Effective operations on partial functions

Commented transcriptions

Original authors: Nigel Cutland, Paul Young

Commented by Armando Matos

This note was written for personal use. It consists on commented transcriptions of: (i) part of Chapter 10 of [2], (ii) [4]. Previously, the reference [1] was also commented in this note.

Cutland, [2], Chapter 10

Once we have studied effectively computable operations on *numbers* it is natural to ask whether there is a comparable notion for operations on *functions*. The essential difference between functions and numbers as basic objects is that functions are usually infinite rather than finite. With this in mind, in §1 of this chapter we discuss the features we might reasonably expect of an *effective* operator on partial functions: this leads to the formulation of the definition of *recursive operators* on partial functions.

In §2 we shall see that there is a close connection between recursive operators and those effective operations on computable functions that we discussed in Chapter 5, §3. In §3 we prove the important fixed point theorem for recursive operators known as the first Recursion theorem. The final part of this chapter provides a discussion of some of the applications of this theorem in computability and the theory of programming.

1. Recursive operators

Let us denote by $\mathcal{F}_n$, $(n \geq 1)$ the class of all partial functions from $\mathbb{N}^{\mathbb{N}}$ to $\mathbb{N}$. We use the word *operator* to describe a function $\Phi : \mathcal{F}_m \to \mathcal{F}_n$; the letters Φ, Ψ,. . . will invariably denote operators in this chapter. We shall confine our attention to totally defined operators $\Phi : \mathcal{F}_m \to \mathcal{F}_n$; i.e. such that the domain of Φ is the whole[1] of $\mathcal{F}_m$.

[1] At this stage we are dealing with *mathematical* partial functions. When referring to recursive

The chief problem when trying to formulate the idea of a computable (or effective) operator $\Phi : \mathcal{F}_1 \to \mathcal{F}_1$, say, is that both an "input" function f and the "output" function $\Phi(f)$ are likely to be infinite objects, and hence incapable of being given in a finite time. Yet our intuition about effective processes is that in some sense they should [sometimes] be completed within a finite time.

To see how this problem can be overcome, consider the following operators from $\mathcal{F}_1$ to $\mathcal{F}_1$:

(a) $\Phi_1(f) = 2f$.

(b) $\Phi_2(f) = g$, where $g(x) = \sum_{y \le x} f(y)$.

These operators are certainly down to earth and explicit. Intuitively we might regard them as effective operators: but why? Let $f \in \mathcal{F}_1$ and let $g_1 = \Phi_1(f)$; notice that any particular value $g_1(x)$ (if defined) can be calculated in finite time from the single value $f(x)$ of f; if we set $g_2 = \Phi_2(f)$, then to calculate $g_2(x)$ (if defined) we need to know the finite number of values $f(0)$, $f(1)$, $f(x)$. Thus in both cases any defined value of the output function ($\Phi_1(f)$ or $\Phi_2(f)$) can be effectively calculated in a finite time using only a finite part of the input function f. This is essentially the definition of a recursive operator given below[2].

One consequence of the definition will be the following: suppose that $\Phi(f)(x) = y$ is calculated using only a finite part θ of f; then if g is any other function having θ as a finite part we must expect that $\Phi(g)(x) = y$ also[3].

To frame our definition precisely there are some technical considerations. First, let us agree that by a "finite part" of a function f we mean a finite function θ extended by f. (We say that θ is a finite function if its domain is a finite set.) For convenience we adopt the convention

θ always denotes a finite function in this chapter.

The above discussion shows that the definition of recursive operator will involve effective calculations with finite functions. We make this precise by coding each

partial functions, we can see an operator Φ as a function transforming indices into indices. In the book, the partial recursive functions are denoted by $\mathcal{R}$.

[2]Here, the author is talking about recursive (partial recursive) functions.

[3]Notice that the finite part of f which is used, denoted by θ below, depends in general on the value of x. Thing of the the functional $\Phi(f)(x) = f^{(f(1))}(f(0)) = f(\ldots f(f(0)))$, where the last expression has $f(1)+1$ symbols f.

finite function θ by a number $\tilde{\theta}$ and using ordinary computability. A suitable coding for our purposes is defined as follows: suppose that $\theta \in \mathcal{F}_n$. The n-tuple $\overline{x} = (x_1, x_2, \ldots, x_n)$ is coded by the number $\langle \overline{x} \rangle = p_1^{x_1+1} p_2^{x_2+1} \ldots p_n^{x_n+1}$; then define the code $\tilde{\theta}$ for θ by[4]

$$\begin{cases} \tilde{\theta} = \prod_{\overline{x} \in \mathrm{Dom}(\theta)} p_{\langle \overline{x} \rangle}^{\theta(\overline{x})+1} & \text{provided that } \mathrm{Dom}(\theta) \neq \emptyset, \\ \tilde{\theta} = 0 & \text{if } \mathrm{Dom}(\theta) = \emptyset \\ & \text{(in which case } \theta = f_\emptyset) \end{cases}$$

[From now on we drop the bar over the tuple variables, writing x instead of $\overline{x}$, etc.]
There is a simple effective procedure to decide for any number z whether $z = \tilde{\theta}$ for some finite function θ; and if so, to decide whether a given x belongs to $\mathrm{Dom}(\theta)$, and calculate $\theta(x)$ if it does.

Now we have our definition:

1.1. Definition
Let $\Phi : \mathcal{F}_m \rightarrow \mathcal{F}_n$. Then Φ is a *recursive operator* if there is a computable function $\phi(z, x)$ such that for all $f \in \mathcal{F}_m$ and $x \in \mathbb{N}^n$, $y \in \mathbb{N}$

$$[\Phi(f)(x) = y] \text{ iff } [\exists \text{ finite } \theta \subseteq f : \phi(\tilde{\theta}, x) \simeq y].$$

(Note that ϕ is not required to be total.)[5] □

1.2. Example
The operator $\Phi(f) = 2f$ is a recursive operator: to see this define $\phi(z, x)$ by

$$\phi(z, x) = \begin{cases} 2\theta(x) & \text{if } z = \tilde{\theta} \text{ and } x \in \mathrm{Dom}(\theta), \\ \text{undefined} & \text{otherwise} \end{cases}$$

By Church's thesis, ϕ is computable: now for any f, x, y we have

$$\begin{aligned} \Phi(f)(x) \simeq y \quad &\Leftrightarrow \quad x \in \mathrm{Dom}(f) \text{ and } y = 2f(x) \\ &\Leftrightarrow \quad \text{there is } \theta \subseteq f \text{ with } x \in \mathrm{Dom}(f) \text{ and } y = 2\theta(x) \\ &\Leftrightarrow \quad \text{there is } \theta \subseteq f \text{ such that } \phi(\tilde{\theta}, x) \simeq y \end{aligned}$$

[4] In the second alternative why do not define $\tilde{\theta} = 1$? This would avoid the need for an alternative.
[5] I would prefer: the function ϕ^f with an oracle for f, given x, computes $[\Phi(f)](x)$, if any. Of course, in any finite computation, only a finite part of f is used.

Hence Φ is a recursive operator.

Further examples will be given in 1.6 below.

An important feature of recursive operators is that they are *continuous* and *monotone* in the following sense.

Definition 1.3

Let $\Phi : \mathcal{F}_m \to \mathcal{F}_n$ be an operator.

(a) Φ is *continuous* if for any $f \in \mathcal{F}_m$ and all x, y:
$\Phi(f)(x) \simeq y$ iff there is finite $\theta \subseteq f$ with $\Phi(\theta)(x) \simeq y$;

(b) Φ is *monotone* if whenever $f, g \in \mathcal{F}_m$ with $f \subseteq g$, then
$\Phi(f) \subseteq \Phi(g)$. □

[Notes.

(1) In general, the "finite θ" above depends on x and y.

(2) Notice that (a) implies that, if $\Phi(f)(x) \simeq y$, then a finite part of f is sufficient to "compute" y. In the effective world, we would add: f and ϕ are partial recursive functions; θ can be effectively computed from x (and y?).]

These properties are easily established for recursive operators, and as we shall see they aid the recognition of such operators.

Theorem 1.4

A recursive operator is continuous and monotone.

Proof. Let $\Phi : \mathcal{F}_m \to \mathcal{F}_n$ be a recursive operator, with computable function ϕ as required by the definition. Suppose that $\Phi(f)(x) \simeq y$, and let $\theta \subseteq f$ be such that $\phi(\tilde{\theta}, x) \simeq y$. Since $\theta \subseteq \theta$, it follows immediately that $\Phi(\theta)(x) \simeq y$. Conversely, if $\theta \subseteq f$ and $\Phi(\theta)(x) \simeq y$, there is $\theta_1 \subseteq \theta$ such that $\phi(\tilde{\theta}_1, x) \simeq y$; but then $\theta_1 \subseteq f$, so we have that $\Phi(f)(x) \simeq y$. Hence Φ is continuous.

Monotonicity follows directly from continuity: suppose that $f \subseteq g$ and $\Phi(f)(x) \simeq y$. Take $\theta \subseteq f$, such that $\Phi(\theta)(x) \simeq y$; then $\theta \subseteq g$, so by continuity, $\Phi(g)(x) \simeq y$. □

The use of the term *continuous* to describe the property 1.3(a) is justified informally as follows. Suppose that $\Phi : \mathcal{F}_1 \to \mathcal{F}_1$ satisfies 1.3(a) and $f \in \mathcal{F}_1$. Then given any $x_1, \ldots, x_k$ for which $\Phi(f)(x_i)$ $(1 \leq i \leq k)$ are defined, using 1.3(a) we can obtain a finite[6] $\theta \subseteq f$ such that $\Phi(\theta)(x_i) = \Phi(f)(x_i)$ $(1 \leq i \leq k)$. Thus,

[6]Each of the x_i for $1 \leq i \leq k$ defines a finite function $\theta_i \subseteq f$. Of course, the same finite function

whenever $g \supseteq \theta$, by 1.3(a) again, we have $\Phi(g)(x_i) = \Phi(f)(x_i)$ $(1 \leq i \leq k)$ i.e. if g is "near" to f (in the sense that they agree on the finite set $\mathrm{Dom}(\theta)$ then $\Phi(g)$ "near" to $\Phi(f)$ (in the sense that they agree on the finite set $x_1, \ldots, x_k$). Thus, informally, Φ is continuous.

The continuity property 1.3(a) specifies that a value $\Phi(f)(x)$ is determined (if at all) by a finite amount of positive information about f. This means information asserting that f is defined at certain points and takes certain values there, as opposed to negative information that would indicate points where f is not defined. Using this idea the term continuous can be rigorously justified as follows.

The *positive information topology*[7] on $\mathcal{F}_m$ is defined by taking as base of open neighbourhoods sets of the form

$$U_\theta = \{f : \theta \subseteq f\} \quad \theta \in \mathcal{F}_m, \text{ finite.}$$

Thus f belongs to U_θ iff θ is correct positive information about f. It is then an easy exercise to see that an operator is continuous with respect to the positive information topology precisely when it possesses property 1.3(a).

The following characterisation of recursive operators using continuity will make it easy to establish recursiveness of various operators.

Theorem 1.5.
Let $\Phi : \mathcal{F}_m \to \mathcal{F}_n$ be an operator. Then Φ is a recursive operator iff
(a) Φ is continuous,
(b) the function $\phi(z, x)$ given by

$$\begin{cases} \phi(\tilde{\theta}, x) \simeq \Phi(\theta)(x) & \text{for } \theta \in \mathcal{F}_m, \\ \phi(z, x) \text{ is undefined} & \text{for all other } z, \end{cases}$$

is computable.

Proof. Suppose that Φ is recursive with computable function ϕ_1 such that

$$\Phi(f)(x) \simeq y \text{ iff } \exists \theta : \theta \subseteq f \text{ and } \phi_1(\tilde{\theta}, x) \simeq y$$

$\cup_{1 \leq i \leq k} \theta_i$ may be "used" for each of those x_i.

[7]1 The reader unfamiliar with topology will lose nothing in further development by omitting this paragraph.

Then taking Φ as given in the theorem, we have

$$\phi(\tilde{\theta}, x) \simeq y \iff \exists \theta_1 : \theta_1 \subseteq \theta \text{ and } \phi_1(\tilde{\theta}_1, x) \simeq y;$$

the relation on the right is partially decidable, so ϕ is computable by theorem 6-6.13.

Conversely, suppose that conditions (a) and (b) of the theorem hold; then

$$\begin{aligned} \Phi(f)(x) \simeq y \quad &\Leftrightarrow \quad \exists \theta : \theta \subseteq f \text{ and } \Phi(\theta, x) \simeq y \;\; \text{(by (a))} \\ &\Leftrightarrow \quad \exists \theta : \theta \subseteq f \text{ and } \phi(\tilde{\theta}, x) \simeq y \;\; \text{(by (b))} \end{aligned}$$

whence Φ is a recursive operator. □

This theorem enables us to show quite easily that the following operators are all recursive:

Examples 1.6.

(a) (The diagonalisation operator) $\Phi(f)(x) \simeq f(x, x)$ ($f \in \mathcal{F}_2$) is obviously continuous, and $\phi(\tilde{\theta}, x) \simeq \theta(x, x)$ is computable.

(b) $\Phi(f)(x) \simeq \sum_{y \leq x} f(y)$ ($f \in \mathcal{F}_1$).
This is the second example discussed at the beginning of this section. We saw there that Φ is continuous; and clearly $\phi(\tilde{\theta}, x) \simeq \sum_{y \leq x} \theta(y)$ is computable.

(c) Let $g \in \mathcal{F}_1$ be computable. Define $\Phi : \mathcal{F}_n \to \mathcal{F}_n$ by $\Phi(f) = g \circ f$. Obviously Φ is continuous, and $\phi(\tilde{\theta}, x) = g(\theta(x))$ is computable.

(d) (The Ackermann operator). Let $\Phi : \mathcal{F}_2 \to \mathcal{F}_2$ be given by

$$\begin{aligned} &\Phi(f)(0, y) = y + 1 \\ &\Phi(f)(x + 1, 0) \simeq f(x, 1) \\ &\Phi(f)(x + 1, y + 1) \simeq f(x, f(x + 1, y)) \end{aligned}$$

To see that Φ is continuous, note that $\Phi(f)(x, y)$ depends on at most two particular values of f. For recursiveness, it is immediate by Church's thesis that the function ϕ given by

$$\begin{aligned} &\phi(\tilde{\theta}, 0, y) = y + 1 \\ &\phi(\tilde{\theta}, x + 1, 0) \simeq \theta(x, 1) \\ &\phi(\tilde{\theta}, x + 1, y + 1) \simeq \theta(x, \theta(x + 1, y)) \end{aligned}$$

is computable.

(e) (The μ-operator.) Consider $\Phi : \mathcal{F}_{n+1} \to \mathcal{F}_n$, given by $\Phi(f)(x) \simeq \mu y$:

$[f(x,y)=0]$. It is immediate that this operator is continuous, and that the function ϕ given by $\phi(\tilde{\theta},x)\simeq \mu y : [f(x,y)=0]$ is computable.

When the definition 1.1 of a recursive operator $\Phi : \mathcal{F}_m \to \mathcal{F}_n$ is extended to the case $n=0$, we have what is called a recursive functional. The members of $\mathcal{F}_0$ are 0-ary functions; i.e. constants. Just as $\mathcal{F}_n$ $(n \geq 1)$ includes the function that is defined nowhere, $\mathcal{F}_0$ includes the "undefined" constant, which is denoted by ω. Thus $\mathcal{F}_0 = \mathbb{N} \cup \{\omega\}$, and an operator $\Phi : \mathcal{F}_m \to \mathcal{F}_0$ a *recursive functional* if there is a computable function $\phi(x)$ such that for any $f \in \mathcal{F}_m \to \mathcal{F}_0$, and $y \in \mathbb{N}$:

$$\Phi(f) \simeq y \text{ iff } [\exists \theta : \theta \subseteq f \text{ and } \phi(\tilde{\theta}) = y$$

We write $\Phi(f) = \omega$ if $\Phi(f)$ is undefined; this emphasises that Φ is still thought of as being a total operator.

We should point out that in some texts the term *partial recursive functional* $\mathcal{F}_m \to \mathcal{F}_n$ is used to describe recursive operators, including the case $n=0$. In such contexts the word "partial" describes the kind of object being operated on rather than the domain of definition of the operation.

We shall not discuss here the extension of the ideas of this section to partially defined operators and the corresponding partial recursive operators $\Phi : \mathcal{F}_m \to \mathcal{F}_n$. The reader is referred to [3] for a full discussion of these and related matters.

Exercises 1.7

1. Show that the following operators are recursive.
   (a) $\Phi(f) = f^2$ $(f \in \mathcal{F}_1)$.
   (b) $\Phi(f) = g$ $(f \in \mathcal{F}_n)$, where g is a fixed computable function in $\mathcal{F}_n$.
   (c) $\Phi(f) = f \circ g$ $(f \in \mathcal{F}_1)$, where g is a fixed computable function in $\mathcal{F}_n$.
   (d) Let $h \in \mathcal{F}_{n+1}$ be a fixed computable function; define $\Phi : \mathcal{F}_{n+1} \to \mathcal{F}_{n+1}$ by

   $$\Phi(f)(x,y) \simeq \begin{cases} 0 & \text{if } h(x,y)=0, \\ f(x+1,y)+1 & \text{if } h(x,y) \text{ is defined and } \neq 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

   (The significance of this operator will be seen later.)
2. Prove that if Φ is a recursive operator and f is computable then so is $\Phi(f)$.

3. Decide whether the following operators $\Phi : \mathcal{F}_1 \to \mathcal{F}_1$ are (i) monotonic, (ii) continuous, (iii) recursive.

   (a)

$$\Phi(f)(x) = \begin{cases} f(x) & \text{if Dom}(f) \text{ is finite,} \\ \text{undefined} & \text{if Dom}(f) \text{ is infinite.} \end{cases}$$

2. Effective operations on computable functions

In chapter 5 §3 we considered that certain operations on computable functions should be called effective because they can be given by total computable functions *acting on indices*. For instance, in example 5-3.1(2) we saw that there is a total computable function g such that for all $e \in \mathbb{N}$, $(\phi_e)^2 = \phi_{g(e)}$.

We shall see in this section that any recursive operator Φ, when restricted to computable functions, yields an effective operation of this kind on indices. This is the first part of a theorem of Myhill and Shepherdson. They proved, moreover, that all such operations on indices of computable functions arise in this way.

We shall prove the two parts of the Myhill-Shepherdson result separately, taking the easier part first.

Theorem 2.1 (Myhill-Shepherdson, part I)
Suppose that $\Psi : \mathcal{F}_m \to \mathcal{F}_n$ is a recursive operator. Then there is a total computable function h such that

$$\Psi(\phi_e^{(m)}) = \phi_{h(e)}^{(n)} \quad (e \in \mathbb{N})$$

Proof. Let ψ be a computable function showing that Ψ is a recursive operator according to definition 1.1. Then for any e we have

$$\Psi(\phi_e^{(m)})(x) \simeq y \;\Leftrightarrow\; [\exists\theta : \theta \subseteq \phi_e^{(m)} \text{ and } \psi(\tilde{\theta}, x) \simeq y.$$

We shall show that the function g defined by

$$g(e, x) \simeq \Psi(\phi_e^{(m)})(x)$$

is computable, by showing that the relation $g(e, x) \simeq y$ is partially decidable. To this end, consider the relation $R(z, e, x, y)$ given by

$$R(z, e, x, y) = \exists\theta : z = \tilde{\theta} \text{ and } \theta \subseteq \phi_e^{(m)} \text{ and } \psi(\Theta, x) \simeq y.$$

Then R is partially decidable, with the following informal partial decision procedure.

(1) Decide whether $z = \tilde{\theta}$ for some θ; if so obtain $x_1\, x_2, \ldots, x_k \in \mathbb{N}^m$ and $y_1, y_2, \ldots, y_k$ such that $\mathrm{Dom}(\theta) = \{x_1, \ldots, x_k\}$ and $\theta(x_i) = y_i$ $(1 \le i \le k)$; then

(2) for $i = 1, \ldots, k$ compute $\phi_e^{(m)}(x_i)$; if, for $1 \le i \le k$, $\phi_e^{(m)}(x_i)$ is defined and equals y_i, then

(3) compute $\psi(z, x)$ and if defined check whether it equals y.

If $R(z, e, x, y)$ holds, this is a mechanical procedure that will tell us so in finite time, as required.

Since $R(z, e, x, y)$ is partially decidable, so is the relation $\exists z : R(z, e, x, y)$ (by Theorem 6-6.5): but

$$\begin{aligned} \exists z : R(z, e, x, y) &\Leftrightarrow \Psi(\phi_e^{(m)})(x) \simeq y \;\; \text{(from the definition of } R) \\ &\Leftrightarrow g(e, x) \simeq y \;\; \text{(from the definition of } g) \end{aligned}$$

Thus $g(e, x) \simeq y$ is partially decidable, so by theorem 6-6.13, g is computable.

Now the *s-m-n* theorem provides a total computable function h such that

$$\begin{aligned} \phi_{h(e)}^{(n)}(x) &\simeq g(e, x) \\ &\simeq \Psi(\phi_e^{(m)})(x) \end{aligned}$$

from which we have $\phi_{h(e)}^{(n)} = \Psi(\phi_e^{(m)})$. □

Notice that the function h given by this theorem for a recursive operator $\Psi : \mathcal{F}_1 \to \mathcal{F}_1$ is *extensional* in the following sense.

Definition 2.2

A total function $h : \mathbb{N} \to \mathbb{N}$ is extensional if for all a, b, if $\phi_a = \phi_b$ then $\phi_{h(a)} = \phi_{h(b)}$. □

Now we can state the other half of Myhill and Shepherdson's result.

Theorem 2.3 (Myhill-Shepherdson, part II)

Suppose that $h$is an extensional total computable function. Then there is a unique recursive operator Ψ such that $\Psi(\phi_e) = \phi_{h(e)}$ for all e.

Proof. At the heart of our proof lies an application of the Rice-Shapiro theorem (theorem 7-2.16).

Let h be an extensional total computable function. Then h defines an operator $\Psi_0 : \mathcal{C}_1 \to \mathcal{C}_1$ by

$$\Psi_0(\phi(e)) = \phi_{h(e)}$$

Ψ_0 is well defined since h is extensional. We have to show that there is a unique recursive operator $\Psi : \mathcal{F}_1 \to \mathcal{F}_1$ that extends Ψ_0.

First note that $\Psi_0(\theta)$ is defined for all finite θ, since finite functions are computable. Thus any recursive operator Ψ extending Ψ_0, being continuous, must be defined by

$$[\Psi(f)(x) \simeq y] \;\equiv\; [\exists\theta : \theta \subseteq f \text{ and } \Psi_0(\theta)(x) \simeq y.] \tag{2.4}$$

So such a Ψ, if it exists, is unique. To prove the theorem we must now show that

(i) (2.4) *does* define an operator Ψ,

(ii) Ψ extends Ψ_0.

(iii) Ψ is recursive.

We first use the Rice-Shapiro theorem to show that Ψ_0 is continuous in the following sense: for computable functions f

$$[\Psi_0(f)(x) \simeq y] \;\Leftrightarrow\; [\exists\theta : \theta \subseteq f \text{ and } \Psi_0(\theta)(x) \simeq y] \tag{2.5}$$

To see this, fix x, y and let $\mathcal{A} = \{f \in \mathcal{C}_1 : \Psi_0(f)(x) \simeq y\}$. Then the set $A = \{e : \phi_e \in \mathcal{A}\} = \{e : \phi_{h(e)}(x) \simeq y\}$ is r.e.; so by the Rice-Shapiro theorem, if f is computable then

$$f \in \mathcal{A} \;\Leftrightarrow\; [\exists\theta : \theta \subseteq f \text{ and } \theta \in \mathcal{A}]$$

which is precisely (2.5).

Now we establish (i), (ii), (iii) above.

(i) Let f be any partial function; we must show that for any x, (2.4) defines $\Psi(f)(x)$ uniquely (if at all). Suppose then that $\theta_1, \theta_2 \subseteq f$ and $\Psi_0(\theta_1)(x) \simeq$

y_1 and $\Psi_0(\theta_2)(x) \simeq y_2$. Take a finite function $\theta \supseteq \theta_1,\ \theta_2$ (say, $\theta = f_{\mathrm{Dom}(\theta_1)\cup\mathrm{Dom}(\theta_2)}$; by (2.5)

$$y_1 \simeq \Psi_0(\theta_1)(x) \simeq \Psi_0(\theta)(x) \simeq \Psi_0(\theta_2)(x) \simeq y_2$$

Thus (2.4) defines an operator Ψ unambiguously.

(ii) This is immediate from (2.5) and the definition (2.4).

(iii) We show that Ψ satisfies the conditions of theorem 1.5. Clearly Ψ is continuous, from the definition. For the other condition we must show that the function ψ given by

$$\psi(\tilde{\theta}, x) \simeq \Psi(\theta)(x)$$
$$\psi(z, x) \text{ is undefined if } z \neq \tilde{\theta}$$

is computable. Now it is easily seen by using Church's thesis that there is a computable function c such that for any finite function θ, $c(\tilde{\theta})$ is an index for θ; i.e. $\theta = \phi_{c(\tilde{\theta})}$. Thus

$$\begin{aligned}\psi(\tilde{\theta}, x) &\simeq \Psi(\phi_{c(\tilde{\theta})})(x) \\ &\simeq \phi_{h(c(\tilde{\theta}))}(x)\end{aligned}$$

so ψ is computable, since h and c are. Hence Ψ is a recursive operator. □

Remarks

1. The proof of theorem 2.3 actually shows that for any extensional computable h there is a unique *continuous* operator $\Psi : \mathcal{F}_1 \to \mathcal{F}_1$ such that $\Psi(\phi_e) = \phi_{h(e)}$, all e, and that this operator is recursive.

2. Theorem 2.3 extends in a natural way to cover operators from $\mathcal{F}_m \to \mathcal{F}_n$. The proof is almost identical, using the natural extension of the Rice-Shapiro Theorem to subsets of $\mathcal{C}_m$; see exercise 2.6(2) below.

Exercises 2.6

1. Suppose that Φ, Ψ are recursive operators $\mathcal{F}_1 \to \mathcal{F}_1$; knowing that $\Phi \circ \Psi$ is continuous (exercise 1.7(7)) use the two parts of the Myhill-Shepherdson Theorem together with the first remark above to show that $\Phi \circ \Psi$ is recursive.

2. State and prove a general version of theorem 2.3 for operators from $\mathcal{C}_m \to \mathcal{C}n$.

3. Formulate and prove versions of the Myhill-Shepherdson Theorem (both parts) appropriate for the operators you have defined (a) in exercise 1.7(5), (b) in exercise 1.7(8).

3. The first Recursion Theorem

The first Recursion Theorem of Kleene is a fixed point theorem for recursive operators, and is often referred to as the Fixed point Theorem (of recursion theory). We shall see later that it is a very useful result.

3.1. The first Recursion Theorem (Kleene)

Suppose that $\Phi : \mathcal{F}_n \to \mathcal{F}_m$ is a recursive operator. Then there is a computable function f_Φ that is the least fixed point of Φ; i.e.

(a) $\Phi(f_\Phi) = f_\Phi$.

(b) if $\Phi(g) = g$, then $f_\Phi \subseteq g$.

Hence, if f_Φ is total, it is the only fixed point of Φ.

Proof. We use the continuity and monotonicity of Φ to construct the least fixed point f_Φ as follows. Define a sequence of functions $\{f_n\}$ $(n \in \mathbb{N})$ by

$$\begin{aligned} f_0 &= f_\emptyset \text{ (the function with empty domain)}, \\ f_{n+1} &= \Phi(f_n) \end{aligned}$$

Then $f_0 = f_\emptyset \subseteq f_1$; and if $f_n \subseteq f_{n+1}$, by monotonicity we have that $f_{n+1} = \Phi(f_n) \subseteq \Phi(f_{n+1}) = f_{n+2}$. Hence $f_n \subseteq f_{n+1}$ for all n. Now let

$$f_\Phi = \bigcup_{n \in \mathbb{N}} f_n$$

by which we mean

$$f_\Phi(x) \simeq y \text{ iff } \exists n \text{ such that } f_n(x) \simeq y.$$

We shall show that f_Φ is a fixed point for Φ.

For all n, $f_n \subseteq f_\Phi$

hence

$$f_{n+1} = \Phi(f_n) \subseteq \Phi(f_\Phi)$$

thus

$$f_\Phi \subseteq \Phi(f_\Phi)$$

Conversely, suppose that $\Phi(f_Phi)(x) \simeq y$; then there is finite $\theta \subseteq f_\Phi$ such that $\Phi(\theta)(x) \simeq y$; take n such that $\theta \subseteq f_n$; then by continuity $\Phi(f_n)(x) \simeq y$. That is, $f_{n+1}(x) \simeq y$. Hence $f_\Phi(x) \simeq y$. Thus $\Phi(f_\Phi) \subseteq f_\Phi$, and so $\Phi(f_\Phi) = f_\Phi$ as required.

To see that f_Φ is the least fixed point of Φ, suppose that $\Phi(g) = g$; then clearly $f_0 = f_\emptyset \subseteq g$, and by induction we see that $\_n \subseteq g$ for all n. Hence $f_\Phi \subseteq g$, as required. Moreover, if f_Φ is total, then $f_\Phi = g$, so f_Phi is the only fixed point of Φ.

Finally we show that f_Φ is computable. Use theorem 2.1 to obtain a total computable function h such that for all e

$$\Phi(f_e) = \phi_{h(e)}$$

Let e_0 be an index for f_0; define a computable function k by

$$\begin{aligned} k(0) &= e_0 \\ k(n+1) &= h(k(n)) \end{aligned}$$

Then $f_n = \phi_{k(n)}$ for each n; thus

$$f_\Phi(x) \simeq y \;\Leftrightarrow\; \exists n : \phi_{k(n)}(x) \simeq y$$

The relation on the right hand side is partially decidable, and hence f_ϕ is computable. □

Remark. The recursiveness of the operator Φ was used in this proof only in showing that f_Φ is computable. The first part of the proof shows that any *continuous* operator has a least fixed point.

We shall see in the following examples that a recursive operator may have many fixed points, and that the least fixed point is not necessarily a total function.

3.2. Examples

1. Let Φ be the recursive operator given by

$$\begin{aligned} \Phi(f)(0) &= 1 \\ \Phi(f)(x+1) &= f(x+2) \end{aligned}$$

   Then the least fixed point is $\begin{cases} f_\Phi(0) = 0 \\ f_\Phi(x+1) = \text{ undefined} \end{cases}$

Other fixed points of Φ take the form $\begin{cases} f_\Phi(0) = 0 \\ f_\Phi(x+1) = a \end{cases}$

2. Recall the definition of the Ackermann function 0 in example 2-5.5:

$$\begin{cases} \psi(0,y) = y+1 \\ \psi(x+1,0) \simeq \psi(x,1) \\ \psi(x+1,y+1) \simeq \psi(x,\psi(x+1,y)) \end{cases}$$

The first Recursion Theorem gives a neat proof that these equations do define a unique function ψ and that ψ is total and computable. Let Φ be the Ackermann operator given in example 1.6(d). The fixed points of Φ are the functions that satisfy the above equations. Let $\psi = f_\Phi$; then ψ is a computable function satisfying these equations, so we have only to show that ψ is total. Clearly, $\psi(0,y)$ is defined for all y; if $\psi(x,y)$ is defined for all y, then by induction on y we see that $\psi(x+1,y)$ is defined for all y. Hence $\psi(x,y)$ is defined for all x, y; i.e. ψ is total.

3. Let $h(x,y)$ be a fixed computable function and let Φ be the recursive operator given in exercises 1.7(1d). Then the least fixed point f_Φ is a computable function satisfying

$$f_\Phi(x,y) = \begin{cases} 0 & \text{if } h(x,y) = 0 \\ f_\Phi(x+1,y)+1 & \text{if } h(x,y) \text{ is defined and not } 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

But what is this rather strange looking function? We can quite easily check that

$$f_\Phi(x,y) \simeq \mu z : h(x+z,y) = 0$$

as follows. First suppose that $[\mu z : h(x+z,y) = 0] = m$; then $h(x+z,y)$ is defined and not 0 for all $z < m$, and $h(x+m,y) = 0$. Hence

$$f_\Phi(x,y) = f_\Phi(x+1,y)+1 = \ldots = f_\Phi(x+z,y)+z(z \leq m) = f_\Phi(x+m,y)+m = 0+m = m$$

Suppose on the other hand that $f_\Phi(x,y) = m$; then from the equations this must be because

$$m = f_\Phi(x,y) = f_\Phi(x+1,y)+1 = \ldots = f_\Phi(x+m,y)+m$$

and $h(x+z,y)$ is defined and not 0 for $z < m$; then $f_\Phi(x+m,y) = 0$, so $h(x+m,y) = 0$. Thus $m = \mu z : h(x+z,y) = 0$.

We can infer from this example that the function $f_\Phi(0,y) \simeq (\mu z : h(z,y) = 0)$ is computable; of course, there is no use pretending that we have a new and clever proof of the closure of $\mathcal{C}$ under the μ operator, since we have used this property of $\mathcal{C}$ implicitly in our proof of the first Recursion Theorem. (In Kleene's equation calculus approach (see chapter 3 §1), however, the first Recursion Theorem is proved without the use of the μ-operator, so closure under the μ-operator is established by this example.)

We can see from the above examples why the first Recursion Theorem is so called. The general idea of recursion is that of defining a function "in terms of itself". A simple instance of this is primitive recursion, discussed in chapter 2. We have seen more general forms of recursion in the definitions of Ackermann's function, and the function f_Φ, in example 3.2(3) above.

We were able to see quite easily in Chapter 2 that primitive recursive definitions are meaningful, but with more complex recursive definitions this is not so obvious; conceivably there are no functions satisfying the proposed definition. This is where the first Recursion theorem comes in. Very general kinds of definition by recursion are represented by an equation of the form

$$f = \Phi(f) \tag{3.3}$$

where Φ is a recursive operator. The first Recursion Theorem shows that such a definition is meaningful; there is even a *computable* function satisfying it. Since in mathematics we require that definitions define things uniquely, we can say that the recursive definition (3.3) defines the least fixed point of the operator Φ. Thus, according to the first Recursion Theorem, the class of computable functions is closed under a very general form of definition by recursion.

1 A Paper by Paul Young

This is a commented transcription of the paper

Paul R. Young,
An effective operator, continuous but not partial recursive,
Proceedings of the American Mathematical Society. 02/1968; 19(1).

Introduction. It is known that under many conditions, effective operators will be partial recursive, ([MS], [KLS], [L]). On the other hand, certain pathological examples have been constructed by Friedberg [F] and Pour-El [P] to show that effective operators are not always partial recursive. Pour-El has observed that although it is well known that all partial recursive operators are continuous, the effective but not partial recursive operators of [F] and [P] are not continuous, and she has raised the question of the existence of effective operators which are continuous but not partial recursive. It is easy to see that all partial recursive operators are not just continuous, but are in fact "effectively continuous". This enables us to answer Pour-El's question by constructing an effective operator which is continuous but not "effectively continuous". Since it is continuous, our example of an effective but not partial recursive operator is perhaps less pathological than earlier examples[8].

Notation and definitions. $\mathbb{N}$ is the set of all nonnegative integers. $\mathcal{P}$ is the set of all partial functions mapping $\mathbb{N}$ to $\mathbb{N}$, and $\mathcal{P}^r$ is the set of all partial recursive elements of $\mathcal{P}$. $\{\varphi_e\}$ is a standard effective enumeration of $\mathcal{P}^r$. In this paper we will be concerned only with operators mapping subsets of $\mathcal{P}^r$ into $\mathcal{P}^r$.

> A function is *finite* if its domain is finite.
> If f and g are partial functions mapping $\mathbb{N}^m$ to $\mathbb{N}$ we say that $f \subseteq g$ if, whenever $f(x)$ is defined and has the value y, then $g(x)$ is also defined and has the value y.
> We will topologize the functions $f : \mathbb{N}^m \to \mathbb{N}$ by defining as basic open sets the "cylinders" $\phi_h \stackrel{\text{def}}{=} \{\psi : h \subseteq \psi\}$, where the function h is finite.

We will let f be a fixed total recursive function for which $\varphi_{f(i)}$ is a one-one enumeration of all finite members of $\mathcal{P}$ (equivalently of $\mathcal{P}^r$) and for which each $f(i)$ is a canonical index; i.e., we can effectively compute the cardinality of $\varphi_{f(i)}$ from $f(i)$.

[8]The referee informs me that C. E. M. Yates has also constructed an example (unpublished) of an effective operator which is continuous but not partial recursive.

> The set of all finite functions can be effectively enumerated. Of course, every finite function is partial recursive. Let the enumeration be $\phi_{f(0)}$, $\phi_{f(1)}\ldots$ From $i \in \mathbb{N}$ we can get the properties of $\phi_{f(i)}$, such as the cardinality of the domain, the value of $\phi_{f(i)}(n)$ for n in the domain, etc. Note however that, given an $i \in \mathbb{N}$ it is undecidable if ϕ_i (based on a standard enumeration of all partial recursive functions) if a finite function. That is, although we can effectively enumerate all finite functions (through indices) as a subsequence of the standard (index) enumerations (and, if we want, without function repetitions), we cannot effectively use an index enumeration which is the subsequence of *all* the indices that correspond to the partial recursive functions that are finite.
> A similar situation occurs for other effectively enumerable classes of total recursive functions, such as the class of primitive recursive functions.

$\mathcal{P}$ is topologized by taking as basic open sets all sets of the form $\{\psi : \varphi_{f(i)} \subseteq \psi\}$ and we denote $\{\psi : \varphi_{f(i)} \subseteq \psi\}$ by $\phi_{f(i)}$. This topology yields a relative topology on $\mathcal{P}^r$ and we denote $\phi_{f(i)} \cap \mathcal{P}^r = \{\varphi_e : \varphi_{f(i)} \subseteq \varphi_e\}$ also by $\phi_{f(i)}$, relying on the context to make the usage clear.

> Intuitively, we are simultaneously dealing with two very different concepts: mathematical partial functions and effective partial functions. Later in the paper the author limits its attention to (i) effective partial functions; (ii) enumerable (using f, see above) basic open sets; (iii) effective open sets, that is, open sets that can be effectively enumerated as a sequence of basic open sets, (iv) Effectively continuous operators (which need not be continuous)...

Intuitively, we want an operator Φ with domain $\mathcal{D} \subseteq \mathcal{P}$ to be effectively continuous if, given $\psi \in \mathcal{D}$ and a neighbourhood M of $\Phi(\psi)$, we can effectively find a neighbourhood M' of ψ such that $\psi' \in M' \cap \mathcal{D}$ implies that $\Phi(\psi') \in M$. However we restrict ourselves to working with domains in $\mathcal{P}^r$ and to basic open sets as neighbourhoods.

> This is the usual definition of continuity. Instead "neighbourhood" we can use "basic open set".

For the purpose of this paper, we adopt the following

Definition. An operator Φ on the domain $\mathcal{D} \subseteq \mathcal{P}^r$ is *effectively continuous* if there is a partial recursive function $c(e,y)$ such that, if $\phi_e \in \mathcal{D}$ and $\Phi(\phi_e) \in \phi_{f(y)}$, then $c(e,y)$ is defined, $\phi_e \in \phi_{f(c(e,y))}$, and if $\phi_x \in \phi_{f(c(e,y))} \cap \mathcal{D}$ then $\Phi(\phi_x) \in \phi_{f(y)}$. $\square$

> Consider a partial function ϕ_e whose transform $\Phi(\phi_e)$ is in $\phi_{f(y)}$, the basic open set defined by the index y; in symbols: $\Phi(\phi_e) \in \phi_{f(y)}$. Then there is a basic open set $\phi_{f(c(e,y))}$ determined by the index $c(e,y)$, such that $\Phi(\phi_{f(c(e,y))})$ is included in $\phi_{f(y)}$.
> More simply: for every basic open set O containing the image of a function ϕ_e we can effectively obtain a basic open set containing ϕ_e whose image is entirely contained in O.

We remark in passing that an effectively continuous operator need not be an effective operator, even when its range is contained in $\mathcal{P}^r$. To see this we consider the domain $\mathcal{D}$ consisting of all finite functions and consider operators mapping $\mathcal{D}$ into $\mathcal{D}$. For each set $S \subseteq \mathbb{N}$ we define the operator Φ_S by

$$\Phi_s(\phi_e) = \phi_e/S$$

where ϕ_e/S denotes the restriction of to S. One easily verifies that distinct sets S and S' give rise to distinct operators Φ_S and $\Phi_{S'}$.

Furthermore each such operator is effectively continuous because the identity function $c(e,y) = y$ witnesses the effective continuity of each Φ_S. Since there are uncountably many operators Φ_S but only countably many effective operators on any given domain, not all effectively continuous operators can be effective operators. On the other hand, it may be that placing additional conditions on the operator will assure that effective continuity of an operator will imply effectiveness of the operator. For example, we do not know whether every effectively continuous operator mapping *all* of $\mathcal{P}^r$ into $\mathcal{P}^r$ is effective. We also lack an example of an effectively continuous effective operator which is not partial recursive.

Definition [MS]. An operator Φ is effective on the domain $\mathcal{D} \subseteq \mathcal{P}^r$ if there is a total recursive function g such that for all $\phi_e \in \mathcal{D}$, $\Phi(\phi_e) = \phi_{g(e)}$. $\square$

Remark. An immediate consequence of this definition is that if $\phi_e \in \mathcal{D}$ and $\phi_x = \phi_e$ then $\phi_{g(x)} = \phi_{g(e)}$.

Main result.

Lemma 1. Let Φ be a partial recursive operator on the domain $\mathcal{D}$. If $\mathcal{D}' = \{\phi_x : \phi_x \subseteq \phi_e \text{ for some } \phi_e \in \mathcal{D}\}$, Φ can be extended to a partial recursive operator on $\mathcal{D}'$ and there is a total recursive function g such that, for all $\phi_e \in \mathcal{D}'$, $\Phi(\phi_e) = \phi_{g(e)}$. Finally, if t is a recursive function such that

$$\phi_{t(n)} \subseteq \phi_e \in \mathcal{D}', \text{ and } \lim_n \phi_{t(n)} = \phi_e,$$

then

$$\phi_{g(t(n))} \subseteq \phi_{g(e)}, \text{ and } \lim_n \phi_{g(t(n))} = \phi_{g(e)}.$$

Proof. This is well known. E.g., it is an immediate consequence of the definition of partial recursive operator and Lemmas 3.1' and 3.2' given in [L], (bearing in mind that every partial recursive operator is a Banach-Mazur operator). Since we will be working only with domains and ranges contained in $\mathcal{P}^r$, the reader unfamiliar with partial recursive operators may take the existence of such extensions as the defining property for partial recursive operators. □

Lemma 2. Every partial recursive operator is effectively continuous on its domain.

Proof. We simply give the usual proof of continuity, observing that the calculations are effective: Let Φ be partial recursive on the domain $\mathcal{D}$. Let g and $\mathcal{D}'$ be as in Lemma 1. Given ϕ_e and $\phi_{f(y)}$, begin enumerating ϕ_e, letting

$$\phi_{f(h(n,c))} = \phi_e^{(n)} = \text{the set of elements of } \phi_e \text{ enumerated in } \phi_e \text{ by stage } n.$$

By Lemma 1,

$$\phi_{g(e)} = \lim_n \phi_{g(f(h(n,e)))}$$

if $\phi_e \in \mathcal{D}'$, so if $\phi_{f(y)} \subseteq \phi_{g(e)}$ we eventually find n_0 such that $\phi_{f(y)} \subseteq \phi_{g(f(h(n_0,e)))}$. Also by Lemma 1, if $\phi_{f(h(n_0,e))} \subseteq \phi_z$ and $\phi_z \in \mathcal{D}'$ then $\phi_{g(f(h(n_0,e)))} \subseteq \phi_{g(z)}$. Thus if we define $\phi_{c(e,y)} = \phi_{f(h(n_0,e))}$, c will witness the effective continuity Φ. □

The proof of our result blends two techniques. One is the technique introduced by Friedberg to construct effective operators which are not partial recursive. The other is a rate-of-growth argument: Given a basic open set $\phi_{f(y)}$ in the range of the operator Φ we are going to find a basic open set $\phi_{f(c(e,y))}$ in the domain of Φ in such a way that the function c will establish the continuity of Φ. However, to establish continuity, for some fixed e the rate of growth of $|\phi_{(f(e,y))}|$ with respect to $|\phi_{f(y)}|$ will have to be so great that c cannot be a recursive function. ($|\phi_{f(x)}|$ is the cardinality of $\phi_{f(x)}$.)

Theorem. (A) There is an effective operator, Φ, which is continuous on its domain but which is not effectively continuous. (A fortiori, it is not partial recursive.)
((B) The operator Φ of (A) has the following property: It is the union of a partial recursive operator Φ_0 on a completely recursively enumerable domain, $\mathcal{C}$, together with the trivial operator $\Phi_1(\omega) = \omega$ defined only on a certain constant function ω.)

(Remark. The relevance of (B), whose proof the reader may ignore in proving (A), is that any effective operator Φ_0' on a completely recursively enumerable domain $\mathcal{C}$ may be trivially extended to an effective operator Φ_0'' on all $\mathcal{P}^r$ simply by defining $\Phi_0'(\phi_x)$ to be the nowhere defined function for $\phi_x \notin \mathcal{C}$. By [MS], any effective operator defined on all of $\mathcal{P}^r$ is in fact partial recursive, and hence effectively continuous. Thus Φ, which differs almost trivially from Φ_0, is not effectively continuous even though Φ is continuous and Φ_0 can be extended to an effectively continuous operator on all of $\mathcal{P}^r$.

Proof. Let ψ be a partial recursive function which can be majorized by no total recursive function. (E.g., if $\psi(x) = \phi_x(x) + 1$, the assumption that ϕ_e is a total function majorizing ψ leads to an immediate contradiction.)

An element m in of the domain of ψ is called *maximal* if $n < m$ implies $\psi(n) < \psi(m)$ wherever $\psi(m)$ is defined. It is easy to see that any function with a largest maximal element is bounded. Therefore Ψ has no largest maximal element. Since $n = \mu y : [\psi(y)$ is defined$]$ is maximal, ψ has infinitely many maximal elements.

We cannot define the operator Φ which we are seeking directly from the enumerations of the ϕ_e's, for if we did Φ would be partial recursive. Consequently we adopt the technique introduced by Friedberg in [F] to produce effective operators which are not partial recursive. We let $R = \{e : \phi_e(x) = 0$ for all $x \leq e\}$ and let ω be the function $\omega(x) = 0$ for all x. We let ω_n be the function $\{\langle 0,0\rangle, \langle 1,0\rangle, \ldots, \langle n,0\rangle\} = \omega/\{0,1,\ldots,n\}$.

We now construct the total recursive function g which computes Φ.

We find it convenient to use a marker, Λ, in the course of construction. Although distinct members may be simultaneously marked by Λ, once Λ is introduced beside a number it is never moved from the number nor are priority methods used in the construction.

First begin enumerating R; whenever we find $e \in R$ we place ω into $\phi_{g(e)}$.

We also enumerate ψ and whenever we find m in the domain of ψ we look for the smallest $r > 0$ such that

$$\phi_{f(i)} \stackrel{\text{def}}{=} \omega_{\psi(m)-1} \cup \{\langle \psi(m), r\rangle\} \tag{$\star$}$$

is not yet known to have an extension ϕ_e, with $e \in R$ and such that the marker λ does not appear beside the canonical index $f(i)$. We then place the marker λ beside $f(i)$ and we place ω_{m-1} into $\phi_{g(a)}$ for every extension ϕ_a of $\phi_{f(i)}$.

For each $f(i)$ with the marker λ beside it, $\phi_{f(i)}$ defined by ($\star$), we also do the following: if we find $e \in R$ with ϕ_e an extension of $\phi_{f(i)}$, we place ω into $\phi_{g(a)}$ for every extension ϕ_a of $\phi_{p(i)}$. When this occurs, we also find the smallest $r' \geq 0$ for which

$$\phi_{f(i')} \stackrel{\text{def}}{=} \omega_{\psi(m)-1} \ \cup \ \{\langle \psi(m), r'\rangle\}$$

is not yet known to have an extension ϕ_e with $e \in R$ and such that the marker Λ has not been placed beside the canonical index $f(i')$. We place the maker Λ beside $f(i')$ and we place ω_{m-1} into $\phi_{g(a)}$ for every extension ϕ_a of $\phi_{f(i')}$

This completes our description of g.

Let $\mathcal{C}$ be the set of the ϕ_e such that ϕ_e extends some $\phi_{f(i)}$ where $f(i)$ has the marker, Λ, placed beside it in the course of the construction. (Since $\mathcal{C}$ is the class of all r.e. supersets of a r.e. sequence of canonically enumerable finite sets, $\mathcal{C}$ is completely recursively enumerable by a standard characterisation of completely enumerable classes given in [MS] and [R].)

It is clear from our construction of g that $\phi_x = \phi_y$ and if $\phi_x \in \mathcal{C}$, then $\phi_{g(x)} = \phi_{g(y)}$. (In fact we know that either $\phi_{g(x)} = \omega$ or $\phi_{g(x)} = \omega_{m-1}$ for some m in the domain of ψ.) Also, if $\phi_e = \omega$, then $e \in R$, so that $\phi_{g(e)} = \omega$. Thus g determines an effective operator, Φ, on $\mathcal{C} \cup \omega$.

We now show that Φ is continuous at each point of its domain, $\mathcal{C} \cup \{\omega\}$ If $\phi_x \in \mathcal{C}$, then for some z belonging to the domain of ψ and for some $r > 0$, ϕ_x extends $\omega_{\psi(z)-1} \cup \{(\psi(z), r)\}$. Furthermore, for all y such that ψ_y extends

$\omega_{\psi(z)-1} \cup \{(\psi(z), r)\}$, $\phi_x \in \mathcal{C}$ and

$$\Phi(\phi_y) = \Phi(\omega_{\psi(z)-1} \cup \{(\psi(z), r)\})$$

Thus if $\phi_{f(i)} \subseteq \Phi(\phi_x)$ and $\omega_{\psi(z)-1} \cup \{(\psi(z), r)\}) \subseteq \phi_y$, then $\phi_{f(i)} \subseteq \Phi(\phi_y)$.

This establishes that Φ is continuous at each point of $\mathcal{C}$. To establish continuity at ω, suppose $\phi_{f(i)} \subseteq \Phi(\omega)$ $(= \omega)$. Let n_i be a maximal element of the domain of ψ such that $\phi_{f(i)} \subseteq \omega_{n_i-1}$. Clearly $\omega_{\psi(n_i)-1} \subseteq \omega$. Suppose $\phi_x \in \mathcal{C}$ and $\omega_{\psi(n_i)-1} \subseteq \phi_x$. Then there is an element z belonging to the domain of ψ and an $r > 0$ such that $\omega_{\psi(z)-1} \cup \{(\psi(z), r)\} \subseteq \phi_x$.

By the construction,

$$\omega_{z-1} \subseteq \Phi(\omega_{\psi(z)-1} \cup \{(\psi(z), r)\} = \Phi(\phi_x)$$

Since $\psi(n_i) \leq\leq \psi(z)$ and n_i is maximal, $n_i \leq z$. Thus

$$\phi_{f(i)} \subseteq \omega_{n_i-1} \subseteq \omega_{z-1} \subseteq \Phi(\phi_x)$$

establishing the continuity of Φat ω.

It remains to show that Φ is not effectively continuous on $\mathcal{C} \cup \{\omega\}$.

We first show that for each m in the domain of ψ there is some extension of $\omega_{\psi(m)-1}$ which gets mapped to ω_{m-1}. It is in fact clear from our construction that this will happen unless $\omega_{\psi(m)-1} \cup \{\langle\psi(m), r\rangle\}$ gets mapped to ω for infinitely many $r > 0$. But for $\omega_{\psi(m)-1} \cup \{\langle\psi(m), r\rangle\}$ to get mapped to w $(r > 0)$, there must be some ϕ_e, extending $\omega_{\psi(m)-1} \cup \{\langle\psi(m), r\rangle\}$ with $e \in R$, i.e. with $\phi_e(x) = 0$ for all $x \leq e$. Since this implies that $e < \psi(m)$, there are at most finitely many such e's, and so for each m in the domain of ψ, there is some $r > 0$ with $\omega_{\psi(m)-1} \cup \{\langle\psi(m), r\rangle\}$ mapped to ω_{m-1}.

Now suppose that Φ were effectively continuous. Since $\Phi(\omega = \omega$, given n, since $\omega_n \leq \omega$, we could effectively find $\phi_{f(s(n))} \subseteq \omega$ such that each extension of $\phi_{s(n)}$ in the domain of Φ has an image which extends ω_n. Letting t be the total recursive function such that

$$t(n) = \max\{y : \langle y, 0\rangle \in \phi_{f(s(n))}\}$$

we would have that $\omega_{t(n)} \subseteq \phi_e$ and $\phi_e \in \mathcal{C} \cup \{\omega\}$ implies

$\omega_n \subseteq \Phi(\phi_e)$. But for m in the domain of ψ there is some $r > 0$ such that

$$\Phi(\omega_{\psi(m)-1} \cup \{\langle \psi(m), r \rangle\}) = \omega_{m-1}$$

Since $t(m) \leq \psi(m)$ implies

$$\omega_{t(m)} \subseteq \omega_{\psi(m)-1} \cup \{\langle \psi(m), r \rangle\}$$

$t(m) < \psi(m)$ implies $\omega_m \subseteq \omega_{m-1}$, a contradiction. Thus $t(m) \geq \psi(m)$ for all m for which $\psi(m)$ is defined. This means that t majorizes ψ, and this contradiction shows that Φ is not effectively continuous, completing our proof. □

Young's paper references

[F] R. S. Friedberg, *Un contre-example relatif aux fonctionelles recursives*, C. R. Acad. Sci, Paris 247 (1958), pp. 852-854.

[KLS] G. Kreisel, D. Lacombe, and J. R.Shoenfield, "Partial recursive functionals and effective operations", in *Constructivity in Mathematics*, North-Holland, Amsterdam, 1959, pp. 290-297.

[L] A. N. Lachlan, *Effective operations in a general setting*, J. Symbolic Logic **29** (1964), pp. 163-178.

[MS] J. Myhill and J. C. Shepherdson, *Effective operations on partial recursive functions*, Z. Math. Logik Grundlagen Math. **1** (1955), pp. 310-317.

[P] M. B. Pour-El, *A comparison of five computable operators*, Z. Math. Logik Grundlagen Math. **6** (1960), pp. 325-340.

[R] H. G. Rice, *Completely recursively enumerable classes and their key arrays*, J. Symbolic Logic **74** (1953), pp. 358-366.

References

[1] M.J. Beeson. The nonderivability in intuitionistic formal systems of theorems on the continuity of effective operations. *Journal of Symbolic Logic*, 41:321–346, 1975.

[2] N. Cutland. *Computability: An Introduction to Recursive Function Theory.* Cambridge University Press, 1980.

[3] Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* MIT Press Cambridge, MA, 1987. Original edition 1967. Third printing (1992).

[4] Paul R. Young. An effective operator, continuous but not partial recursive. *Proceedings of the American Mathematical Society*, 19(1):103–108, 1968.