Proof. (i) Consider the natural transformation $\varepsilon: \mathbf{CL} \to \mathrm{id}$ defined for each $\mathscr{A}$ in $\mathbf{Cart}_N$ by $\varepsilon(\mathscr{A}): \mathbf{CL}(\mathscr{A}) \to \mathscr{A}$ as follows:

An object of $\mathbf{CL}(\mathscr{A})$ is a type of $\mathbf{L}(\mathscr{A})$, that is, an object of $\mathscr{A}$. Put $\varepsilon(\mathscr{A})(A) = A$.

An arrow $B \to C$ in $\mathbf{CL}(\mathscr{A})$ has the form $f = (y\in B, \varphi(y))$, where $\varphi(y)\in C$ in $\mathbf{L}(\mathscr{A})$. Put $\varepsilon(\mathscr{A})$ $(f) =$ the unique arrow $g: B \to C$ such that $gy \underset{y}{=} \varphi(y)$, using functional completeness.

It is easily verified that $\varepsilon(\mathscr{A})$ is an arrow in $\mathbf{Cart}_N$. Moreover, in view of functional completeness, it establishes a one-to-one correspondence between $\mathrm{Hom}_{\mathbf{CL}(\mathscr{A})}(B, C)$ and $\mathrm{Hom}_{\mathscr{A}}(B, C)$. Thus $\varepsilon(\mathscr{A})$ is an isomorphism.

(ii) Consider the natural transformation $\eta: \mathrm{id} \to \mathbf{LC}$ defined for each $\mathscr{L}$ in λ-**Calc** by $\eta(\mathscr{L}): \mathscr{L} \to \mathbf{LC}(\mathscr{L})$ as follows:

$$\eta(\mathscr{L})(A) \equiv A;$$
$$\eta(\mathscr{L})(\varphi(x_1,\ldots,x_n)) \equiv (z\in 1, \varphi(x_1,\ldots,x_n)) \text{ in } \mathbf{C}(\mathscr{L}(x_1,\ldots,x_n)).$$

Note that we have identified $\mathbf{C}(\mathscr{L})[x_1,\ldots,x_n]$ with $\mathbf{C}(\mathscr{L}(x_1,\ldots,x_n))$ as is justified by Proposition 11.2. It is easily verified that $\eta(\mathscr{L})$ is an arrow in λ-**Calc.** To see that $\eta(\mathscr{L})$ is an isomorphism, construct its inverse, which sends $(z\in 1, \varphi(z))$ onto $\varphi(*)$.

Corollary 11.4. $\mathbf{C}(\mathscr{L}_0)$, the free cartesian closed category with weak natural numbers object generated by the pure typed λ-calculus, is an initial object in $\mathbf{Cart}_N$.

The initial object of $\mathbf{Cart}_N$ may also be obtained by the methods of Section 4.

We end this section with a remark concerning the problem of how to interpret languages in categories. In the present context this is explained quite easily: an *interpretation* of a typed λ-calculus $\mathscr{L}$ in a cartesian closed category $\mathscr{A}$ with weak natural numbers object is just a translation $\mathscr{L} \to \mathbf{L}(\mathscr{A})$. By Theorem 11.3 (or just by adjointness, see Exercise 3 below), this is essentially the same as a cartesian closed functor $\mathbf{C}(\mathscr{L}) \to \mathscr{A}$. As already observed after Proposition 10.7, $\mathscr{L}_0$ has a unique interpretation in any cartesian closed category with weak natural numbers object.

Exercises

1. Show how to obtain the free cartesian closed category with a weak natural numbers object generated by any classification. (See Exercise 2 of Section 10.)
2. In the spirit of this section, find a new method for constructing the free cartesian closed category with a weak natural numbers object generated by a graph.
3. Show that $\mathbf{C}$ is left adjoint to $\mathbf{L}$ with adjunction η and ε.
4. Prove that $I_B\langle\langle y, v\rangle, x\rangle = (t\in 1,\ \ I(y,v,x))$ in $\mathbf{C}(\mathscr{L}(y,v,x))$.

12 The decision problem for equality

Let us look at the cartesian closed category with weak natural numbers object freely generated by the empty graph, as in Section 4, but with weak natural numbers object, or as in Exercise 2 of Section 11. Since both are initial objects in $\mathbf{Cart}_N$ (see Corollary 11.4), they are isomorphic. We shall write $\mathscr{C}_0$ for this initial object. $\mathscr{C}_0$ is of interest to logicians, as it gives a version of Gödel's primitive recursive functionals of finite type, and to categorists, as it is related to the so-called 'coherence problem' for $\mathbf{Cart}_N$. This problem asks when diagrams in a category commute or, equivalently, when two arrows between two given objects are equal. Indeed if one wants to compute $\mathrm{Hom}(A, B)$ in $\mathscr{C}_0$, two problems arise:

(I) Find an algorithm for obtaining all arrows $A \to B$ in $\mathscr{C}_0$ (that is, all proofs $A \to B$ in the corresponding deductive system).

(II) Find an algorithm for deciding when two arrows $A \to B$ are equal (better: when two proofs describe the same arrow).

We shall here address ourselves to the second problem. Looking at the proof of the distributive law

$$\langle f, g\rangle h = \langle fh, gh\rangle$$

for cartesian closed categories given in Section 3, we note that both sides must be expanded to be shown equal. It seems easier to consider $\mathscr{C}_0$ as given by $\mathbf{C}(\mathscr{L}_0)$ rather than as constructed by the method of Section 4.

Two arrows $f, g: A \to B$ in $\mathscr{C}_0$ are thus given by two terms $\varphi(x)$ and $\psi(x)$ of type B in $\mathscr{L}_0$ with a free variable x of type A. We want to decide whether $\varphi(x) \underset{x}{=} \psi(x)$ holds, or equivalently, $\lambda_{x\in A}\phi(x) = \lambda_{x\in A}\psi(x)$ holds in $\mathscr{L}_0$. Let us call two terms a and b of $\mathscr{L}_0$ whose free variables are contained in X *convertible* if the equation $a \underset{X}{=} b$ holds in $\mathscr{L}_0$. Terms of $\mathscr{L}_0$ are, of course, defined inductively, as the reader will recall. Thus Problem (II) has been reduced to deciding when two closed terms of type B in $\mathscr{L}_0(x)$ or $\mathscr{L}_0$ are convertible. In view of the fact that there are closed terms of each type in $\mathscr{L}_0$, we need not distinguish between $\underset{X}{=}$ and $=$, as was pointed out in Section 10.

Actually, we shall solve the decision problem for convertibility not in $\mathscr{L}_0$ but in $\mathscr{L}'_0$, which is like $\mathscr{L}_0$ but without type 1. In other words, $\mathscr{L}'_0$ is a variant of pure typed λ-calculus in which the only basic type is N. This may be done without loss in generality for the following reason: a closed term of type B in $\mathscr{L}_0$ or $\mathscr{L}_0(x)$ corresponds to an arrow $1 \to B$ in $\mathscr{C}_0$ or $\mathscr{C}_0[x]$, where the object B is canonically isomorphic to either 1 (which case may be dismissed as uninteresting) or an object whose inductive construction does not contain 1 at all. This is so in view of the canonical isomorphisms $C \times 1 \cong C \cong 1 \times C$, $C^1 \cong C$ and $1^C \cong 1$. The last mentioned isomorphism presupposes that $\mathrm{Hom}(1, C)$ is not empty, which is the case in $\mathscr{C}_0$, as there are closed terms of each type in $\mathscr{L}_0$.

To solve the decision problem for convertibility of terms in $\mathscr{L}'_0$, we shall replace convertibility by a finer relation, that of reducibility. However, it becomes tedious to distinguish between terms which differ only in the choice of bound variables. We shall call two such terms a and a' *congruent* and write $a \equiv a'$.

First we shall define a relation $a > a'$ between terms of type A in $\mathscr{L}_0$ or $\mathscr{L}'_0$ (actually, congruence classes) and say 'a basically reduces to a''. There are eight basic reductions; in each of the basic equations of typed λ-calculus the left hand side *basically reduces* to the right hand side:

B1.	$a > *$	$(a \in 1, a \not\equiv *)$; (not used in $\mathscr{L}'_0$)
B2.	$\pi(\langle a, b\rangle) > a$	$(a \in A, b \in B)$;
B3.	$\pi'(\langle a, b\rangle) > b$	$(a \in A, b \in B)$;
B4.	$\langle \pi(c), \pi'(c)\rangle > c$	$(c \in A \times B)$;
B5.	$\lambda_{x \in A}\varphi(x)^{\mathrm{f}} a > \varphi(a)$	$(a \in A)$;
B6.	$\lambda_{x \in A}(f^{\mathrm{f}} x) > f$	$(f \in B^A$, x not free in $f)$;
B7.	$I(a, h, 0) > a$	$(a \in A, h \in A^A)$;
B8.	$I(a, h, S(n)) > h^{\mathrm{f}} I(a, h, n)$	$(a \in A, h \in A^A, n \in N)$.

We shall say that b *reduces to* b' *in one step* and write $b \underset{1}{>} b'$ provided b' is obtained from b by replacing a single occurrence of a subterm a in b by a', where $a > a'$. For example,

$$\lambda_{x \in A}\langle \pi(\langle x, y\rangle), y\rangle \underset{1}{>} \lambda_{x \in A}\langle x, y\rangle$$

because $\pi(\langle x, y\rangle) > x$.

We shall say that b *reduces to* b' *in* n *steps* and write $b \underset{n}{>} b'$ provided

$$b \equiv b_0 \underset{1}{>} b_1 \underset{1}{>} \cdots \underset{1}{>} b_n \equiv b'.$$

In particular, $b \underset{0}{>} b'$ means that $b \equiv b'$. We shall also say b *reduces to* b' and write $b \geqslant b'$ provided there is a natural number n such that $b \underset{n}{>} b'$.

The convertibility relation between terms is of course the smallest

equivalence relation containing $\geqslant$, that is, the equivalence relation generated by $\geqslant$. This takes a particularly simple form in view of the following:

Proposition 12.1. (Church–Rosser Theorem). In $\mathscr{L}'_0$, if $b \geqslant c$ and $b \geqslant d$ then there exists a term e such that $c \geqslant e$ and $d \geqslant e$.

We postpone the proof of this until later and only note its consequence:

Corollary 12.2. If b and b' are terms of type B, then $b = b'$ holds in $\mathscr{L}_0$ if and only if there is a term $d \in B$ such that $b \geqslant d$ and $b' \geqslant d$.

Proof. It suffices to check that the relation between b and b' which holds whenever there exists d such that $b \geqslant d$ and $b' \geqslant d$ is an equivalence relation.

We shall call a term b *irreducible*, or *in normal form*, if there does not exist a term b' such that $b \underset{1}{>} b'$, that is, if for no subterm a of b there exists a' with $a > a'$. Another way of saying this is that $b \geqslant b'$ implies $b \equiv b'$ for all terms b'.

Remark 12.3. In $\mathscr{L}_0$ there are irreducible closed terms $\kappa(A)$ of each type A, defined inductively as follows:

$$\kappa(1) \equiv *, \kappa(N) \equiv 0, \kappa(A \times B) \equiv \langle \kappa(A), \kappa(B)\rangle, \kappa(B^A) \equiv \lambda_{x \in A}\kappa(B).$$

We shall leave the easy verification of this to the reader and pass on to some further obvious consequences of the Church–Rosser Theorem.

Clearly, a sufficient condition for $b = b'$ to hold in $\mathscr{L}_0$ or $\mathscr{L}'_0$ is that b and b' reduce to congruent irreducible terms (or have congruent normal forms). Call b *normalizable* if there exists an irreducible b^* such that $b \geqslant b^*$.

Corollary 12.4. In $\mathscr{L}'_0$, if b is normalizable, then its normal form is unique up to congruence. Two normalizable terms are convertible if and only if they have congruent normal forms.

One might think that this gives a decision procedure for convertibility of normalizable terms: reduce each to normal form and see whether these irreducible terms are congruent. Unfortunately, there is still a problem of how to reduce a given term to normal form. While one sequence of one-step reductions may end up with an irreducible term after a finite number of steps, it is conceivable that another sequence of one-step reductions will never terminate, and we may have no way of telling beforehand whether we are on the right track.

We shall call a term *bounded* (some authors say 'strongly normalizable') if there is a number n so that no sequence of one-step reductions has more than n steps. The *bound* of t, written $\mathrm{bd}(t)$, is the smallest such n. For example, the bound of an irreducible term is 0. Clearly, if a term is bounded, every sequence of one-step reductions will terminate after a finite number of steps. (The converse of this statement is also true, in view of König's Lemma; but we shall not need this.) In particular, every bounded term is normalizable. Note that if $t \underset{1}{>} t'$ then $\mathrm{bd}(t) > \mathrm{bd}(t')$.

We thus have an algorithm for deciding convertibility of two bounded terms in $\mathscr{L}'_0$: just reduce both of them at random until irreducible terms are reached and then compare these to see whether they are congruent.

We shall prove in Section 13 that the Church–Rosser Theorem holds for bounded terms and in Section 14 that all terms are bounded. We shall thus obtain an algorithm for deciding convertibility of terms in $\mathscr{L}'_0$ and therefore for deciding equality of arrows in $\mathscr{C}_0 = \mathrm{C}(\mathscr{L}_0)$.

For the moment, let us just make an observation that will be useful later.

Lemma 12.5. Suppose $\varphi(x)$ is a term in $\mathscr{L}_0$ with no free variables other than x of type A and a is a closed term of type A such that $\varphi(a)$ is bounded. Then $\varphi(x)$ is bounded.

Proof. If $\varphi(x) \underset{1}{>} \psi(x)$ by virtue of B2 to B8, then surely also $\varphi(a) \underset{1}{>} \psi(a)$. However, when the basic reduction $x \underset{1}{>} *$ is used, then $\varphi(*) \equiv \psi(*)$ are the same terms. This unfortunate exception complicates the proof somewhat. Still, $\varphi(x) \geqslant \psi(x)$ implies $\varphi(a) \geqslant \psi(a)$. Consider the set Γ of all terms $\psi(x)$ such that $\varphi(x) \geqslant \psi(x)$. For any $\psi(x)$ in Γ it thus follows that $\varphi(a) \geqslant \psi(a)$. Since $\varphi(a)$ is bounded, the set Δ of all terms b such that $\varphi(a) \geqslant b$ is finite. (Remember that we do not distinguish between congruent terms, that is, terms that differ only in the choice of bound variables.) Moreover, for each b in Δ, the set Γ_b of all $\psi(x)$ such that $\psi(a) \equiv b$ is finite. Hence $\Gamma \subseteq \bigcup_{b \in \Delta} \Gamma_b$ is also finite, and therefore $\psi(x)$ is bounded.

Exercises

1. Show that all irreducible closed terms of $\mathscr{L}_0$ have the form $*$, $S^k(0)$, $\langle a, b\rangle$ (where a and b are necessarily irreducible) or $\lambda_{x \in A}\varphi(x)$ (where $\varphi(x)$ is necessarily irreducible). Thus closed terms of the form $\pi(c), \pi'(c), f^{\varsigma} a$ or $I(a, h, n)$ are never irreducible.
2. Show that $\langle a, b\rangle$ is bounded if and only if a and b are bounded.

13 The Church–Rosser theorem for bounded terms

In this section we shall prove the Church–Rosser Theorem (Proposition 12.1) for the special case when b is a bounded term of $\mathscr{L}'_0$ or,

more generally, of $\mathscr{L}_0$ but without any subterm of type 1 other than $*$. As we shall prove in Section 14 that every term is bounded, this will establish Proposition 12.1.

Proposition 13.1. If b is bounded and $b \underset{m}{>} c$ and $b \underset{n}{>} d$, then there is a term e such that $c \geqslant e$ and $d \geqslant e$.

Proof. We argue by induction on the bound of b and reduce the problem to the case $m \leqslant 1, n \leqslant 1$. The case $m = 1$, $n = 1$, is handled by Lemma 13.2 below. If $m = 0$, or $n = 0$, there is nothing to prove.

So suppose $m > 1$ or $n > 1$. We then have

$$b \underset{1}{>} c_1 >_{m-1} c, \quad b \underset{1}{>} d_1 >_{n-1} d,$$

where $m - 1 > 0$ or $n - 1 > 0$. By Lemma 13.2 below we find e_1 so that $c_1 \geqslant e_1$ and $d_1 \geqslant e_1$. Now c_1 and d_1 have smaller bound than b; so, by inductional assumption, we can find c_2 and d_2 so that $c \geqslant c_2, e_1 \geqslant c_2$, $e_1 \geqslant d_2$ and $d \geqslant d_2$. Again, e_1 has smaller bound than b; so we can find e such that $c_2 \geqslant e$ and $d_2 \geqslant e$. By transitivity, $c \geqslant e$ and $d \geqslant e$.

This proof is illustrated by the following diagram:

$$\begin{tikzcd}
 & & b \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & & \\
 & c_1 \arrow[dl, dash, "m-1"'] \arrow[dr, dash] & & d_1 \arrow[dl, dash] \arrow[dr, dash, "n-1"] & \\
c \arrow[dr, dash] & & e_1 \arrow[dl, dash] \arrow[dr, dash] & & d \arrow[dl, dash] \\
 & c_2 \arrow[dr, dash] & & d_2 \arrow[dl, dash] & \\
 & & e & &
\end{tikzcd}$$

It remains to prove the lemma.

Lemma 13.2. If $b \underset{1}{>} c$ and $b \underset{1}{>} d$ then there is a term e such that $c \geqslant e$ and $d \geqslant e$.

Proof. The reduction of b to c depends on the basic reduction of a subterm a of b to a', and the reduction of b to d depends on the basic reduction of a subterm f of b to f'. If a and f do not overlap, we have

$$b \equiv \ldots a \ldots f \ldots,$$
$$c \equiv \ldots a' \ldots f \ldots,$$
$$d \equiv \ldots a \ldots f' \ldots.$$

If we now take

$$e \equiv \ldots a' \ldots f' \ldots,$$

then clearly $c \geqslant e$ and $d \geqslant e$.

If the subterms a and f of b do overlap, one of them must be a subterm of the other, say f is a subterm of a. Without loss in generality, we may assume that $a \equiv b$. So b reduces in two ways: an 'outer' reduction on the whole term b and an 'inner' reduction on the subterm f. Thus we have achieved a reduction of the problem to the following special case:

If $b \underset{1}{>} c$ (*outer* reduction) and $b \underset{1}{>} d$ (*inner* reduction) then there is a term e such that $c \geqslant e$ and $d \geqslant e$. (Recall that $b \underset{1}{>} c$ means that there is a basic reduction of b to c.)

There are now eight cases for $b \underset{1}{>} c$ according to the eight basic reductions in Section 12. The following diagrams illustrate what we do in these eight cases. We always put the outer reduction on the left and the inner reduction on the right.

B1.

$$\begin{tikzcd}
 & b \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
* \arrow[dr, dash] & & d \equiv * \arrow[dl, dash] \\
 & * &
\end{tikzcd}$$

B2.

$$\begin{tikzcd}
 & \pi(\langle a, b\rangle) \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
a \arrow[dr, dash] & & \pi(\langle a', b'\rangle) \arrow[dl, dash] \\
 & a' &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \pi(\langle \pi(c), \pi'(c)\rangle) \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
\pi(c) \arrow[dr, dash] & & \pi(c) \arrow[dl, dash] \\
 & \pi(c) &
\end{tikzcd}$$

In the first subcase of B2, the inner reduction takes a to a' (whence $b' \equiv b$) or b to b' (whence $a' \equiv a$). In the second subcase of B2, the inner reduction takes $\langle a, b\rangle$ to c, provided $a \equiv \pi(c)$ and $b \equiv \pi'(c)$.

B3. Similar

B4.

$$\begin{tikzcd}
 & \langle \pi(c), \pi'(c)\rangle \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
c \arrow[dr, dash] & & \langle \pi(c'), \pi'(c')\rangle \arrow[dl, dash] \\
 & c' &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \langle \pi(\langle a, b\rangle), \pi'(\langle a, b\rangle)\rangle \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
\langle a, b\rangle \arrow[dr, dash] & & \langle a, \pi'(\langle a, b\rangle)\rangle \arrow[dl, dash] \\
 & \langle a, b\rangle &
\end{tikzcd}$$

In the first subcase of B4, the inner reduction takes c to c'. In the second subcase of B4, the inner reduction takes $\pi(c)$ to a, provided $c \equiv \langle a, b\rangle$. There is another subcase of B4, not shown, in which the inner reduction takes $\pi'(c)$ to b.

B5.

$$\begin{tikzcd}
 & \lambda_{x\in A}\, \varphi(x)\text{‘}a \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
\varphi(a) \arrow[dr, dash] & & \lambda_{x\in A}\, \varphi'(x)\text{‘}a' \arrow[dl, dash] \\
 & \varphi'(a') &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \lambda_{x\in A}\, (f\text{‘}x)\text{‘}a \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
f\text{‘}a \arrow[dr, dash] & & f\text{‘}a \arrow[dl, dash] \\
 & f\text{‘}a &
\end{tikzcd}$$

In the first subcase of B5, the inner reduction takes $\varphi(x)$ to $\varphi'(x)$ (whence $a' \equiv a$) or a to a' (whence $\varphi'(x) \equiv \varphi(x)$). Note that if $\varphi(x) \geqslant \varphi'(x)$ then $\varphi(a) \geqslant \varphi'(a)$. In the second subcase of B5, the inner reduction takes $\lambda_{x\in A}\varphi(x)$ to f, provided $\varphi(x) \equiv f\text{‘}x$ and x is not free in f.

B6.

$$\begin{tikzcd}
 & \lambda_{x\in A}\, (f\text{‘}x) \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
f \arrow[dr, dash] & & \lambda_{x\in A}\, (f'\text{‘}x) \arrow[dl, dash] \\
 & f' &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \lambda_{x\in A}\, (\lambda_{y\in A}\, \varphi(y)\text{‘}x) \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
\lambda_{x\in A}\, \varphi(y) \arrow[dr, dash] & & \lambda_{x\in A}\, \varphi(x) \arrow[dl, dash] \\
 & \lambda_{x\in A}\, \varphi(x) &
\end{tikzcd}$$

In the first subcase of B6, the inner reduction takes f to f'. In the second subcase of B6, the inner reduction takes $f^{\varsigma}x$ to $\varphi(x)$, provided $f \equiv \lambda_{y\in A}\varphi(y)$.

B7.

$$\begin{tikzcd}
& I(a, h, 0) \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
a \arrow[dr, dash] & & I(a', h', 0) \arrow[dl, dash] \\
& a' &
\end{tikzcd}$$

Here the inner reduction takes a to a' (whence $h' \equiv h$) or h to h' (whence $a' \equiv a$).

B8.

$$\begin{tikzcd}
& I(a, h, S(n)) \arrow[dl, dash, "1"'] \arrow[dr, dash, "1"] & \\
h^{\varsigma}I(a, h, n) \arrow[dr, dash] & & I(a', h', S(n')) \arrow[dl, dash] \\
& h'^{\varsigma}I(a', h', n') &
\end{tikzcd}$$

Here the inner reduction takes a to a' (whence $h' \equiv h$ and $n' \equiv n$) or h to h' (whence $a' \equiv a$ and $n' \equiv n$) or n to n' (whence $a' \equiv a$ and $h' \equiv h$).

The proof of Lemma 13.2 is now complete, hence so is the proof of Proposition 13.1.

Exercise

(Obtulowicz). Show that the argument in case B4 breaks down if $\pi(c)$ or $\pi'(c)$ is of type 1 and that the argument in case B6 breaks down if $f^{\varsigma}x$ has type 1. In particular, show that the Church–Rosser Theorem fails for $\mathscr{L}_0$ by taking c or f to be a variable.

14 All terms are bounded

One wants to prove that all terms of $\mathscr{L}_0$ are bounded. Clearly all irreducible terms are bounded, in particular variables, 0 and $*$. We may list the following partial results.

Remark 14.1.

(1) $\langle a,b\rangle$ is bounded if a and b are.
(2) $\pi(c)$ and $\pi'(c)$ are bounded if c is.
(3) $\lambda_{x\in A}\varphi(x)$ is bounded if $\varphi(x)$ is.
(4) $S(n)$ is bounded if n is.

Proof. We shall prove (1), for example, the others being similar. We argue by induction on $\mathrm{bd}(a) + \mathrm{bd}(b)$. Clearly, it suffices to show that c is bounded whenever $\langle a,b\rangle \underset{1}{>} c$. If $c \equiv \langle a',b\rangle$ with $a \underset{1}{>} a'$ or $c \equiv \langle a,b'\rangle$ with $b \underset{1}{>} b'$, c is bounded by inductional assumption. The only other case is $a \equiv \pi(c)$, $b \equiv \pi'(c)$, but then c is bounded because it is a subterm of a.

Unfortunately, this kind of argument does not extend to terms of the form $b^{\varsigma}a$ and $I(a,h,n)$. Note that in the basic reductions

$$\lambda_{x\in A}\varphi(x)^{\varsigma}a > \varphi(a), \quad I(a,h,S(n)) > h^{\varsigma}I(a,h,n)$$

the right hand side may be more complicated than the left hand side.

We shall follow Tait and replace boundedness by an apparently stronger notion, that of 'computability', which is defined by induction on types. We first confine attention to closed terms.

Definition 14.2. A closed term c is *computable* provided one of the following cases holds:

(i) $c\in 1$ or N and c is bounded;
(ii) $c\in A \times B$ and $\pi(c)$ and $\pi'(c)$ are computable;
(iii) $c\in B^A$, c is bounded and $c^{\varsigma}a$ is computable for all computable closed $a\in A$.

Here are two immediate consequences of the definition.

Lemma 14.3. Assume c and c' closed.

(1) If c is computable, then c is bounded.
(2) If c is computable and $c \geqslant c'$, then c' is computable.

Proof. We shall prove (1) and leave (2) as an exercise, as it is never used. The proof of (1) goes by induction on types. We need only look at the case $c\in A \times B$. Since c is computable, so is $\pi(c)\in A$, by Definition 14.2. By inductional assumption, the result holds for A, hence $\pi(c)$ is bounded. Therefore c, being a subterm of $\pi(c)$, is also bounded.

Lemma 14.4. (I) A closed term c is computable if one of the following three cases holds:

(1) $c \equiv \langle a,b\rangle$ and a and b are computable;

(2) $c \equiv \lambda_{x \in A}\varphi(x)$ and $\varphi(a)$ is computable for all computable closed $a \in A$,

(3) c is neither of the above and, for all closed c', if $c \underset{1}{>} c'$ then c' is computable.

(II) For all types C, $\kappa(C)$ is computable.

Proof. For the purpose of this proof only we shall make two definitions. Call a closed term *a pre-computable* if it satisfied (1), (2) or (3) above. Call a type C *nice* if all pre-computable $c \in C$ are computable. We may thus restate (I) as:

(I′) All types are nice.

Before proving this, let us make an observation:

(III) If C and all subtypes of C are nice then $\kappa(C)$ is computable.

'Being a subtype' is of course the transitive relation generated by: A and B are subtypes of $A \times B$ and B^A.

Indeed, (III) is easily shown by induction on the type C. By Definition 14.2, $\kappa(1) \equiv *$ and $\kappa(N) \equiv 0$ are clearly computable, because they are bounded. Moreover, $\kappa(A \times B) \equiv \langle \kappa(A), \kappa(B) \rangle$ is computable by (1) if $\kappa(A)$ and $\kappa(B)$ are, and $\kappa(B^A) \equiv \lambda_{x \in A}\kappa(B)$ is computable by (2) if $\kappa(B)$ is, so that the induction hypothesis applies.

As we are planning to prove (I′), (II) will follow immediately from (III). It remains to prove (I′), which we shall do by induction on types. To this purpose, let us adopt the following assumption:

> **(A)** All proper subtypes of C are nice and the closed term $c \in C$ is pre-computable.

We wish to establish the following conclusion:

> **(C)** c is computable.

We shall look at the cases $C = 1, N, A \times B$ and B^A separately, but first let us note this preliminary conclusion:

> **(P)** c is bounded.

Indeed, by assumption (A), c satisfies (1), (2) or (3). In case (1), c is bounded by Remark 14.1 and Lemma 14.3, because a and b are bounded. In case (2), it will follow from Remark 14.1 that c is bounded if we show that $\varphi(x)$ is bounded. Now $\kappa(A)$ is computable by the assumption (A) and (III). Therefore, $\varphi(\kappa(A))$ is bounded by (2) and Lemma 14.3, hence $\varphi(x)$ is bounded by Lemma 12.5. In case (3), c is evidently bounded, because, whenever $c \underset{1}{>} c'$, then c' is bounded by Lemma 14.3.

We are now ready to prove the conclusion (**C**). When $C = 1$ or N, computable means bounded, and so we refer to the preliminary conclusion (**P**).

Suppose $C = A \times B$. According to Definition 14.2, we must show that $\pi(c) \in A$ and $\pi'(c) \in B$ are computable, for example, the former. We shall proceed by induction on bd(c). By assumption (**A**), A is nice, so we need only show that $\pi(c)$ is pre-computable. Since $\pi(c)$ is neither a pair nor a λ-term, we only have to check (3).

So suppose $\pi(c) \underset{1}{>} a'$, we must show that a' is computable. There are two cases. If $a' \equiv \pi(c')$ and $c \underset{1}{>} c'$, a' is computable by inductional assumption, since bd$(c') <$ bd(c). If $a' \equiv a$ and $c \equiv a$ and $c \equiv \langle a, b \rangle$, a' is computable by (1).

Next, suppose $C = B^A$. According to Definition 14.2, we must show that $c^{\ulcorner}a \in B$ is computable for all computable closed $a \in A$, as we already know that c is bounded by (**P**). We shall proceed by induction on bd(c) + bd(a). By assumption B is nice, so we need only show that $c^{\ulcorner}a$ is pre-computable. Since $c^{\ulcorner}a$ is neither a pair nor a λ-term, we only have to check (3).

So suppose $c^{\ulcorner}a \underset{1}{>} b'$, we must show that b' is computable. There are two cases. If $b' \equiv c'^{\ulcorner}a$ with $c \underset{1}{>} c'$ or $b' \equiv c^{\ulcorner}a'$ with $a \underset{1}{>} a'$, b' is computable by inductional assumption, since bd$(a') <$ bd(a) and bd$(c') <$ bd(c). If $b' \equiv \varphi(a)$ and $c \equiv \lambda_{x \in A}\varphi(x)$, b' is computable by (2).

We have thus established the conclusion (**C**) and the proof of Lemma 14.4 is complete.

Lemma 14.5. If $a \in A$, $h \in A^A$ and $n \in N$ are computable closed terms, then $I(a, h, n)$ is computable.

Proof. We proceed by induction on bd(a) + bd(h) + bd(n) + $\sigma(n)$, where $\sigma(n)$ is the number of occurrences of the symbol S in the normal form of n. (Recall that n computable implies n bounded.) Since $I(a, h, n)$ is neither a pair nor a λ-term, we need only check case (3) of Lemma 14.4.

So suppose $I(a, h, n) \underset{1}{>} d$; we must show that d is computable. There are three cases. If $d \equiv I(a', h, n)$ with $a \underset{1}{>} a'$ or $d \equiv I(a, h', n)$ with $h \underset{1}{>} h'$ or $d \equiv I(a, h, n')$ with $n \underset{1}{>} n'$, d is computable by inductional assumption, since bd$(a') <$ bd(a), bd$(h') <$ bd(h) and bd$(n') <$ bd(n) but $\sigma(n') = \sigma(n)$. If $d \equiv a$ with $n \equiv 0$, d is given to be computable. Finally, if $d \equiv h^{\ulcorner}I(a, h, m)$ with $n \equiv S(m)$, we have $\sigma(m) < \sigma(n)$ and bd(m) = bd(n), so $I(a, h, m)$ is computable by inductional assumption. Since h is given to be computable, d is computable by Definition 14.2.

We now extend the notion of computability to open terms.

Definition 14.6. A term $t \equiv \varphi(x_1,\ldots,x_n)$, with no free variables other than $x_1,\ldots,x_n$, is *computable* provided, for all computable closed terms $a_1,\ldots,a_n$ of appropriate types, the closed term $\bar{t} \equiv \varphi(a_1,\ldots,a_n)$ is computable.

Theorem 14.7. All terms of $\mathscr{L}_0$ are computable, hence bounded.

Proof. We proceed by induction on the length of terms. For the constants $*$ and 0 and for all variables the result holds trivially. It remains to prove the following six statements.

(1) If a and b are computable, so is $\langle a,b\rangle$.

Indeed, let $\bar{a}$ and $\bar{b}$ be computable, then so is $\langle \bar{a},\bar{b}\rangle$, by Lemma 14.4.

(2) If c is computable, then so are $\pi(c)$ and $\pi'(c)$.

Indeed, let $\bar{c}$ be computable, then so are $\pi(\bar{c})$ and $\pi'(\bar{c})$ by Definition 14.2.

(3) If $f\in B^A$ and $a\in A$ are computable, then so is $f{}^{\varsigma}a$.

Indeed, let $\bar{f}$ and $\bar{a}$ be computable, then so is $\bar{f}{}^{\varsigma}\bar{a}$, by Definition 14.2 and Lemma 14.3.

(4) If $\varphi(x,x_1,\ldots,x_n)$ is computable, so is $\lambda_{x\in A}\varphi(x,x_1,\ldots,x_n)$.

Indeed, let $\bar{\varphi}(x) \equiv \varphi(x,a_1,\ldots,a_n)$ for computable closed $a_1,\ldots,a_n$ and assume that $\bar{\varphi}(a)\in B$ is computable for all computable closed $a\in A$. Then $\lambda_{x\in A}\bar{\varphi}(x)$ is computable, by Lemma 14.4.

(5) If $n\in N$ is computable, so is $S(n)$.

Indeed, let $\bar{n}$ be computable, that is, bounded. Then so is $S(\bar{n})$.

(6) If $a\in A$, $h\in A^A$ and $n\in N$ are computable, then so is $I(a,h,n)$.

Indeed, let $\bar{a},\bar{h}$ and $\bar{n}$ be computable. Then so is $I(\bar{a},\bar{h},\bar{n})$, by Lemma 14.5.

The first person to prove Theorem 14.7 in essentially the generality given here was R.C. de Vrijer. Our proof is closer to Tait's original proof, but depends crucially on an idea of de Vrijer, which is here embedded in condition (3) of Lemma 14.4 and the use that is made of it.

Exercises

1. Prove (2) to (4) of Remark 14.1.
2. Prove (2) of Lemma 14.3.

15 C-monoids

A small category with one object is a *monoid*, that is, a semigroup with a unity element. (See Part 0, Section 1, Example C2′.) If a small cartesian closed category has only one object, it is a rather uninteresting monoid. For, if 1 is the terminal object, Hom(1, 1) has only one element. However, if we delete the terminal object, we obtain an interesting class of monoids.

A *C-monoid* (C for Curry, Church, combinatory or cartesian) is a monoid $\mathscr{M}$ with extra structure $(\pi,\pi',\varepsilon,*,\langle\,\rangle)$, where π,π', and ε are elements of $\mathscr{M}$ (nullary operations), $(-)^*$ is a unary operation and $\langle -,-\rangle$ is a binary operation satisfying the following identities:

C1. $\pi\langle a,b\rangle = a,$

C2. $\pi'\langle a,b\rangle = b,$

C3. $\langle \pi c,\pi' c\rangle = c,$

C4. $\varepsilon\langle h^*\pi,\pi'\rangle = h,$

C5. $(\varepsilon\langle k\pi,\pi'\rangle)^* = k,$

for all a,b,c,h and k. These are the axioms of a cartesian closed category without terminal object, with the type subscripts erased.

We list some easy consequences of the above definition:

C3a. $\langle a,b\rangle c = \langle ac,bc\rangle,$

C3b. $\langle \pi,\pi'\rangle = 1,$

C4a. $\varepsilon\langle h^*a,b\rangle = h\langle a,b\rangle,$

C5a. $h^*k = (h\langle k\pi,\pi'\rangle)^*,$

C5b. $\varepsilon^* = 1,$

for all a,b,c,h and k.

Proof.

$$\begin{aligned}
\langle a,b\rangle c &= \langle \pi\langle a,b\rangle c,\pi'\langle a,b\rangle c\rangle && \text{by C3,}\\
&= \langle ac,bc\rangle && \text{by C3.}\\
\langle \pi,\pi'\rangle &= \langle \pi 1,\pi' 1\rangle = 1 && \text{by C3.}\\
\varepsilon\langle h^*a,b\rangle &= \varepsilon\langle h^*\pi\langle a,b\rangle,\pi'\langle a,b\rangle\rangle && \text{by C1 and C2,}\\
&= \varepsilon\langle h^*\pi,\pi'\rangle\langle a,b\rangle && \text{by C3a,}\\
&= h\langle a,b\rangle && \text{by C4.}\\
h^*k &= (\varepsilon\langle h^*k\pi,\pi'\rangle)^* && \text{by C5,}\\
&= (\varepsilon\langle h^*\pi\langle k\pi,\pi'\rangle,\pi'\langle k\pi,\pi'\rangle\rangle)^* && \text{by C1 and C2,}
\end{aligned}$$