,.\, (\cdots((U q_1) q_2) \cdots q_k) \,. \tag{1.10}$$

Once again, it is significant that we use the sequence of variables $\mathfrak{q}$ to form this term, and not arbitrary variables of the same type. The special case is $\mathscr{U}_{\mathbf{1}}^{\mathfrak{q}} \in \mathcal{L}(\Gamma_{\mathfrak{q}}\mathbf{1}, \mathbf{1}) = \mathcal{L}(\mathbf{1}, \mathbf{1})$ given by $\mathscr{U}_{\mathbf{1}}^{\mathfrak{q}} = \star$.

**Proposition 1.12.** *Given types $\tau_1, \dots, \tau_k, \sigma, \rho$ and a permutation $\theta \in S_k$, the term*

$$\begin{gathered} P_\theta : \big(\tau_1 \to \cdots \to \tau_k \to \rho\big) \to \big(\tau_{\theta(1)} \to \cdots \to \tau_{\theta(k)} \to \rho\big) \\ P_\theta = \lambda U^{\tau_1 \to \cdots \to \tau_k \to \rho} v_1^{\tau_{\theta(1)}} v_2^{\tau_{\theta(2)}} \cdots v_k^{\tau_{\theta(k)}} \,.\, \big(\cdots((U v_{\theta^{-1}(1)}) v_{\theta^{-1}(2)}) \cdots v_{\theta^{-1}(k)}\big) \end{gathered}$$

*is an isomorphism in $\mathcal{L}$ between the objects*

$$\big(\tau_1 \to \cdots \to \tau_k \to \rho\big) \cong \big(\tau_{\theta(1)} \to \cdots \to \tau_{\theta(k)} \to \rho\big) \,.$$

With the notation of the proposition:

**Corollary 1.13.** *There is a bijection*

$$\Lambda_{\tau_1 \to \cdots \to \tau_k \to \rho} / =_{\beta\eta} \xrightarrow{\ \cong\ } \Lambda_{\tau_{\theta(1)} \to \cdots \to \tau_{\theta(k)} \to \rho} / =_{\beta\eta} \ .$$

*Proof.* We have, by the proposition

$$\begin{aligned} \Lambda_{\tau_1 \to \cdots \to \tau_k \to \rho} / =_{\beta\eta} &= \mathcal{L}(\mathbf{1}, \tau_1 \to \cdots \to \tau_k \to \rho) \\ &\cong \mathcal{L}(\mathbf{1}, \tau_{\theta(1)} \to \cdots \to \tau_{\theta(k)} \to \rho) \\ &= \Lambda_{\tau_{\theta(1)} \to \cdots \to \tau_{\theta(k)} \to \rho} / =_{\beta\eta} \ . \end{aligned}$$

□

### 1.1 Structural rules and monads

As above, let $\mathcal{L}_{cl}$ denote the category of closed lambda terms. Throughout this section, $A \subseteq Y$ is finite and so there is a right adjoint $\Gamma_{\mathfrak{a}}$ to the inclusion $I$ for any ordering $\mathfrak{a}$ of $A$:

$$\mathcal{L}_{cl} \underset{\Gamma_{\mathfrak{a}}}{\overset{I}{\rightleftarrows}} \mathcal{L}_A \ . \tag{1.11}$$

**Definition 1.14.** Denote by $T_{\mathfrak{a}}$ the composition $\Gamma_{\mathfrak{a}} \circ I$ on $\mathcal{L}_{cl}$.

In the case where $\mathfrak{a} = \{x : \alpha\}$ we define the monad $T_{\mathfrak{a}}$ to have multiplication $\mu$ given by

$$\mu_\sigma = \lambda u^{\alpha \to (\alpha \to \sigma)} x^\alpha . ((ux)x) : (\alpha \to (\alpha \to \sigma)) \to (\alpha \to \sigma)$$

and unit $\xi$ given by

$$\xi_\sigma = \lambda w^\sigma x^\alpha . w : \sigma \to (\alpha \to \sigma)\,.$$

Let $\mathfrak{a}, \mathfrak{b}$ be *disjoint* finite ordered sets of variables, and $T_{\mathfrak{a}}, T_{\mathfrak{b}}$ the associated monads on $\mathcal{L}_{cl}$. There is a distributive law between these two monads, and their composition as functors is therefore naturally equipped with the structure of a monad. For simplicity, we write down the propositions only in the case where $\mathfrak{a} = \{x : \alpha\}$ and $\mathfrak{b} = \{y : \beta\}$ are singletons.

**Lemma 1.15.** *With the induced monad structure the composite $T_{\mathfrak{a}} T_{\mathfrak{b}}$ is isomorphic, as a monad, to $T_{\mathfrak{a}:\mathfrak{b}}$ where $\mathfrak{a} : \mathfrak{b}$ denotes concatenation of sequences.*

## 2 Questions

**Question 1.** *Prove Lemma 1.4, you may use Lemma 1.2 in your proof.*

**Question 2.** *Prove that $\mathscr{U}^{\mathfrak{q}}$ is a natural transformation $I_Q \circ \Gamma_{\mathfrak{q}} \longrightarrow 1_{\mathcal{L}}$ in the special case where $\mathfrak{q} = \{q : \tau\}$.*

### 2.1 Extension questions (requires adjoints and monads)

**Question 3.** *Prove that $\Gamma_{\mathfrak{q}}$ is right adjoint to $I_Q$ with counit $\mathscr{U}^{\mathfrak{q}}$ by showing that for types $\sigma, \rho$ there are natural bijections*

$$\mathcal{L}(\sigma, \rho) = \mathcal{L}(I_Q(\sigma), \rho) \cong \mathcal{L}_Q(\sigma, \Gamma_{\mathfrak{q}} \rho)\,, \tag{2.1}$$

$$\mathcal{L}(\mathbf{1}, \rho) = \mathcal{L}(I_Q(\mathbf{1}), \rho) \cong \mathcal{L}_Q(\mathbf{1}, \Gamma_{\mathfrak{q}} \rho)\,. \tag{2.2}$$

*You can use Corollary 1.13 in your proof.*

**Question 4.** *Prove that the monads $T_{\mathfrak{a}}, T_{\mathfrak{b}}$ admit a distributive law*

$$\chi : T_{\mathfrak{a}} T_{\mathfrak{b}} \longrightarrow T_{\mathfrak{b}} T_{\mathfrak{a}}$$

$$\chi_\sigma = \lambda z^{\alpha \to (\beta \to \sigma)} y^\beta x^\alpha . ((zx)y)\,.$$

# A Background on lambda calculus

**Definition A.1.** *Let $\mathscr{V}$ be a (countably) infinite set of variables, and let $\mathscr{L}$ be the language consisting of $\mathscr{V}$ along with the special symbols*

$$\lambda \qquad . \qquad ( \qquad )$$

*Let $\mathscr{L}^*$ be the set of words of $\mathscr{L}$, more precisely, an element $w \in \mathscr{L}^*$ is a finite sequence $(w_1, ..., w_n)$ where each $w_i$ is in $\mathscr{L}$, for convenience, such an element will be written as $w_1...w_n$. Now let $\Lambda'$ denote the smallest subset of $\mathscr{L}^*$ such that*

- *if $x \in \mathscr{V}$ then $x \in \Lambda'$,*
- *if $M, N \in \Lambda'$ then $(MN) \in \Lambda'$,*
- *if $x \in \mathscr{V}$ and $M \in \Lambda'$ then $(\lambda x.M) \in \Lambda'$*

*$\Lambda'$ is the set of* ***preterms****. A preterm $M$ such that $M \in \mathscr{V}$ is a* ***variable****, if $M = (M_1M_2)$ for some preterms $M_1, M_2$, then $M$ is an* ***application****, and if $M = (\lambda x, M')$ for some $x \in \mathscr{V}$ and $M' \in \Lambda'$ then $M$ is an* ***abstraction****.*

**Definition A.2.** ***Single step** $\beta$**-reduction** $\to_\beta$ is the smallest relation on $\Lambda$ satisfying:*

- *the* ***reduction axiom****:*
  - *for all variables $x$ and $\lambda$-terms $M, M'$, $(\lambda x.M)M' \to_\beta M[x := M']$, where $M[x := M']$ is the term given by replacing every free occurrence of $x$ in $M$ with $M'$,*
- *the following* ***compatibility axioms****:*
  - *if $M \to_\beta M'$ then $(MN) \to_\beta (M'N)$ and $(NM) \to_\beta (NM')$,*
  - *if $M \to_\beta M'$ then for any variable $x$, $\lambda x.M \to_\beta \lambda xM'$.*

*A subterm of the form $(\lambda x.M)M'$ is a $\beta$**-redex**, and $(\lambda x.M)M'$* ***single step** $\beta$**-reduces** *to $M'$.*

**Definition A.3.** ***Multi step** $\beta$**-reduction** $\twoheadrightarrow$ (or simply $\beta$**-reduction**) is the smallest relation on $\Lambda$ satisfying*

- *the* ***reduction axiom****:*
  - *if $M \to_\beta M'$ then $M \twoheadrightarrow M'$,*
- ***reflexivity****:*
  - *if $M = M'$ then $M \twoheadrightarrow M'$,*

- ***transitivity***:
  - *if* $M_1 \twoheadrightarrow M_2$ *and* $M_2 \twoheadrightarrow M_3$ *then* $M_1 \twoheadrightarrow M_3$

*If* $M \twoheadrightarrow M'$*, then* $M$ ***multistep*** $\beta$***-reduces*** *to* $M[x := M']$*.*

*The reflexive, symmetric closure of multistep* $\beta$*-reduction is* $\beta$***-equivalence****. That is, the smallest relation containing multi step* $\beta$*-reduction which is reflexive and symmetric.*

There is also $\eta$-expansion, which is defined similarly, we are more terse in Definition A.4 than in Definition A.3.

**Definition A.4.** ***Single step*** $\eta$***-expansion*** $\longrightarrow_\eta$ *is the smallest, compatible relation on* $\Lambda$ *satisfying:*

$$M \longrightarrow_\eta \lambda x.Mx \tag{A.1}$$

*where* $x$ *is a variable not in the free variable set of* $M$*.* ***Multi step*** $\eta$***-expansion*** *is the reflexive closure of single step* $\eta$*-expansion.* $\eta$***-equivalence*** *is the reflexive, symmetric symmetric closure of multi step* $\eta$*-expansion.*

$\beta\eta$***-equivalence*** *is the union of* $\eta$*-equivalence and* $\beta$*-equivalence.*

In the simply-typed lambda calculus [1, Chapter 3] there is an infinite set of **atomic types** and the set $\Phi_\to$ of **simple types** is built up from the atomic types using $\to$. Let $\Lambda'$ denote the set of untyped lambda calculus preterms in these variables, as defined in [1, Chapter 1]. We define a subset $\Lambda'_{wt} \subseteq \Lambda'$ of **well-typed** preterms, together with a function $t : \Lambda'_{wt} \longrightarrow \Phi_\to$ by induction:

- all variables $x : \sigma$ are well-typed and $t(x) = \sigma$,
- if $M = (P\,Q)$ and $P, Q$ are well-typed with $t(P) = \sigma \to \tau$ and $t(Q) = \sigma$ for some $\sigma, \tau$ then $M$ is well-typed and $t(M) = \tau$,
- if $M = \lambda x\,.\,N$ with $N$ well-typed, then $M$ is well-typed and $T(M) = t(x) \to t(N)$.

We define $\Lambda'_\sigma = \{M \in \Lambda'_{wt} \,|\, t(M) = \sigma\}$ and call these **preterms of type** $\sigma$. Next we observe that $\Lambda'_{wt} \subseteq \Lambda'$ is closed under the relation of $\alpha$-equivalence on $\Lambda'$, as long as we understand $\alpha$-equivalence type by type, that is, we take

$$\lambda x\,.\,M =_\alpha \lambda y\,.\,M[x := y]$$

as long as $t(x) = t(y)$. Denoting this relation by $=_\alpha$, we may therefore define the sets of **well-typed lambda terms** and **well-typed lambda terms of type** $\sigma$, respectively:

$$\Lambda_{wt} = \Lambda'_{wt} / =_\alpha \tag{A.2}$$

$$\Lambda_\sigma = \Lambda'_\sigma / =_\alpha \;. \tag{A.3}$$

Note that $\Lambda_{wt}$ is the disjoint union over all $\sigma \in \Phi_\to$ of $\Lambda_\sigma$. We write $M : \sigma$ as a synonym for $[M] \in \Lambda_\sigma$, and call these equivalence classes **terms of type** $\sigma$. Since terms are, by definition, $\alpha$-equivalence classes, the expression $M = N$ henceforth means $M =_\alpha N$ unless indicated otherwise. We denote the set of free variables of a term $M$ by $\mathrm{FV}(M)$.

**Definition A.5.** The substitution operation on lambda terms is a family of functions

$$\left\{ \text{subst}_\sigma : Y_\sigma \times \Lambda_\sigma \times \Lambda_{wt} \longrightarrow \Lambda_{wt} \right\}_{\sigma \in \Phi_\rightarrow}$$

We write $M[x := N]$ for $\text{subst}_\sigma(x, N, M)$ and this term is defined inductively (on the structure of $M$) as follows:

- if $M$ is a variable then either $M = x$ in which case $M[x := N] = N$, or $M \neq x$ in which case $M[x := N] = M$.
- if $M = (M_1\, M_2)$ then $M[x := N] = \big(M_1[x := N]\, M_2[x := N]\big)$.
- if $M = \lambda y.L$ we may assume by $\alpha$-equivalence that $y \neq x$ and that $y$ does not occur in $N$ and set $M[x := N] = \lambda y.L[x := N]$.

Note that if $x \notin \text{FV}(M)$ then $M[x := N] = M$.

# References


[1] M. Sørensen and P. Urzyczyn, *Lectures on the Curry-Howard isomorphism*, Studies in Logic and the Foundations of Mathematics Vol. 149, Elsevier New York, (2006).

[2] W. Troiani, *An Introduction to the Untyped λ-Calculus and the Church-Rosser Theorem*, https://williamtroiani.github.io/pdfs/ChurchRosserTheorem.pdf