1

The type-free λ-calculus

The λ-calculus is a family of prototype programming languages invented by a logician, Alonzo Church, in the 1930's. Their main feature is that they are *higher-order*; that is, they give a systematic notation for operators whose input and output values may be other operators. Also they are *functional*, that is they are based on the notion of *function* or *operator* and include notation for function-application and abstraction.

This book will be about the simplest of these languages, the *pure* λ-calculus, in which λ-terms are formed by application and abstraction from variables only. No atomic constants will be allowed.

1A λ-terms and their structure

1A1 Definition (λ-terms) An infinite sequence of ***term-variables*** is assumed to be given. Then linguistic expressions called ***λ-terms*** are defined thus:

(i) each term-variable is a λ-term, called an ***atom*** or ***atomic term***;
(ii) if M and N are λ-terms then (MN) is a λ-term called an ***application***;
(iii) if x is a term-variable and M is a λ-term then $(\lambda x \cdot M)$ is a λ-term called an ***abstract*** or a ***λ-abstract***.

A ***composite*** λ-term is a λ-term that is not an atom.

1A1.1 *Notation* *Term-variables* are denoted by "u", "v", "w", "x", "y", "z", with or without number-subscripts. Distinct letters denote distinct variables unless otherwise stated.

Arbitrary λ-terms are denoted by "L", "M", "N", "P", "Q", "R", "S", "T", with or without number-subscripts. For "*λ-term*" we shall usually say just "*term*".

Syntactic identity: "$M \equiv N$" will mean that M is the same expression as N (if M and N are terms or other expressions). But for identity of numbers, sets, etc. we shall say "=" as usual.

Parentheses and repeated λ's will often be omitted in such a way that, for example,

$$\lambda xyz \cdot M \equiv (\lambda x \cdot (\lambda y \cdot (\lambda z \cdot M))), \quad MNPQ \equiv (((MN)P)Q).$$

(The rule for restoring parentheses omitted from $MNPQ$ is called *association to the left.*)

1A2 Definition The ***length***, $|M|$, of a λ-term M is the number of occurrences of variables in M; in detail, define

$$|x| = 1, \quad |MN| = |M| + |N|, \quad |\lambda x \cdot M| = 1 + |M|.$$

1A2.1 *Example* $|(\lambda x \cdot yx)(\lambda z \cdot x)| = 5.$

1A3 Definition (Subterms) The ***subterms*** of a term M are defined by induction on $|M|$ as follows:

(i) an atom is a subterm of itself;
(ii) if $M \equiv \lambda x \cdot P$, its subterms are M and all subterms of P;
(iii) if $M \equiv P_1 P_2$, its subterms are all the subterms of P_1, all those of P_2, and M itself.

1A3.1 *Example* If $M \equiv (\lambda x \cdot yx)(\lambda z \cdot x(yx))$ its subterms are x, y, yx, $\lambda x \cdot yx$, $x(yx)$, $\lambda z \cdot x(yx)$ and M itself. (But not z.)

1A4 Notation (Occurrences, components) A subterm of a term M may have more than one occurrence in M; for example the term

$$(\lambda x \cdot yx)(\lambda z \cdot x(yx))$$

contains two occurrences of yx and three of x. The precise definition of "occurrence" is written out in 9A2, but the reader who already has a good intuitive idea of the occurrence-concept will go a long way without needing to look at this definition.

In this book occurrences will be underlined to distinguish them from subterms; for example we may say

"Let $\underline{P}$ be any occurrence of P in M".

An occurrence of λx will be called an ***abstractor***, and the occurrence of x in it will be called a ***binding occurrence of*** x.

All the occurrences of terms in M, other than binding occurrences of variables, will be called ***components of*** M.

1A5 Definition (Body, scope, covering abstractors) Let $\underline{\lambda x \cdot P}$ be a component of a term M. The displayed component $\underline{P}$ is called the ***body of*** $\underline{\lambda x \cdot P}$ or the ***scope of the abstractor*** $\underline{\lambda x}$.

The ***covering abstractors*** of a component $\underline{R}$ of M are the abstractors in M whose scopes contain $\underline{R}$.

1A6 Definition (Free, bound) A non-binding variable-occurrence $\underline{x}$ in a term M is said to be ***bound in*** M iff it is in the scope of an occurrence of λx in M, otherwise it is ***free in*** M.

A variable x is said to be ***bound in*** M iff M contains an occurrence of λx; and x is said to be ***free in*** M iff M contains a free occurrence of x. The set of all variables free in M is called

$$FV(M).$$

1A6.1 *Warning* Two distinct concepts have been defined here, free/bound occurrences and free/bound variables. A variable x may be both free and bound in M, for example if $M \equiv x(\lambda x \cdot x)$, but a particular occurrence of x in M cannot be both free and bound.

Also note that x is said to be bound in $\lambda x \cdot y$ even though its only occurrence there is a binding one.

1A7 Definition (Substitution) Define $[N/x]M$ to be the result of substituting N for each free occurrence of x in M and making any changes of bound variables needed to prevent variables free in N from becoming bound in $[N/x]M$. More precisely, define for all N, x, P, Q and all $y \not\equiv x$

$$
\begin{array}{rlll}
\text{(i)} & [N/x]x & \equiv N, & \\
\text{(ii)} & [N/x]y & \equiv y, & \\
\text{(iii)} & [N/x](P,Q) & \equiv (([N/x]P)([N/x]Q)), & \\
\text{(iv)} & [N/x](\lambda x \cdot P) & \equiv \lambda x \cdot P, & \\
\text{(v)} & [N/x](\lambda y \cdot P) & \equiv \lambda y \cdot P & \textit{if } x \notin FV(P), \\
\text{(vi)} & [N/x](\lambda y \cdot P) & \equiv \lambda y \cdot [N/x]P & \textit{if } x \in FV(P) \textit{ and } y \notin FV(N), \\
\text{(vii)} & [N/x](\lambda y \cdot P) & \equiv \lambda z \cdot [N/x][z/y]P & \textit{if } x \in FV(P) \textit{ and } y \in FV(N).
\end{array}
$$

(In (vii) z is the first variable in the sequence given in 1A1 which does not occur free in NP.)

1A7.1 *Notation* (Simultaneous substitution) For any $N_1, \ldots, N_n$ and any distinct $x_1, \ldots, x_n$, the result of simultaneously substituting N_1 for x_1, N_2 for $x_2, \ldots$ in M, and changing bound variables to avoid clashes, is defined similarly to $[N/x]M$. (For a neat definition see Stoughton 1988 §2.) It is called

$$[N_1/x_1, \ldots, N_n/x_n]M.$$

1A8 Definition (Changing bound variables, α-conversion) Let $y \notin FV(M)$; then we say

$$(\alpha) \qquad \lambda x \cdot M \equiv_\alpha \lambda y \cdot [y/x]M,$$

and the act of replacing an occurrence of $\lambda x \cdot M$ in a term by $\lambda y \cdot [y/x]M$ is called a ***change of bound variables***. If P changes to Q by a finite (perhaps empty) series of changes of bound variables we say P ***α-converts to*** Q or

$$P \equiv_\alpha Q.$$

1A8.1 *Note* Some basic lemmas about α-conversion and substitution are given in HS 86 §1B. Two simple properties that will be needed here are

(i) $P \equiv_\alpha Q \implies |P| = |Q|$,

(ii) $P \equiv_\alpha Q \implies FV(P) = FV(Q)$.

1A9 Definition A term M ***has a bound-variable clash*** iff M contains an abstractor $\underline{\lambda x}$ and a (free, bound or binding) occurrence of x that is not in its scope.

Examples of terms with bound-variable clashes are

$$x(\lambda x \cdot N), \quad \lambda x \cdot \lambda y \cdot \lambda x \cdot N, \quad (\lambda x \cdot P)(\lambda x \cdot Q).$$

We shall be mainly interested in terms *without* such clashes.

1A9.1 *Lemma* *Every term can be α-converted to a term without bound-variable clashes.*

Proof By the lemmas in HS 86 §1B. □

1A10 Definition (Closed terms) A ***closed term*** or ***combinator*** is a term in which no variable occurs free.

1A10.1 *Example* The following closed terms will be used in examples and results throughout this book.

$$\begin{array}{lll}
\mathbf{B} \equiv \lambda xyz\cdot x(yz), & \mathbf{B}' \equiv \lambda xyz\cdot y(xz), \\
\mathbf{C} \equiv \lambda xyz\cdot xzy, & \mathbf{I} \equiv \lambda x\cdot x, \\
\mathbf{K} \equiv \lambda xy\cdot x, & \mathbf{S} \equiv \lambda xyz\cdot xz(yz), \\
\mathbf{W} \equiv \lambda xy\cdot xyy, & \mathbf{Y} \equiv \lambda x\cdot(\lambda y\cdot x(yy))(\lambda y\cdot x(yy)), \\
\bar{0} \equiv \lambda xy\cdot y, & \bar{1} \equiv \lambda xy\cdot xy, \\
\bar{n} \equiv \lambda xy\cdot x^n y \equiv \lambda xy\cdot x(x(\ldots(xy)\ldots)) & (n\ x\text{'s applied to } y) .
\end{array}$$

($\mathbf{Y}$ is Curry's ***fixed-point combinator***, see HS 86 Ch.3 §3B for background; the terms $\bar{n}$ are the ***Church numerals*** for $n = 0, 1, 2, \ldots$, see HS 86 Def. 4.2.)

1B β-reduction and β-normal forms

This section outlines the definition and main properties of the term-rewriting procedure called β-reduction. Further details can be found in many other books, for example HS 86 Chs. 1–6 and Barendregt 1984 Chs. 3 and 11–14.

1B1 Definition (β-contraction) A ***β-redex*** is any term $(\lambda x \cdot M)N$; its ***contractum*** is $[N/x]M$ and its ***re-write rule*** is

$$(\lambda x \cdot M)N \ \triangleright_{1\beta}\ [N/x]M.$$

Iff P contains a β-redex-occurrence $\underline{R} \equiv \underline{(\lambda x \cdot M)N}$ and Q is the result of replacing this by $[N/x]M$, we say P ***β-contracts to*** Q ($P \triangleright_{1\beta} Q$) and we call the triple $\langle P, \underline{R}, Q\rangle$ a ***β-contraction of*** P.

1B1.1 *Lemma* $P \triangleright_{1\beta} Q \implies FV(P) \supseteq FV(Q)$.

1B2 Definition (β-reduction) A ***β-reduction*** of a term P is a finite or infinite sequence of β-contractions with form

(i) $$\langle P_1, \underline{R}_1, Q_1\rangle, \quad \langle P_2, \underline{R}_2, Q_2\rangle, \ \ldots$$

where $P_1 \equiv_\alpha P$ and $Q_i \equiv_\alpha P_{i+1}$ for $i = 1, 2, \ldots$. (The empty sequence is allowed.) We say a finite reduction is ***from*** P ***to*** Q iff either it has $n \geq 1$ contractions and $Q_n \equiv_\alpha Q$

or it is empty and $P \equiv_\alpha Q$. A reduction from P to Q is said to ***terminate*** or ***end*** at Q. Iff there is a reduction from P to Q we say P ***β-reduces to*** Q, or

$$P \rhd_\beta Q.$$

Note that α-conversions are allowed in a β-reduction.

1B3 Definition The ***length*** of a β-reduction is the number of its β-contractions (finite or ∞). A reduction with ***maximal length*** is one that continues as long as there are redexes to contract (i.e. one that either is infinite or ends at a term containing no redexes).

1B4 Definition (β-conversion) Iff we can change P to Q by a finite sequence of β-reductions and reversed β-reductions, we say P ***β-converts to*** Q, or P ***is β-equal to*** Q, or

$$P =_\beta Q.$$

A reversed β-reduction is called a ***β-expansion.***

1B4.1 *Exercise* For every term F let $X_F \equiv \mathbf{Y}F$ where $\mathbf{Y}$ is the fixed-point combinator defined in 1A10.1; show that

$$FX_F =_\beta X_F.$$

1B5 Church-Rosser Theorem for β (i) *If* $M \rhd_\beta P$ *and* $M \rhd_\beta Q$ *(see Fig. 1B5a) then there exists* T *such that*

$$P \rhd_\beta T, \quad Q \rhd_\beta T.$$

(ii) *If* $P =_\beta Q$ *(see Fig. 1B5b) then there exists* T *such that*

$$P \rhd_\beta T, \quad Q \rhd_\beta T.$$

Proof of 1B5 (i) See HS 86 Appendix 1 or Barendregt 1984 §3.2. (ii) This is deduced from (i) as suggested in Fig. 1B5b. □

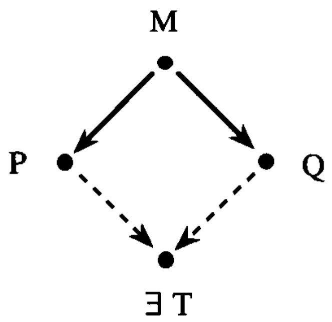

Fig. 1B5a.

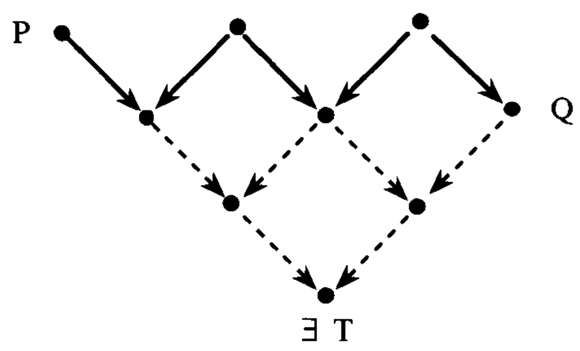

Fig. 1B5b.

1B6 Definition (β-normal forms) A ***β-normal form*** is a term that contains no β-redexes. The class of all β-nf's is called ***β-nf***. We say a term M ***has β-nf*** N iff

$$M \rhd_\beta N \text{ and } N \in \beta\text{-}nf.$$

1B6.1 *Note* Roughly speaking, a reduction can be thought of as a computation and a β-nf as its result. One main aim when designing a type-theory is to give it the property that every computation can be pursued to a result if the operator wishes, i.e. that every term with a type has a β-nf. This gives normal forms even more significance in a type-theory than they already have in a type-free theory.

(Terms in general do not necessarily have β-nf's of course. The simplest term without one is $(\lambda x \cdot xx)(\lambda x \cdot xx)$.)

1B7 NF-Uniqueness Lemma *Modulo α-conversion, a term M has at most one β-nf.*

Proof An easy application of the Church-Rosser theorem. □

1B7.1 *Notation* If M has a β-nf it will be called $M_{*\beta}$.

1B8 Definition (Leftmost reductions) The ***leftmost β-redex-occurrence*** in a term P is the β-redex-occurrence whose leftmost parenthesis is to the left of all the parentheses in all the other β-redex-occurrences in P.

The leftmost β-reduction of a term P is a β-reduction of P with maximal length, say

$$\langle P_1, \underline{R}_1, Q_1\rangle, \quad \langle P_2, \underline{R}_2, Q_2\rangle, \ \ldots,$$

such that $\underline{R}_i$ is the leftmost β-redex-occurrence in P_i for all $i \geq 1$ (and P_1 α-converts to P and P_{i+1} α-converts to Q_i for all $i \geq 1$).

1B9 Leftmost-reduction Theorem *A term M has a β-nf $M_{*\beta}$ iff the leftmost β-reduction of M is finite and ends at $M_{*\beta}$.*

Proof See Curry and Feys 1958 §4E Cor. 1.1. (In fact this result is an immediate corollary of a slightly deeper result called the *standardization theorem*; for the latter see Curry and Feys 1958 §4E Thm. 1 or Barendregt 1984 Thm. 11.4.7, or the particularly clear proof in Mitschke 1979 Thm. 7.) □

1B9.1 *Example* The leftmost reduction of the fixed-point combinator **Y** in 1A10.1 is easily seen to be infinite, so **Y** has no β-nf.

1B9.2 *Note* (Seeking β-normal forms) The leftmost reduction of a term M is completely determined by M, so by 1B9 it gives an algorithm for seeking $M_{*\beta}$: if $M_{*\beta}$ exists the leftmost reduction of M will end at $M_{*\beta}$, and if not, this reduction will be infinite. Of course this algorithm does not decide in finite time whether M has a β-nf; and in fact this cannot be done, as the set of terms with normal forms is not recursive. (See e.g. HS 86 Cor 5.6.2 or Barendregt 1984 Thm. 6.6.5.)

1B10 Lemma (Structure of a β-normal form) *Every β-nf N can be expressed uniquely in the form*

(i) $$N \equiv \lambda x_1 \ldots x_m \cdot y N_1 \ldots N_n \qquad (m \geq 0, n \geq 0),$$

where $N_1, \ldots, N_n$ are β-nf's. And if N is closed then $y \in \{x_1, \ldots, x_m\}$.

Proof Easy induction on $|N|$. Note the uniqueness. □

1B10.1 *Note* The following special cases of 1B10 are worth mention:

$$\begin{array}{llcll}
m = n = 0: & N & \equiv & y & \text{(an atom);}\\
m = 0, n \geq 1: & N & \equiv & y N_1 \ldots N_n & \text{(an application);}\\
m \geq 1: & N & \equiv & \lambda x_1 \ldots x_m \cdot P & \text{(an abstract);}\\
m \geq 1, n = 0: & N & \equiv & \lambda x_1 \ldots x_m \cdot y & \text{(called an \textbf{\textit{abstracted atom}}).}
\end{array}$$

1B10.2 *Exercise* Prove that β-nf is the smallest class of terms satisfying (i) and (ii) below:

(i) all variables are in β-nf;
(ii) for all $m, n \geq 0$ with $m + n \geq 1$, and all $x_1, \ldots, x_m, y$,

$$N_1, \ldots, N_n \in \beta\text{-}nf \quad \Longrightarrow \quad \lambda x_1 \ldots x_m \cdot y N_1 \ldots N_n \in \beta\text{-}nf.$$

1C η- and $\beta\eta$-reductions

This section sketches the most basic properties of η- and $\beta\eta$-reductions. For more details see HS 86 Ch. 7 and Barendregt 1984 §15.1.

1C1 Definition (η-reduction, η-conversion) An ***η-redex*** is any term $\lambda x \cdot M x$ with $x \notin FV(M)$; its re-write rule is

$$\lambda x \cdot M x \rhd_{1\eta} M.$$

Its ***contractum*** is M. The definitions of ***η-contracts***, ***η-reduces*** ($\rhd_\eta$), ***η-converts*** ($=_\eta$), etc. are like those of the corresponding β-concepts in 1B.

1C2 Lemma *All η-reductions are finite; in fact an η-reduction $P \triangleright_\eta Q$ must have length $\leq |P|/2$.*

Proof Each η-contraction reduces $|P|$ to $|P|-2$. □

1C3 Definition The ***η-family*** $\{P\}_\eta$ of a term P is the set of all terms Q such that $P \triangleright_\eta Q$.

1C3.1 *Note* By 1C2, $\{P\}_\eta$ is finite.

1C4 Church-Rosser Theorem for η *If $P =_\eta Q$ then there exists T such that*

$$P \triangleright_\eta T, \qquad Q \triangleright_\eta T.$$

Proof Straightforward. (Barendregt 1984 Lemma 3.3.7.) □

1C5 Definition (βη-reduction, βη-conversion) A ***βη-redex*** is any β- or η-redex. The definitions of ***βη-contracts***, ***βη-reduces*** ($\triangleright_{\beta\eta}$), ***βη-converts*** ($=_{\beta\eta}$) are like those of the corresponding β-concepts in 1B.

1C5.1 *Lemma* $\quad P \triangleright_{\beta\eta} Q \quad \Longrightarrow \quad FV(P) \supseteq FV(Q).$

1C5.2 *Note* A $\beta\eta$-reduction may have α-steps as well as β and η. The following theorem says that all its η-steps can be postponed to the end of the reduction.

1C6 η-Postponement Theorem *If $M \triangleright_{\beta\eta} N$ then there exists a term P such that*

$$M \triangleright_\beta P \triangleright_\eta N.$$

Proof Nederpelt 1973 Thm. 7.28 or Barendregt 1984 Cor. 15.1.6. □

1C7 Commuting Lemma *If $M \triangleright_\beta P$ and $M \triangleright_\eta Q$ (see Fig. 1C7a) then there exists a term T such that*

$$P \triangleright_\eta T, \qquad Q \triangleright_\beta T.$$

Proof Barendregt 1984 Lemma 3.3.8. □

1C7.1 *Corollary* *If $M \triangleright_{\beta\eta} P$ and $M \triangleright_\beta Q$ then there exists a term T such that*

$$P \triangleright_\beta T, \qquad Q \triangleright_{\beta\eta} T.$$

Proof By 1B5, 1C4 and 1C7. □

1C8 Church-Rosser Theorem for βη (i) *If $M \triangleright_{\beta\eta} P$ and $M \triangleright_{\beta\eta} Q$ then there exists T such that*

$$P \triangleright_{\beta\eta} T, \qquad Q \triangleright_{\beta\eta} T.$$

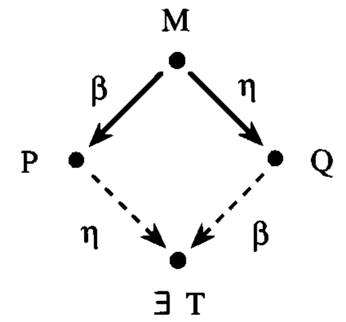

Fig. 1C7a.

(ii) *If* $P =_{\beta\eta} Q$ *then there exists* T *such that*

$$P \rhd_{\beta\eta} T, \qquad Q \rhd_{\beta\eta} T.$$

Proof (i) From 1B5, 1C4, 1C6, 1C7. (ii) From (i) as in Fig. 1B5b. □

1C9 Definition ($\beta\eta$- and η-normal forms) A ***$\beta\eta$-normal form ($\beta\eta$-nf)*** is a term without $\beta\eta$-redexes. The class of all $\beta\eta$-nf's is called ***$\beta\eta$-nf***. We say M ***has $\beta\eta$-nf*** N iff

$$M \rhd_{\beta\eta} N, \qquad N \in \beta\eta\text{-}nf.$$

Similarly we define ***η-normal form***, ***η-nf***, and M ***has η-nf*** N.

1C9.1 *Notation* The $\beta\eta$-nf and η-nf of a term M are unique modulo $\equiv_\alpha$ by the Church-Rosser theorems for $\beta\eta$ and η; they will be called

$$M_{*\beta\eta}, \qquad M_{*\eta}.$$

1C9.2 *Lemma* (i) *An η-reduction of a β-nf cannot create new β-redexes; more precisely*

$$M \in \beta\text{-}nf \text{ and } M \rhd_\eta N \implies N \in \beta\text{-}nf.$$

(ii) *For every* $M, M_{*\beta\eta}$ *is the η-nf of* $M_{*\beta}$; *i.e.* $M_{*\beta\eta} \equiv (M_{*\beta})_{*\eta}$.

Proof (i) It is easy to check all possible cases. (ii) By 1C2, $M_{*\beta}$ has an η-nf $(M_{*\beta})_{*\eta}$, and this is a $\beta\eta$-nf by (i). □

1C9.3 *Corollary* *If N is a β-nf then all the members of its η-family are β-nf's and exactly one of them is a $\beta\eta$-nf, namely* $N_{*\eta}$.

1C9.4 *Lemma* *A term has a $\beta\eta$-nf iff it has a β-nf.*

Proof For "only if", see Curry et al. 1972 §11E Lemma 13.1 or Barendregt 1984 Cor. 15.1.5. For "if", see 1C9.2. (By the way, do not confuse the present lemma with a claim that a term is *in* β-nf iff it is *in* $\beta\eta$-nf, which is of course false!) □

1C9.5 *Note* (Seeking $\beta\eta$-normal-forms) To seek for $M*_{\beta\eta}$, reduce M by its leftmost β-reduction. If this is finite, it must end at $M*_\beta$ and then the leftmost η-reduction will reach an η-nf in $\leq |M*_\beta|/2$ steps, by 1C2. If the leftmost β-reduction of M is infinite, $M*_\beta$ does not exist and hence by 1C9.4 neither does $M*_{\beta\eta}$. Of course this procedure does not decide in finite time whether $M*_{\beta\eta}$ exists; see the comment in 1B9.2.

1D Restricted λ-terms

The following restricted classes of λ-terms will play a role later in the correspondence between type-assignment and propositional logic.

1D1 Definition (λI-terms) A λ-term P is called a ***λI-term*** iff, for each subterm with form $\lambda x \cdot M$ in P, x occurs free in M at least once.

1D1.1 *Note* The λI-terms are the terms that were originally studied by Church. They have the property that if a λI-term has a normal form, so have all its subterms (Church 1941 §7, Thm. 7 XXXII). Church restricted his system to λI-terms because he regarded terms without normal forms as meaningless and preferred that meaningful terms did not have meaningless subterms. The λI-terms are discussed in detail in Barendregt 1984 Ch. 9.

The standard example of a non-λI-term is $\mathbf{K} \equiv \lambda xy \cdot x \equiv \lambda x \cdot (\lambda y \cdot x)$.

1D1.2 *Notation* Sometimes unrestricted λ-terms are called ***λK-terms***, and the unrestricted λ-calculus the ***λK-calculus***, to contrast with λI-terms and to emphasise the absence of restriction.

1D2 Definition (BCKλ-terms) A ***BCKλ-term*** is a λ-term P such that

(i) *for each subterm $\lambda x \cdot M$ of P, x occurs free in M at most once,*
(ii) *each free variable of P has just one occurrence free in P.*

1D2.1 *Examples* Of the terms in the list in 1A10.1 the following are BCKλ-terms:

$$\mathbf{B} \equiv \lambda xyz \cdot x(yz), \qquad \mathbf{B}' \equiv \lambda xyz \cdot y(xz), \qquad \mathbf{C} \equiv \lambda xyz \cdot xzy,$$
$$\mathbf{I} \equiv \lambda x \cdot x, \qquad \mathbf{K} \equiv \lambda xy \cdot x, \qquad \bar{n} \equiv \lambda xy \cdot x^n y \ (n = 0 \text{ or } 1).$$

And the following are not:

$$\mathbf{S} \equiv \lambda xyz \cdot xz(yz), \qquad \mathbf{W} \equiv \lambda xy \cdot xyy,$$
$$\mathbf{Y} \equiv \lambda x \cdot (\lambda y \cdot x(yy))(\lambda y \cdot x(yy)), \qquad \bar{n} \equiv \lambda xy \cdot x^n y \ (n \geq 2).$$

1D2.2 *Lemma The class of all BCKλ-terms is closed under abstraction, i.e. if M is a BCKλ-term then so is $\lambda x \cdot M$ for every variable x.*

Proof By 1D2(ii), x occurs free at most once in M. □

1D2.3 *Notes* (i) In contrast to the above lemma the class of all λI-terms is only closed under abstractions $\lambda x \cdot M$ such that x occurs free in M.

(ii) The BCKλ-terms are so called because the closed terms in this class correspond to combinations of three combinators called "**B**", "**C**" and "**K**" in combinatory logic (see 9F for details). They have also sometimes been called *linear λ-terms* but this name is nowadays usually applied to the following class.

1D3 Definition (BCIλ-terms) A ***BCIλ-term*** or ***linear λ-term*** is a λ-term P such that

(i) *for each subterm $\lambda x \cdot M$ of P, x occurs free in M exactly once,*
(ii) *each free variable of P has just one occurrence free in P.*

Clearly every BCIλ-term is a BCKλ-term, but the BCKλ-term **K** is not a BCIλ-term; in fact a term is a BCIλ-term iff it is both a λI-term and a BCKλ-term. The closed BCIλ-terms correspond to combinations of the combinators called **B**, **C** and **I** in combinatory logic; details are in 9F.

1D4 Lemma *Each of the three classes (λI-terms, BCKλ-terms and BCIλ-terms) is closed under $\beta\eta$-reduction, i.e. every term obtained by $\beta\eta$-reducing a member of the class is also in the class.*

Proof Straightforward. □

1D5 Definition A β-contraction $(\lambda x \cdot M)N \triangleright_{1\beta} [N/x]M$ is said to ***cancel*** N iff x does not occur free in M; it is said to ***duplicate*** N iff x has at least two free occurrences in M.

A β-reduction is ***non-duplicating*** iff none of its contractions duplicates; it is ***non-cancelling*** iff none cancels.

1D6 Lemma *Every β-reduction of a λI-term is non-cancelling; every one of a BCKλ-term is non-duplicating, and every one of a BCIλ-term is both.*