of the theory, and even after over 20 years it is still essential for the intending specialist. (It does not cover type theory.)

[Kri93] is a sophisticated and smooth introduction (originally published in French). It covers less than the present book but treats several topics that will only be mentioned in passing here, such as intersection-types and Böhm's theorem. It also treats Girard's type-system F.

[Han04] is a short computer-science-oriented introduction. Its core topics overlap the present book. They are covered in less detail, but some useful extra topics are also included.

[Rév88] is a computer-science-oriented introduction demanding slightly less mathematical experience from the reader than the present book and covering less material. There are some exercises (but no answers). In Section 2.5 there is an interesting variant of β-reduction which generates the same equality as the usual one, and is confluent, but does not depend on a preliminary definition of substitution.

[Tak91] is a short introduction for Japanese readers on about the same level as the present book. It also contains an introduction to recursive functions, but does not treat types or combinatory logic.

[Wol04] is a Russian-language textbook of which a large part is an introduction to λ-calculus and combinators, covering the first five chapters of the present book as well as some more special topics such as types.

[Rez82] is a bibliography of all the literature up to 1982 on λ-calculus and combinators, valuable for the reader interested in history. It has very few omissions, and includes many unpublished manuscripts.

[Bet99] is a bibliography of works published from 1980 to 1999, based largely on items reviewed in the journal *Mathematical Reviews*. It is an electronic '.ps' file, for on-screen reading. (Printing-out is not recommended; it has over 500 pages!)

2
Combinatory logic

2A Introduction to CL

Systems of combinators are designed to do the same work as systems of λ-calculus, but without using bound variables. In fact, the annoying technical complications involved in substitution and α-conversion will be avoided completely in the present chapter. However, for this technical advantage we shall have to sacrifice the intuitive clarity of the λ-notation.

To motivate combinators, consider the commutative law of addition in arithmetic, which says

$$(\forall x, y)\;\; x + y \;=\; y + x.$$

The above expression contains bound variables 'x' and 'y'. But these can be removed, as follows. We first define an addition operator A by

$$A(x,y) \;=\; x + y \qquad \text{(for all } x, y\text{)},$$

and then introduce an operator $\mathbf{C}$ defined by

$$(\mathbf{C}(f))(x,y) \;=\; f(y,x) \qquad \text{(for all } f, x, y\text{)}.$$

Then the commutative law becomes simply

$$A \;=\; \mathbf{C}(A).$$

The operator $\mathbf{C}$ may be called a *combinator*; other examples of such operators are the following:

$\mathbf{B}$,	which composes two functions:	$(\mathbf{B}(f,g))(x) \;=\; f(g(x))$;
$\mathbf{B}'$,	a reversed composition operator:	$(\mathbf{B}'(f,g))(x) \;=\; g(f(x))$;
$\mathbf{I}$,	the identity operator:	$\mathbf{I}(f) \;=\; f$;
$\mathbf{K}$,	which forms constant functions:	$(\mathbf{K}(a))(x) \;=\; a$;

S, a stronger composition operator: $(\mathbf{S}(f,g))(x) = f(x, g(x))$;
W, for doubling or 'diagonalizing': $(\mathbf{W}(f))(x) = f(x,x)$.

Instead of trying to define 'combinator' rigorously in this informal context, we shall build up a formal system of terms in which the above 'combinators' can be represented. Just as in the previous chapter, the system to be studied here will be the simplest possible one, with no syntactical complications or restrictions, but with the warning that systems used in practice are more complicated. The ideas introduced in the present chapter will be common to all systems, however.

Definition 2.1 (Combinatory logic terms, or CL-terms) Assume that there is given an infinite sequence of expressions v_0, v_{00}, v_{000}, ... called *variables*, and a finite or infinite sequence of expressions called *atomic constants*, including three called *basic combinators*: **I**, **K**, **S**. (If **I**, **K** and **S** are the only atomic constants, the system will be called *pure*, otherwise *applied*.) The set of expressions called *CL-terms* is defined inductively as follows:

(a) all variables and atomic constants, including **I**, **K**, **S**, are CL-terms;
(b) if X and Y are CL-terms, then so is (XY).

An *atom* is a variable or atomic constant. A *non-redex constant* is an atomic constant other than **I**, **K**, **S**. A *non-redex atom* is a variable or a non-redex constant. A *closed term* is a term containing no variables. A *combinator* is a term whose only atoms are basic combinators. (In the pure system this is the same as a closed term.)

Examples of CL-terms (the one on the left is a combinator):

$$((\mathbf{S}(\mathbf{KS}))\mathbf{K}), \qquad ((\mathbf{S}(\mathbf{K}\mathrm{v}_0))((\mathbf{SK})\mathbf{K})).$$

Notation 2.2 Capital Roman letters will denote CL-terms in this chapter, and 'term' will mean 'CL-term'.

'CL' will mean 'combinatory logic', i.e. the study of systems of CL-terms. (In later chapters, particular systems will be called 'CLw', 'CLξ', etc., but never just 'CL'.)

The rest of the notation will be the same as in Chapter 1. In particular 'x', 'y', 'z', 'u', 'v', 'w' will stand for variables (distinct unless otherwise stated), and '$\equiv$' for syntactic identity of terms. Also parentheses will be omitted following the convention of association to the left, so that $(((UV)W)X)$ will be abbreviated to $UVWX$.

Definition 2.3 The *length of* X (or $lgh(X)$) is the number of occurrences of atoms in X:

(a) $lgh(a) = 1$ for atoms a;
(b) $lgh(UV) = lgh(U) + lgh(V)$.

For example, if $X \equiv x\mathbf{K}(\mathbf{SS}xy)$, then $lgh(X) \equiv 6$.

Definition 2.4 The relation X *occurs in* Y, or X *is a subterm of* Y, is defined thus:

(a) X occurs in X;
(b) if X occurs in U or in V, then X occurs in (UV).

The set of all variables occurring in Y is called $\mathrm{FV}(Y)$. (In CL-terms all occurrences of variables are free, because there is no λ to bind them.)

Example 2.5 Let $Y \equiv \mathbf{K}(x\mathbf{S})((x\mathbf{S}yz)(\mathbf{I}x))$. Then $x\mathbf{S}$ and x occur in Y (and $x\mathbf{S}$ has two occurrences and x has three). Also

$$\mathrm{FV}(Y) \equiv \{x, y, z\}.$$

Definition 2.6 (Substitution) $[U/x]Y$ is defined to be the result of substituting U for every occurrence of x in Y: that is,

(a) $[U/x]x \equiv U$,
(b) $[U/x]a \equiv a$ for atoms $a \not\equiv x$,
(c) $[U/x](VW) \equiv ([U/x]V\ [U/x]W)$.

For all $U_1, \ldots, U_n$ and mutually distinct $x_1, \ldots, x_n$, the result of simultaneously substituting U_1 for x_1, U_2 for x_2, ..., U_n for x_n in Y is called

$$[U_1/x_1, \ldots, U_n/x_n]Y.$$

Example 2.7

(a) $[(\mathbf{SK})/x](yxx) \equiv y(\mathbf{SK})(\mathbf{SK})$,
(b) $[(\mathbf{SK})/x, (\mathbf{KI})/y](yxx) \equiv \mathbf{KI}(\mathbf{SK})(\mathbf{SK})$.

Exercise 2.8* (a) Give a definition of $[U_1/x_1, \ldots, U_n/x_n]Y$ by induction on Y.

(b) An example in Remark 1.23 shows that the identity

$$[U_1/x_1, \ldots, U_n/x_n]Y \equiv [U_1/x_1]([U_2/x_2](\ldots [U_n/x_n]Y)\ldots)$$

can fail. State a non-trivial condition sufficient to make this identity true.

2B Weak reduction

In the next section, we shall see how **I**, **K** and **S** can be made to play a rôle that is essentially equivalent to 'λ'. We shall need the following reducibility relation.

Definition 2.9 (Weak reduction) Any term $\mathbf{I}X$, $\mathbf{K}XY$ or $\mathbf{S}XYZ$ is called a *(weak) redex*. *Contracting* an occurrence of a weak redex in a term U means replacing one occurrence of

$$\mathbf{I}X \text{ by } X, \quad \text{or}$$
$$\mathbf{K}XY \text{ by } X, \quad \text{or}$$
$$\mathbf{S}XYZ \text{ by } XZ(YZ).$$

Iff this changes U to U', we say that U *(weakly) contracts to* U', or

$$U \rhd_{1w} U'.$$

Iff V is obtained from U by a finite (perhaps empty) series of weak contractions, we say that U *(weakly) reduces to* V, or

$$U \rhd_w V.$$

Definition 2.10 A *weak normal form* (or *weak nf* or *term in weak normal form*) is a term that contains no weak redexes. Iff a term U weakly reduces to a weak normal form X, we call X a *weak normal form of* U.

(Actually the Church–Rosser theorem later will imply that a term cannot have more than one weak normal form.)

Example 2.11 Define $\mathbf{B} \equiv \mathbf{S}(\mathbf{KS})\mathbf{K}$. Then $\mathbf{B}XYZ \rhd_w X(YZ)$ for all terms X, Y and Z, since

$$\begin{array}{lll} \mathbf{B}XYZ \equiv & \mathbf{S}(\mathbf{KS})\mathbf{K}XYZ & \\ \rhd_{1w} & \mathbf{KS}X(\mathbf{K}X)YZ & \text{by contracting } \mathbf{S}(\mathbf{KS})\mathbf{K}X \text{ to } \mathbf{KS}X(\mathbf{K}X) \\ \rhd_{1w} & \mathbf{S}(\mathbf{K}X)YZ & \text{by contracting } \mathbf{KS}X \text{ to } \mathbf{S} \\ \rhd_{1w} & \mathbf{K}XZ(YZ) & \text{by contracting } \mathbf{S}(\mathbf{K}X)YZ \\ \rhd_{1w} & X(YZ) & \text{by contracting } \mathbf{K}XZ. \end{array}$$

Example 2.12 Define $\mathbf{C} \equiv \mathbf{S}(\mathbf{BBS})(\mathbf{KK})$. Then $\mathbf{C}XYZ \rhd_w XZY$, since

$$\begin{array}{lll} \mathbf{C}XYZ \equiv & \mathbf{S}(\mathbf{BBS})(\mathbf{KK})XYZ & \\ \rhd_{1w} & \mathbf{BBS}X(\mathbf{KK}X)YZ & \text{by contracting } \mathbf{S}(\mathbf{BBS})(\mathbf{KK})X \\ \rhd_{1w} & \mathbf{BBS}X\mathbf{K}YZ & \text{by contracting } \mathbf{KK}X \\ \rhd_{w} & \mathbf{B}(\mathbf{S}X)\mathbf{K}YZ & \text{by 2.11} \\ \rhd_{w} & \mathbf{S}X(\mathbf{K}Y)Z & \text{by 2.11} \\ \rhd_{1w} & XZ(\mathbf{K}YZ) & \text{by contracting } \mathbf{S}X(\mathbf{K}Y)Z \\ \rhd_{1w} & XZY & \text{by contracting } \mathbf{K}YZ. \end{array}$$

Incidentally, in line 4 of this reduction, a redex $\mathbf{K}YZ$ seems to occur; but this is not really so, since, when all its parentheses are inserted, $\mathbf{B}(\mathbf{S}X)\mathbf{K}YZ$ is really $((((\mathbf{B}(\mathbf{S}X))\mathbf{K})Y)Z)$.

Exercise 2.13* Reduce the following CL-terms to normal forms:

(i) $\mathbf{SIK}x$, (ii) $\mathbf{SSK}xy$, (iii) $\mathbf{S}(\mathbf{SK})xy$,
(iv) $\mathbf{S}(\mathbf{KS})\mathbf{S}xyz$, (v) $\mathbf{SBBI}xy$.

Lemma 2.14 (Substitution lemma for $\rhd_w$)

(a) $X \rhd_w Y \implies \mathrm{FV}(X) \supseteq \mathrm{FV}(Y)$;
(b) $X \rhd_w Y \implies [X/v]Z \rhd_w [Y/v]Z$;
(c) $X \rhd_w Y \implies [U_1/x_1,\ldots,U_n/x_n]X \rhd_w [U_1/x_1,\ldots,U_n/x_n]Y$.

Proof For (a): for all terms U, V, W, we have: $\mathrm{FV}(\mathbf{I}U) \supseteq \mathrm{FV}(U)$, $\mathrm{FV}(\mathbf{K}UV) \supseteq \mathrm{FV}(U)$, and $\mathrm{FV}(\mathbf{S}UVW) \supseteq \mathrm{FV}(UW(VW))$.

For (b): any contractions made in X can also be made in the substituted X's in $[X/v]Z$.

For (c): if R is a redex and contracts to T, then $[U_1/x_1,\ldots,U_n/x_n]R$ is also a redex and contracts to $[U_1/x_1,\ldots,U_n/x_n]T$. □

Theorem 2.15 (Church–Rosser theorem for $\rhd_w$) *If $U \rhd_w X$ and $U \rhd_w Y$, then there exists a CL-term T such that*

$$X \rhd_w T \quad \text{and} \quad Y \rhd_w T.$$

Proof Appendix A2, Theorem A2.13. □

Corollary 2.15.1 (Uniqueness of nf) *A CL-term can have at most one weak normal form.*

Exercise 2.16 Prove that $\mathbf{SKK}X \triangleright_w X$ for all terms X. (Hence, by letting $I \equiv \mathbf{SKK}$, we obtain a term composed only of $\mathbf{S}$ and $\mathbf{K}$ which behaves like the combinator $\mathbf{I}$. Thus CL could have been based on just two atoms, $\mathbf{K}$ and $\mathbf{S}$. However, if we did this, a very simple correspondence between normal forms in CL and λ would fail; see Remark 8.23 and Exercise 9.19 later.)

Exercise 2.17* (Tricky) Construct combinators $\mathbf{B}'$ and $\mathbf{W}$ such that

$$\begin{array}{lll} \mathbf{B}'XYZ \triangleright_w Y(XZ) & & \text{(for all } X, Y, Z\text{)}, \\ \mathbf{W}XY \triangleright_w XYY & & \text{(for all } X, Y\text{)}. \end{array}$$

2C Abstraction in CL

In this section, we shall define a CL-term called '$[x].M$' for every x and M, with the property that

$$([x].M)N \triangleright_w [N/x]M. \qquad (1)$$

Thus the term $[x].M$ will play a role like $\lambda x.M$. It will be a combination of $\mathbf{I}$'s, $\mathbf{K}$'s, $\mathbf{S}$'s and parts of M, built up as follows.

Definition 2.18 (Abstraction) For every CL-term M and every variable x, a CL-term called $[x].M$ is defined by induction on M, thus:

(a) $[x].M \equiv \mathbf{K}M$ if $x \notin \mathrm{FV}(M)$;

(b) $[x].x \equiv \mathbf{I}$;

(c) $[x].Ux \equiv U$ if $x \notin \mathrm{FV}(U)$;

(f) $[x].UV \equiv \mathbf{S}([x].U)([x].V)$ if neither (a) nor (c) applies.[1]

Example 2.19

$$\begin{array}{rcll} [x].xy & \equiv & \mathbf{S}([x].x)([x].y) & \text{by 2.18(f)} \\ & \equiv & \mathbf{SI}(\mathbf{K}y) & \text{by 2.18 (b) and (a).} \end{array}$$

[1] These clauses are from [CF58, Section 6A, clauses(a)–(f)], deleting (d)–(e), which are irrelevant here. The notation '$[x]$' is from [CF58, Section 6A]. In [Ros55], [Bar84] and [HS86] the notation '$\lambda^* x$' was used instead, to stress similarities between CL and λ-calculus. But the two systems have important differences, and '$\lambda^* x$' has since acquired some other meanings in the literature, so the '$[x]$' notation is used here.

Warning 2.20 In λ-calculus an expression λx can be part of a λ-term, for example the term $\lambda x.xy$. But in CL, the corresponding expression $[x]$ is not part of the formal language of CL-terms at all. In the above example, the expression $[x].xy$ is not itself a CL-term, but is merely a short-hand to denote the CL-term $\mathbf{SI}(\mathbf{K}y)$.

Theorem 2.21 *The clauses in Definition 2.18 allow us to construct $[x].M$ for all x and M. Further, $[x].M$ does not contain x, and, for all N,*

$$([x].M)N \triangleright_w [N/x]M.$$

Proof By induction on M we shall prove that $[x].M$ is always defined, does not contain x, and that

$$([x].M)\,x \triangleright_w M.$$

The theorem will follow by substituting N for x and using 2.14(c).

Case 1: $M \equiv x$. Then Definition 2.18(b) applies, and

$$([x].x)\,x \equiv \mathbf{I}x \triangleright_w x.$$

Case 2: M is an atom and $M \not\equiv x$. Then 2.18(a) applies, and

$$([x].M)\,x \equiv \mathbf{K}Mx \triangleright_w M.$$

Case 3: $M \equiv UV$. By the induction hypothesis, we may assume

$$([x].U)\,x \triangleright_w U, \qquad ([x].V)\,x \triangleright_w V.$$

Subcase 3(i): $x \notin \mathrm{FV}(M)$. Like Case 2.
Subcase 3(ii): $x \notin \mathrm{FV}(U)$ and $V \equiv x$. Then

$$\begin{array}{rcll} ([x].M)\,x & \equiv & ([x].Ux)\,x & \\ & \equiv & Ux & \text{by 2.18(c),} \\ & \equiv & M. & \end{array}$$

Subcase 3(iii): Neither of the above two subcases applies. Then

$$\begin{array}{rcll} ([x].M)\,x & \equiv & \mathbf{S}([x].U)([x].V)\,x & \text{by 2.18(f)} \\ & \triangleright_{1w} & ([x].U)\,x\,(([x].V)\,x) & \\ & \triangleright_w & UV & \text{by induction hypothesis} \\ & \equiv & M. & \end{array}$$

(Note how the redexes and contractions for $\mathbf{I}$, $\mathbf{K}$, and $\mathbf{S}$ in 2.9 fit in with the cases in this proof; in fact this is their purpose.) □

Exercise 2.22* Evaluate

$$[x].u(vx), \qquad [x].x(\mathbf{S}y), \qquad [x].uxxv.$$

Remark 2.23 There are several other possible definitions of abstraction besides the one in Definition 2.18. For example, [Bar84, Definition 7.1.5] omits 2.18(c). But this omission enormously increases the lengths of terms $[x_1].(\ldots([x_n].M)\ldots)$ for most $x_1, \ldots, x_n, M$. Some alternative definitions of abstraction will be compared in Chapter 9.

Definition 2.24 For all variables $x_1, \ldots, x_n$ (not necessarily distinct),

$$[x_1, \ldots, x_n].M \equiv [x_1].([x_2].(\ldots([x_n].M)\ldots)).$$

Example 2.25

(a)
$$\begin{aligned} [x,y].x &\equiv [x].([y].x) \\ &\equiv [x].(\mathbf{K}x) && \text{by 2.18(a) for } [y] \\ &\equiv \mathbf{K} && \text{by 2.18(c).} \end{aligned}$$

(b)
$$\begin{aligned} [x,y,z].xz(yz) &\equiv [x].([y].([z].xz(yz))) \\ &\equiv [x].([y].(\mathbf{S}([z].xz)([z].yz))) && \text{by 2.18(f) for } [z] \\ &\equiv [x].([y].\mathbf{S}xy) && \text{by 2.18(c) for } [z] \\ &\equiv [x].\mathbf{S}x && \text{by 2.18(c) for } [y] \\ &\equiv \mathbf{S} && \text{by 2.18(c).} \end{aligned}$$

Exercise 2.26* Evaluate

$$[x,y,z].xzy, \qquad [x,y,z].y(xz), \qquad [x,y].xyy.$$

Compare $[x,y,z].xzy$ with the combinator **C** in Example 2.12. Note that $[x,y,z].y(xz)$ and $[x,y].xyy$ give answers to Exercise 2.17, combinators **B′** and **W**. There are other possible answers to that exercise, but the the abstraction algorithm in Definition 2.18 has changed the formerly tricky task of finding an answer into a routine matter.

Theorem 2.27 *For all variables $x_1, \ldots, x_n$ (mutually distinct),*

$$([x_1, \ldots, x_n].M)\, U_1 \ldots U_n \;\triangleright_w\; [U_1/x_1, \ldots, U_n/x_n]M.$$

Proof By 2.14(c) it is enough to prove $([x_1, \ldots, x_n].M)x_1 \ldots x_n \triangleright_w M$. And this comes from 2.21 by an easy induction on n. □

Lemma 2.28 (Substitution and abstraction)

(a) $\mathrm{FV}([x].M) = \mathrm{FV}(M) - \{x\}$ *if* $x \in \mathrm{FV}(M)$;

(b) $[y].[y/x]M \equiv [x].M$ *if* $y \notin \mathrm{FV}(M)$;

(c) $[N/x]([y].M) \equiv [y].[N/x]M$ *if* $y \notin \mathrm{FV}(xN)$.

Proof Straightforward induction on M. □

Comment Part (b) of Lemma 2.28 shows that the analogue in CL of the λ-calculus relation $\equiv_\alpha$ is simply identity. Part (c) is an approximate analogue of Definition 1.12(f).

The last few results have shown that $[x]$ has similar properties to λx. But it must be emphasized again that, in contrast to λx, $[x]$ is not part of the formal system of terms; $[x].M$ is defined in the meta-theory by induction on M, and is constructed from **I**, **K**, **S**, and parts of M.

2D Weak equality

Definition 2.29 (Weak equality or weak convertibility) We shall say X is *weakly equal* or *weakly convertible* to Y, or $X =_w Y$, iff Y can be obtained from X by a finite (perhaps empty) series of weak contractions and reversed weak contractions. That is, $X =_w Y$ iff there exist $X_0, \ldots, X_n$ $(n \geq 0)$ such that

$$(\forall i \leq n-1)\ (\ X_i \triangleright_{1w} X_{i+1} \ \text{ or } \ X_{i+1} \triangleright_{1w} X_i\ ),$$

$$X_0 \equiv X, \qquad X_n \equiv Y.$$

Exercise 2.30* Prove that, if **B**, **W** are the terms in Example 2.11 and Exercise 2.17, then

$$\mathbf{BWBI}x =_w \mathbf{SII}x.$$

Lemma 2.31

(a) $X =_w Y \implies [X/v]Z =_w [Y/v]Z$;

(b) $X =_w Y \implies [U_1/x_1, \ldots, U_n/x_n]X =_w [U_1/x_1, \ldots, U_n/x_n]Y$.

Theorem 2.32 (Church–Rosser theorem for $=_w$) *If $X =_w Y$, then there exists a term T such that*

$$X \triangleright_w T \quad \text{and} \quad Y \triangleright_w T.$$

Proof From 2.15, like the proof of 1.41 from 1.32. □

Corollary 2.32.1 *If $X =_w Y$ and Y is a weak normal form, then we have $X \triangleright_w Y$.*

Corollary 2.32.2 *If $X =_w Y$, then either X and Y have no weak normal form, or they both have the same weak normal form.*

Corollary 2.32.3 *If X and Y are distinct weak normal forms, then $X \neq_w Y$; in particular $\mathbf{S} \neq_w \mathbf{K}$. Hence $=_w$ is non-trivial in the sense that not all terms are weakly equal.*

Corollary 2.32.4 (Uniqueness of nf) *A term can be weakly equal to at most one weak normal form.*

Corollary 2.32.5 *If a and b are atoms other than $\mathbf{I}$, $\mathbf{K}$ and $\mathbf{S}$, and $aX_1 \dots X_m =_w bY_1 \dots Y_n$, then $a \equiv b$ and $m = n$ and $X_i =_w Y_i$ for all $i \leq m$.*

Warning 2.33 Although the above results show that $=_w$ in CL behaves very like $=_\beta$ in λ, the two relations do not correspond exactly. The main difference is that $=_\beta$ has the property which [CF58] calls (ξ), namely

$$(\xi) \qquad X =_\beta Y \implies \lambda x.X =_\beta \lambda x.Y.$$

(This holds in λ because any contraction or change of bound variable made in X can also be made in $\lambda x.X$.) When translated into CL, (ξ) becomes

$$X =_w Y \implies [x].X =_w [x].Y.$$

But for CL-terms, $[x]$ is not part of the syntax, and (ξ) fails. For example, take

$$X \equiv \mathbf{S}xyz, \qquad Y \equiv xz(yz);$$

then $X =_w Y$, but

$$[x].X \equiv \mathbf{S}(\mathbf{SS}(\mathbf{K}y))(\mathbf{K}z),$$
$$[x].Y \equiv \mathbf{S}(\mathbf{SI}(\mathbf{K}z))(\mathbf{K}(yz)).$$

These are normal forms and distinct, so by 2.32.3 they are not weakly equal.

For many purposes the lack of (ξ) is no problem and the simplicity of weak equality gives it an advantage over λ-calculus. This is especially true if all we want to do is define a set of functions in a formal theory, for example the recursive functions in Chapter 5. But for some other purposes (ξ) turns out to be indispensable, and weak equality is too weak. We then either have to abandon combinators and use λ, or add new axioms to weak equality to make it stronger. Possible extra axioms will be discussed in Chapter 9.

Exercise 2.34 *

(a) Construct a *pairing-combinator* $\mathbf{D}$ and two *projections* $\mathbf{D}_1$, $\mathbf{D}_2$ such that

$$\mathbf{D}_1(\mathbf{D}xy) \triangleright_w x, \qquad \mathbf{D}_2(\mathbf{D}xy) \triangleright_w y.$$

(b) Show that there is no combinator that distinguishes between atoms and composite terms; i.e. show that there is no A such that

$$AX =_w \mathbf{S} \quad \text{if } X \text{ is an atom,}$$
$$AX =_w \mathbf{K} \quad \text{if } X \equiv UV \text{ for some } U, V.$$

(Operations involving decisions that depend on the syntactic structure of terms can hardly ever be done by combinators.)

(c) Prove that a term X is in weak normal form iff X is minimal with respect to weak reduction, i.e. iff

$$X \triangleright_w Y \implies Y \equiv X.$$

(Contrast λ-calculus, 1.27(d).) Show that this would be false if there were an atom $\mathbf{W}$ with an axiom-scheme

$$\mathbf{W}XY \triangleright_w XYY.$$

Extra practice 2.35

(a) Reduce the following CL-terms to weak normal forms. (For some of them, use the reductions for $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{W}$ shown in Examples 2.11 and 2.12 and Exercise 2.17.)

(i) $\mathbf{KS}uxyz$, (ii) $\mathbf{S}(\mathbf{K}x)(\mathbf{KI}y)z$,
(iii) $\mathbf{CSI}xy$, (iv) $\mathbf{S}(\mathbf{CI})xy$,

(v) $\mathbf{B(BS)B}xyzu$, (vi) $\mathbf{BB(BB)}uvwxy$,
(vii) $\mathbf{B(BW(BC))(BB(BB))}xyzu$.

(b) Evaluate the following:

$[x].xu(xv)$, $[y].ux(uy)$, $[x,y].ux(uy)$.

(c) Prove that $\mathbf{SK}xy =_w \mathbf{KI}xy$. (Cf. Example 8.16(a).)

Further reading

There are many informative websites: just type 'combinatory' into a search engine. Also several introductions to λ include CL as well. The following are some references that focus mainly on CL.

[Ste72], [Bun02] and [Wol03] are introductions to CL aimed at about the same level as the present book. If the reader is dissatisfied with this book, he or she might find one of these more useful!

[Bar84] contains only one chapter on CL explicitly (Chapter 7). But most of the ideas in that book apply to CL as well as λ.

[Smu85] contains a humorous and clever account of combinators and self-application, and is especially good for examples and exercises on the interdefinability of various combinators.

[Sch24] is the first-ever exposition of combinators, by the man who invented them, and is a very readable non-technical short sketch.

[CF58] was the only book on CL for many years, and is still valuable for a few things, for example its discussion of particular combinators and interdefinability questions (Chapter 5), alternative definitions of $[x]$ (Section 6A), strong equality and reduction (Sections 6B–6F), and historical comments at the ends of chapters.

[CHS72] is a continuation and updating of [CF58], and contains proofs of the main properties of weak reduction (Section 11B). Definitions of $[x]$ are discussed in Section 11C. References for other topics will be given as they crop up later in the present book.

[Bac78] has historical interest; it is a strong plea for a functional style of programming, using combinators as an analogy, and led to an upsurge of interest in combinators, and to several combinator-based programming languages. (But Backus was not the first to advocate this; some precursors were [Fit58], [McC60], [Lan65], [Lan66], [BG66] and [Tur76].)

3
The power of λ and combinators

3A Introduction

The purpose of this chapter and the next two is to show some of the expressive power of both λ and CL.

The present chapter describes three interesting theorems which hold for both λ and combinators, and are used frequently in the published literature: the *fixed-point theorem*, *Böhm's theorem*, and a theorem which helps in proving that a term has no normal form.

After these results, Section 3E will outline the history of λ and CL, and will discuss the question of whether they have any meaning, or are just uninterpretable formal systems.

Then Chapter 4 will show that all recursive functions are definable in both systems, and Chapter 5 will deduce from this a general undecidability theorem.

Notation 3.1 This chapter is written in a neutral notation, which may be interpreted in either λ or CL, as follows.

Notation	*Meaning for λ*	*Meaning for CL*
term	λ-term	CL-term
$X \equiv Y$	$X \equiv_\alpha Y$	X is identical to Y
$X \triangleright_{\beta,w} Y$	$X \triangleright_\beta Y$	$X \triangleright_w Y$
$X =_{\beta,w} Y$	$X =_\beta Y$	$X =_w Y$
λx	λx	$[x]$

Definition 3.2 A *combinator* is (in λ) a closed pure term, i.e. a term containing neither free variables nor atomic constants, and (in CL) a