

Chapter 2

Untyped Lambda Calculus

We assume the existence of a denumerable set **VAR** of (object) variables x_0, x_1, x_2, \dots , and use x, y, z to range over these variables. Given two variables x_1 and x_2 , we write $x_1 = x_2$ if both x_1 and x_2 denote the same x_n for some natural number n ; similarly, we write $x_1 < x_2$ ($x_1 \leq x_2$) if x_1 and x_2 denote x_{n_1} and x_{n_2} , respectively, for some natural numbers n_1 and n_2 satisfying $n_1 < n_2$ ($n_1 \leq n_2$); we write $x_1 > x_2$ ($x_1 \geq x_2$) to mean $x_2 < x_1$ ($x_2 \leq x_1$).

Definition 2.0.1 (λ -terms) *The (pure) λ -terms are formally defined below:*

$$\text{terms } t ::= x \mid \lambda x.t \mid t_1(t_2)$$

We use **TERM** for the set of all λ -terms. Given a λ -term t , t is either a variable, or a λ -abstraction of the form $\lambda x.t_1$, or an application of the form $t_1(t_2)$. We write $t_1 \equiv t_2$ to mean that t_1 and t_2 are syntactically the same.

When giving examples, We often use I for $\lambda x.x$, K for $\lambda x.\lambda y.x$, K' for $\lambda x.\lambda y.y$, and S for $\lambda x.\lambda y.\lambda z.x(z)(y(z))$. Many other special λ -terms are to be introduced.

Definition 2.0.2 (Size of λ -terms) *We define a unary function $size(\cdot)$ to compute the size of a given λ -term:*

$$\begin{aligned} size(x) &= 0 \\ size(\lambda x.t) &= 1 + size(t) \\ size(t_1(t_2)) &= 1 + size(t_1) + size(t_2) \end{aligned}$$

Clearly, we have $size(I) = 1$, $size(K) = 2$ and $size(S) = 6$.

There is often a need to refer to a subterm in a given λ -term. For this purpose, we introduce *paths* defined as finite sequences of natural numbers:

$$\text{paths } p ::= \emptyset \mid n.p$$

We use \emptyset for the empty sequence and $n.p$ for the sequence whose head and tail are n and p , respectively, where n ranges over natural numbers. Given two paths p_1 and p_2 , we write $p_1 @ p_2$ for the concatenation of p_1 and p_2 . We say that p_1 is a prefix of p_2 if $p_2 = p_1 @ p_3$ for some path p_3 ; this prefix is proper if p_3 is not empty. We say that p_1 and p_2 are incompatible if neither of them is the prefix of the other.

We use **PATH** for the set of all paths and \bar{p} to range over finite sets of paths. Given n and \bar{p} , we use $n.\bar{p}$ for the set $\{n.p \mid p \in \bar{p}\}$. Given p_0 and \bar{p} , the sets $p_0@p$ and $\bar{p}@p_0$ are $\{p_0@p \mid p \in \bar{p}\}$ and $\{p@p_0 \mid p \in \bar{p}\}$, respectively.

Definition 2.0.3 We define as follows a partial binary function $subterm(\cdot, \cdot)$ from **(TERM, PATH)** to **TERM**:

$$\begin{aligned} subterm(t, \emptyset) &= t \\ subterm(t_1(t_2), 0.p) &= subterm(t_1, p) \\ subterm(t_1(t_2), 1.p) &= subterm(t_2, p) \\ subterm(\lambda x.t, 0.p) &= subterm(t, p) \end{aligned}$$

Given two λ -terms t_1, t_2 and a path p , we say that t_1 is a *subterm of t_2 at p* if $subterm(t_2, p) = t_1$; this subterm is proper if p is not empty. We may simply say that t_1 is a subterm of t_2 if $subterm(t_2, p) = t_1$ for some path p . Also, we may say that t_1 has an occurrence in t_2 (at p) if t_1 is a subterm of t_2 (at p). Note that for a λ -term of the form $\lambda x.t$, the variable x following the binder λ does not count as an occurrence (in the formal sense).

Given a λ -term, we use $paths(t)$ for the set of paths such that $p \in paths(t)$ if and only if $subterm(t, p)$ is defined. Clearly, for every λ -term t , we have

- $\emptyset \in paths(t)$, and
- $p_0 \in paths(t)$ implies that $p \in paths(t)$ holds for every prefix p of p_0 .

Note that for every λ -term t , $p \in paths(t)$ implies p being a sequence of 0's and 1's.

Definition 2.0.4 (Variable Set) We define a function $vars$ as follows that maps λ -terms to finite sets of variables:

$$\begin{aligned} vars(x) &= \{x\} \\ vars(\lambda x.t) &= vars(t) \cup \{x\} \\ vars(t_1(t_2)) &= vars(t_1) \cup vars(t_2) \end{aligned}$$

Clearly, for every λ -term t_0 , $x \in vars(t_0)$ if and only if t_0 has a subterm of the form x or $\lambda x.t$.

Definition 2.0.5 (Free Variable Set) We define a function FV as follows that maps λ -terms to finite sets of variables:

$$\begin{aligned} FV(x) &= \{x\} \\ FV(\lambda x.t) &= FV(t) \setminus \{x\} \\ FV(t_1(t_2)) &= FV(t_1) \cup FV(t_2) \end{aligned}$$

Given a λ -term t , we refer to $FV(t)$ as the set of free variables in t . We say that a variable x is free in t if and only if $x \in FV(t)$ holds.

Given a λ -term t_0 and a variable x , an occurrence of x in t_0 at p_0 is a free occurrence if $subterm(t_0, p)$ is not of the form $\lambda x.t$ for any prefix p of p_0 . It is clear from the definition of FV that $x \in FV(t)$ if and only if x has at least one free occurrence in t .

Definition 2.0.6 (Variable Replacement) Given a λ -term t and two variables x and y , we define $t[y/x]$ as follows by structural induction on t :

$$\begin{aligned} x[y/x] &::= y \\ x'[y/x] &::= x' \text{ if } x' \text{ is not } x \\ t_1(t_2)[y/x] &::= t_1[y/x](t_2[y/x]) \\ (\lambda x.t)[y/x] &::= \lambda x.t \\ (\lambda x'.t)[y/x] &::= \lambda x'.t[y/x] \text{ if } x \neq x' \end{aligned}$$

We refer to $t[y/x]$ as the λ -term obtained from replacing (free occurrences) of x with y in t .

Clearly, $\text{size}(t[y/x]) = \text{size}(t)$ for all λ -terms t and variables x and y .

Proposition 2.0.7 We have the following.

1. $t[x/x] \equiv t$.
2. $t[y/x] \equiv t$ if $x \notin FV(t)$.
3. $t[y/x][z/y] \equiv t[z/x]$ if $y \notin \text{vars}(t)$.

Proof Both (1) and (2) are straightforward. We prove (3) by structural induction on t .

- t is x . Then both $t[y/x][z/y] \equiv z$ and $t[z/x] \equiv z$ hold, and we are done.
- t is x' for some variable $x' \neq x$. Then $x' \neq y$ also holds as $y \notin \text{vars}(t)$. So $t[y/x][z/y] \equiv x'$ and $t[z/x] \equiv x'$, and we are done.
- t is $t_1(t_2)$. For $i = 1, 2$, we have $t_i[y/x][z/y] \equiv t_i[z/x]$ by induction hypotheses on t_i . Therefore, $t[y/x][z/y] \equiv t[z/x]$ holds as well.
- t is $\lambda x.t_0$. Then $t[y/x][z/y] \equiv t[z/y]$, and $t[z/x] \equiv t$. By (2), $t[z/y] \equiv t$ holds, and we are done.
- t is $\lambda x'.t_0$ for some $x' \neq x$. We have $t_0[y/x][z/y] \equiv t_0[z/x]$ by induction hypothesis on t_0 . Note that $x' \neq y$ since $y \notin \text{vars}(t)$. So we have $t[y/x][z/y] \equiv t[z/x]$.

We conclude the proof as all the cases are covered. ■

Definition 2.0.8 We use Γ for a sequence of variables defined as follows:

$$\Gamma ::= \emptyset \mid \Gamma, x$$

We write $x \in \Gamma$ to indicate that x occurs in Γ , and $|\Gamma|$ for the length of Γ , that is, the number of variables in Γ . If $x \in \Gamma$ holds, we define $\Gamma(x)$ as follows: $\Gamma(x) = |\Gamma|$ if $\Gamma = \Gamma_1, x$ and $\Gamma(x) = \Gamma_1(x)$ if $\Gamma = \Gamma_1, x_1$ for some $x_1 \neq x$.

Definition 2.0.9 (α -normal forms) We use \underline{t} for α -normal forms defined as follows:

$$\alpha\text{-normal forms } \underline{t} ::= x \mid n \mid \lambda(\underline{t}) \mid \underline{t}_1(\underline{t}_2)$$

where n ranges over positive integers.

Given an α -normal form \underline{t} , $shift(\underline{t})$ is the α -normal form obtained from increasing each n in \underline{t} by 1. Formally, we have

$$shift(x) = x; shift(n) = n + 1; shift(\underline{t}_1(\underline{t}_2)) = shift(\underline{t}_1)(shift(\underline{t}_2)); shift(\lambda(\underline{t}_1)) = \lambda(shift(\underline{t}_1))$$

Definition 2.0.10 (α -equivalence) Given a sequence Γ of variables and a term t , $NF_\alpha(\Gamma; t)$ is defined inductively as follows:

$$NF_\alpha(\Gamma; t) = \begin{cases} x & \text{if } t = x \text{ for some } x \notin \Gamma; \\ \Gamma(x) & \text{if } t = x \text{ for some } x \in \Gamma; \\ \lambda(\underline{t}_0) & \text{if } t = \lambda x.t_0 \text{ and } \underline{t}_0 = NF_\alpha(\Gamma, x; t_0); \\ \underline{t}_1(\underline{t}_2) & \text{if } t = t_1(t_2) \text{ and } \underline{t}_1 = NF_\alpha(\Gamma; t_1) \text{ and } \underline{t}_2 = NF_\alpha(\Gamma; t_2). \end{cases}$$

We use $NF_\alpha(t)$ as a shorthand for $NF_\alpha(\emptyset; t)$. Given two terms t_1 and t_2 , we say that t_1 and t_2 are α -equivalent if $NF_\alpha(t_1) \equiv NF_\alpha(t_2)$ holds, and we use $t_1 \equiv_\alpha t_2$ to indicate that t_1 and t_2 are α -equivalent. Note that \equiv_α is an equivalence relation, that is, \equiv_α is reflexive, symmetric and transitive.

Clearly, we have $NF_\alpha(I) = \lambda(1)$, $NF_\alpha(K) = \lambda(\lambda(1))$, and $NF_\alpha(S) = \lambda(\lambda(\lambda(1(3)(2(3))))))$, and note that $\lambda(shift(NF_\alpha(I))) = \lambda(\lambda(2)) = NF_\alpha(K')$.

Given \underline{t} and x , we use $\underline{t}[1/x]$ and $\underline{t}[y/x]$ for the α -normal forms obtained from replacing each occurrence of x in \underline{t} with 1 and y , respectively. For brevity, the formal definitions for these replacements are omitted.

Proposition 2.0.11 For every λ -abstract $\lambda x.t$, we have

$$NF_\alpha(\lambda x.t) = \lambda(shift(NF_\alpha(t))[1/x])$$

Proof Let us first establish the following equation for all sequences Γ :

$$NF_\alpha(x, \Gamma; t) = shift(NF_\alpha(\Gamma, t))[1/x]$$

We proceed by structural induction on t :

- t is x . If $x \in \Gamma$, then both sides of the equation equal $\Gamma(x) + 1$. Otherwise, both sides of the equation equal 1.
- t is some variable y that is distinct from x . If $y \in \Gamma$, then both sides of the equation equal $\Gamma(y) + 1$. Otherwise, both sides of the equation equal y .
- t is of the form $\lambda x_1.t_1$. By definition, $NF_\alpha(x, \Gamma; t) = \lambda(NF_\alpha(x, \Gamma, x_1; t_1))$. By induction hypothesis on t_1 , we have the following:

$$NF_\alpha(x, \Gamma, x_1; t_1) = shift(NF_\alpha(\Gamma, x_1; t_1))[1/x]$$

Note that we have:

$$\lambda(shift(NF_\alpha(\Gamma, x_1; t_1))[1/x]) = shift(\lambda(NF_\alpha(\Gamma, x_1; t_1)))[1/x]$$

Hence, $NF_\alpha(x, \Gamma; t) = shift(NF_\alpha(\Gamma; t))[1/x]$ holds.

- t is of the form $t_1(t_2)$. This is straightforward based on the properties of $NF_\alpha(\cdot)$, $shift(\cdot)$ and variable replacement.

We conclude the inductive proof as all the cases are covered. Let Γ be \emptyset , and we have $NF_\alpha(x; t) = shift(NF_\alpha(t))[1/x]$. Therefore, we have $NF_\alpha(\lambda x.t) = \lambda(shift(NF_\alpha(t))[1/x])$. ■

Given a λ -abstraction $\lambda x.t$, let us choose a variable y not in $vars(t)$. By Proposition 2.0.11, we have

$$NF_\alpha(\lambda x.t) = \lambda(shift(NF_\alpha(t))[1/x]) \quad \text{and} \quad NF_\alpha(\lambda y.t[y/x]) = \lambda(shift(NF_\alpha(t[y/x]))[1/y])$$

It should be easy to note that $shift(NF_\alpha(t))[1/x] = shift(NF_\alpha(t[y/x]))[1/y]$. Therefore, $\lambda x.t$ and $\lambda y.t[y/x]$ are α -equivalent.

Definition 2.0.12 (*Substitutions*) We use θ for substitutions, which are finite mappings from variables to λ -terms:

$$\text{substitutions } \theta ::= [] \mid \theta[x \mapsto t]$$

We may use $[]$ for the empty mapping and $\theta[x \mapsto t]$ for the mapping that extends θ with a link from x to t , where x is assumed to be not in $\mathbf{dom}(\theta)$. We use $\mathbf{dom}(\theta)$ for the (finite) domain of θ and $vars(\theta)$ for the following (finite) set of variables:

$$\mathbf{dom}(\theta) \cup (\cup_{x \in \mathbf{dom}(\theta)} vars(\theta(x)))$$

We may use $[x_1 \mapsto t_1, \dots, x_n \mapsto t_n]$ for the substitution θ such that $\mathbf{dom}(\theta) = \{x_1, \dots, x_n\}$ and $\theta(x_i) = t_i$ for $1 \leq i \leq n$, where x_1, \dots, x_n are assumed to be distinct variables.

Definition 2.0.13 Given a λ -term t and a substitution θ , we use $t[\theta]$ for the result of applying the substitution θ to t , which is formally defined below as a function by induction on the size of t :

- $t[\theta] = \theta(x)$ if t is some x in $\mathbf{dom}(\theta)$.
- $t[\theta] = x$ if t is some x not in $\mathbf{dom}(\theta)$.
- $t[\theta] = \lambda y.(t_1[y/x])[\theta]$ if t is $\lambda x.t_1$, where y is the first variable not in $vars(t_1) \cup vars(\theta)$. Note that the reason for choosing y in such a manner is to guarantee that applying a substitution θ to a term t can be done deterministically.
- $t[\theta] = t_1[\theta](t_2[\theta])$ if $t = t_1(t_2)$.

Given two substitutions θ_1 and θ_2 , we write $\theta_1 \equiv_\alpha \theta_2$ to mean that $\theta_1(x) \equiv_\alpha \theta_2(x)$ holds for every $x \in \mathbf{dom}(\theta_1) = \mathbf{dom}(\theta_2)$. We are to prove that $t[\theta] \equiv_\alpha t'[\theta']$ whenever $\theta \equiv_\alpha \theta'$ and $t \equiv_\alpha t'$, that is, the operation of applying a substitution to a term is well-defined modulo α -equivalence.

Let us use $\underline{\theta}$ for finite mappings from variables to α -normal forms and $shift(\underline{\theta})$ be the mapping $\underline{\theta}'$ such that $\mathbf{dom}(\underline{\theta}') = \mathbf{dom}(\underline{\theta})$ and $\underline{\theta}'(x) = shift(\underline{\theta}(x))$ for each $x \in \mathbf{dom}(\underline{\theta}')$. Given \underline{t} , we define $\underline{t}[\underline{\theta}]$ as follows:

$$\underline{t}[\underline{\theta}] = \begin{cases} \underline{\theta}(x) & \text{if } t = x; \\ n & \text{if } t = n; \\ \lambda(\underline{t}_1[\underline{\theta}']) & \text{if } t = \lambda(\underline{t}_1) \text{ and } \underline{\theta}' = shift(\underline{\theta}); \\ \underline{t}_1[\underline{\theta}](\underline{t}_2[\underline{\theta}]) & \text{if } t = t_1(t_2). \end{cases}$$

Proposition 2.0.14 *Given x , t and θ , if y is a variable not in $\text{vars}(t) \cup \text{vars}(\theta)$, then we have the following equation:*

$$\text{shift}(\underline{t}[y/x][\underline{\theta}])[1/y] = \text{shift}(\underline{t})[1/x][\text{shift}(\underline{\theta})]$$

Proof We proceed by structural induction on \underline{t} . For brevity, we only consider the case where t is of the form $\lambda(\underline{t}_1)$. Note that $\text{shift}(\underline{t}[y/x][\underline{\theta}]) = \lambda(\text{shift}(\underline{t}_1[y/x][\underline{\theta}']))$ in this case, where $\underline{\theta}' = \text{shift}(\underline{\theta})$. By induction hypothesis on \underline{t}_1 , we have:

$$\text{shift}(\underline{t}_1[y/x][\underline{\theta}'])[1/y] = \text{shift}(\underline{t}_1)[1/x][\text{shift}(\underline{\theta}')$$

Note that $\text{shift}(\underline{t})[1/x][\text{shift}(\underline{\theta})] = \lambda(\text{shift}(\underline{t}_1)[1/x][\underline{\theta}']) = \lambda(\text{shift}(\underline{t}_1)[1/x][\text{shift}(\underline{\theta}'))$, and we have

$$\text{shift}(\underline{t}[y/x][\underline{\theta}])[1/y] = \lambda(\text{shift}(\underline{t}_1[y/x][\underline{\theta}'])[1/y]) = \text{shift}(\underline{t})[1/x][\text{shift}(\underline{\theta})]$$

All of the other cases can be readily handled. ■

Proposition 2.0.15 *We have the following equation:*

$$NF_\alpha(t[\theta]) = NF_\alpha(t)[NF_\alpha(\theta)]$$

In other words, the substitution function given in Definition 2.0.13 is well-defined modulo the α -equivalence relation.

Proof Let $\underline{\theta} = NF_\alpha(\theta)$. We proceed by induction on the size of t . The only interesting case is the one where t is of the form $\lambda x_1.t_1$. By definition, $t[\theta] = \lambda y.t_1[y/x_1][\theta]$, where y is some variable not appearing in $\text{vars}(t_1) \cup \text{vars}(\theta)$. By induction hypothesis on $t_1[y/x_1]$, $NF_\alpha(t_1[y/x_1][\theta]) = NF_\alpha(t_1[y/x_1][\underline{\theta}])$ holds. Let $\underline{t}_1 = NF_\alpha(t_1)$, and we have $NF_\alpha(t_1[y/x_1]) = \underline{t}_1[y/x_1]$. By Proposition 2.0.14, we have

$$\lambda(\text{shift}(NF_\alpha(t_1[y/x_1][\theta]))[1/y]) = \lambda(\text{shift}(\underline{t}_1[y/x_1][\underline{\theta}])[1/y]) = \lambda(\text{shift}(\underline{t}_1)[1/x_1][\text{shift}(\underline{\theta})])$$

Note that $NF_\alpha(t) = \lambda(\text{shift}(\underline{t}_1)[1/x_1])$, which leads to $NF_\alpha(t)[\underline{\theta}] = \lambda(\text{shift}(\underline{t}_1)[1/x_1][\text{shift}(\underline{\theta})])$. So we have $NF_\alpha(t[\theta]) = NF_\alpha(t)[NF_\alpha(\theta)]$ in this case. All of the other cases can be readily handled. ■

Given θ_1 and θ_2 , we use $\theta_2 \circ \theta_1$ for the substitution θ such that $\text{dom}(\theta) = \text{dom}(\theta_1) \cup \text{dom}(\theta_2)$, and for each $x \in \text{dom}(\theta)$, $\theta(x) = x[\theta_1][\theta_2]$.

Lemma 2.0.16 $(t[\theta_1])[\theta_2] \equiv_\alpha t[\theta_2 \circ \theta_1]$.

Proof As an exercise. ■

Given a λ -abstraction $\lambda x.t$ and a finite set of variables, we can also choose another λ -abstraction $\lambda x'.t'$ that is α -equivalent to $\lambda x.t$ while guaranteeing that x' does not occur in the given finite set of variables. This is often called α -conversion or α -renaming (of a bound variable).

Definition 2.0.17 (β -redexes) *A λ -term t is a β -redex if it is of the form $\lambda x.t_1(t_2)$, and its contractum is $t_1[x := t_2]$. We may also refer to the contractum of a β -redex as the reduct of the β -redex.*

Given a λ -term t , \mathcal{R} is a set of β -redexes in t if \mathcal{R} a finite set of paths such that $\text{subterm}(t, p)$ is a β -redex for each $p \in \mathcal{R}$.

Definition 2.0.18 (λ -term Contexts)

$$\text{contexts } C ::= [] \mid \lambda x.C \mid C(t) \mid t(C)$$

Given a context C and a λ -term t , we use $C[t]$ for the λ -term obtained from replacing the hole $[]$ in C , which is formally defined below:

$$C[t] = \begin{cases} t & \text{if } C \text{ is } []; \\ \lambda x.(C_0[t]) & \text{if } C \text{ is } \lambda x.C_0; \\ C_1[t](t_2) & \text{if } C \text{ is } C_1(t_2); \\ t_1((C_2[t])) & \text{if } C \text{ is } t_1(C_2). \end{cases}$$

Given a context C and a path p , we use $\text{subterm}(C, p)$ for either a context or a term defined below:

$$\begin{aligned} \text{subterm}(C, \emptyset) &= C \\ \text{subterm}(C(t), 0.p) &= \text{subterm}(C, p) \\ \text{subterm}(t(C), 0.p) &= \text{subterm}(t, p) \\ \text{subterm}(C(t), 1.p) &= \text{subterm}(t, p) \\ \text{subterm}(t(C), 1.p) &= \text{subterm}(C, p) \\ \text{subterm}(\lambda x.C, 0.p) &= \text{subterm}(C, p) \end{aligned}$$

Definition 2.0.19 (β -reduction) Given two λ -terms t_1, t_2 and a path p , we write $[p] : t_1 \rightarrow_\beta t_2$ if $t_1 \equiv C[t]$ for some context C and β -redex t , where $\text{subterm}(C, p) = []$, and $t_2 \equiv C[t']$ for the reduct t' of t . A reduction $[p]$ is a top reduction if $p = \emptyset$, and it is a head reduction if $p = 0 \dots 0.\emptyset$.

We may write $t_1 \rightarrow_\beta t_2$ to mean $[p] : t_1 \rightarrow_\beta t_2$ for some p . We refer to the binary relation \rightarrow_β as (one-step) β -reduction, and use \rightarrow_β^+ and \rightarrow_β^* for the transitive closure and the reflexive and transitive closure of \rightarrow_β , respectively. We may also refer to \rightarrow_β^* as multi-step β -reduction. In addition, we use \equiv_β for the minimal equivalence relation containing \rightarrow_β .

Definition 2.0.20 (β -reduction Sequences) We use σ for (finite) β -reduction sequences defined as follows:

$$\beta\text{-reduction sequences } \sigma ::= \emptyset \mid [p] + \sigma$$

where \emptyset stands for the empty β -reduction sequence. Note that we may omit writing the trailing \emptyset in β -reduction sequence.

We write $\sigma : t \rightarrow_\beta^* t'$ to mean that σ is a β -reduction sequence from t to t' , that is, σ is of the form $[p_1] + \dots + [p_n] + \emptyset$ and there are λ -terms $t = t_1, \dots, t_{n+1} = t'$ such that $[p_i] : t_i \rightarrow_\beta t_{i+1}$ holds for each $1 \leq i \leq n$. We write $\sigma : t$ to mean $\sigma : t \rightarrow_\beta^* t'$ for some t' , which can be denoted by $\sigma(t)$.

Proposition 2.0.21 Assume $\sigma : t \rightarrow_\beta^* t'$ and $\sigma = [p_1] + [p_2]$. If p_1 and p_2 are incompatible, then we have $\sigma' : t \rightarrow_\beta^* t'$ for $\sigma' = [p_2] + [p_1]$.

Proof By structural induction on t . ■

Let σ be a β -reduction sequence from t such that each $[p]$ in σ implies $p = 0.p_1$ or $p = 1.p_1$ for some p_1 . By applying Proposition 2.0.21 repeatedly, we can arrange σ into σ' of the form $0.\sigma'_1 + 1.\sigma'_1$ such that $\sigma(t) = \sigma'(t)$.

Proposition 2.0.22 Assume $\sigma : t \rightarrow_{\beta}^* t'$. Then for every substitution θ , we also have $\sigma : t[\theta] \rightarrow_{\beta}^* t'[\theta]$.

Proof It is straightforward to verify that $[p] : t \rightarrow_{\beta} t'$ implies $[p] : t[\theta] \rightarrow_{\beta} t'[\theta]$. Then the proposition follows immediately. ■

Lemma 2.0.23 Assume that $\sigma : t$. If $\sigma = \sigma_1 + [\emptyset]$ for some σ_1 containing no head reduction, that is, $[\emptyset]$ does not appear in σ_1 , then there exists $\sigma' : t$ such that $\sigma' = [\emptyset] + \sigma'_1$ for some σ'_1 and $\sigma'(t) = \sigma(t)$.

Proof Since σ contains a head reduction, t must be of the form $\lambda x.t_1(t_2)$. Since σ_1 contains no head reduction, we can assume by Proposition 2.0.21 that $\sigma_1 = 0.0.\sigma_{11} + 1.\sigma_{12}$ for some $\sigma_{11} : t_1$ and $\sigma_{12} : t_2$. Clearly, $\sigma_1(t) = \lambda x.\sigma_{11}(t_1)(\sigma_{12}(t_2))$, and $\sigma(t) = \sigma_{11}(t_1)[x := \sigma_{12}(t_2)]$. Let p_1^x, \dots, p_n^x be an enumeration of the positions of all free occurrences of x in $\sigma(t_1)$, and

$$\sigma' = [\emptyset] + \sigma_{11} + p_1^x @ \sigma_{12} + \dots + p_n^x @ \sigma_{12}$$

Clearly, we have $\sigma' : t$ and $\sigma'(t) = \sigma(t)$. ■

2.1 Developments

Definition 2.1.1 (Residuals under β -reduction) Let t_0 be $C[(\lambda x.t_1)t_2]$, where the hole $[]$ in C is at some position p_0 , and $t'_0 = C[t_1[x := t_2]]$. For each β -redex in t_0 at some position p , the residuals of the β -redex, denoted by $\mathbf{Res}(t_0, p_0, p)$, is a set of subterms in t'_0 defined as follows:

1. If $p = p_0$, then $\mathbf{Res}(t_0, p_0, p) = \emptyset$.
2. If $p = (p_0.0)@p'$, then $\mathbf{Res}(t_0, p_0, p) = \{p_0 @ p'\}$.
3. If $p = (p_0.1)@p'$, then $\mathbf{Res}(t_0, p_0, p) = \{p_0 @ p_x @ p'\}$, where p_x ranges over all paths such that $\text{subterm}(t_1, p_x)$ is a free occurrence of x in t_1 .
4. Otherwise, p_0 is not a prefix of p . In this case, $\mathbf{Res}(t_0, p_0, p) = \{p\}$.

Clearly, the residuals of a β -redex are β -redexes themselves.

Definition 2.1.2 (λ -terms with marked β -redexes) Given a λ -term t and a set \mathcal{R} of β -redexes in t , we write t/\mathcal{R} a λ -term with marked β -redexes, or a marked λ -term for short.

Definition 2.1.3 (Developments) Assume that t is a λ -term and \mathcal{R} is a set of β -redexes in t . A β -reduction sequence σ is from t/\mathcal{R} is called a development if σ is empty, or $\sigma = [p] + \sigma_1$ for some $p \in \mathcal{R}$ and $[p]$ reduces t/\mathcal{R} to t_1/\mathcal{R}_1 and σ_1 is a development of t_1/\mathcal{R}_1 .

A finite development σ of t/\mathcal{R} is complete if $\sigma(t/\mathcal{R}) = t'/\emptyset$ for some t' .

Lemma 2.1.4 Let P be a unary predicate on marked λ -terms. Assume that P is modulo α -equivalence, that is, $P(t_1/\mathcal{R})$ implies $P(t_2/\mathcal{R})$ whenever $t_1 \equiv_{\alpha} t_2$ holds, and

1. $P(x/\emptyset)$ holds for every variable x .

2. For t/\mathcal{R} , $P(t/\mathcal{R})$ implies $P(\lambda x.t/0.\mathcal{R})$.
3. For t_1/\mathcal{R}_1 and t_2/\mathcal{R}_2 , $P(t_1/\mathcal{R}_1)$ and $P(t_2/\mathcal{R}_2)$ implies $P(t/\mathcal{R})$, where $t = t_1(t_2)$ and $\mathcal{R} = 0.\mathcal{R}_1 \cup 1.\mathcal{R}_2$
4. For t_1/\mathcal{R}_1 and t_2/\mathcal{R}_2 , $P(t_1/\mathcal{R}_1)$, $P(t_2/\mathcal{R}_2)$ and $P(t_1/\mathcal{R}_1[x := t_2/\mathcal{R}_2])$ implies $P(t/\mathcal{R} \cup \{\emptyset\})$, where $t = (\lambda x.t_1)(t_2)$ and $\mathcal{R} = \{\emptyset\} \cup 0.0.\mathcal{R}_1 \cup 1.\mathcal{R}_2$.

Then $P(t/\mathcal{R})$ holds for every marked λ -term t/\mathcal{R} .

Proof We first prove that for every marked λ -term t/\mathcal{R} , $P(t/\mathcal{R}[\theta])$ holds for every substitution θ that maps variables to marked λ -terms satisfying P , that is, $P(\theta(x))$ for each $x \in \mathbf{dom}(\theta)$. We proceed by structural induction on t .

- t is some variable x . Then $t[\theta]$ is either x or $\theta(x)$. So $P(t[\theta])$ holds.
- t is $\lambda x_0.t_0$ for some λ -term t_0 . Then $\mathcal{R} = 0.\mathcal{R}_0$ for some set \mathcal{R}_0 of redexes in t_0 . Given that P is modulo α -equivalence, we may assume $t[\theta] \equiv \lambda x_0.t_0[\theta]$ without loss of generality. By induction hypothesis on t_0 , we have $P(t_0[\theta])$. By (2), we have $P(t[\theta]/\mathcal{R})$.
- t is $t_1(t_2)$ and $\emptyset \notin \mathcal{R}$. Then $\mathcal{R} = 0.\mathcal{R}_1 \cup 1.\mathcal{R}_2$ for some sets \mathcal{R}_0 and \mathcal{R}_1 of redexes in t_1 and t_2 , respectively. Clearly, $t[\theta] = t_1[\theta](t_2[\theta])$. By induction hypothesis, both $P(t_1/\mathcal{R}_0[\theta])$ and $P(t_2/\mathcal{R}_1[\theta])$ hold. By (3), we have $P(t/\mathcal{R}[\theta])$.
- t is $(\lambda x.t_1)(t_2)$ and $\emptyset \in \mathcal{R}$. Then $\mathcal{R} = \{\emptyset\} \cup 0.0.\mathcal{R}_1 \cup 1.\mathcal{R}_2$ for some \mathcal{R}_1 and \mathcal{R}_2 . By induction hypothesis on t_2 , $P(t_2/\mathcal{R}_2[\theta])$ holds. Let θ' be $\theta[x \mapsto t_2/\mathcal{R}_2[\theta]]$. By induction hypothesis on t_1 , $P(t_1/\mathcal{R}_1[\theta'])$ holds. Note that $t_1/\mathcal{R}_1[\theta']$ is α -equivalent to $t_1/\mathcal{R}_1[\theta][x := t_2/\mathcal{R}_2[\theta]]$. By (4), $P(t/\mathcal{R}[\theta])$ holds.

Therefore, by structural induction, $P(t/\mathcal{R}[\theta])$ for all λ -terms t and all substitutions θ such that $P(\theta(x))$ holds for each $x \in \mathbf{dom}(\theta)$. Let θ be the empty substitution, and we have $P(t/\mathcal{R})$ for all λ -terms t . ■

Theorem 2.1.5 (Finite Developments) For each λ -term t , there exists a number n such that the length of every development from t is less than or equal to n .

Proof Let $P(t/\mathcal{R})$ be the statement that there is a number n such that the length of every development from t/\mathcal{R} is less than or equal to n .

- Assume $t = x$ for some variable x . Then we can choose n to be 0.
- Assume $t = \lambda x.t_1$. Then $\mathcal{R} = 1.\mathcal{R}_1$ for some set \mathcal{R}_1 of β -redexes in t_1 . Note in this case that each development from t/\mathcal{R} is of the form $0.\sigma_1$ for some development σ_1 from t_1/\mathcal{R}_1 . Hence, $P(t_1/\mathcal{R}_1)$ implies $P(t/\mathcal{R})$.
- Assume $t = t_1(t_2)$ and $\emptyset \notin \mathcal{R}$. Then $\mathcal{R} = 0.\mathcal{R}_1 \cup 1.\mathcal{R}_2$, and each development from t/\mathcal{R} can essentially be written as $0.\sigma_1 + 1.\sigma_2$, where σ_1 and σ_2 are some developments from t_1/\mathcal{R}_1 and t_2/\mathcal{R}_2 , respectively. Clearly, $P(t_1/\mathcal{R}_1)$ and $P(t_2/\mathcal{R}_2)$ implies $P(t/\mathcal{R})$ in this case.

- Assume $t = (\lambda x.t_1)(t_2)$ and $\emptyset \in \mathcal{R}$. Let σ be any development from t/\mathcal{R} . We may assume that $[\emptyset]$ appears in σ for otherwise we can simply take $\sigma + [\emptyset]$ instead. Let $\sigma = \sigma_1 + [\emptyset] + \sigma_2$. By studying the proof of Lemma 2.0.23, we see that there is a development σ' from t/\mathcal{R} that is of the form $[\emptyset] + \sigma'_1 + \sigma_2$. Assume $[\emptyset] : t/\mathcal{R} \rightarrow_{\beta} t'/\mathcal{R}'$. Then we clearly have $P(t_1/\mathcal{R}_1)$, $P(t_2/\mathcal{R}_2)$ and $P(t'/\mathcal{R}')$ implies $P(t/\mathcal{R})$.

By Lemma 2.1.4, we have $P(t/\mathcal{R})$ for all t/\mathcal{R} . Given t , let \mathcal{R}_t be the set of all β -redexes in t . Then every development from t is a development from t/\mathcal{R}_t . Hence, we are done. ■

Given t/\mathcal{R} , let $\mu_0(t/\mathcal{R})$ be the maximum of $length(\sigma)$, where σ ranges over all the developments of t/\mathcal{R} . Let $\mu_0(t)$ be $\mu_0(t/\mathcal{R}_t)$, where \mathcal{R}_t is the set of all β -redexes in t . Theorem 2.1.5 simply states that $\mu_0(t) < \infty$ for all t .

Lemma 2.1.6 *Assume that σ_1 and σ_2 are two complete developments from t/\mathcal{R} . Then $\sigma_1(t/\mathcal{R}) = \sigma_2(t/\mathcal{R})$.*

Proof Let $P(t/\mathcal{R})$ be the statement that $\sigma_1(t/\mathcal{R}) = \sigma_2(t/\mathcal{R})$ for every pair of complete developments σ_1 and σ_2 from t/\mathcal{R} .

- Assume $t = x$ for some variable x . Clearly, $P(t/\mathcal{R})$ holds.
- Assume $t = \lambda x.t_1$. Then $\mathcal{R} = 0.\mathcal{R}_1$ for some set β -redexes in t_1 . Clearly $P(t_1/\mathcal{R}_1)$ implies $P(t/\mathcal{R})$.
- Assume $t = t_1(t_2)$ and $\emptyset \notin \mathcal{R}$. Then $\mathcal{R} = 0.\mathcal{R}_1 \cup 1.\mathcal{R}_2$, where \mathcal{R}_1 and \mathcal{R}_2 are some sets of β -redexes in t_1 and t_2 , respectively. Clearly, $P(t_1/\mathcal{R}_1)$ and $P(t_2/\mathcal{R}_2)$ implies $P(t/\mathcal{R})$.
- Assume $t = (\lambda x.t_1)(t_2)$ and $\emptyset \in \mathcal{R}$. Let $\sigma_1 = \sigma_{11} + [\emptyset] + \sigma_{12}$ and $\sigma_2 = \sigma_{21} + [\emptyset] + \sigma_{22}$. By studying the proof of Lemma 2.0.23, we see that there is a complete development σ'_1 from t/\mathcal{R} that is of the form $[\emptyset] + \sigma'_{11} + \sigma_{12}$, and $\sigma'_1(t) = \sigma_1(t)$. Similarly, there is a complete development σ'_2 from t/\mathcal{R} that is of the form $[\emptyset] + \sigma'_{21} + \sigma_{22}$, and $\sigma'_2(t) = \sigma_2(t)$. Assume $[\emptyset] : t/\mathcal{R} \rightarrow_{\beta} t'/\mathcal{R}'$. Note that $\sigma'_{11} + \sigma_{12}$ and $\sigma'_{21} + \sigma_{22}$ are complete developments from t'/\mathcal{R}' . Hence, $P(t'/\mathcal{R}')$ implies $P(t/\mathcal{R})$.

By Lemma 2.1.4, we have $P(t/\mathcal{R})$ for all t/\mathcal{R} , which yields this lemma. ■

Lemma 2.1.7 *Assume σ_1 and σ_2 are developments from t . Then there exists σ'_1 and σ'_2 such that $\sigma_1 + \sigma'_2$ and $\sigma_2 + \sigma'_1$ are developments from t to some term t' .*

Proof Assume σ_1 and σ_2 are developments from t/\mathcal{R}_1 and t/\mathcal{R}_2 , respectively. Then σ_1 and σ_2 are also developments from t/\mathcal{R} for $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$. By Theorem 2.1.5, there exists σ'_1 and σ'_2 such that both $\sigma_1 + \sigma'_2$ and $\sigma_2 + \sigma'_1$ are complete developments from t/\mathcal{R} . By Lemma 2.1.6, we have

$$(\sigma_1 + \sigma'_2)(t/\mathcal{R}) = (\sigma_2 + \sigma'_1)(t/\mathcal{R})$$

This concludes the proof. ■

Definition 2.1.8 *(Standard Developments)* A standard development σ from t/\mathcal{R} is standard if

1. σ is empty, or
2. $\sigma = [p] + \sigma_1$ for the leftmost p in \mathcal{R} and σ_1 is a standard development of $[p](t/\mathcal{R})$.

2.2 Fundamental Theorems of λ -calculus

Lemma 2.2.1 *Assume that σ_1 is a development from t . For every finite β -reduction sequence σ_2 from t , there exists a development σ'_1 and a β -reduction sequence σ'_2 such that $(\sigma_1 + \sigma'_2)(t) = (\sigma_2 + \sigma'_1)(t)$.*

Proof We proceed by induction on the length of σ_2 .

- $\sigma_2 = \emptyset$. Let $\sigma'_1 = \sigma_1$ and $\sigma'_2 = \emptyset$, and we are done.
- $\sigma_2 = \sigma_{20} + \sigma_{21}$, where σ_{20} is a nonempty development. Let σ_{10} be σ_1 . By Lemma 2.1.7, there exists two developments σ'_{10} and σ'_{20} such that $(\sigma_{10} + \sigma'_{20})(t) = (\sigma_{20} + \sigma'_{10})(t)$. By induction hypothesis on $\sigma(t)$, there exist a development σ''_{10} and a β -reduction sequence σ'_{21} such that $(\sigma'_{10} + \sigma'_{21})(\sigma_{20}(t)) = (\sigma_{21} + \sigma''_{10})(\sigma_{20}(t))$. Let $\sigma'_1 = \sigma''_{10}$ and $\sigma'_2 = \sigma'_{20} + \sigma'_{21}$, and we are done.

If $\sigma_2 = \sigma_{21} + \dots + \sigma_{2n}$ for some developments $\sigma_{21}, \dots, \sigma_{2n}$, then it is clear from the proof that σ'_2 can be written in the form of $\sigma'_{21} + \dots + \sigma'_{2n}$, where σ'_{2i} are all development for $1 \leq i \leq n$. In other words, if σ_2 is a concatenation of n developments, then σ'_2 can also be chosen to be a concatenation of n developments. ■

Lemma 2.2.1 is often referred to as the *Strip Lemma* for the obvious reason.

Theorem 2.2.2 (Church-Rosser) *Assume that $t \equiv_{\beta} t'$, where \equiv_{β} is the minimal equivalence relation containing \rightarrow_{β} . Then there exists $\sigma_1 : t$ and $\sigma_2 : t'$ such that $\sigma_1(t) = \sigma_2(t')$.*

Proof $t \equiv_{\beta} t'$ implies the existence of λ -terms t_0, t_1, \dots, t_n for some $n \geq 1$ such that $t = t_0$ and $t' = t_n$ and for each $0 \leq i < n$, either $t_i \rightarrow_{\beta} t_{i+1}$ or $t_{i+1} \rightarrow_{\beta} t_i$ holds. We proceed by induction on n .

- Assume $n = 1$. We omit this trivial case for brevity.
- Assume $n > 0$. By induction hypothesis, we have σ_{11} and σ_{12} such that $\sigma_{11}(t_1) = \sigma_{12}(t_n)$.
 - Assume $[p] : t_0 \rightarrow_{\beta} t_1$ for some p . Let $\sigma_1 = [p] + \sigma_{11}$ and $\sigma_2 = \sigma_{12}$, and we are done.
 - Assume $[p] : t_1 \rightarrow_{\beta} t_0$ for some p . Clearly, $[p]$ is a development, By Lemma 2.2.1, we have σ_{10} and σ_{20} such that

$$([p] + \sigma_{10})(t_1) = (\sigma_{11} + \sigma_{20})(t_1)$$

Let $\sigma_1 = \sigma_{10}$ and $\sigma_2 = \sigma_{12} + \sigma_{20}$, and we are done.

We conclude the induction proof as all the cases are covered. ■

Lemma 2.2.3 *Assume $\sigma = \sigma_1 + \sigma_2$ is finite β -reduction sequence for a λ -term t , where σ_1 is a standard development and σ_2 is a standard β -reduction sequence. Then we can construct a standard (finite) β -reduction sequence σ' from t such that $\sigma(t) = \sigma'(t)$.*

Proof We are to define a binary function std_2 that takes the arguments σ_1 and σ_2 and returns σ' . ■

Theorem 2.2.4 (Standardization)

Given a λ -term t , we use $norm_\beta(t)$ for the (possibly infinite) reduction sequence σ from t such that each β -reduction step in σ is leftmost and $\sigma(t)$ is in normal form σ is finite.

Theorem 2.2.5 (Normalization) *Assume $\nu(t) < \infty$. Then $norm_\beta(t)$ is finite.*

Lemma 2.2.6 *Assume $\mu(u[x := v](t_1) \dots (t_n)) < \infty$ and $\mu(v) < \infty$. Then we have:*

$$\mu((\lambda x.u)(v)(t_1) \dots (t_n)) \leq 1 + \mu(u[x := v](t_1) \dots (t_u)) + \mu(v)$$

Proof Let $t^* = u[x := v](t_1) \dots (t_n)$, Clearly, $\mu(t^*) < \infty$ implies that $\mu(u), \mu(t_1), \dots, \mu(t_n)$ are all finite. We proceed by induction on $\mu(u) + \mu(v) + \mu(t_1) + \dots + \mu(t_n)$. Let $t = \mu((\lambda x.u)(v)(t_1) \dots (t_n))$. Assume $[p] : t \rightarrow_\beta t'$, and we do a case analysis on p .

- Assume p in u . Then $t' = (\lambda x.u')(v)(t_1) \dots (t_n)$ for some u' such that $u \rightarrow_\beta u'$ holds. By induction hypothesis,

$$\mu(t') \leq 1 + \mu(u'[x := v](t_1) \dots (t_u)) + \mu(v) \leq \mu(t^*) + \mu(v)$$

- Assume p in v . Then $t' = (\lambda x.u)(v')(t_1) \dots (t_n)$ for some v' such that $v \rightarrow_\beta v'$ holds. By induction hypothesis,

$$\mu(t') \leq 1 + \mu(u[x := v'](t_1) \dots (t_u)) + \mu(v') \leq \mu(t^*) + \mu(v)$$

- Assume p in t_i for some $1 \leq i \leq n$. Then $t' = (\lambda x.u)(v)(t_1) \dots (t'_i) \dots (t_n)$ for some t'_i such that $t_i \rightarrow_\beta t'_i$ holds. By induction hypothesis,

$$\mu(t') \leq 1 + \mu(u[x := v](t_1) \dots (t'_i) \dots (t_u)) + \mu(v) \leq \mu(t^*) + \mu(v)$$

- Assume that p is the outmost β -redex $(\lambda x.u)(v)$. Then $t^* = t'$. So, $\mu(t') \leq \mu(t^*) + \mu(v)$.

So $\mu(t') \leq \mu(t^*) + \mu(v)$ for each t' such that $t \rightarrow_\beta t'$ holds, and this yields $\mu(t) \leq 1 + \mu(t^*) + \mu(v)$. ■

Definition 2.2.7 (λ_I -terms and β_I -redexes) *A λ -term t_0 is a λ_I -term if for every subterm t of t_0 , t being of the form $\lambda x.t_1$ implies $x \in FV(t_1)$. Moreover, a β -redex $\lambda x.t_1(t_2)$ is a β_I -redex if $x \in FV(t_1)$.*

Theorem 2.2.8 (Conservation) *Assume $[p] : t \rightarrow_\beta t'$ and $\mu(t') < \infty$. If subterm(t, p) is β_I -redex, then $\mu(t) < \infty$.*

Proof We proceed by induction on $\langle size(t'), \mu(t') \rangle$, lexicographically ordered. ■

Corollary 2.2.9 *A λ_I -term is strongly normalizing if and only if it is weakly normalizing.*

Proof If t is strongly normalizing, then it is obviously weakly normalizing. If t is weakly normalizing, then there exists a finite reduction sequence $\sigma : t \rightarrow_\beta^* t'$ such that t' is in β -normal form. It is trivial to verify that each β -redex reduced in σ is a β_I -redex. By Theorem 2.2.8, t is strongly normalizing. ■

2.3 Exercises

Exercise 1 Assume $t \equiv_{\alpha} t'$. Please show that $FV(t) = FV(t')$ holds.

Exercise 2 Assume that y is a variable not contained in $\text{vars}(t)$. Let t' be $t[y/x]$, that is, the term obtained from replacing (not substituting) y for x in t . Please show that $FV(t') = FV(t)$ if $x \notin FV(t)$, and $FV(t') = (FV(t) \setminus \{x\}) \cup \{y\}$ if $x \in FV(t)$.

Exercise 3 Assume $t \equiv_{\alpha} t'$. Please show $t[y/x] \equiv_{\alpha} t'[y/x]$ holds for any x if y does not occur in $\text{vars}(t) \cup \text{vars}(t')$.

Exercise 4 Please show $\lambda x.t \equiv_{\alpha} \lambda y.t[x := y]$ if $y \notin FV(t)$.

Exercise 5 Assume that σ is a development of t . Please show $\text{size}(\sigma(t)) < 2^{|t|}$, where $|\cdot|$ be a unary function defined as follows:

$$|x| = 1; |\lambda x.t| = |t|; |(t_1)t_2| = |t_1| + |t_2|$$

Exercise 6 Prove Lemma 2.1.7 by employing Lemma 2.1.4 directly.

Exercise 7 Assume that σ is a standard development of t/\mathcal{R} that is also complete. Please show that $\text{length}(\sigma) = \mu_0(t/\mathcal{R})$ if \mathcal{R} contains only β_I -redexes.

