Notes on Foundations of Programming Languages
Reduction Systems

Sandra Alves

September 17, 2023

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 1.1 *A* reduction system *is a pair* $\langle \mathcal{A}, \to_{\mathrm{R}} \rangle$*, where* $\mathcal{A}$ *is a set of terms and* $\to_{\mathrm{R}}$[1] *is a relation on* $\mathcal{A}$*. We call* R *a* notion of reduction*. We will sometimes refer to* $\to_{\mathrm{R}}$ *as just* R*.*

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

$$\mathrm{R} : \mathrm{M} \to \mathrm{N} \ \textit{if} \ \cdots$$

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

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

We will always present notions of reduction as contraction rules.

Definition 1.2 *Let* $\mathcal{A}$ *be a set of terms. A* context *(denoted by* $\mathrm{C}[\ ]$*) is a 'term' containing one or more occurrences of* $[\ ]$*, denoting holes, and such that if* $\mathrm{M} \in \mathcal{A}$*, then replacing the holes in* $\mathrm{C}[\ ]$ *by* M*, is the term* $\mathrm{C}[\mathrm{M}] \in \mathcal{A}$*.*

A relation R is compatible if it can be lifted upon contexts.

Definition 1.3 *1. A binary relation* R *on a set of terms* $\mathcal{A}$ *is* compatible *if*

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

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

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

Definition 1.4 *Let* R *be a notion of reduction on* $\mathcal{A}$*. Then* R *induces the following binary relations on* $\mathcal{A}$*:*

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

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

- *The* R*-reduction denoted by* $\twoheadrightarrow_{\mathrm{R}}$*. The* $\twoheadrightarrow_{\mathrm{R}}$ *relation is the reflexive, transitive closure of* $\to_{\mathrm{R}}$*, and is inductively defined as follows:*

$$\begin{array}{lcl} \mathrm{M} \to_{\mathrm{R}} \mathrm{N} & \Rightarrow & \mathrm{M} \twoheadrightarrow_{\mathrm{R}} \mathrm{N} \\ \mathrm{M} \twoheadrightarrow_{\mathrm{R}} \mathrm{M} & & \\ \mathrm{M} \twoheadrightarrow_{\mathrm{R}} \mathrm{N}, \mathrm{N} \twoheadrightarrow_{\mathrm{R}} \mathrm{P} & \Rightarrow & \mathrm{M} \twoheadrightarrow_{\mathrm{R}} \mathrm{P} \end{array}$$

The relation $\to_{\mathrm{R}}$ is, by definition, a compatible relation. The relation $\twoheadrightarrow_{\mathrm{R}}$ is the reflexive transitive closure of $\to_{\mathrm{R}}$ and therefore a reduction relation.

We will sometimes omit R, when it is clear from the context which notion of reduction R represents.

Definition 1.5 *Let* $\langle \mathcal{A}, \to_{\mathrm{R}} \rangle$ *be a reduction system.*

[1] We will often refer to $\to_{\mathrm{R}}$ as just R.

- *A term* M *in* $\mathcal{A}$ *is called an* R-redex, *if* $(M, N) \in R$ *for some* N *in* $\mathcal{A}$. *The term* N *is called an* R-contractum *of* M.
- *A term* M *is said to be in* R-normal form (R-nf) *if* M *does not contain (as a subterm) any* R-*redex.*
- *A term* M *has a (*R-*nf), if* M R-*reduces to* N, *and* N *is a (*R-*nf).*
- *We write* NF_R *to denote the set of terms in* R-*normal form.*

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

Definition 1.6 *Let* R *be a notion of reduction.*

- R *satisfies the* diamond property *(see Figure 1) if:*

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

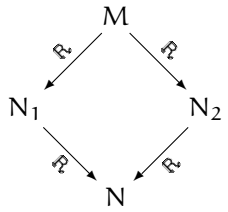


Figure 1: Diamond property

- R *is said to be* Church-Rosser *(CR) if* $\twoheadrightarrow_R$ *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 1.7

- *Let* Δ *be a* R-*redex with contractum* Δ'. *We write*

$$M \xrightarrow{\Delta}_R N$$

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

- *An* R-reduction (path) *is a sequence (possible infinite)*

$$M_0 \xrightarrow{\Delta_0}_R M_1 \xrightarrow{\Delta_1}_R M_2 \rightarrow_R \cdots$$

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

Definition 1.8 *Let* $\langle \mathcal{A}, R \rangle$ *be a reduction system. The* R-reduction graph *of a term* $M \in \mathcal{A}$ *(denoted by* $\mathcal{G}_R(M)$*) is the set*

$$\{N \in \mathcal{A} \mid M \twoheadrightarrow_R N\}$$

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

Definition 1.9 *The reduction system* $\langle \mathcal{A}, \mathrm{R} \rangle$ *is* strongly normalising *(SN), if for every* $M_0 \in \mathcal{A}$*, every reduction sequence*

$$M_0 \to_R M_1 \to_R \cdots$$

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

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

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

Definition 1.11 *A strategy* F *is* normalising *if*

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

Definition 1.12 *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 The λ-calculus

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

2.1 Syntax

Definition 2.1 *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 2.2 *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 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.

Definition 2.3 *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 2.4 *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 2.5 (Substitution) *The result of substituting the free occurrences of x by L in M (denoted by $M[L/x]$) is defined as:*

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

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 2.6 (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.

2.3.1 Substitution and α-equivalence

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

$$((\lambda xy.x)MN \twoheadrightarrow M.$$

But if we take $M \equiv y$ then

$$((\lambda xy.x)yN \twoheadrightarrow N.$$

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 does not occur free in any subterm of M of the form $\lambda y.P$, and $y \in fv(N)$. This condition is ensured if the set of bound variables of M is disjoint from the set of free variables of N:

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

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

Definition 2.7 *A* change of bound variable x *in a term* M *is the substitution of a subterm 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 2.8 (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.*

In any given context, we will always assume the sets of free and bound variables of any term to be disjoint[2]. Therefore any substitution $M[N/x]$ is valid. Moreover, we do not distinguish terms that are α-congruent (for instance $\lambda x.x \equiv \lambda y.y$).

Definition 2.9 *Let* β *be the notion of reduction in Definition 2.6, and* $\rightarrow_\beta$ *and* $\twoheadrightarrow_\beta$ *be the binary relations induced by* β *as described in Definition 1.4. We say that:*

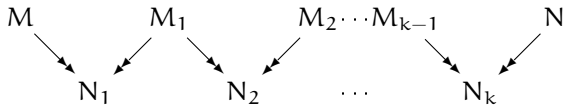
- M β-reduces to N in one step*, and write* $M \rightarrow_\beta N$*;*
- M β-reduces to N*, and write* $M \twoheadrightarrow_\beta N$.

Based on the relations defined above, we can build reduction-graphs for terms in Λ.

Remark 2.10 *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.*

2.3.2 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 2.11 *Let* $M, N \in \Lambda$*. The notion of equality* $M = N$ *is formalised by the rules in Figure 2.*

[2]this is known as the Barendregt variable convention

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

Figure 2: The notion of equality of λ-terms

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

Figure 3: The $\rightrightarrows$ reduction relation

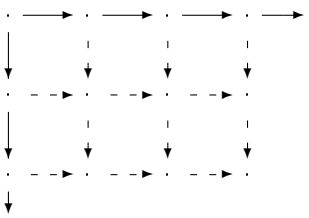
2.4 Confluence

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 2.12 *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 $\twoheadrightarrow_\beta$ satisfies the diamond property, by defining the parallel reduction relation $\rightrightarrows$ and proving that $\rightrightarrows$ satisfies the diamond property and that $\twoheadrightarrow_\beta$ is the transitive closure of $\rightrightarrows$.

Definition 2.13 *We define a binary relation $\rightrightarrows$ on the set of λ-terms inductively as indicated in Figure 3.*

Lemma 2.14 *If $M \rightrightarrows M'$ and $N \rightrightarrows N'$, then $M[N/x] \rightrightarrows M'[N'/x]$.*

Proof: By induction on the definition of $\rightrightarrows$.

- $M \rightrightarrows M'$ is $x \rightrightarrows x$. Then the result follows trivial by hypothesis since

$$M[N/x] \equiv x[N/x] \equiv N \rightrightarrows N' \equiv x[N'/x] \equiv M'[N'/x].$$

- $M \rightrightarrows M'$ is $\lambda y.P \rightrightarrows \lambda y.P'$, and is a consequence of $P \rightrightarrows P'$. By induction hypothesis $P[N/x] \rightrightarrows P'[N'/x]$. Then $\lambda y.(P[N/x]) \rightrightarrows \lambda y.(P'[N'/x])$ by the definition of 2.13, thus $(\lambda y.P)[N/x] \rightrightarrows (\lambda y.P')[N'/x]$, by the definition of substitution.
- $M \rightrightarrows M'$ is $PQ \rightrightarrows P'Q'$, and is a consequence of $P \rightrightarrows P'$ and $Q \rightrightarrows Q'$. Then by induction hypothesis $P[N/x] \rightrightarrows P'[N'/x]$ and $Q[N/x] \rightrightarrows Q'[N'/x]$. Therefore $PQ[N/x] \equiv P[N/x]Q[N/x]Q \rightrightarrows P'[N'/x]Q'[N'/x] \equiv P'Q'[N'/x]$.
- $M \rightrightarrows M'$ is $(\lambda y.P)Q \rightrightarrows P'[Q'/y]$, and is a consequence of $P \rightrightarrows P'$ and $Q \rightrightarrows Q'$, then:

$$\begin{array}{rcl} M[N/x] & \equiv & (\lambda y.P[N/x])Q[N/x] \\ & \rightrightarrows & P'[N'/x][Q'[N'/x]/y] \\ & \equiv & (P'[Q'/y])[N'/x] \\ & \equiv & M'[N'/x] \end{array}$$

□

Lemma 2.15 $\rightrightarrows$ *satisfies the diamond property.*

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

- $M \rightrightarrows M_1$ is $x \rightrightarrows x$. Then trivially $M_2 \equiv x$ and $M_3 \equiv x$.
- $M \rightrightarrows M_1$ is $\lambda x.P \rightrightarrows \lambda x.P'$ and is a consequence of $P \rightrightarrows P'$. Then $M_2 \equiv \lambda x.P''$. By induction hypothesis there is a term P''' with $P' \rightrightarrows P'''$ and $P'' \rightrightarrows P'''$, thus take $M_3 \equiv \lambda x.P'''$.
- $M \rightrightarrows M_1$ is $PQ \rightrightarrows P'Q'$ and is a consequence of $P \rightrightarrows P'$ and $Q \rightrightarrows Q'$. Then we have two subcases:
  - $M_2 \equiv P''Q''$, such that $P \rightrightarrows P''$ and $Q \rightrightarrows Q''$. Then by induction hypothesis there is a term P''' with $P' \rightrightarrows P'''$ and $P'' \rightrightarrows P'''$ 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 $P_1 \rightrightarrows P_1''$ and $Q \rightrightarrows Q''$. Then one has $P' \equiv \lambda x.P_1'$ with $P_1 \rightrightarrows P_1'$. By induction hypothesis and Lemma 2.14 one can take $M_3 \equiv P_1'''[Q'''/x]$.
- $M \rightrightarrows M_1$ is $(\lambda x.P)Q \rightrightarrows P'[Q'/x]$ and is a consequence of $P \rightrightarrows P'$ and $Q \rightrightarrows Q'$. Then we have two subcases:
  - $M_2 \equiv (\lambda x.P'')Q''$. Then by induction hypothesis and Lemma 2.14 one can take $M_3 = P'''[Q'''/x]$.
  - $M_2 \equiv P''[Q''/x]$ with $P \rightrightarrows P''$ and $Q \rightrightarrows Q''$. By induction hypothesis and Lemma 2.14 one can can take $M_3 \equiv P'''[Q'''/x]$.

□

Lemma 2.16 $\twoheadrightarrow_\beta$ *is the transitive closure of* $\rightrightarrows$.

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

$$\longrightarrow_\beta \subseteq \rightrightarrows \subseteq \twoheadrightarrow_\beta$$

Since $\twoheadrightarrow_\beta$ is this transitive closure of $\longrightarrow_\beta$, so it is of $\rightrightarrows$. □

Theorem 2.17 (Church-Rosser) *Let* M *be a* λ*-term, if* $M \twoheadrightarrow_\beta N_1$ *and* $M \twoheadrightarrow_\beta N_2$*, there is a* λ*-term* N *such that* $N_1 \twoheadrightarrow_\beta N$ *and* $N_2 \twoheadrightarrow_\beta N$*. Thus* $\longrightarrow_\beta$ *is Church-Rosser.*

Proof: By Lemma 2.15 $\rightrightarrows$ satisfies the diamond property. By Lemma 2.16 $\twoheadrightarrow_\beta$ is the transitive closure of $\rightrightarrows$. The result follows by Lemma 2.12. □

2.4.1 The notion of η-reduction

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

Definition 2.18 *1. The notion of* η*-reduction is given by*

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

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

Theorem 2.19 $\beta\eta$ *is Church-Rosser.*

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 2.20 F_L *is defined as follows:*

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

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

Theorem 2.21 *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 2.22 F_∞ *is defined as follows:*

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

Theorem 2.23 *The reduction strategy* F_∞ *is maximal.*

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 2.24 *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 2.25 *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 2.26 *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 $\langle \Lambda_{\mathcal{L}}, \beta \rangle$.

2.7 The *de Brujin* notation

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

Definition 2.27 *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.

Definition 2.28 *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 2.29 *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:*

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

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

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

Exercises

1 For any of the following λ-terms reduce them to β normal for (if possible):

(a) xy

(b) $(\lambda x.x)(yz)$

(c) $(\lambda x.xy)(\lambda z.z)$

(d) $(\lambda xy.xy)$

(e) $(\lambda x.x\lambda x.xy)(yy)$

(f) $(\lambda x.xx)(\lambda x.xx)$

(g) $(\lambda x.xxx)(\lambda x.xxx)$

(h) $(\lambda xy.y)((\lambda x.xx)(\lambda x.xx))b$

(i) $M = AAx$ where $A = \lambda axz.z(aax)$

2 What is wrong with the following reductions?

(a) $(\lambda xy.yx)y \longrightarrow_\beta \lambda y.yy$

(b) $(\lambda xx.xx)y \longrightarrow_\beta \lambda x.yy$

(c) $(\lambda x.xx)(\lambda y.ay)k \longrightarrow_\beta (\lambda x.xx)ak \longrightarrow_\beta akak$

3 Let $S = \lambda xyz.xz(yz)$, $K = \lambda xy.x$, $B = \lambda xyz.x(yz)$ and $I = \lambda x.x$. Show that

(a) $S(KI) \longrightarrow^*_{\beta\eta} I$

(b) $BI \longrightarrow^*_{\beta\eta} I$

(c) $SKK \longrightarrow^*_{\beta\eta} I$

(d) $S(KS)K \longrightarrow^*_{\beta\eta} B$

4 Let $T_1 = (\lambda x.xx)(\lambda x.xx)$ and $T_2 = (\lambda xy.yx)(\lambda x.xx)(\lambda x.x)$. Can T_1 and T_2 be reduced to the same λ-term T?

5 If M and N are two λ-terms, then $M \equiv_\alpha N$ if they are the same λ-term or if one can be obtained from the other by a change of bound variables. For example $(\lambda x.x)z \equiv_\alpha (\lambda y.y)z$ and $(\lambda x.x)z \not\equiv_\alpha (\lambda x.y)z$. Prove, by strutural inductionof a λ-term M that if $x \neq y$ and if $x \notin FV(L)$ then:

$$M[N/x][L/y] \equiv_\alpha M[L/y][N[L/y]/x]$$

6 Show that if x does not occur free in the λ-term M, $\forall N.M[N/x] = M$.

7 Consider the $\twoheadrightarrow_1$ in Λ, inductively defined in the following way:

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

Prove, by induction on M, that if $N \twoheadrightarrow_1 N'$, then $M[N/x] \twoheadrightarrow_1 M[N'/x]$.

8 Prove, by induction on the definition of $\twoheadrightarrow_1$, the following properties:

(a) $\lambda x.M \twoheadrightarrow_1 N$, implies that $N \equiv \lambda x.M'$ and $M \twoheadrightarrow_1 M'$

(b) $MN \twoheadrightarrow_1 L$, implies that

$$L \equiv M'N' \text{ and } M \twoheadrightarrow_1 M',\ N \twoheadrightarrow_1 N'$$

or

$$M \equiv \lambda x.P,\ L \equiv P'[N'/x] \text{ and } P \twoheadrightarrow_1 P',\ N \twoheadrightarrow_1 N'$$

9 Write the following λ-terms using the De Brujin notation:

(a) $(\lambda xy.y)((\lambda x.xx)(\lambda x.xx))(\lambda y.y)$

(b) $(\lambda fx.f(f(x)))(\lambda zw.z(z(zw)))$

(c) $M = \lambda x.AAx$ where $A = \lambda axz.z(aax)$

10 Reduce to normal form the following De Brujin terms:

(a) $(\lambda(\lambda 2)1)(\lambda 321)$

(b) $(\lambda\lambda.2(2(21)))(\lambda\lambda 2(21))$

(c) $(\lambda\lambda\lambda 31(21))(\lambda\lambda 2)(\lambda 1)$

11 Consider the set Λ^*, of De Brujin terms. Define a function $BL : \Lambda^* \longrightarrow \Lambda$ that converts terms in the De Brujin notation, to λ-calculus.

12 Consider the following set of term Λ' defined inductively as:

$$\begin{array}{rcl} x \in \mathcal{V} & \Rightarrow & x \in \Lambda' \\ M, N \in \Lambda' & \Rightarrow & (MN) \in \Lambda' \\ M \in \Lambda',\ x \in \mathcal{V} & \Rightarrow & (\lambda x M) \in \Lambda' \\ M, N \in \Lambda',\ x \in \mathcal{V},\ i \in \mathbb{N} & \Rightarrow & ((\lambda_i x M)N) \in \Lambda' \end{array}$$

and the following functions $|\cdot|, \varphi(\cdot) : \Lambda' \to \Lambda$:

$$\begin{array}{rcl} |x| & = & x \\ |MN| & = & |M||N| \\ |\lambda x.M| & = & \lambda x.|M| \\ |(\lambda_i x.M)N| & = & (\lambda x|M|)|N| \end{array}$$

$$\begin{array}{rcl} \varphi(x) & = & x \\ \varphi(MN) & = & \varphi(M)\varphi(N),\ \text{if } M \not\equiv (\lambda_i x P), \\ \varphi(\lambda x.M) & = & \lambda x.\varphi(M) \\ \varphi((\lambda_i x.M)N) & = & \varphi(M)[\varphi(N)/x] \end{array}$$

Let $\beta' = \beta_0 \cup \beta_1$ as defined in lectures.

Prove the following properties:

1. If $M \twoheadrightarrow_\beta N$ and $M = |M'|$ then there exists N' such that $M' \twoheadrightarrow_{\beta'} N'$ and $N = |N'|$.
2. If $M' \twoheadrightarrow_{\beta'} N'$ then $|M'| \twoheadrightarrow_\beta |N'|$.
3. $\varphi(M[N/x]) \equiv \varphi(M)[\varphi(N)/x]$.
4. If $M \twoheadrightarrow_{\beta'} N$ then $\varphi(M)| \twoheadrightarrow_\beta \varphi(N)$.
5. If $M \in \Lambda'$ then $|M| \twoheadrightarrow_\beta \varphi(M)$.

References

[Barendregt, 1984] Barendregt, H. P. (1984). *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, Amsterdam, second, revised edition.

[Barendregt et al., 1976] Barendregt, H. P., Bergstra, J., Klop, J. W., and Volken, H. (1976). Degrees, reductions and representability in the lambda calculus. Technical Report Preprint no.22, University of Utrecht, Department of Mathematics.

[Bruijn, 1972] Bruijn, N. G. D. (1972). Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. *INDAG. MATH*, 34:381–392.

[Church and Rosser, 1936] Church, A. and Rosser, J. B. (1936). Some Properties of Conversion. *Transactions of the American Mathematical Society*, 39(3):472–482.

[Curry and Feys, 1958] Curry, H. and Feys, R. (1958). *Combinatory Logic*, volume 1. North-Holland, Amsterdam.

[Dershowitz and Jouannaud, 1990] Dershowitz, N. and Jouannaud, J.-P. (1990). Rewrite Systems. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, chapter 6, pages 243–320. North-Holland, Amsterdam.

[Klop, 1992] Klop, J. W. (1992). Term Rewriting Systems. In Abramsky, S., Gabbay, D. M., and Maibaum, T. S. E., editors, *Handbook of Logic in Computer Science: Background - Computational Structures (Volume 2)*, pages 1–116. Clarendon Press, Oxford.

[van Raamsdonk et al., 1999] van Raamsdonk, F., Severi, P., Sorensen, M. H. B., and Xi, H. (1999). Perpetual reductions in λ-calculus. *Information and Computation*, 149(2):173–225.