2
The Lambda-Calculus Reduction System

2.1 Reduction Systems

In this section we present basic notions on reduction systems. For a more detailed study see [Klop, 1992, Dershowitz and Jouannaud, 1990].

Definition 2.1

A *reduction system* is a pair $\langle \mathcal{A}, \rightarrow_{\boldsymbol{R}} \rangle$, where $\mathcal{A}$ is a set of terms and $\boldsymbol{R}$ is a relation on $\mathcal{A}$. We call $\boldsymbol{R}$ a *notion of reduction*.

A notion of reduction can be introduced as a set of *contraction rules*:

$$\boldsymbol{R} : M \rightarrow N \ \textit{if} \ \cdots$$

This corresponds to the following relation on $\mathcal{A}$:

$$\boldsymbol{R} = \{(M, N) \mid \cdots\}$$

We will always present notions of reduction as contraction rules.

Definition 2.2

Let $\mathcal{A}$ be a set of terms. A *context* (denoted by $C[\ ]$) is a 'term' containing one or more occurrences of $[\ ]$, denoting holes, and such that if $M \in \mathcal{A}$, then replacing the holes in $C[\ ]$ by M, is the term $C[M] \in \mathcal{A}$.

A relation $\boldsymbol{R}$ is compatible if it can be lifted upon contexts.

Definition 2.3

1. A binary relation $\boldsymbol{R}$ on a set of terms $\mathcal{A}$ is *compatible* if

$$(M, N) \in \boldsymbol{R} \Rightarrow (C[M], C[N]) \in \boldsymbol{R}$$

for all $M, N \in \mathcal{A}$ and all contexts $C[\,]$ with one hole.

2. A compatible, reflexive and transitive relation on $\mathcal{A}$ is called a *reduction* relation on $\mathcal{A}$.

Definition 2.4

Let $\boldsymbol{R}$ be a notion of reduction on $\mathcal{A}$. Then $\boldsymbol{R}$ induces the following binary relations on $\mathcal{A}$:

– The one step $\boldsymbol{R}$-reduction denoted by $\rightarrow_{\boldsymbol{R}}$. The $\rightarrow_{\boldsymbol{R}}$ relation is the compatible closure of $\boldsymbol{R}$, and is inductively defined as follows:

$$\begin{array}{lcl} (M, N) \in \boldsymbol{R} & \Rightarrow & M \rightarrow_{\boldsymbol{R}} N \\ M \rightarrow_{\boldsymbol{R}} N & \Rightarrow & C[M] \rightarrow_{\boldsymbol{R}} C[N] \end{array}$$

– The $\boldsymbol{R}$-reduction denoted by $\longrightarrow^*_{\boldsymbol{R}}$. The $\longrightarrow^*_{\boldsymbol{R}}$ relation is the reflexive, transitive closure of $\rightarrow_{\boldsymbol{R}}$, and is inductively defined as follows:

$$\begin{array}{lcl} M \rightarrow_{\boldsymbol{R}} N & \Rightarrow & M \longrightarrow^*_{\boldsymbol{R}} N \\ M \longrightarrow^*_{\boldsymbol{R}} M & & \\ M \longrightarrow^*_{\boldsymbol{R}} N, N \longrightarrow^*_{\boldsymbol{R}} P & \Rightarrow & M \longrightarrow^*_{\boldsymbol{R}} P \end{array}$$

The relation $\rightarrow_{\boldsymbol{R}}$ is, by definition, a compatible relation. The relation $\longrightarrow^*_{\boldsymbol{R}}$ is the reflexive transitive closure of $\rightarrow_{\boldsymbol{R}}$ and therefore a reduction relation.

We will sometimes omit $\boldsymbol{R}$, when it is clear from the context which notion of reduction $\boldsymbol{R}$ represents.

Definition 2.5

Let $\langle \mathcal{A}, \boldsymbol{R} \rangle$ be a notion of reduction.

– A term M in $\mathcal{A}$ is called an $\boldsymbol{R}$-*redex*, if $(M, N) \in \boldsymbol{R}$ for some N in $\mathcal{A}$. The term N is called an $\boldsymbol{R}$-*contractum* of M.

– A term M is said to be in $\boldsymbol{R}$-*normal form (*$\boldsymbol{R}$*-nf)* if M does not contain (as a subterm) any $\boldsymbol{R}$-redex.

– A term M has a ($\boldsymbol{R}$-nf), if M $\boldsymbol{R}$-reduces to N, and N is a ($\boldsymbol{R}$-nf).

– We write $\mathrm{NF}_{\boldsymbol{R}}$ to denote the set of terms in $\boldsymbol{R}$-normal form.

We now discuss the property of confluence, which will be required in the reduction systems we will consider.

Definition 2.6

Let $\boldsymbol{R}$ be a notion of reduction.

– $\boldsymbol{R}$ satisfies the *diamond property* (see Figure 2.1) if:

$$\forall M, N_1, N_2.(M \ \boldsymbol{R}\ N_1 \wedge M \ \boldsymbol{R}\ N_2 \Rightarrow \exists N.(N_1 \ \boldsymbol{R}\ N \wedge N_2 \ \boldsymbol{R}\ N))$$

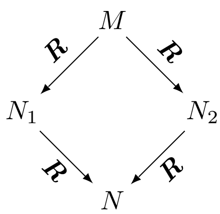


Figure 2.1 Diamond property

– $\boldsymbol{R}$ is said to be *Church-Rosser* (CR) if $\longrightarrow^*_{\boldsymbol{R}}$ satisfies the diamond property.

If a reduction is Church-Rosser then it is not possible to reduce any λ-term to two distinct normal forms. Therefore, if a term has a normal form, the normal form is unique.

Definition 2.7

– Let $\varDelta$ be a $\boldsymbol{R}$-redex with contractum $\varDelta'$. We write

$$M \xrightarrow{\varDelta}_{\boldsymbol{R}} N$$

if $M \equiv C[\varDelta]$, and $N \equiv C[\varDelta']$.

– An *$\boldsymbol{R}$-reduction (path)* is a sequence (possible infinite)

$$M_0 \xrightarrow{\varDelta_0}_{\boldsymbol{R}} M_1 \xrightarrow{\varDelta_1}_{\boldsymbol{R}} M_2 \rightarrow_{\boldsymbol{R}} \cdots$$

We will sometimes leave out the redexes Δ_i, when denoting a reduction sequence.

Definition 2.8

Let $\langle \mathcal{A}, \boldsymbol{R} \rangle$ be a reduction system. The ***R**-reduction graph* of a term $M \in \mathcal{A}$ (denoted by $\mathcal{G}_{\boldsymbol{R}}(M)$) is the set

$$\{N \in \mathcal{A} \mid M \longrightarrow^*_{\boldsymbol{R}} M\}$$

directed by $\rightarrow_{\boldsymbol{R}}$. This defines a multigraph since, if several redexes give rise to $M_0 \rightarrow_{\boldsymbol{R}} M_1$, then that many directed arcs connect M_0 to M_1 in $\mathcal{G}_{\boldsymbol{R}}(M)$.

Definition 2.9

The reduction system $\langle \mathcal{A}, \boldsymbol{R} \rangle$ is *strongly normalising* (SN), if for every $M_0 \in \mathcal{A}$, every reduction sequence

$$M_0 \rightarrow_{\boldsymbol{R}} M_1 \rightarrow_{\boldsymbol{R}} \cdots$$

reaches a normal form.

In terms of the graph representation of reductions, if a reduction system is strongly normalising, then the reduction-graph for all the terms in the system is both acyclic and finite.

Reduction-graphs represent all the possible ways to reduce a term. A *reduction strategy* defines a way to travel trough the reduction-graphs, thus providing a choice of how to reduce a term. We will now define this notion, and discuss some of its properties.

Definition 2.10

Let $\langle \mathcal{A}, \boldsymbol{R} \rangle$ be a reduction system. A *reduction strategy* F is a map

$$F : \mathcal{A} \rightarrow \mathcal{A}$$

such that, for all $M \in \mathcal{A}$, $M \rightarrow_{\boldsymbol{R}} F(M)$ if M is not in normal form.

Definition 2.11

A strategy F is *normalising* if

$$M \text{ has a normal form} \Rightarrow \exists\ n\ F^n(M) \text{ is a normal form.}$$

Definition 2.12

A strategy F is *maximal* if the minimum number of applications of F needed to reach the normal form is equal to the length of the longest finite reduction path.

2.2 The λ-calculus

In this section we briefly present the type free λ-calculus. For a more detailed reference, see [Barendregt, 1984].

2.2.1 Syntax

Definition 2.13

Let $\mathcal{V}$ be an infinite set of variables. The set of λ-*terms*, Λ is inductively defined from $\mathcal{V}$ the following way:

$$\begin{array}{rcll} x \in \mathcal{V} & \Rightarrow & x \in \Lambda & \\ M, N \in \Lambda & \Rightarrow & (MN) \in \Lambda & \text{(Application)} \\ M \in \Lambda,\, x \in \mathcal{V} & \Rightarrow & (\lambda x.M) \in \Lambda & \text{(Abstraction)} \end{array}$$

Notation:. We use the symbol $\equiv$ to denote syntactic equality between terms.

We consider application to be left associative, and abstraction to be right associative, and use the following abbreviations to simplify notation:

$$\begin{array}{rcl} (M_1 M_2 \ldots M_n) & \equiv & (\ldots(M_1 M_2)\ldots M_n) \\ (\lambda x_1 x_2 \ldots x_n.M) & \equiv & (\lambda x_1.(\lambda x_2.(\ldots(\lambda x_n.M)\ldots))) \end{array}$$

We define formally the notion of contexts in the λ-calculus.

Definition 2.14

A *context* $C[\;]$ in the λ-calculus is inductively defined in the following way:

– x is a context

– $[\;]$ is a context

– if $C_1[\;]$ and $C_2[\;]$ are contexts, then $C_1[\;]C_2[\;]$ and $\lambda x.C_1[\;]$ are also contexts.

2.2.2 Variables and Substitutions

A variable x occurs *free* in a term M if x is not in the scope of an abstraction λx in M. Otherwise x occurs *bound* in M.

For example, in the term $(\lambda zx.yx)$, the variable x occurs *bound* and the variable y occurs *free*.

Definition 2.15

Let M be in Λ, the set $\mathsf{fv}(M)$ of free variables of M is inductively defined as follows:

$$\begin{array}{rcl} \mathsf{fv}(x) & = & \{x\} \\ \mathsf{fv}(MN) & = & \mathsf{fv}(M) \cup \mathsf{fv}(N) \\ \mathsf{fv}(\lambda x.M) & = & \mathsf{fv}(M) \setminus \{x\} \end{array}$$

Definition 2.16

Let M be in Λ, the set $\mathsf{bv}(M)$ of bound variables of M is inductively defined as follows:

$$\begin{array}{rcl} \mathsf{bv}(x) & = & \emptyset \\ \mathsf{bv}(MN) & = & \mathsf{bv}(M) \cup \mathsf{bv}(N) \\ \mathsf{bv}(\lambda x.M) & = & \mathsf{bv}(M) \cup \{x\} \end{array}$$

A λ-term is closed if and only if $\mathsf{fv}(M) = \emptyset$. The set of closed λ-terms is denoted by $\Lambda^0 \subset \Lambda$.

Note that the sets of free and bound variables of a term are not necessarily disjoint: x occurs both *free* and *bound* in $x(\lambda xy.x)$.

Definition 2.17 (Substitution)

The result of substituting the free occurrences of x by L in M (denoted by $M[L/x]$) is defined as:

$$\begin{array}{rcl} y[L/x] & \equiv & \begin{cases} L & \text{if } x \equiv y \\ y & \text{otherwise} \end{cases} \\ (MN)[L/x] & \equiv & (M[L/x])(N[L/x]) \\ (\lambda y.M)[L/x] & \equiv & \begin{cases} \lambda y.M & \text{if } x \equiv y \\ \lambda y.(M[L/x]) & \text{otherwise} \end{cases} \end{array}$$

Some care is needed with substitution to avoid the problem of *variable capture*.

Consider the term $\mathsf{first} \equiv (\lambda xy.x)$. For any given λ-terms M and N, we would expect

$$((\lambda xy.x)[M/x])[N/y] \equiv M.$$

But if we take $M \equiv y$ then

$$((\lambda xy.x)[y/x])[N/y] \equiv N.$$

The problem arises when the free variable y in M enters the scope of a (λy) in $(\lambda xy.x)$. To avoid variable capture, a substitution $M[N/x]$ should only be allowed, in which case we say that x is substitutable by N in M, if x doesn't occur free in any subterm of M of the form $\lambda y.P$, and $y \in \mathsf{fv}(N)$. This condition is ensured if the set of bound variables of M is disjoint from the set of free variables of N:

$$\mathsf{bv}(M) \cap \mathsf{fv}(N) = \emptyset.$$

The previous condition can be always be ensure by renaming, when necessary, the bound variables in M. This operation is called α-conversion.

Definition 2.18

A *change of bound variable* x in a term M is the substitution of all the subterms of M of the form $\lambda x.N$ by $\lambda y.(N[y/x])$, where y does not occur in N.

The change of bound variables preserves the meaning of the term, in the sense that it represents the same function. This notion is called *α-congruence*:

Definition 2.19 (α-congruence)

The terms M and N are α-congruent, (notation $M \equiv_\alpha N$), if N can be obtained from M, by a series of changes of bound variables, and vice-versa.

For example:

$$\lambda x.xy \equiv_\alpha \lambda z.zy \not\equiv_\alpha \lambda y.yy.$$

We will always assume the sets of free and bound variables of any term to be disjoint[1]. Therefore any substitution $M[N/x]$ is valid. Moreover, we do not distinguish terms that are α-congruent (for instance $\lambda x.x \equiv \lambda y.y$).

[1] this is known as the Barendregt variable convention

2.2.3 The $\langle \Lambda, \beta \rangle$ Reduction System

We will now present a reduction relation on Λ. See [Barendregt, 1984] for more details on this, and other notions of reduction in the λ-calculus. This notion, together with Λ, defines a reduction system for which some properties will be discussed.

Definition 2.20 (β-reduction)

The notion of *β-reduction* on Λ is defined by the following contraction rule:

$$\beta : (\lambda x.M)N \to M[N/x], \quad M, N \in \Lambda$$

A λ-term of the form $(\lambda x.M)N$ is called a β-redex and $M[N/x]$ is his β-contractum.

Example 2.21

The term $(\lambda x.z)((\lambda xy.xy)(\lambda x.x))$ has a β-redex $(\lambda xy.xy)(\lambda x.x)$, thus it is not in β-nf form. This term has the β-nf z.

Definition 2.22

Let β be the notion of reduction in Definition 2.20, and $\to_\beta$ and $\longrightarrow^*_\beta$ be the binary relations induced by β as described in Definition 2.4. We say that:

– *M β-reduces to N in one step*, and write $M \to_\beta N$;

– *M β-reduces to N*, and write $M \longrightarrow^*_\beta N$.

Based on the relations defined above, we can build reduction-graphs for terms in Λ. We now show an example of such a graph.

Example 2.23

For the λ-term $(\lambda x.(\lambda y.xy)(\lambda x.x))(\lambda z.w)$, we obtain the reduction graph in Figure 2.2.

Remark 2.24

Note that, the fact that a term M has a β-normal form is not implied neither implies $\mathcal{G}_\beta(M)$ to be finite (the reduction graph for $\Omega \equiv (\lambda x.xx)(\lambda y.yy)$ is finite and the term does not have a normal form.

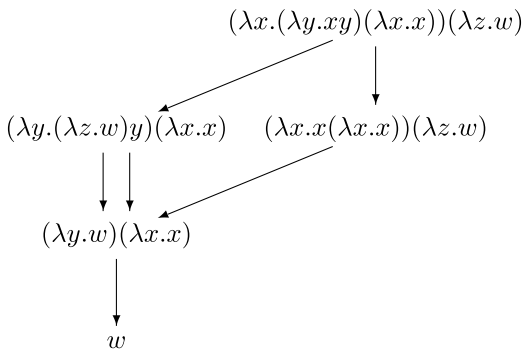
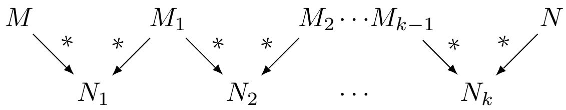


Figure 2.2 Reduction-graph for $(\lambda x.(\lambda y.xy)(\lambda x.x))(\lambda z.w)$

2.2.3.1 Equality of λ-terms. Based on the notion of β-reduction, we define equality between λ-terms. Informaly, we say that two λ-terms are equal if M can be transformed into N by a series of reductions/expansions (M expands to N, if $N \longrightarrow M$). A pictorial representation of the notion of equality is:



Definition 2.25

Let $M, N \in \Lambda$. The notion of equality $M = N$ is formalised by the rules in Figure 2.3.

For example, $a((\lambda y.by)c) = (\lambda x.ax)(bc)$, but neither $a((\lambda y.by)c)$ reduces to $(\lambda x.ax)(bc)$, nor $(\lambda x.ax)(bc)$ reduces to $a((\lambda y.by)c)$, but they have the same *"value"* $a(bc)$.

$$
\begin{array}{rcc}
(\lambda x.M)N = M[N/x] & & \\
M = M & & \\
M = N & \Rightarrow & N = M \\
M = N,\ N = L & \Rightarrow & M = L \\
M = N & \Rightarrow & ML = NL \\
M = N & \Rightarrow & LM = LN \\
M = N & \Rightarrow & \lambda x.M = \lambda x.N
\end{array}
$$

Figure 2.3 The notion of equality of λ-terms

2.2.4 Confluence

The β-reduction is Church-Rosser [Barendregt, 1984, Church and Rosser, 1936]. We will show a proof of confluence for the λ-calculus with the β notion of reduction, due to W. Tait and P. Martin-Löf.

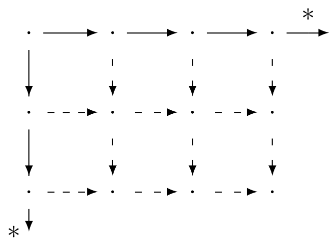
We first recall the following result from [Barendregt, 1984].

Lemma 2.26

If a binary relation satisfies the diamond property, then the transitive closure of that relation also satisfies the diamond property.

Proof

As suggested by the following diagram:



□

We will show that $\longrightarrow_\beta^*$ satisfies the diamond property, by defining the parallel reduction relation $\twoheadrightarrow_1$ and proving that $\twoheadrightarrow_1$ satisfies the diamond property and that $\longrightarrow_\beta^*$ is the transitive closure of $\twoheadrightarrow_1$.

Definition 2.27

We define a binary relation $\twoheadrightarrow_1$ on the set of λ-terms inductively as indicated in Figure 2.4.

$$\begin{array}{lll} M \twoheadrightarrow_1 M & & \\ M \twoheadrightarrow_1 M' & \Rightarrow & \lambda x.M \twoheadrightarrow_1 \lambda x.M' \\ M \twoheadrightarrow_1 M', N \twoheadrightarrow_1 N' & \Rightarrow & MN \twoheadrightarrow_1 M'N' \\ M \twoheadrightarrow_1 M', N \twoheadrightarrow_1 N' & \Rightarrow & (\lambda x.M)N \twoheadrightarrow_1 M'[N'/x] \end{array}$$

Figure 2.4 The $\twoheadrightarrow_1$ reduction relation

Lemma 2.28

If $M \twoheadrightarrow_1 M'$ and $N \twoheadrightarrow_1 N'$, where $\mathsf{fv}(M) \cap \mathsf{fv}(N) = \varnothing$, then $M[N/x] \twoheadrightarrow_1 M'[N'/x]$.

Proof

By induction on the definition of $\twoheadrightarrow_1$.

- $M \twoheadrightarrow_1 M'$ is $M \twoheadrightarrow_1 M$. Then one has to show that $M[N/x] \twoheadrightarrow_1 M[N'/x]$. This follows easily by induction on the structure of M. $P \twoheadrightarrow_1 P'$.
- $M \twoheadrightarrow_1 M'$ is $\lambda y.P \twoheadrightarrow_1 \lambda y.P'$, and is a consequence of $P \twoheadrightarrow_1 P'$. By induction hypothesis $P[N/x] \twoheadrightarrow_1 P'[N'/x]$. Then $\lambda y.(P[N/x]) \twoheadrightarrow_1 \lambda y.(P'[N/x])$ by the definition of 2.27, thus $(\lambda y.P)[N/x] \twoheadrightarrow_1 (\lambda y.P')[N'/x]$, by the definition of substitution.
- $M \twoheadrightarrow_1 M'$ is $PQ \twoheadrightarrow_1 P'Q'$, and is a consequence of $P \twoheadrightarrow_1 P'$ and $Q \twoheadrightarrow_1 Q'$. Then we have two subcases:
  - $x \in \mathsf{fv}(P)$. By induction hypothesis $P[N/x] \twoheadrightarrow_1 P'[N'/x]$. Then $P[N/x]Q \twoheadrightarrow_1 P'[N'/x]Q'$, thus $PQ[N/x] \twoheadrightarrow_1 P'Q'[N'/x]$.
  - $x \in \mathsf{fv}(Q)$. By induction hypothesis $Q[N/x] \twoheadrightarrow_1 Q'[N'/x]$. Then $PQ[N/x] \twoheadrightarrow_1 P'Q'[N'/x]$, thus $PQ[N/x] \twoheadrightarrow_1 P'Q'[N'/x]$.
- $M \twoheadrightarrow_1 M'$ is $(\lambda y.P)Q \twoheadrightarrow_1 P'[Q'/y]$, and is a consequence of $P \twoheadrightarrow_1 P'$ and $Q \twoheadrightarrow_1 Q'$, then:

$$\begin{array}{rcl} M[N/x] & = & (\lambda y.P[N/x])Q \\ & \twoheadrightarrow_1 & P'[N'/x][Q'/y] \\ & \overset{\text{(I.H.)}}{\equiv} & (P'[Q'/y])[N'/x] \\ & = & M'[N'/x] \end{array}$$

□

Lemma 2.29

$\twoheadrightarrow_1$ satisfies the diamond property.

Proof

By induction on the definition of $\twoheadrightarrow_1$, one can show that if $M \twoheadrightarrow_1 M_1$ and $M \twoheadrightarrow_1 M_2$, then there is a term M_3 such that $M_1 \twoheadrightarrow_1 M_3$ and $M_2 \twoheadrightarrow_1 M_3$.

- $M \twoheadrightarrow_1 M_1$ is $M \twoheadrightarrow_1 M$. Then take $M_3 \equiv M_2$.

- $M \twoheadrightarrow_1 M_1$ is $\lambda x.P \twoheadrightarrow_1 \lambda x.P'$ and is a consequence of $P \twoheadrightarrow_1 P'$. Then $M_2 \equiv \lambda x.P''$. By induction hypothesis there is a term P''' with $P' \twoheadrightarrow_1 P'''$ and $P'' \twoheadrightarrow_1 P'''$, thus take $M_3 \equiv \lambda x.P'''$.
- $M \twoheadrightarrow_1 M_1$ is $NP \twoheadrightarrow_1 P'Q'$ and is a consequence of $P \twoheadrightarrow_1 P'$ and $Q \twoheadrightarrow_1 Q'$. Then we have two subcases:
  - $M_2 \equiv P''Q''$. Then by induction hypothesis there is a term P''' with $P' \twoheadrightarrow_1 P'''$ and $P'' \twoheadrightarrow_1 P'''$ and the same for Q, thus take $M_3 = P'''Q'''$.
  - $P \equiv \lambda x.P_1$, $M_2 \equiv P_1''[Q''/x]$ with $P_1 \twoheadrightarrow_1 P_1''$ and $Q \twoheadrightarrow_1 Q''$. Then one has $P' \equiv \lambda x.P_1'$ with $P_1 \twoheadrightarrow_1 P_1'$. By induction hypothesis and Lemma 2.28 one can take $M_3 \equiv P_1'''[P'''/x]$.
- $M \twoheadrightarrow_1 M_1$ is $(\lambda x.P)Q \twoheadrightarrow_1 P'[Q'/x]$ and is a consequence of $P \twoheadrightarrow_1 P'$ and $Q \twoheadrightarrow_1 Q'$. Then we have two subcases:
  - $M_2 \equiv (\lambda x.P'')Q''$. Then by induction hypothesis and Lemma 2.28 one can take $M_3 = P'''[Q'''/x]$.
  - $M_2 \equiv P''[Q''/x]$ with $P \twoheadrightarrow_1 P''$ and $Q \twoheadrightarrow_1 Q''$. By induction hypothesis and Lemma 2.28 one can can take $M_3 \equiv P'''[Q'''/x]$.

□

Lemma 2.30

$\longrightarrow^*$ is the transitive closure of $\twoheadrightarrow_1$.

Proof

If we represent reduction as a relation (thus as a set of pairs), we have:

$$\longrightarrow_= \subseteq \twoheadrightarrow_1 \subseteq \longrightarrow^*$$

Since $\longrightarrow^*$ is this transitive closure of $\rightarrow_=$, so it is of $\twoheadrightarrow_1$. □

Theorem 2.31 (Church-Rosser)

Let M be a λ-term, if $M \longrightarrow_\beta^* N_1$ and $M \longrightarrow_\beta^* N_2$, there is a λ-term N such that $N_1 \longrightarrow_\beta^* N$ and $N_2 \longrightarrow_\beta^* N$. Thus $\longrightarrow$ is Church-Rosser.

Proof

By Lemma 2.29 $\twoheadrightarrow_1$ satisfies the diamond property. By Lemma 2.30 $\longrightarrow^*$ is the transitive closure of $\twoheadrightarrow_1$. The result follows by Lemma 2.26. □

2.2.4.1 The notion of η reduction. We consider another important notion of reduction, which is called η-reduction.

Definition 2.32

1. The notion of η-reduction is given by

$$\eta : \lambda x.Mx \longrightarrow M, \text{ with } x \notin \mathsf{fv}(M).$$

2. $\beta\eta = \beta \cup \eta$.

Theorem 2.33

$\beta\eta$ is Church-Rosser.

2.2.5 Normalisation

The $\langle \Lambda, \beta \rangle$ reduction system is not strongly normalising. Consider $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$, and $M \equiv (\lambda xy.y)\Omega$. The reduction path

$$(\lambda xy.y)\Omega \xrightarrow{M}_{\beta} \lambda y.y$$

ends with the normal form of the term, erasing Ω. However, if we try to normalise the subterm Ω, we obtain the following infinite reduction sequence:

$$(\lambda xy.y)\Omega \xrightarrow{\Omega}_{\beta} (\lambda xy.y)\Omega \xrightarrow{\Omega}_{\beta} (\lambda xy.y)\Omega \xrightarrow{\Omega}_{\beta} \cdots$$

A λ-term M that has a normal form, which admits an infinite reduction sequence, reaches the normal form if all the subterms of M that do not have normal forms, are erased. This is obtained if the following strategy is used.

Definition 2.34

F_{L} is defined as follows:

$$\begin{array}{rcll} F_{\mathrm{L}}(M) & = & M & \text{if } M \text{ is in normal form,} \\ & = & M' & \text{if } M \xrightarrow{\Delta}_{\beta} M' \text{ and } \Delta \text{ is the leftmost redex in } M. \end{array}$$

The following was proved in [Curry and Feys, 1958].

Theorem 2.35

The reduction strategy F_L is normalising.

F_L is called the "normal order" strategy and it is related to the call-by-name strategy in programming languages.

The following strategy is due to Barendregt et al. [Barendregt et al., 1976] and was proved to be maximal in [van Raamsdonk et al., 1999].

Definition 2.36

F_∞ is defined as follows:

$$\begin{array}{lcll} F_\infty(x\vec{P}Q\vec{R}) & = & x\vec{P}F_\infty(Q)\vec{R} & if \vec{P} \in \mathrm{NF}_\beta, Q \notin \mathrm{NF}_\beta \\ F_\infty(\lambda x.P) & = & \lambda x.F_\infty(P) & \\ F_\infty(\lambda x.P)Q\vec{R} & = & P[Q/x]\vec{R} & \text{if } x \in \mathsf{fv}(P), \text{ or } Q \in \mathrm{NF}_\beta \\ F_\infty(\lambda x.P)Q\vec{R} & = & (\lambda x.P)F_\infty(Q)\vec{R} & \text{if } x \notin \mathsf{fv}(P), \text{ and } Q \notin \mathrm{NF}_\beta \end{array}$$

Theorem 2.37

The reduction strategy F_∞ is maximal.

2.2.6 Subsystems of the lambda calculus

Several subsystems of the λ-calculus can be obtained by restricting the set of terms. Here we present three of those systems: the *$\lambda_{\mathcal{I}}$-calculus*, the *affine λ-calculus* and the *linear λ-calculus*. These systems are obtained by imposing restrictions on variable occurrences on the terms.

In the $\lambda_{\mathcal{I}}$ calculus, in every term of the form $\lambda x.M$, x occurs in M. Therefore β-reduction in $\lambda_{\mathcal{I}}$, never erases terms.

Definition 2.38

Let $\mathcal{V}$ be an infinite set of variables. The set of *$\lambda_{\mathcal{I}}$-terms*, $\Lambda_{\mathcal{I}}$ is inductively defined from $\mathcal{V}$ in the following way:

$$\begin{array}{rcll} x \in \mathcal{V} & \Rightarrow & x \in \Lambda_{\mathcal{I}} & \\ M, N \in \Lambda_{\mathcal{I}} & \Rightarrow & (MN) \in \Lambda_{\mathcal{I}} & \text{(Application)} \\ M \in \Lambda_{\mathcal{I}},\, x \in \mathsf{fv}(M) & \Rightarrow & (\lambda x.M) \in \Lambda_{\mathcal{I}} & \text{(Abstraction)} \end{array}$$

In the affine λ-calculus β-reduction never duplicates terms. Thus, in every term M, every variable occurs free at most once in any subterm of M.

Definition 2.39

Let $\mathcal{V}$ be an infinite set of variables. The set of *affine* λ*-terms*, $\Lambda_{\mathcal{A}}$ is inductively defined from $\mathcal{V}$ in the following way:

$$\begin{array}{rcll} x \in \mathcal{V} & \Rightarrow & x \in \Lambda_{\mathcal{A}} & \\ M, N \in \Lambda_{\mathcal{A}},\ \mathsf{fv}(M) \cap \mathsf{fv}(N) = \emptyset & \Rightarrow & (MN) \in \Lambda_{\mathcal{A}} & \text{(Application)} \\ M \in \Lambda_{\mathcal{A}},\, x \in \mathcal{V} & \Rightarrow & (\lambda x.M) \in \Lambda_{\mathcal{A}} & \text{(Abstraction)} \end{array}$$

The linear λ-calculus is the intersection of the $\lambda_{\mathcal{I}}$ and the affine λ-calculus. In the linear λ-calculus, in every term M every variable occurs free exactly once in any subterm of M.

Definition 2.40

Let $\mathcal{V}$ be an infinite set of variables. The set of *linear* λ*-terms*, $\Lambda_{\mathcal{L}}$ is inductively defined from $\mathcal{V}$ in the following way:

$$\begin{array}{rcll} x \in \mathcal{V} & \Rightarrow & x \in \Lambda_{\mathcal{L}} & \\ M, N \in \Lambda_{\mathcal{L}},\ \mathsf{fv}(M) \cap \mathsf{fv}(N) = \emptyset & \Rightarrow & (MN) \in \Lambda_{\mathcal{L}} & \text{(Application)} \\ M \in \Lambda_{\mathcal{L}},\, x \in \mathsf{fv}(M) & \Rightarrow & (\lambda x.M) \in \Lambda_{\mathcal{L}} & \text{(Abstraction)} \end{array}$$

All the notions defined for Λ, are defined in an analogous way for $\Lambda_{\mathcal{I}}$, $\Lambda_{\mathcal{A}}$, and $\Lambda_{\mathcal{L}}$. The sets of terms $\Lambda_{\mathcal{I}}$, $\Lambda_{\mathcal{A}}$, and $\Lambda_{\mathcal{L}}$, with the β-reduction notion respectively define the reduction systems $\langle \Lambda_{\mathcal{I}}, \beta \rangle$, $\langle \Lambda_{\mathcal{A}}, \beta \rangle$, and $\langle \Lambda_{\mathcal{L}}, \beta \rangle$.

2.2.7 The de Brujin notation

We recall a notation for representing terms in the λ-calculus , which eliminates the necessary for using names of variables. This notation is due to Nicolaas Govert de Bruijn [Bruijn, 1972].

Definition 2.41

1. The set of nameless terms Λ^* has the following alphabet:

$$\lambda, (,), 1, 2, 3, \ldots$$

2. Λ^* is defined inductively in the following way:

$$\begin{array}{rcl} n \in \mathbb{N} \setminus \{0\} & \Rightarrow & n \in \Lambda^* \\ A, B \in \Lambda^* & \Rightarrow & (AB) \in \Lambda^* \\ A \in \Lambda^* & \Rightarrow & \lambda A \in \Lambda^* \end{array}$$

Each De Bruijn index is a natural number that represents an occurrence of a variable in a λ-term, and denotes the number of binders that are in scope between that occurrence and its corresponding binder.

Example 2.42

The term $\lambda z.(\lambda y.y(\lambda x.x))(\lambda x.zx)$ is written as $\lambda(\lambda 1(\lambda 1))(\lambda 21)$.

Definition 2.43

The notion of β reduction for nameless terms is defined by the following reduction rule.

$$(\beta) : (\lambda P)Q \longrightarrow P[Q/1]$$

Substituition has to be defined in an appropriate way. In the β-reduction $(\lambda M)N$, three aspects need to be consider:

1. find the variables $n_1, n_2, \ldots, n_k$ in M under the range of the λ in λM;
2. decrease the free variables of M taking into account the removal of the outer binder;
3. replace $n_1, n_2, \ldots, n_k$ with N, suitably increasing the free variables occurring in N each time, to match the number of λ-binders the corresponding variable occurs under when substituted.

Definition 2.44

Let $M, N \in \Lambda^*$ and $n \in \mathbb{N} \setminus \{0\}$. Substitution $M[N/n$ is inductevly defined as:

$$\begin{array}{rcl} m[N/n] & \equiv & \begin{cases} m & \text{if } m < n \\ m-1 & \text{if } m > n \\ \mathsf{remane}_{(m,1)}(N) & \text{if } m = n \end{cases} \\ (M_1M_2)[N/n] & \equiv & (M_1[N/n])(M_2[N/n]) \\ (\lambda M)[N/n] & \equiv & \lambda(M[N/n+1]) \end{array}$$

and $\mathsf{remane}_{(n,i)}(M)$ is inductively defined as:

$$\begin{array}{lcl} \mathsf{remane}_{(n,i)}(j) & \equiv & \begin{cases} j & \text{if } i < i \\ j+m-1 & \text{if } j \geq i \end{cases} \\ \mathsf{remane}_{(n,i)}(M_1M_2) & \equiv & (\mathsf{remane}_{(n,i)}(M_1))(\mathsf{remane}_{(n,i)}(M_2)) \\ \mathsf{remane}_{(n,i)}(\lambda M) & \equiv & \lambda(\mathsf{remane}_{(n,i+1)}(M)) \end{array}$$

We can can define a translation from Λ to Λ^*:

$$
\begin{array}{lcl}
\mathsf{DB}\ x\ (x_1,\ldots,x_n) & \equiv & i, \text{ where } i \text{ is the minimum such that } x \equiv x_i \\
\mathsf{DB}\ (\lambda x.M)\ (x_1,\ldots,x_n) & \equiv & \lambda(\mathsf{DB}\ M\ (x,x_1,\ldots,x_n)) \\
\mathsf{DB}\ (MN)\ (x_1,\ldots,x_n) & \equiv & (\mathsf{DB}\ M\ (x_1,\ldots,x_n))(\mathsf{DB}\ N\ (x_1,\ldots,x_n))
\end{array}
$$