CHAPTER 2

CONVERSION

2.1. Lambda terms and conversion

The principal object of study in the λ-calculus is the set of lambda terms modulo convertibility. These notions will be introduced in this section.

2.1.1. DEFINITION. (i) *Lambda terms* are words over the following alphabet:

$v_0, v_1, \ldots$	variables,
λ	abstractor,
(,)	parentheses.

(ii) The set of λ-terms Λ is defined inductively* as follows:

(1) $x \in \Lambda$;

(2) $M \in \Lambda \Rightarrow (\lambda x M) \in \Lambda$;

(3) $M, N \in \Lambda \Rightarrow (MN) \in \Lambda$;

where x in (1) or (2) is an arbitrary variable.

2.1.2. NOTATION. (i) $M, N, L, \ldots$ denote arbitrary λ-terms.

(ii) $x, y, z, \ldots$ denote arbitrary variables.

(iii) Outermost parentheses are not written.

(iv) The symbol $\equiv$ denotes syntactic equality.

The following notation is used constantly. It results from Schönfinkel's reduction of functions of many variables to those of one variable.

2.1.3. NOTATION. (i) Let $\vec{x} \equiv x_1, \ldots, x_n$. Then

$$\lambda x_1 \cdots x_n.M \equiv \lambda \vec{x}.M \equiv \lambda x_1(\lambda x_2(\cdots (\lambda x_n(M))..)).$$

(ii) Let $\vec{N} \equiv N_1, \ldots, N_n$. Then

$$MN_1 \cdots N_n \equiv M\vec{N} \equiv (\cdots ((MN_1)N_2) \cdots N_n)$$

(*association to the left*).

(iii) $\| M \|$ is the *length* of M, that is, the number of symbols in M.

* If an inductive definition is given, it will always be understood that the class defined is the *least* class satisfying the conditions.

EXAMPLES. The following are λ-terms:

(i) xx,

(ii) $\lambda x.xx$,

(iii) $\lambda xy.yx (\equiv \lambda x(\lambda y(yx)))$,

(iv) $\lambda xy.yx(\lambda z.z)(\equiv \lambda x(\lambda y((yx)(\lambda z.z))))$.

It will not always be necessary to rewrite terms like (iii), (iv) above in their unabbreviated form. Once lemma 2.1.23 is proved the applicative behaviour of such terms is immediate.

The basic equivalence relation on λ-terms is that of *convertibility*. This relation will be generated by axioms. In order to formulate these axioms, a substitution operator is needed. $M[x:=N]$ denotes the result of substituting N for x in M. As in the case of predicate logic (or of programming), some care is needed in defining this operation in order to avoid confusion between free and bound variables. This care will be postponed for a moment.

2.1.4. DEFINITION. The theory $\boldsymbol{\lambda}$ has as formulas

$$M = N$$

where $M, N \in \Lambda$ and is axiomatized by the following axioms and rules:

(I) $(\lambda x.M)N = M[x:=N]$, (β-conversion);

(II.1) $M = M$;

(II.2) $M = N \Rightarrow N = M$;

(II.3) $M = N,\ N = L \Rightarrow M = L$;

(II.4) $M = N \Rightarrow MZ = NZ$;

(II.5) $M = N \Rightarrow ZM = ZN$;

(II.6) $M = N \Rightarrow \lambda x.M = \lambda x.N$, (rule ξ).

Provability in $\boldsymbol{\lambda}$ of an equation is denoted by $\boldsymbol{\lambda} \vdash M = N$ or often just by $M = N$. If $\boldsymbol{\lambda} \vdash M = N$, then M and N are called *convertible*. Note that $M \equiv N \Rightarrow M = N$, but not conversely.

The names "β-conversion" and "rule ξ" are historical. In the literature the theory $\boldsymbol{\lambda}$ appears under various names:

λ-calculus,
λβ-calculus,
λ*K*-calculus,
λ*K*β-calculus.

The *K* is to distinguish $\boldsymbol{\lambda}$ from a restricted theory $\boldsymbol{\lambda I}$ introduced in §2.2.

Note that $\boldsymbol{\lambda}$ is logic free: it is an equational theory. Connectives and quantifiers will be used in the informal metalanguage discussing about $\boldsymbol{\lambda}$. For example,

$$\forall M \quad (\lambda x.x)M = M,$$

$$M = N \Rightarrow MM = NN \wedge MN = NM$$

the meaning being

$$\forall M \in \Lambda \quad \boldsymbol{\lambda} \vdash (\lambda x.x)M = M,$$

$$\boldsymbol{\lambda} \vdash M = N \Rightarrow \boldsymbol{\lambda} \vdash MM = NN \text{ and } \boldsymbol{\lambda} \vdash MN = NM.$$

The following quite useful theorem gives an illustration of what can be proved in $\boldsymbol{\lambda}$.

2.1.5. Fixed point theorem. $\forall F\, \exists X\; FX = X$.

Proof. Let $W \equiv \lambda x.F(xx)$ and $X \equiv WW$. Then

$$X \equiv WW \equiv (\lambda x.F(xx))W = F(WW) = FX. \quad \square$$

Some syntactic notions and notations

2.1.6. Definition. A variable x occurs *free* in a λ-term M if x is not in the scope of a λx; x occurs *bound* otherwise.

For example in $M \equiv x(\lambda y.xy)$ the variable x occurs free (twice) and y occurs bound. A variable may occur both free and bound in a λ-term: $y(\lambda y.y)$.

In this respect λx has the same binding properties as $\forall x$ in predicate logic or $\int_0^1 \cdots \mathrm{d}x$ in calculus.

2.1.7. Definition. (i) $\mathrm{FV}(M)$ is the set of free variables in M and can be defined inductively as follows:

$$\mathrm{FV}(x) = \{x\},$$

$$\mathrm{FV}(\lambda x.M) = \mathrm{FV}(M) - \{x\},$$

$$\mathrm{FV}(MN) = \mathrm{FV}(M) \cup \mathrm{FV}(N).$$

(ii) M is *closed* or a *combinator* if $\mathrm{FV}(M) = \emptyset$.
(iii) $\Lambda^0 = \{M \in \Lambda | M \text{ is closed}\}$.
(iv) $\Lambda^0(\vec{x}) = \{M \in \Lambda | \mathrm{FV}(M) \subseteq \{\vec{x}\}\}$.
(v) A *closure* of $M \in \Lambda$ is $\lambda\vec{x}.M$, where $\{\vec{x}\} = \mathrm{FV}(M)$.
E.g. $\lambda xy.xy \in \Lambda^0$; $\lambda xy.xyz \in \Lambda^0(z)$.
Note that a closure of M depends on the order of the $\vec{x}$.

2.1.8. Definition. (i) M is a *subterm* of N(notation $M \subset N$) if $M \in \mathrm{Sub}(N)$, where $\mathrm{Sub}(N)$, the collection of subterms of N, is defined inductively as follows:

$$\begin{aligned}&\mathrm{Sub}(x) = \{x\},\\ &\mathrm{Sub}(\lambda x.N_1) = \mathrm{Sub}(N_1) \cup \{\lambda x.N_1\},\\ &\mathrm{Sub}(N_1N_2) = \mathrm{Sub}(N_1) \cup \mathrm{Sub}(N_2) \cup \{N_1N_2\}.\end{aligned}$$

(ii) A subterm may *occur* several times; e.g. $M \equiv \lambda x.x\mathbf{I}(x\mathbf{I})$ has two *occurrences* of the subterm $\mathbf{I} \equiv \lambda y.y$.

(iii) Let N_1, N_2 be subterm occurrences of M. Then N_1, N_2 are *disjoint* if N_1 and N_2 have no common symbol occurrences.

(iv) A subterm occurrence N of M is *active* if N occurs as $(NZ) \subset M$ for some Z; otherwise N is *passive*.

Example. (i) Let $M \equiv \lambda x.xy(\lambda z.y)$. Then $xy \subset M$, but $z \not\subset M$ and $y(\lambda z.y) \not\subset M$, since $M \equiv \lambda x.((xy)(\lambda z.y))$.

(ii) The occurrences of x and $(\lambda z.y)$ in M are disjoint. So are the first occurrences of y and $(\lambda z.y)$.

(iii) x, xy are active subterms of M; y, $\lambda z.y$ are passive ones.

2.1.9. Definition. Let F, $M \in \Lambda$. Then

(i) $F^0M \equiv M$; $F^{n+1}M \equiv F(F^nM)$,

(ii) $FM^{\sim 0} \equiv F$; $FM^{\sim n+1} \equiv FM^{\sim n}M$.

Terms modulo a change of bound variables

Now it will be shown why some care is needed for substitution.

2.1.10. Fallacy. $\forall MN\ M = N$.

Proof. Let $F \equiv \lambda xy.yx$. Then for all M,N

$$FMN \equiv ((\lambda x(\lambda y.yx))M)N = (\lambda y.yM)N = NM.$$

In particular $Fyx = xy$. But

$$Fyx = ((\lambda x(\lambda y.yx))y)x = (\lambda y.yy)x = xx.$$

Hence $xy = xx$. From this it is not difficult to derive any equation. See exercise 2.4.2 (iii). □

Refutation. The apparent contradiction results from the step

$$(\lambda x(\lambda y.yx))y = \lambda y.yy.$$

The free variable y becomes bound after substitution for x in $\lambda y.yx$. This should not be allowed. In predicate logic one expresses this by "y is not substitutable for x in $\lambda y.yx$", and in programming by "confusion of local and global variables." □

This problem will be avoided in 2.1.11–2.1.14 along the following lines. See appendix C for the details.

(1) Identify two terms if each can be transformed to the other by a renaming of its bound variables.

(2) Consider a λ-term as a representative of its equivalence class.

(3) Interpret substitution $M[x:=N]$ as an operation on the equivalence classes of M and N. This operation can be performed using representatives, provided that the bound variables are named properly as formulated in the variable convention below.

2.1.11. Definition. (i) A *change of bound variables* in M is the replacement of a part $\lambda x.N$ of M by $\lambda y.(N[x:=y])$, where y does not occur (at all) in N. (Because y is fresh there is no danger in the substitution $N[x:=y]$).

(ii) M is *α-congruent* with N, notation $M \equiv_\alpha N$, if N results from M by a series of changes of bound variables. For example

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

$$\lambda x.x(\lambda x.x) \equiv_\alpha \lambda x'.x'(\lambda x.x) \equiv_\alpha \lambda x'.x'(\lambda x''.x'').$$

In Church [1941] the renaming of bound variables was built into the conversion rules: there the theory $\boldsymbol{\lambda}$ was extended by the axiom scheme

$$\lambda x.M = \lambda y.M[x:=y] \qquad (\alpha\text{-conversion})$$

where y is not free or bound in M.

We prefer to identify α-congruent terms on a syntactic level.

2.1.12. Convention. Terms that are α-congruent are identified.

So now we write $\lambda x.x \equiv \lambda y.y$, etcetera.

2.1.13. Variable convention. If $M_1, \ldots, M_n$ occur in a certain mathematical context (e.g. definition, proof), then in these terms all bound variables are chosen to be different from the free variables.

Examples of the use of the variable convention.

(1) Already in the proof of the fixed point theorem 2.1.5 the variable convention was implicitly used. In defining $W \equiv \lambda x.F(xx)$, it is essential that $x \notin \mathrm{FV}(F)$.

(2) Let $\mathbf{K} \equiv \lambda xy.x$. Then

$$\mathbf{K}M \equiv (\lambda x(\lambda y.x))M = \lambda y.M.$$

Hence

$$(*) \qquad \forall M \quad \mathbb{K}M = \lambda y.M.$$

But

$$\mathbf{K}y \neq \lambda y.y.$$

This is so, because by the variable convention ($*$) has to be interpreted as

$$\forall M \quad \mathbb{K}M = \lambda y.M, \quad \text{where } y \notin \mathrm{FV}(M).$$

Hence by the identification of α-congruent terms

$$\mathbf{K}y = \lambda y'.y.$$

2.1.14. Moral. Using conventions 2.1.12 and 2.1.13 one can work with λ-terms in the naive way.

Naive means that substitution and other operations on terms can be performed without questioning whether they are allowed. The moral will be proved in appendix C.

Substitution

2.1.15. Definition. The result of substituting N for the free occurrences of x in M (notation $M[x:=N]$) is defined as follows.

$$\begin{aligned}
&x[x:=N] \equiv N;\\
&y[x:=N] \equiv y, \quad \text{if } x \not\equiv y;\\
&(\lambda y.M_1)[x:=N] \equiv \lambda y.(M_1[x:=N]);\\
&(M_1M_2)[x:=N] \equiv (M_1[x:=N])(M_2[x:=N]);
\end{aligned}$$

In the third clause it is not needed to say "provided that $y \not\equiv x$ and $y \notin \mathrm{FV}(N)$". By the variable convention this is the case.

2.1.16. Substitution lemma. If $x \not\equiv y$ and $x \notin \mathrm{FV}(L)$, then

$$M[x:=N][y:=L] \equiv M[y:=L][x:=N[y:=L]].$$

Proof. By induction on the structure of M.

Case 1. M is a variable.

Case 1.1. $M \equiv x$. Then both sides equal $N[y:=L]$ since $x \not\equiv y$.

Case 1.2. $M \equiv y$. Then both sides equal L, for $x \notin \mathrm{FV}(L)$ implies $L[x:= \cdots] \equiv L$.

Case 1.3. $M \equiv z \not\equiv x,y$. Then both sides equal z.

Case 2. $M \equiv \lambda z.M_1$. By the variable convention we may assume that $z \not\equiv x,y$ and z is not free in N,L. Then by the induction hypothesis

$$\begin{aligned}(\lambda z.M_1)[x:=N][y:=L] &\equiv \lambda z.M_1[x:=N][y:=L]\\ &\equiv \lambda z.M_1[y:=L][x:=N[y:=L]]\\ &\equiv (\lambda z.M_1)[y:=L][x:=N[y:=L]].\end{aligned}$$

Case 3. $M \equiv M_1M_2$. Then the statement follows again from the induction hypothesis. □

2.1.17. Proposition. (i) $M = M' \Rightarrow M[x:=N] = M'[x:=N]$.

(ii) $N = N' \Rightarrow M[x:=N] = M[x:=N']$.

(iii) $M = M'$, $N = N' \Rightarrow M[x:=N] = M'[x:=N']$.

Proof. (i) By induction on the length of proof of $M = M'$. Since this kind of proof occurs often, this one will be spelled out in detail. Later on such details will be left to the reader.

Case 1. $M = M'$ is an axiom $(\lambda y.A)B = A[y:=B]$. Then

$$\begin{aligned}M[x:=N] &\equiv (\lambda y.A[x:=N])(B[x:=N])\\ &= A[x:=N][y:=B[x:=N]]\\ &\equiv A[y:=B][x:=N]\\ &\equiv M'[x:=N]\end{aligned}$$

by the substitution lemma 2.1.16 (by the variable convention $y \not\equiv x$, $y \notin \mathrm{FV}(N)$).

Case 2. $M = M'$ is an axiom because $M \equiv M'$. Then the result follows immediately.

Case 3. $M = M'$ is $ZM_1 = ZM_1'$ and is a direct consequence of $M_1 = M_1'$. Hence

$$\begin{aligned}M[x:=N] &\equiv Z[x:=N]M_1[x:=N]\\ &= Z[x:=N]M_1'[x:=N] \quad \text{by the induction hypothesis,}\\ &\equiv M'[x:=N].\end{aligned}$$

The other derivation rules are treated similarly.

(i) Alternative proof.

$$\begin{aligned} M = M' &\Rightarrow \lambda x.M = \lambda x.M' \\ &\Rightarrow (\lambda x.M)N = (\lambda x.M')N \\ &\Rightarrow M[x:=N] = M'[x:=N]. \end{aligned}$$

(ii) Use induction on the structure of M.
(iii) By (i) and (ii)

$$M[x:=N] = M'[x:=N] = M'[x:=N']. \quad \square$$

Is the following true?

$$N = N' \Rightarrow \lambda x.x(\lambda y.N) = \lambda x.x(\lambda y.N').$$

It is, but this does not follow from proposition 2.1.17 (ii) since $(\lambda x.x(\lambda y.N))$ cannot be written as $(\lambda x.x(\lambda y.z))[z:=N]$ if x or y is free in N. Therefore the following notion of context is useful.

2.1.18. Definition. (i) A *context* $C[\ \ ]$ is a term with some holes in it. More formally:

x is a context,
$[\ \ ]$ is a context,
if $C_1[\ \ ]$ and $C_2[\ \ ]$ are contexts, then so are $C_1[\ \ ]C_2[\ \ ]$ and $\lambda x.C_1[\ \ ]$.

(ii) If $C[\ \ ]$ is a context and $M \in \Lambda$, then $C[M]$ denotes the result of placing M in the holes of $C[\ \ ]$. In this act free variables of M may become bound in $C[M]$.

Example. $C[\ \ ] \equiv \lambda x.x(\lambda y.[\ \ ])$ is a context. If $M \equiv xy$, then $C[M] \equiv \lambda x.x(\lambda y.xy)$.

Contexts are *not* considered modulo α-congruence. The essential feature of a context $C[\ \ ]$ is that a free variable in M may become bound in $C[M]$. *Par abus de langage* we write $C[\ \ ] \in \Lambda$ to indicate that $C[\ \ ]$ is a context.

2.1.19. Proposition. *Let* $C[\ \ ] \in \Lambda$. *Then*

$$N = N' \Rightarrow C[N] = C[N'].$$

Proof. Use induction on the structure of $C[\ \ ]$. $\square$

2.1.20. Lemma. (i) $\forall C[\ \ ]\ \forall \vec{x}\ \exists F\ \forall M \in \Lambda^0(\vec{x})\ C[M] = F(\lambda\vec{x}.M)$.
(ii) $\forall C[\ \ ]\ \forall M\ \exists \vec{x}\ \exists F\ C[M] = F(\lambda\vec{x}.M)$.

PROOF. (i) Use induction on the structure of $C[\ ]$.
(ii) By (i). $\square$

Sometimes simultaneous substitution is needed.

2.1.21. DEFINITION. (i) Let $\vec{N} \equiv N_1, \ldots, N_m$; $\vec{x} \equiv x_1, \ldots, x_n$. Then $\vec{N}$ *fits in* $\vec{x}$ if $m = n$ and the $\vec{x}$ do not occur in $\mathrm{FV}(\vec{N})$.
(ii) Let $\vec{N} \equiv N_1, \ldots, N_m$; $\vec{L} \equiv L_1, \ldots, L_n$. Then

$$\vec{N} = \vec{L} \quad \text{if} \quad n = m \text{ and } N_i = L_i \text{ for } 1 \leqslant i \leqslant n.$$

Similarly $\vec{N} \equiv \vec{L}$ is defined.
(iii) Let $\vec{N}$ fit in $\vec{x} \equiv x_1, \ldots, x_n$. Then

$$M[\vec{x} := \vec{N}] \equiv M[x_1 := N_1] \cdots [x_n := N_n].$$

[Here it is needed that the $\vec{x}$ do not occur in $\mathrm{FV}(\vec{N})$. For an alternative definition, see exercise 2.4.8.]
(iv) Let $M \in \Lambda$. As in predicate logic sometimes we write $M \equiv M(\vec{x})$, to indicate substitution:
if $M \equiv M(\vec{x})$ and $\vec{N}$ fits in $\vec{x}$, then $M(\vec{N}) \equiv M[\vec{x} := \vec{N}]$.

2.1.22. PROPOSITION. *Let $\vec{N}_i$ fit in $\vec{x}$. Then*

$$M_1(\vec{x}) = M_2(\vec{x}) \wedge \vec{N}_1 = \vec{N}_2 \Rightarrow M_1(\vec{N}_1) = M_2(\vec{N}_2).$$

PROOF. By proposition 2.1.17 (iii). $\square$

Combinatory completeness

2.1.23. LEMMA. *Let $\vec{x} \equiv x_1, \ldots, x_n$. Then $(\lambda \vec{x}.M)\vec{x} = M$.*

PROOF. If $n = 1$, then

$$(\lambda x_1 . M)x_1 = M[x_1 := x_1] \equiv M.$$

If $n = 2$, then

$$\begin{aligned}(\lambda \vec{x}.M)\vec{x} &\equiv ((\lambda x_1.(\lambda x_2.M))x_1)x_2 \\ &= (\lambda x_2.M)x_2 \quad \text{by the case } n = 1 \text{ for } \lambda x_2.M \\ &= M.\end{aligned}$$

The general case follows similarly (by induction on n). $\square$

2.1.24. COROLLARY (Combinatory Completeness). *Let* $M \equiv M(\vec{x})$. *Then*:
 (i) $\exists F\ F\vec{x} = M(\vec{x})$.
 (ii) $\exists F\ \forall \vec{N}\ F\vec{N} = M(\vec{N})$, *where* $\vec{N}$ *fits in* $\vec{x}$.
 (iii) *In* (i), (ii) *one can take* $F \equiv \lambda\vec{x}.M$.

PROOF. By the lemma, using proposition 2.1.22. □

EXAMPLE. $\exists F\ \forall MN\ FMN = \lambda z.zNM$. Indeed, take

$$F \equiv \lambda xy.(\lambda z.zyx) \equiv \lambda xyz.zyx.$$

Three combinators are of particular importance.

2.1.25. DEFINITION.

$$\mathbf{I} \equiv \lambda x.x, \qquad \mathbf{K} \equiv \lambda xy.x,$$

$$\mathbf{S} \equiv \lambda xyz.xz(yz).$$

2.1.26. COROLLARY. *For all* $M,\ N,\ L \in \Lambda$
 (i) $\mathbf{I}M = M$,
 (ii) $\mathbf{K}MN = M$,
 (iii) $\mathbf{S}MNL = ML(NL)$.

PROOF. By Corollary 2.1.24. □

It will be shown that **I**, **K**, **S** generate the set Λ^0 using only application, see §8.1. Note that $\mathbf{SKK} = \mathbf{I}$, hence just **K**, **S** generate Λ^0.

Extensionality

Lambda terms denote processes. Different terms may denote the same process. For example, $\lambda x.Mx$ and M both yield the same result MN when applied to a term N. Therefore the following rule is introduced.

2.1.27. DEFINITION. (i) Extensionality is the following derivation rule

$$Mx = Nx \Rightarrow M = N \qquad (\textit{ext})$$

provided that $x \notin \mathrm{FV}(MN)$.
 (ii) The theory $\boldsymbol{\lambda}$ extended by this rule is denoted by $\boldsymbol{\lambda} + \textbf{\textit{ext}}$.

The question rises what can be proved in $\boldsymbol{\lambda} + \textbf{\textit{ext}}$ but not in $\boldsymbol{\lambda}$. We have seen the example $\lambda x.Mx = M$. The following shows that this is essentially the only difference.

2.1.28. Definition. Consider the following axiom scheme $\boldsymbol{\eta}$

$$\lambda x.Mx = M \qquad (\eta\text{-conversion})$$

provided that $x \notin \mathrm{FV}(M)$. $\boldsymbol{\lambda\eta}$ is the theory $\boldsymbol{\lambda}$ extended with $\boldsymbol{\eta}$.

2.1.29. Theorem (Curry). *The theories* $\boldsymbol{\lambda} + \boldsymbol{ext}$ *and* $\boldsymbol{\lambda\eta}$ *are equivalent.*

Proof. First we show $\boldsymbol{\lambda} + \boldsymbol{ext} \vdash \boldsymbol{\eta}$. Indeed

$$(\lambda x.Mx)x = Mx$$

Hence if $x \notin \mathrm{FV}(M)$, then by $\boldsymbol{ext}$

$$\lambda x.Mx = M.$$

Conversely, $\boldsymbol{\lambda\eta}$ is closed under the rule $\boldsymbol{ext}$. Let

$$Mx = Nx,$$

with $x \notin \mathrm{FV}(MN)$. Then by rule $\boldsymbol{\xi}$

$$\lambda x.Mx = \lambda x.Nx.$$

Hence by $\boldsymbol{\eta}$ one has $M = N$. $\square$

Note that the rule $\boldsymbol{\xi}$ plays an essential role in the equivalence between $\boldsymbol{\lambda} + \boldsymbol{ext}$ and $\boldsymbol{\lambda\eta}$. Therefore rule $\boldsymbol{\xi}$ is sometimes referred to as the rule of *weak extensionality*.

The extensional λ-calculus will usually be denoted by $\boldsymbol{\lambda\eta}$. The following other names appear in the literature:

$\lambda\eta$-calculus,
$\lambda\beta\eta$-calculus,
$\lambda K\eta$-calculus,
$\lambda K\beta\eta$-calculus.

One of the reasons for considering $\boldsymbol{\lambda\eta}$ is that it enjoys a certain completeness property, see theorem 2.1.40.

Consistency

Since the theory $\boldsymbol{\lambda}$ is logic free, the notion of consistency has to be taken in the following sense

2.1.30. Definition. (i) An *equation* is a formula of the form $M = N$ with $M, N \in \Lambda$; the equation is *closed* if $M, N \in \Lambda^0$.

(ii) Let $\mathfrak{T}$ be a formal theory with equations as formulas. Then $\mathfrak{T}$ is *consistent* (notation Con($\mathfrak{T}$)) if $\mathfrak{T}$ does not prove every closed equation. In the opposite case $\mathfrak{T}$ is *inconsistent*.

(iii) If $\mathfrak{T}$ is a set of equations, then $\boldsymbol{\lambda} + \mathfrak{T}$ is the theory obtained from $\boldsymbol{\lambda}$ by adding the equations of $\mathfrak{T}$ as axioms. $\mathfrak{T}$ is called *consistent* (notation Con($\mathfrak{T}$)) if Con($\boldsymbol{\lambda} + \mathfrak{T}$).

The reader may be worried by the fallacy 2.1.10. Moreover the proof of the fixed point theorem 2.1.5 is in fact a diagonal argument often leading to inconsistencies. Therefore it is not obvious that the theory $\boldsymbol{\lambda}$ is consistent.

The concept of reduction, introduced in chapter 3, will provide an important proof theoretic tool for the theory $\boldsymbol{\lambda}$ and some extensions. Using this one can prove the following.

2.1.31. FACT. The theories $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda\eta}$ are consistent.

PROOF. See theorems 3.2.10(ii) and 3.3.11(ii). See also corollary 2.1.38. □

The theory $\boldsymbol{\lambda}$ extended by a single axiom may become inconsistent.

2.1.32. DEFINITION. Let $M, N \in \Lambda$. Then M and N are *incompatible*, notation $M \# N$, if $\neg$Con($M = N$).

2.1.33. EXAMPLE. $\mathbf{K} \# \mathbf{S}$.

PROOF. Reason in $\boldsymbol{\lambda} + \mathbf{K} = \mathbf{S}$: for all $X, Y, Z \in \Lambda$

$$\mathbf{K}XYZ = \mathbf{S}XYZ;$$

hence

$$XZ = XZ(YZ).$$

Take $X \equiv Z \equiv \mathbf{I}$. Then for all $Y \in \Lambda$

$$\mathbf{I} = Y\mathbf{I}.$$

By taking $Y \equiv \mathbf{K}M$, M arbitrary, it follows that for all M

$$\mathbf{I} = M.$$

Therefore $\boldsymbol{\lambda} + \mathbf{K} = \mathbf{S} \vdash M = \mathbf{I} = N$ for arbitrary M, N, in other words, $\neg$Con($\mathbf{K} = \mathbf{S}$). □

Normal forms

Consider a term like

$$(\lambda x.xa)\mathbf{I}.$$

This term can be "computed" to yield

$$\mathbf{I}a$$

and this gives

$$a.$$

The term a is called a normal form, for it does not "compute" any further. This notion is made precise as follows.

2.1.34. Definition. Let $M \in \Lambda$.

(i) *M is a β-normal form* (abbreviated as β-nf or just nf) if M has no subterm $(\lambda x.R)S$.

(ii) *M has a β*-nf if there exists an N such that $N = M$ and N is a β-nf.

If M is a nf, it is also said that M is *in* nf.

Examples. (i) $\mathbf{I}$ is in nf.

(ii) $\mathbf{KI}$ has a nf (namely $\lambda y.\mathbf{I}$).

(iii) Let $\mathbf{\Omega} \equiv (\lambda x.xx)(\lambda x.xx)$. Then $\mathbf{\Omega}$ has no nf as will be proved in chapter 3.

In the extensional theory $\boldsymbol{\lambda\eta}$ the notion of nf is a bit different, since there also terms such as $\lambda x.ax$ "want to be computed".

2.1.35. Definition. Let $M \in \Lambda$.

(i) M is a $\beta\eta$-nf if M has no subterm $(\lambda x.P)Q$ or $(\lambda x.Rx)$ with $x \notin \mathrm{FV}(R)$.

(ii) M has a $\beta\eta$-nf if

$$\exists N[\lambda\eta \vdash M = N \text{ and } N \text{ is a } \beta\eta\text{-nf}].$$

Examples. (i) $\mathbf{K}, \mathbf{S}$ are $\beta\eta$-nf's.

(ii) $\lambda x.x(\lambda z.xz)$ is not a $\beta\eta$-nf, but has as $\beta\eta$-nf the term $\lambda x.xx$.

2.1.36. Fact (Curry et al. [1972]). M has a $\beta\eta$-nf $\Leftrightarrow M$ has a β-nf.

Proof. See Corollary 15.1.5. □

Normal forms and consistency are connected as follows.

2.1.37. FACT. (i) If M,N are different β-nf's, then

$$\lambda \nvdash M = N.$$

(ii) Similarly for $\beta\eta$-nf's and provability in $\lambda\eta$.

PROOF. By theorems 3.2.10 and 3.3.11 and propositions 3.2.1 and 3.3.2. $\square$

2.1.38. COROLLARY. *The theories* $\boldsymbol{\lambda}$ *and* $\boldsymbol{\lambda\eta}$ *are consistent.*

PROOF. By fact 2.1.37 one has $\boldsymbol{\lambda\eta} \nvdash \mathbf{K} = \mathbf{S}$, since both $\mathbf{K}$ and $\mathbf{S}$ are in $\beta\eta$-nf . $\square$

On the other hand one has the following.

2.1.39. FACT (Böhm [1968]). If M, N are different $\beta\eta$-nf's, then $M \# N$.

PROOF. See corollary 10.4.3. $\square$

It follows that for terms having a nf the theory $\boldsymbol{\lambda\eta}$ is Hilbert-Post complete, cf. definition 4.1.22.

2.1.40. THEOREM. *Suppose* M,N *have a* nf. *Then either* $\boldsymbol{\lambda\eta} \vdash M = N$ *or* $\boldsymbol{\lambda\eta} + M = N$ *is inconsistent.*

PROOF. By fact 2.1.36 the terms M,N have $\beta\eta$-nf's, say M', N'.
Case 1. $M' \equiv N'$. Then

$$\boldsymbol{\lambda\eta} \vdash M = M' = N' = N.$$

Case 2. $M' \not\equiv N'$. Then by fact 2.1.39 one has $\neg\text{Con}(M' = N')$, hence $\neg\text{Con}(\boldsymbol{\lambda\eta} + M = N)$. $\square$

2.2. Some variants of the theory

In this section two variants of the theory λ are considered: combinatory logic and the λI-calculus. The technical development of these systems is given in chapters 7 and 9 respectively. Also a discussion is made about 'significance' of terms.

Combinatory logic

Independently of the λ-calculus a related theory, combinatory logic, was initiated by Schönfinkel [1924] and Curry [1930].