Exercises from *Recursive Functions*

John Peloquin

Chapter 2

Exercises

Definition. $K = \{x \mid \varphi_x(x) \text{ converges}\}$.

Exercise (2-1)**.** For $k \geq 0$, the function

$$g(x) = \begin{cases} 1 & \text{if } x \leq k \\ 0 & \text{if } x > k \end{cases}$$

is primitive recursive.

Proof. Note the following functions are primitive recursive:

Predecessor (truncated) $p(x) = \lambda x[\max(0, x-1)]$

Difference (truncated) $d(x,y) = \lambda xy[\max(0, x-y)]$

Signal $s(x) = \lambda x[\min(x, 1)]$

Indeed, note $p(0) = 0$ and $p(x+1) = x$; $d(x,0) = x$ and $d(x, y+1) = p(d(x,y))$; and $s(x) = d(x, p(x))$. Now $g(x) = s(d(k+1, x))$ since $x \leq k$ iff $0 < (k+1) - x$. □

Exercise (2-2)**.** The function

$$f(x) = \begin{cases} 1 & \text{if } \varphi_x(x) = 1 \\ 0 & \text{otherwise} \end{cases}$$

is not recursive.

Proof. If so, then $g = \lambda x[1 - f(x)]$ is also recursive, say $g = \varphi_k$. But then

$$\varphi_k(k) = 1 \iff g(k) = 1 \iff f(k) = 0 \iff \varphi_k(k) \neq 1$$

—a contradiction. □

Exercise (2-3)**.** Fix an effective enumeration of primitive recursive functions and let f_x denote the $(x+1)$-th function. Then

$$g = \lambda xy[f_x(y)]$$

is recursive but not primitive recursive.

Proof. By Church's Thesis, there is a recursive s such that $f_x = \varphi_{s(x)}$ for all x. Now if u is an index of the universal function for unary recursive functions, then

$$g = \lambda xy[\varphi_{s(x)}(y)] = \lambda xy[\varphi_u^{(2)}(s(x), y)]$$

so g is recursive.

If g is primitive recursive, then so is $h = \lambda x[g(x,x)+1]$. But then $h(x) = f_x(x)+1$ for all x, so $h \neq f_x$ for all x—a contradiction. □

Exercise (2-8)**.** There is no recursive function $f(z,x)$ such that

$$P_z(x) \text{ halts} \iff P_z(x) \text{ halts in at most } f(z,x) \text{ steps}$$

Proof. By Church's Thesis,

$$s(z,x,y) = \begin{cases} 1 & \text{if } P_z(x) \text{ halts in at most } y \text{ steps} \\ 0 & \text{otherwise} \end{cases}$$

is recursive. If f exists, then $h(z,x) = s(z,x,f(z,x))$ is recursive, but

$$h(z,x) = \begin{cases} 1 & \text{if } P_z(x) \text{ halts} \\ 0 & \text{otherwise} \end{cases}$$

—contradicting the halting problem. □

Exercise (2-9)**.** There is no recursive function f with $\text{range}(f) = \{x \mid \varphi_x \text{ is total}\}$.

Proof. If so, set $g = \lambda x[\varphi_{f(x)}(x)+1]$. Then g is recursive, say $g = \varphi_{f(k)}$, but

$$g(k) = \varphi_{f(k)}(k) + 1 = g(k) + 1$$

—a contradiction. □

Exercise (2-10)**.** Let $\mathscr{P}$ denote the class of unary partial recursive functions. Call a surjective map $\pi : \mathbb{N} \to \mathscr{P}$ a *numbering* of $\mathscr{P}$. Let $\pi_0 : \mathbb{N} \to \mathscr{P}$ denote the numbering induced by our effective coding of programs, and call a numbering π *acceptable* if the following conditions hold:

(A1) There is a recursive function f such that $\pi = \pi_0 f$.

(A2) There is a recursive function g such that $\pi_0 = \pi g$.

Let $\varphi_{x,\pi}$ denote $\pi(x)$, with $\varphi_x = \varphi_{x,\pi_0}$ as usual. Then:

(a) Any effective enumeration of programs induces an acceptable numbering.

(b) (A1) is equivalent to the unary enumeration theorem for π, that is, (A1) holds iff the following holds:

> There exists a partial recursive function $\chi(k,x)$ such that for all k, $\varphi_{k,\pi} = \lambda x[\chi(k,x)]$.

(c) (A2) is equivalent to the unary s-m-n theorem for π, that is, (A2) holds iff the following holds:

> For all partial recursive functions $\psi(x,y)$, there exists a recursive function s such that for all x, $\varphi_{s(x),\pi} = \lambda y[\psi(x,y)]$.

(d) (A2) implies that $\pi^{-1}(\psi)$ is infinite for all $\psi \in \mathscr{P}$.

Proof. (a) Immediate by Church's Thesis. Let $\pi(x)$ be the function computed by the $(x+1)$-th program in the enumeration; let $f(x)$ find the $(x+1)$-th program in the enumeration and return its code; and let $g(x)$ find the program in the enumeration coded by x and return $k-1$ where k is its position.

(b)

$\Longrightarrow$ Pull back the enumeration theorem from π_0 to π using f.

Let ψ be universal for π_0. Then for all k,

$$\varphi_{f(k)} = \lambda x[\psi(f(k),x)]$$

Set $\chi(k,x) = \psi(f(k),x)$. Since $\pi = \pi_0 f$, $\varphi_{k,\pi} = \varphi_{f(k)}$ for all k. So

$$\varphi_{k,\pi} = \lambda x[\chi(k,x)]$$

for all k, that is, χ is universal for π.

$\Longleftarrow$ Invert the enumeration theorem for π using the s-m-n theorem for π_0.

Given χ universal for π, define f using the s-m-n theorem for π_0 such that

$$\varphi_{f(k)} = \lambda x[\chi(k,x)]$$

for all k. Then $\varphi_{k,\pi} = \varphi_{f(k)}$ for all k, so $\pi = \pi_0 f$.

(c)

$\Longrightarrow$ Push forward the s-m-n theorem from π_0 to π using g.

Given $\psi(x,y)$, choose r by the s-m-n theorem for π_0 such that

$$\varphi_{r(x)} = \lambda y[\psi(x,y)]$$

for all x. Since $\pi_0 = \pi g$, $\varphi_{r(x)} = \varphi_{g(r(x)),\pi} = \varphi_{(g\circ r)(x),\pi}$ for all x, so $s = g \circ r$ works.

$\Longleftarrow$ Invert the s-m-n theorem for π using the enumeration theorem for π_0.

Let ψ be universal for π_0. Choose g by the s-m-n theorem for π such that

$$\varphi_{g(x),\pi} = \lambda y[\psi(x,y)] = \varphi_x$$

for all x. Then $\pi_0 = \pi g$.

(d) If not, choose $\psi \in \mathscr{P}$ with $\pi^{-1}(\psi) \supseteq g(\pi_0^{-1}(\psi))$ finite. Then it is possible to decide for any x if $\varphi_x = \psi$ by computing $g(x)$ and deciding if $g(x) \in \pi^{-1}(\psi)$—a contradiction. □

Remark. This exercise shows invariance of the enumeration and *s-m-n* theorems under acceptable numberings. Intuitively, this means the details of our particular numbering do not matter so long as we can effectively go back and forth between numbers and programs—that is, code can be effectively identified with data.

Intuitively, (A1) states that we can effectively decode program numbers, and equivalence with the enumeration theorem is natural since universal programs decode. Conversely, (A2) states that we can effectively encode programs as numbers, and equivalence with the *s-m-n* theorem is natural since the proof of the *s-m-n* theorem involves program hacking and encoding.

The exercise and its proof clearly showcase the inverse relationship between the enumeration and *s-m-n* theorems.

Exercise (2-11)**.** Fix a recursive bijection $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Define $\pi : \mathbb{N} \to \mathscr{P}$ by

$$\pi(y) = \lambda x \begin{cases} \begin{cases} \text{divergent} & \text{if } x_1 = 0 \\ x_1 - 1 & \text{if } x_1 \neq 0 \end{cases} & \text{if } x = 0 \\ \varphi_{x_2}(y) & \text{if } x \neq 0 \end{cases}$$

where $y = g(x_1, x_2)$. Then (in the terminology of Exercise 2-10):

(a) π is a numbering.

(b) (A1) holds for π but (A2) does not hold for π.

Proof. (a) Given $\psi \in \mathscr{P}$, write $\psi = \varphi_{x_2}$. If $\psi(0)$ is divergent, set $x_1 = 0$, otherwise set $x_1 = \psi(0) + 1$. By construction $\pi(g(x_1, x_2)) = \psi$. Thus π is surjective.

(b) To see that (A1) holds, note a program code for $\pi(y)$ can be effectively obtained from y by the following procedure:

1. Find x_1, x_2 with $g(x_1, x_2) = y$ (by recursiveness and surjectivity of g).
2. Decode x_2 to obtain the program P_{x_2}.
3. Hack P_{x_2} to do special processing on input $x = 0$: make it diverge (enter an infinite loop) if $x_1 = 0$, and make it return $x_1 - 1$ otherwise.
4. Encode the hacked program.

To see that (A2) does not hold, suppose h is a witness. Then the following procedure decides for any k whether $\varphi_k(0)$ converges—a contradiction:

1. Compute $y = h(k)$.
2. Find x_1, x_2 with $g(x_1, x_2) = y$.
3. Examine x_1. If $x_1 = 0$, $\varphi_k(0)$ diverges, otherwise $\varphi_k(0)$ converges. □

Remark. Intuitively, the numbering in this exercise is incomplete (for algorithms) because it encodes too much information about a function in its number (namely, whether or not the function converges on input 0).

Exercise (2-12)**.** If f is recursive, there exists g recursive with $f = \lambda x[\mu y[g(x, y) = 1]]$.

Proof. By Church's Thesis, since f is total,

$$g(x,y) = \begin{cases} 1 & \text{if } f(x) = y \\ 0 & \text{otherwise} \end{cases}$$

is recursive, and clearly satisfies the desired property. □

Exercise (2-13)**.** There is a partial recursive function ψ such that $\lambda x[\mu y[\psi(x,y) = 1]]$ is not partial recursive.

Proof. Define

$$\psi(x,y) = \begin{cases} 1 & \text{if } y > x \text{ and } y \in K \\ \text{divergent} & \text{otherwise} \end{cases}$$

Suppose the induced μ function is partial recursive. Let t_0 be the least index of a recursive function and define

$$\begin{aligned} t(0) &= t_0 \\ t(x+1) &= \mu y[\psi(t(x), y) = 1] \\ &= \mu y > t(x)[y \in K] \end{aligned}$$

Then t is partial recursive. In fact, since there are infinitely many indices of recursive functions, t is total, and the range of t includes all such indices. Now define

$$f = \lambda x[\varphi_{t(x)}(t(x)) + 1]$$

Then by construction of t, f is recursive but f differs from every recursive function—a contradiction. □

Remark. This exercise shows that the class of partial recursive functions is not closed under application of μ. This is a consequence of our simple definition of μ. In some contexts, μ is defined such that closure holds (see [1], p. 43).

Exercise (2-15)**.** There is a recursive function f such that for all primitive recursive functions g, $f(x) > g(x)$ for all but finitely many x.

Proof. Let f do the following on input x:

1. Generate the first x+1 primitive recursive function derivations, letting $g_0, \ldots, g_x$ denote the functions derived.
2. Calculate successively $g_0(x), \ldots, g_x(x)$.
3. Return $1 + \sum_{i=0}^{x} g_i(x)$.

By Church's Thesis, f is recursive. If g is primitive recursive, then $g = g_x$ for some x, so by construction $f(y) > g(y)$ for all $y \geq x$. □

Remark. In this proof, the input parameter x is used to facilitate varying over the indices of infinitely many functions. In turn, a property is ensured to hold for any given such function almost everywhere (namely, for inputs greater than an index). This technique is reusable and appears often.

Exercise (2-19)**.** There is a partial recursive ψ such that range(ψ) is not decidable.

Proof. Define

$$\psi(x) = \begin{cases} x & \text{if } x \in K \\ \text{divergent} & \text{otherwise} \end{cases}$$

By Church's Thesis, ψ is partial recursive, but range(ψ) $= K$ is undecidable. □

Exercise (2-23)**.** The halting problem reduces to the problem of deciding whether φ_x is total.

Proof. By the s-m-n theorem, there is recursive s such that for all x,

$$\varphi_{s(x)} = \lambda y \begin{cases} 1 & \text{if } x \in K \\ \text{divergent} & \text{otherwise} \end{cases}$$

Then $x \in K$ iff $\varphi_{s(x)}$ is total. □

Exercise (2-27)**.** If there are recursive functions f and g with range(f) $= A$ and range(g) $= \overline{A}$, then A is decidable.

Proof. Decide $x \in A$ by enumerating $f(0), g(0), f(1), g(1), \ldots$ until x is found. □

Exercise (2-29)**.** The set $\{x \mid \text{range}(\varphi_x) \text{ is decidable}\}$ is undecidable.

Proof. By reduction. Choose recursive s by the s-m-n theorem such that

$$\varphi_{s(x)} = \lambda y \begin{cases} \begin{cases} y & \text{if } y \in K \\ \text{divergent} & \text{if } y \notin K \end{cases} & \text{if } x \in K \\ \text{divergent} & \text{if } x \notin K \end{cases}$$

If $x \notin K$, then range($\varphi_{s(x)}$) $= \emptyset$ is trivially decidable. If $x \in K$, then range($\varphi_{s(x)}$) $= K$ is undecidable. So $x \notin K$ iff range($\varphi_{s(x)}$) is decidable. The result follows. □

Exercise (2-30)**.** Let A, B be disjoint sets with $A = \text{domain}(\varphi)$ and $B = \text{domain}(\psi)$.

(a) There is a partial recursive function χ with $\chi(A) = \{0\}$ and $\chi(B) = \{1\}$.

(b) There is not necessarily a recursive function f with $f(A) = \{0\}$ and $f(B) = \{1\}$.

Proof. (a) By Church's Thesis. Have $\chi(x)$ run computations for $\varphi(x)$ and $\psi(x)$ simultaneously and return 0 if $\varphi(x)$ converges or 1 if $\psi(x)$ converges (or diverge otherwise).

(b) Consider

$$A = K_1 = \{x \mid \varphi_x(x) = 1\} \quad \text{and} \quad B = K_0 = \{x \mid \varphi_x(x) = 0\}$$

Clearly A and B are disjoint and by Church's Thesis each is the domain of a partial recursive function. Suppose f is recursive with $f(A) = \{0\}$ and $f(B) = \{1\}$. Assume without loss of generality that range(f) $\subseteq \{0,1\}$ (otherwise apply the signal function). Then $f = \varphi_k$ for some k, and $f(k) \in \{0,1\}$. But

$$f(k) = 0 \iff \varphi_k(k) = 0 \iff k \in K_0 \iff f(k) = 1 \iff f(k) \neq 0$$

—a contradiction. □

Exercise (2-31)**.** The function $\lambda x[\mu y[\varphi_x(y) \text{ diverges}]]$ is not partial recursive.

Proof. Choose recursive s by the s-m-n theorem such that

$$\varphi_{s(x)} = \lambda y \begin{cases} 0 & \text{if } y < x \\ \varphi_x(x) & \text{if } y = x \\ \text{divergent} & \text{if } y > x \end{cases}$$

If the given function is partial recursive, then

$$f = \lambda x[\mu y[\varphi_{s(x)}(y) \text{ diverges}]]$$

is recursive. But $x \in K$ iff $x \neq f(x)$, so K is decidable—a contradiction. □

Exercise (2-33)**.** The set $\{x \mid \varphi_x \text{ is extendable to a recursive function}\}$ is undecidable.

Proof. By reduction. Choose recursive s by the s-m-n theorem such that

$$\varphi_{s(x)} = \lambda y \begin{cases} \varphi_y(y) + 1 & \text{if } x \in K \\ \text{divergent} & \text{if } x \notin K \end{cases}$$

If $x \notin K$, then $\varphi_{s(x)}$ is trivially extendable. If $x \in K$, then $\varphi_{s(x)}$ is not extendable by the proof of Theorem 2-II. Thus $x \notin K$ iff $\varphi_{s(x)}$ is extendable, and the result follows. □

Exercise (2-34)**.** Say B is decidable relative to A if there is a partial recursive φ such that for $x \in A$, $\varphi(x) = 1$ if $x \in B$ and $\varphi(x) = 0$ if $x \notin B$.

Set $T = \{x \mid \varphi_x \text{ is total (i.e., recursive)}\}$.

(a) $\{x \mid \varphi_x \text{ is constant}\}$ is not decidable relative to T.

(b) $\{x \mid \varphi_x \text{ is injective}\}$ is not decidable relative to T.

(c) $\{x \mid \varphi_x(y) = z\}$, for y, z fixed, is decidable relative to T.

(d) $\{x \mid \text{range}(\varphi_x) \text{ is infinite}\}$ is not decidable relative to T.

Proof. For undecidability cases, reduce the halting problem within T.

(a) By the reduction in Exercise 2-26(a), noting that $\varphi_{s(x)}$ is total for all x.

(b) By the reduction in Exercise 2-26(c), noting that $\varphi_{s(x)}$ is total for all x.

(c) Immediate by Church's Thesis.

(d) By the proof of (b). □

Exercise (2-39 (Rice))**.** Let $\mathscr{C}$ be a subset of the unary partial recursive functions. Then $C = \{x \mid \varphi_x \in \mathscr{C}\}$ is undecidable iff $\mathscr{C}$ is nontrivial and proper.

Proof. The forward direction is trivial.

For the reverse, reduce K to C. Assume without loss of generality that the empty function $\varphi_\emptyset$ is not in $\mathscr{C}$ (otherwise use the complement), and fix $\varphi \in \mathscr{C}$. Now by the s-m-n theorem choose recursive s such that

$$\varphi_{s(x)} = \lambda y \begin{cases} \varphi(y) & \text{if } x \in K \\ \text{divergent} & \text{otherwise} \end{cases}$$

If $x \in K$, then $\varphi_{s(x)} = \varphi \in \mathscr{C}$, so $s(x) \in C$. If $x \notin K$, then $\varphi_{s(x)} = \varphi_\emptyset \notin \mathscr{C}$, so $s(x) \notin C$. Thus $x \in K$ iff $s(x) \in C$, and the result follows. □

Chapter 4

Exercises

Exercise (4-1)**.** Let X be a set, G a group of transformations on X (that is, a group of surjective functions on X under composition), and e the identity in G.

(a) e is the identity map on X.

(b) If $g \in G$, $g^{-1} \in G$ is a two-sided inverse map of g. In particular, G consists of permutations of X.

(c) G-isomorphism is an equivalence relation among subsets of X.

Proof. (a) Since $ee = e$ in G, $e(e(x)) = e(x)$ for all $x \in X$. If $y \in X$, choose $x \in X$ with $e(x) = y$ by surjectivity. Then $e(y) = y$. So e is the identity map.

(b) Since $g^{-1}g = e = gg^{-1}$ in G, this follows from (a).

(c) Let $A, B, C \subseteq X$.

Reflexivity By (a), $A \equiv A$ since $e[A] = A$.

Symmetry By (b), if $A \equiv B$ then $B \equiv A$ since if $g[A] = B$ then $g^{-1}[B] = A$.

Transitivity If $A \equiv B$ and $B \equiv C$ then $A \equiv C$ since if $g[A] = B$ and $h[B] = C$ then $hg[A] = h[g[A]] = h[B] = C$. □

Exercise (4-2)**.** There exist nonrecursive permutations.

Proof. Since K and $\overline{K}$ are both infinite, there exists a permutation π mapping evens to K and odds to $\overline{K}$. Clearly π is not recursive, lest K is decidable. □

Exercise (4-3)**.** If ψ is injective and partial recursive, then ψ^{-1} is partial recursive.

Proof. By Church's Thesis. Given y, run the first n steps of computations $\psi(0), \ldots, \psi(n)$ for $n = 0, 1, \ldots$. If a computation halts on input x with output y, then $\psi^{-1}(y) = x$. □

Exercise (4-5)**.** The set $\{x \mid \varphi_x \text{ is a recursive permutation}\}$ is undecidable.

Proof. By the s-m-n theorem, choose recursive s such that

$$\varphi_{s(x)} = \lambda y \begin{cases} y & \text{if } \varphi_x(y) \text{ is convergent} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then φ_x is recursive iff $\varphi_{s(x)}$ is the identity map, a recursive permutation. Hence undecidability follows from undecidability of recursiveness (Theorem 1-VIII). □

Exercise (4-6)**.**

(a) There does not exist an effective enumeration of (Gödel numbers of) all the recursive permutations.

(b) The group of recursive permutations is not finitely generated.

Proof. (a) By diagonalization. Suppose there is such an effective enumeration $\pi_0, \pi_1, \pi_2 \ldots$. Assume without loss of generality that there do not exist k, y such that for all $x \geq k$, $\pi_x(x) = y$. (If this is not the case, swap pairs of permutations arbitrarily far out to obtain an effective enumeration with this property.)

Now construct π recursively as follows:

$$\pi(0) = \mu y[y \neq \pi_0(0)]$$
$$\pi(x+1) = \mu y[y \neq \pi_{x+1}(x+1), \pi(0), \ldots, \pi(x)]$$

Clearly π is recursive and injective. Claim that π is surjective. Indeed, if not, let y be least with $y \notin \text{range}(\pi)$. Then there is least k such that

$$\{0, \ldots, y-1\} \subseteq \pi[\{0, \ldots, k\}]$$

By assumption, there is least $x > k$ such that $\pi_x(x) \neq y$. But then $\pi(x) = y$ by construction, so π is surjective after all.

Now π is a recursive permutation, but $\pi \neq \pi_k$ for all k—a contradiction.

(b) If it is, we can effectively enumerate finite sequences of (Gödel numbers of) recursive permutations such that any recursive permutation is the composite over one of the sequences. But then we can effectively enumerate all recursive permutations (by Theorem 1-VI), contradicting (a).

□

Exercise (4-10)**.** Recursive invariance of properties of sets:

	Property	**Invariant**
(i)	Is empty	Yes
(ii)	Is range of recursive function	Yes
(iii)	Is domain of partial recursive function	Yes
(iv)	Possesses recursive characteristic function	Yes
(v)	Is infinite	Yes
(vi)	Contains the even integers	No
(vii)	Contains a set isomorphic to the even integers	Yes

Proof. For (i) and (v), use cardinality.
For (ii)–(iv) and (vii), look at composites.
For (vi), consider the recursive permutation

$$\pi = (01)(23)\cdots(2k\ 2k+1)\cdots$$

Then for E the set of even integers, $\pi[E]$ is the set of odd integers. □

Exercise (4-12)**.** Recursive invariance of properties of partial functions:

	Property	Invariant
(i)	Has infinite domain	Yes
(ii)	Has domain containing range	Yes
(iii)	Is recursive	Yes
(iv)	Is partial recursive	Yes
(v)	Has domain containing the even integers	No
(vi)	Has domain containing a set isomorphic to the even integers	Yes
(vii)	Has domain with seven members	Yes

Proof. For (i) and (vii), use cardinality.

For (ii)–(iv) and (vi), use Theorem II.

For (v), consider π from Exercise 4-10 and

$$\varphi = \lambda x \begin{cases} 0 & \text{if } x \text{ is even} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Set $\psi = \pi^{-1}\varphi\pi$. Then by Theorem II, $\varphi \equiv \psi$, but domain(φ) is the set of even integers while domain(ψ) is the set of odd integers. □

Exercise (4-14)**.** The recursive permutations do not form a single isomorphism type.

Proof. Consider $\pi = (01)$ and $\rho = (012)$. Every conjugate of π is a transposition, and hence not equal to ρ, so π and ρ are not isomorphic by Theorem II. □

Remark. The property of being a recursive permutation is recursively invariant.

Chapter 5

Exercise (5-2)**.** An infinite set A is recursively enumerable iff A is recursively enumerable without repetitions.

Proof.

$\Longrightarrow$ By Corollary V(d), A is enumerated by an injective partial recursive function whose domain is an initial segment of $\mathbb{N}$. Since A is infinite, the domain is just $\mathbb{N}$.

$\Longleftarrow$ Trivial. □

Exercise (5-3)**.** (a) If A, B are recursive and infinite with infinite complements, then $A \equiv B$. There are $\aleph_0$ such sets.

(b) $\mathscr{T}$ is an isomorphism type of recursive sets iff $\mathscr{T}$ satisfies the following:

(R1) $\mathscr{T}$ is a nonempty set of recursive sets.
(R2) If $X, Y \in \mathscr{T}$, $|X| = |Y|$ and $|\overline{X}| = |\overline{Y}|$.
(R3) If $X \in \mathscr{T}$, Y is recursive, and $|X| = |Y|$ and $|\overline{X}| = |\overline{Y}|$, then $Y \in \mathscr{T}$.

There are $\aleph_0$ recursive isomorphism types.

Proof. (a) The isomorphism follows from (b).

There are at least $\aleph_0$ such sets since the sets $k\mathbb{N}$ $(k > 1)$ are examples, and there are at most $\aleph_0$ such sets since there are $\aleph_0$ recursive sets.

(b) First note if X, Y are recursive, $X \equiv Y$ iff $|X| = |Y|$ and $|\overline{X}| = |\overline{Y}|$. The forward direction is obvious; for the reverse, define $\pi : \mathbb{N} \to \mathbb{N}$ recursively by

$$\pi = \lambda x \begin{cases} \mu y[y \in Y - \{\pi(0), \ldots, \pi(x-1)\}] & \text{if } x \in X \\ \mu y[y \in \overline{Y} - \{\pi(0), \ldots, \pi(x-1)\}] & \text{if } x \in \overline{X} \end{cases}$$

Clearly π is a recursive bijection and $\pi[X] = Y$, so $X \equiv Y$.

Now proceed:

$\Longrightarrow$ (R1) is trivial, and (R2) and (R3) follow from the above.

$\Longleftarrow$ By (R1), there is $X \in \mathscr{T}$. Then by (R1)–(R3) and the above, $\mathscr{T}$ is just the isomorphism type of X.

It is then immediate that there are $\aleph_0$ recursive isomorphism types, since there are $\aleph_0$ sets satisfying (R1)–(R3).

□

Exercise (5-6)**.** The set of all recursive sets is recursively enumerable. (That is, there exists a recursively enumerable set C such that X is recursive iff there exists $x \in C$ with $W_x = X$).

Proof. By Church's Thesis, the following algorithm yields a recursive function r:
Given input x,

1. Construct source code for a program which does the following:

   Given input n, using a universal program and running it on inputs $0, 1, 2\ldots$ successively, search for a sequence $x_0, \ldots, x_n$ with

   (i) $x_0 = 0$ and $\varphi_x(x_0)$ convergent
   (ii) $x_{i+1} = \mu y[y > x_i$ and $\varphi_x(y)$ convergent] for $0 \le i < n$
   (iii) $\varphi_x(x_i) \le \varphi_x(x_{i+1})$ for $0 \le i < n$

   If and when x_n is found, return $\varphi_x(x_n)$ (otherwise never return).

2. Translate the source code so constructed from range instructions to domain instructions (Corollary V(b)).

3. Encode the domain instructions and return the code.

Claim r enumerates indices of the recursive sets.

Indeed, r enumerates only indices of recursive sets, for if φ_x is not recursive, or does not recursively enumerate a set in nondecreasing order, then by construction $W_{r(x)}$ is finite and hence recursive. Conversely, if φ_x recursively enumerates a set in nondecreasing order, then by construction $W_{r(x)} = \text{range}(\varphi_x)$, which is recursive by Theorem III. Moreover, if X is recursive, then by Theorem III again X is recursively enumerated in nondecreasing order, say by φ_x, so $W_{r(x)} = X$. □

Exercise (5-8)**.** There exists an infinite set with no infinite recursively enumerable subset.

Proof. There are $\aleph_0$ infinite recursively enumerable sets, say $A_0, A_1, \ldots$. Define a function f as follows:

$$f(0) = \mu y[y \in A_0]$$
$$f(x+1) = \mu y[y > f(x) + 1 \text{ and } y \in A_{x+1}]$$

Then f is well-defined and increasing since each A_x is infinite.

Set $B = \overline{f[\mathbb{N}]}$. Then B is infinite since $f(x) + 1 \in B$ for all x. But for all x, $A_x \not\subseteq B$ since $f(x) \in A_x - B$. □

Exercise (5-9)**.** Recursiveness and recursive enumerability of sets:

	Set	**(a) recursive**	**(b) r.e.**	**(c) r.e. complement**
(i)	$\{x \mid x$ prime$\}$	Y	Y	Y
(ii)	$\{x \mid \exists$ run of $\ge x$ 7's in $\pi\}$	Y	Y	Y
(iii)	$\{x \mid \exists$ run of $= x$ 7's in $\pi\}$		Y	
(iv)	$\{x \mid W_x = \emptyset\}$	N	N	Y
(v)	$\{x \mid W_x$ infinite$\}$	N	N	N
(vi)	$\{x \mid \varphi_x$ total$\}$	N	N	N
(vii)	$\{x \mid W_x = W_n\}$ $(W_n \neq \emptyset)$	N	N	N
(viii)	$\{x \mid W_x$ recursive$\}$	N	N	N

Proof. For (i), use Church's Thesis.

For (ii) and (iii), use results of §1.3.

For (iv), for (b) use reduction of $\overline{K}$ and for (c) use projection.

For (v), use reduction of the total functions (see (vi)). By the s-m-n theorem, choose recursive s with

$$\varphi_{s(x)} = \lambda y[\sum_{i=0}^{y} \varphi_x(i)]$$

Then φ_x is total iff $W_{s(x)}$ is infinite.

For (vi), for (b) use Exercise 2-9 and for (c) use reduction of $\overline{K}$.

For (vii), use reduction of $\overline{K}$. For (b), if W_n is finite, choose recursive t with

$$\varphi_{t(x)} = \lambda y \begin{cases} \varphi_n(y) & \text{if } P_x(x) \text{ does not converge in } \leq y \text{ steps} \\ 0 & \text{otherwise} \end{cases}$$

If $x \notin K$ then $W_{t(x)} = W_n$, and if $x \in K$ then $W_{t(x)}$ is infinite so $W_{t(x)} \neq W_n$. If W_n is infinite, do a similar reduction with

$$\varphi_{u(x)} = \lambda y \begin{cases} \varphi_n(y) & \text{if } P_x(x) \text{ does not converge in } \leq y \text{ steps} \\ \text{undefined} & \text{otherwise} \end{cases}$$

For (viii), use reduction of $\overline{K}$. For (c), choose recursive v with

$$\varphi_{v(x)} = \lambda y \begin{cases} \varphi_y(y) & \text{if } P_x(x) \text{ does not converge in } \leq y \text{ steps} \\ \text{undefined} & \text{otherwise} \end{cases}$$

If $x \notin K$ then $W_{v(x)} = K$ is not recursive, and if $x \in K$ then $W_{v(x)}$ is finite so recursive. □

Exercise (5-12)**.** There are $\aleph_0$ sets recursively enumerable but not recursive.

Proof. There are at least $\aleph_0$ since the sets

$$n + K = \{ n + x \mid x \in K \} \qquad (n \geq 0)$$

are examples. There are at most $\aleph_0$ since there are $\aleph_0$ recursively enumerable sets. □

Exercise (5-15)**.** Let $\pi : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a recursive injection, and define

$$\psi_z = \lambda x y[\varphi_z(\pi(x, y))]$$

As z varies, ψ_z varies over all binary partial recursive functions.

Proof. Each ψ_z is clearly partial recursive.

Given $\psi(x, y)$ partial recursive, let π_1, π_2 be recursive with $\pi(\pi_1(u), \pi_2(u)) = u$ for all u and consider

$$\varphi = \lambda u[\psi(\pi_1(u), \pi_2(u))]$$

Then φ is partial recursive, so $\varphi = \varphi_z$ for some z and

$$\begin{aligned}\psi_z &= \lambda x y[\varphi_z(\pi(x,y))] \\ &= \lambda x y[\psi(\pi_1(\pi(x,y)), \pi_2(\pi(x,y)))] \\ &= \lambda x y[\psi(x,y)]\end{aligned}$$

So $\psi_z = \psi$. □

Exercise (5-17)**.** Let f be a function and ψ a partial function.

(a) The following are equivalent:

(i) f is recursive
(ii) f as a binary relation is recursive
(iii) f as a binary relation is recursively enumerable

(b) ψ is partial recursive iff ψ as a binary relation is recursively enumerable.

Proof. (a) Clearly (i) $\Longrightarrow$ (ii), and (ii) $\Longrightarrow$ (iii) by Theorem I. Since f is total, f is recursive iff f is partial recursive, so (iii) $\Longrightarrow$ (i) follows from (b).

(b)

$\Longrightarrow$ Define

$$\varphi = \lambda x y \begin{cases} 1 & \text{if } \psi(x) = y \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then $(x,y) \in \psi$ iff $(x,y) \in \text{domain}(\varphi)$, so ψ is recursively enumerable.

$\Longleftarrow$ To compute $\psi(x)$, search for y with $(x,y) \in \psi$. □

Exercise (5-19)**.** Let R, S be k-ary relations. If $R \equiv S$, then $\tau^k[R] \equiv \tau^k[S]$.

Proof. If $R \equiv S$, there is a recursive bijection $g : \mathbb{N} \to \mathbb{N}$ with $g[R] = S$. Let $h : \mathbb{N} \to \mathbb{N}$ to be the 'conjugate' of g by τ^k, that is,

$$h = \lambda x\big[\tau^k(g(\pi_1^k(x)), \ldots, g(\pi_k^k(x)))\big]$$

Then h is a recursive bijection and $h[\tau^k[R]] = \tau^k[S]$, so $\tau^k[R] \equiv \tau^k[S]$. □

Remark. This shows that recursively invariant properties of sets can be extended to recursively invariant properties of relations using codings.

Exercise (5-20)**.** There exist k-ary relations R, S with $R \not\equiv S$ and $\tau^k[R] \equiv \tau^k[S]$.

Proof. Set $R = \{(0,0),(1,0)\}$ and $S = \{(0,0),(0,1)\}$. Then $R \not\equiv S$, but $\tau[R] = \{0,2\}$ and $\tau[S] = \{0,1\}$, so $\sigma = (12)$ is a recursive bijection with $\sigma[\tau[R]] = \tau[S]$, so $\tau[R] \equiv \tau[S]$. □

Remark. This shows that recursively invariant properties of relations may not be recursively invariant properties of their coding sets (cf. Exercise 5-19). The failure is due to the fact that isomorphism between relations is in general stronger than isomorphism between sets, requiring a 'uniformity' across coordinates not implied by isomorphism of sets.

Exercise (5-26)**.** It is not possible to recursively decide, given x with φ_x total and nondecreasing, whether range(φ_x) is finite.

Proof. By reduction of K. By the s-m-n theorem, choose recursive s with

$$\varphi_{s(x)} = \lambda y \begin{cases} t & \text{if } P_x(x) \text{ converges in exactly } t \leq y \text{ steps} \\ y & \text{otherwise} \end{cases}$$

Then $x \in K$ iff range$(\varphi_{s(x)})$ is finite. □

Exercise (5-27(a))**.** There is no recursive f such that for all x, if W_x is a nonempty recursive set, then $\varphi_{f(x)}$ recursively enumerates W_x either in increasing order, or in increasing order up to a point of constant value.

Proof. If so it is possible to recursively find, for any x where W_x is nonempty and recursive, a characteristic index for W_x, contradicting Exercise 5-29(a). □

Exercise (5-29)**.** (a) There is no partial recursive ψ such that for all x, if W_x is finite then $\psi(x)$ is a characteristic index of W_x.

(b) (i) There is a class of finite sets which is recursively enumerable but not characteristically enumerable.

(ii) There is a class of finite sets which is characteristically enumerable but not canonically enumerable.

(iii) The class of recursive sets is not characteristically enumerable.

Proof. (a) By the s-m-n theorem, choose recursive s with

$$\varphi_{s(x)} = \lambda y \begin{cases} \varphi_x(x) & \text{if } y = 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then for all x, $W_{s(x)} = \emptyset$ or $W_{s(x)} = \{0\}$. If ψ exists, then ψs is recursive, and $\varphi_{\psi(s(x))}$ is the characteristic function of $W_{s(x)}$ for all x. Therefore

$$f = \lambda x[\varphi_{\psi(s(x))}(0)]$$

is recursive. But f is the characteristic function of K—a contradiction.

(b) (i) For each $n \in \mathbb{N}$, define a coded 'initial segment' of K:

$$K_n = \{\langle x, 0\rangle \mid x \in K \text{ and } x \leq n\} \cup \{\langle n, 1\rangle\}$$

Set $\mathscr{C} = \{K_n \mid n \in \mathbb{N}\}$. Claim $\mathscr{C}$ is as desired.
Indeed, $\mathscr{C}$ is a set of finite sets, and $\mathscr{C}$ is recursively enumerable by the function which, on input n, returns (by an s-m-n function) an index of the following partial function whose domain is K_n:

$$\lambda z \begin{cases} \varphi_x(x) & \text{if } z = \langle x, 0\rangle \text{ with } x \leq n \\ 0 & \text{if } z = \langle n, 1\rangle \\ \text{undefined} & \text{otherwise} \end{cases}$$

But if $\mathscr{C}$ is characteristically enumerable, K is decidable: given x, go through the list and find a characteristic index for some K_n with $x \leq n$ (by construction, n can be recursively determined from a characteristic index for K_n). Then $x \in K$ iff $\langle x, 0\rangle \in K_n$.

(ii) For each $x \in \mathbb{N}$, define

$$T_x = \begin{cases} \{\langle t, 0\rangle, \langle x, 1\rangle\} & \text{if } P_x(x) \text{ converges in exactly } t \text{ steps} \\ \{\langle x, 1\rangle\} & \text{otherwise} \end{cases}$$

Set $\mathscr{C} = \{ T_x \mid x \in \mathbb{N}\}$. Then $\mathscr{C}$ is a set of finite sets and is characteristically enumerable by the function which, on input x, returns an index of the following characteristic function for T_x:

$$\lambda z \begin{cases} 1 & \text{if } z = \langle t, 0\rangle \text{ and } P_x(x) \text{ converges in exactly } t \text{ steps} \\ 1 & \text{if } z = \langle x, 1\rangle \\ 0 & \text{otherwise} \end{cases}$$

If $\mathscr{C}$ is canonically enumerable, K is decidable: given x, go through the list and find a canonical index for T_x (x can be recursively determined from a canonical index for T_x). Calculate the size of T_x (Theorem XV(a)). Then $x \in K$ iff $|T_x| = 2$.

(iii) By diagonalization. If the recursive sets are characteristically enumerable, it is possible to construct a recursive characteristic function for a set which differs from every recursive set—a contradiction. □

Remark. In this exercise, we used pairs to encode 'metadata' about sets in those sets.

Exercise (5-32)**.** If f is the single-valuizing function (Theorem XVI), then

$$\psi_z = \tau^{-1}[W_{f(z)}]$$

gives an acceptable numbering of the unary partial recursive functions.

Proof. To go from numbers to programs: set $\eta = \varphi_{f'(f(x))}$ with f' as in Corollary V(c), and given z construct a program which computes ψ_z as follows:

> Given input x, compute $\eta(0), \eta(1), \ldots$. If and when k, y are found with $\eta(k) = \langle x, y\rangle$, return y.

To go from programs to numbers: given x_0, construct an index z_0 for

$$\lambda x \begin{cases} 1 & \text{if } x = \langle y, z\rangle \text{ and } \varphi_{x_0}(y) = z \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then $W_{z_0} = \tau[\varphi_{x_0}]$. Since φ_{x_0} is a partial function, W_{z_0} is already single-valued, so $W_{f(z_0)} = W_{z_0}$ and

$$\psi_{z_0} = \tau^{-1}[W_{f(z_0)}] = \tau^{-1}[W_{z_0}] = \tau^{-1}[\tau[\varphi_{x_0}]] = \varphi_{x_0}$$ □

Chapter 6

Exercises

Definition. $K_0 = \{\langle x, y\rangle \mid x \in W_y\}$.

Definition. $A \leq_m B$ iff there exists recursive f such that $x \in A$ iff $f(x) \in B$. $A \equiv_m B$ iff $A \leq_m B$ and $B \leq_m A$.

Exercise (6-3)**.** $K_0 \leq_m K$, so K is complete for $\leq_m$ (among r.e. sets).

Proof. By the s-m-n theorem choose recursive s such that

$$\varphi_{s(z)} = \lambda w \begin{cases} 1 & \text{if } z = \langle x, y\rangle \text{ and } P_x(y) \text{ converges} \\ \text{undefined} & \text{otherwise} \end{cases}$$

Then

$$z \in K_0 \implies W_{s(z)} = \mathbb{N} \implies s(z) \in W_{s(z)} \implies s(z) \in K$$
$$z \notin K_0 \implies W_{s(z)} = \emptyset \implies s(z) \notin W_{s(z)} \implies s(z) \notin K$$

Therefore s witnesses $K_0 \leq_m K$. Since K_0 is complete, K is complete. □

Exercise (6-8)**.** A reducibility relation $\leq_r$ is recursively invariant iff $A \equiv B$ implies $A \equiv_r B$ for all A, B.

Proof.

$\implies$ If $A \equiv B$, then $A \leq_r A$ implies $A \leq_r B$ and $B \leq_r B$ implies $B \leq_r A$, so $A \equiv_r B$.

$\impliedby$ If $A \equiv C$, $B \equiv D$, and $A \leq_r B$, then $C \leq_r A \leq_r B \leq_r D$, so $C \leq_r D$.

□

References

[1] Cutland, N. J. *Computability: An introduction to recursive function theory.* New York: Cambridge, 1980.

[2] Rogers, H. *Theory of Recursive Functions and Effective Computability.* Cambridge: MIT, 1987.