Chapter 7

Recursive and Semi-Recursive Relations

(Rogers chapter 5, Cutland Chapters 6, 7.)

The *characteristic function* χ_R for a k-ary relation R
is a function $f : \mathbf{N}^k \to [0, 1]$ such that for all x,

$$R(x) \Rightarrow f(x) = 1 \ \wedge \ \neg R(x) \Rightarrow f(x) = 0$$

A *verifying function* for a relation R
is a partial function $f : \mathbf{N}^k \to [0, 1]$ such that

$$R(x) \iff \phi(x) \approx 1$$

A *refuting function* for a relation R
is a partial function $f : \mathbf{N}^k \to [0, 1]$ such that

$$\neg R(x) \iff \phi(x) \approx 1$$

A *computably decidable* or *recursive* relation
is a relation whose characteristic function is total recursive.

A *computably verifiable* or *semi-recursive* relation
is a relation with a partial recursive verifying function.

A *computably refutable* or *co-semi-recursive* relation
is a relation with a partial recursive refuting function.

The "computably decidable, verifiable, refutable" notation is not standard, but is much closer to the truth than the standard "recursive, semi-recursive" notation.

Examples

- computationally decidable: Gödel numbers of valid formulas in a propositional language.
- computationally verifiable: Gödel numbers of valid formulas in the language of arithmetic.
- computationally refutable: first-order consistency in the language of arithmetic.
- none of the above: the complete theory of arithmetic.

7.1 Arity and Clutter Reduction

From now on, we will drop the arity parameter when working with unary functions.

$$\phi_n = \phi_n^1$$

We can handle k-ary functions and relations by using a k-ary coding function.

$$\phi_n^k(\vec{x}) \approx \phi_m(\langle\vec{x}\rangle)$$

$$R(\vec{x}) \approx S(\langle\vec{x}\rangle)$$

Exercise 7.1 *Show that there is an effective conversion from index n to index m and conversely. That is, there exist total recursive f, g such that for all k-ary $\vec{x}$,*

$$\phi_n^k(\vec{x}) \approx \phi_{f(n)}(\langle\vec{x}\rangle)$$

$$\phi_{g(n)}^k(\vec{x}) \approx \phi_n(\langle\vec{x}\rangle)$$

Big hint: use the universal construction to set up the application of an index to a coded sequence of arguments like this:

$$\psi(i, \vec{x}) \approx ((\mu z)\ U(i, (z)_0, (z)_1, \langle\langle\vec{x}\rangle\rangle))_1$$

Apply the s-m-n *theorem and see what you get. For the other side, apply the universal construction in a manner that eliminates coding brackets instead of adding them. All the exercises will be like this. Get in the habit of casting for the "right" application of the universal construction and then applying* s-m-n *to the index position.*

7.2 Alternate Characterizations of Verifiability

The following results and concepts must become second nature to you. Here is some standard, but curious notation:

$$W_n = dom(\phi_n)$$

$$E_n = rng(\phi_n)$$

Proposition 7.1 *R is formally verifiable $\Longleftrightarrow$ there exists an n, such that for all x,*

$$R(\vec{x}) \Longleftrightarrow W_n(\langle\vec{x}\rangle)$$

Exercise 7.2 *Prove it. Hint: Use a μ operator to produce infinite loops in the right places and conversely, use sg to cut outputs higher than 1 down to 1.*

You know from set theory that

- A set is enumerable just in case there exists a bijection from **N** to the set (i.e., an enumeration).
- A set is denumerable just in case it is enumerable or finite. This is equivalent to the existence of a surjection from **N** to the set.

What if we "motorize" the concept of enumerability, requiring that the enumeration be a total recursive function? Then we arrive at the concept of a recursively enumerable set:

$$S \text{ is } \textit{recursively enumerable} \text{ (r.e.)} \Longleftrightarrow S = \emptyset$$
$$\vee\ \exists \text{ total recursive } f\ (S = rng(f))$$

The following result says that this amounts to the same thing as computable verifiability.

Proposition 7.2 *(The fundamental theorem of r.e. sets)*

$$S \textit{ is r.e.} \Longleftrightarrow \textit{there exists an } n \textit{ such that } S = W_n$$

Proof. Suppose that S is r.e. If $S = \emptyset$, then $S = dom(\emptyset)$. But $\emptyset =$ the function
defined by $(\mu z)(x \neq x)$, and hence is partial recursive. Now suppose that $S \neq \emptyset$ We must show that $S = rng(f)$, where f is total recursive. Define:

$$\psi(x) = (\mu z)(f(z) = x)$$

This is partial recursive since f is recursive. And $S = dom(\psi)$.

Conversely, suppose that $S = W_w = dom(\phi_n)$. If $S = \emptyset$, then we are done. So suppose $S \neq \emptyset$. So there is at least one k such that $S(k)$. Now we need to produce a total recursive enumeration of S. We can't assume that W_w is infinite, so perhaps at some point we run out of new things to enumerate. We must therefore continue to output some previously enumerated number until a new element of W_w is detected.

$$\begin{aligned} f(0) &= k \\ f(n+1) &= \begin{cases} (n)_2 & \text{if } \forall z \leq n\ ((n)_2 \neq f(z) \wedge U(w, (n)_0, (n)_1, \langle (n)_2 \rangle)) \\ f(n) & \text{otherwise.} \end{cases} \end{aligned}$$

This is a course of values recursion over partial recursive constructions and hence is partial recursive. $\dashv$

Corollary 7.3 *The sequence*

$$W_0, W_1, \ldots, W_n, \ldots$$

is an enumeration of the computably verifiable sets.

The corollary is central to the theory of computability. It provides a way of using our effective numbering of *Part* to number the verifiable sets without having to do it again from scratch. According to the following result, there is an equivalent, effectively intercompilable enumeration based on ranges instead of domains.

Proposition 7.4 *Effective conversions between domains and ranges.*

$$\exists \textit{ total recursive } f \ \forall n \ W_n = E_{f(n)}$$

$$\exists \textit{ total recursive } g \ \forall n \ W_{g(n)} = E_n$$

Exercise 7.3 *Prove it. Hint: use the* s-m-n *theorem over a universal dovetailing construction analogous to the one given in the preceding exercise. Good practice in doing things the elegant, formally valid way!*

Corollary 7.5 *The sequence*

$$E_0, E_1, \ldots, E_n, \ldots$$

is also an enumeration of the computably verifiable sets.

Recall that the primitive recursive relations are closed under bounded existential quantification. The computably verifiable relations are closed under full existential quantification over **N**. And every partial recursive relation results from an unbounded existential quantification over a (primitive) recursive relation.

Proposition 7.6 ***The Projection Theorem*** *S is computably verifiable $\iff$ $\exists$ elementary [r.e.] R such that for all x*

$$S(x) \iff \exists y \ R(x, y)$$

Proof. Let S be computably verifiable. By the fundamental theorem of r.e. sets, $\exists n(S = W_n)$. Thus, for all x,

$$\begin{array}{rcl} S(x) & \iff & W_n(x) \\ & \iff & \phi_n(x)\downarrow \\ & \iff & \exists z \ U(n, (z)_0, (z)_1, \langle x \rangle) \end{array}$$

Recall that U is elementary. Conversely, suppose

$$S(x) \iff \exists z\ R(z,x)$$

where R is r.e. So for some n, $R(\langle z,x\rangle) = W_n(\langle z,x\rangle)$. Define

$$\psi(x) \approx (\mu w)\ U(n,(w)_0,(w)_1,\langle (w)_2,x\rangle)$$

This is partial recursive, so for some m,

$$\phi_m = \psi$$

Now

$$\begin{aligned} W_m(x) &\iff \phi_m(x)\downarrow \\ &\iff \psi(x)\downarrow \\ &\iff (\mu w)\ U(n,(w)_0,(w)_1,\langle (w)_2,x\rangle)\downarrow \\ &\iff \exists w\ U(n,(w)_0,(w)_1,\langle (w)_2,x\rangle) \\ &\iff \exists z\ \phi_n(\langle z,x\rangle)\downarrow \\ &\iff \exists z\ W_n(\langle z,x\rangle) \\ &\iff \exists z\ R(z,x) \\ &\iff S(x) \end{aligned}$$

So S is r.e.$\dashv$

Corollary 7.7 *(Summary) The following are equivalent:*

S is computably verifiable

S is r.e.

$\exists n\ S = E_n$

$\exists n\ S = W_n$

$\exists$ *elementary (recursive, r.e.) R such that* $\forall x(S(x) \leftrightarrow \exists y\ R(x,y))$

7.3 The Halting Problem

What can we do with the W_n notation? Well, by Corollary 7.7, we have a nice, effective enumeration of the computationally verifiable sets. Thus, we can form a two-dimensional table of zeros and ones as follows:

$$T[n,m] = \chi_{W_n}(m)$$

Thus, each row of the table is the characteristic function of a computably verifiable set and the characteristic function of each computably verifiable set occurs

at least once in the table (infinitely often, actually).
Now define the set K whose characteristic function is the diagonal of the table:

$$\begin{aligned} \chi_k(n) &= T[n,n] \\ &= \chi_{W_n}(n) \end{aligned}$$

Then the characteristic function of $\overline{K}$ (or $\neg K$, the complement of K) is the counter-diagonal of the table:

$$\begin{aligned} \chi_{\overline{K}}(n) &= \overline{sg}(T[n,n]) \\ &= \overline{sg}(\chi_{W_n}(n)) \end{aligned}$$

Evidently, $\overline{K}$ is not a computably verifiable set, since its characteristic function is concocted to differ from each row of the table along the diagonal. Thus:

Proposition 7.8 $\overline{K}$ *is not r.e.*

Notice that the proof of proposition 7.8 is almost the same as Cantor's argument that the power set of **N** is not enumerable. The difference is that in Cantor's argument, we know that the diagonal's characteristic function yields a subset of **N** so the enumeration assumption is rejected. In this case, we know from the fundamental theorem of r.e. sets that the effective enumeration $W_0, W_1, W_2, \ldots$ exists, so the diagonal characteristic function yields a non-r.e. set.

Unwinding the definition of K we have:

$$\begin{aligned} K(n) &\iff \chi_{W_n}(n) = 1 \\ &\iff W_n(n) \\ &\iff n \in dom(\phi_n) \\ &\iff \phi_n(n) \downarrow \end{aligned}$$

Thus, K is the set of all indices that return an output or "halt" when given themselves as input. For this reason, K is called the halting problem.

7.4 Alternate Characterizations of the Recursive Sets

Decision intuitively requires that one be able to verify whether yes and verify whether no.

Proposition 7.9 *S is recursive $\iff$ $S, \overline{S}$ are both r.e.*

Exercise 7.4 *Prove it. Hint: One way is immediate using results proved above. The other requires a nice dovetailing construction, because you don't want to commit a suki by focusing on S forever before checking $\overline{S}$. But you can count on one or the other halting because each number is either in S or $\overline{S}$.*

Corollary 7.10 *K, $\overline{K}$ are not recursive.*

The next result is very interesting because it links the straightforwardly epistemological concept of decidability with the function theoretic concept of monotonicity. The idea is that you can use a monotone enumeration to decide the set: once you see a bigger thing than what you are looking for, you know you won't ever find what you are looking for. Conversely, if you can decide a set, then you can make sure that nothing smaller than x will have to be added to the enumeration before you add x. Compare this with the fundamental theorem for r.e. sets. In that case, you can't.

Proposition 7.11 *S is recursive $\iff$ S is finite*
$\wedge\ \exists$ monotone, total recursive f $(S = rng(f))$

Proof: Suppose S is recursive. If S is finite, we are done. So suppose that S is infinite. Define

$$\begin{aligned} f(0) &= (\mu x)\ (\chi_S(x) = 1) \\ f(n+1) &= (\mu x)\ (\chi_S(x) = 1 \wedge \forall_{z \leq n}(f(n) \neq x)) \end{aligned}$$

Conversely, suppose that S is finite. Then a case definition (using one case per element) yields a program for the characteristic function of S, so S is primitive recursive, and hence recursive.

Finally, suppose that S is the range of monotone, total recursive $f \ldots$. $\dashv$

Exercise 7.5 *Complete the proof. Hint: refer to the intuitive motivation above.*

7.5 Closure Laws

These will be very important for hierarchy theory.

Proposition 7.12

1. The r.e. sets form a Boolean algebra that is also closed under unbounded existential quantification.
2. The recursive sets form a Boolean algebra.
3. The recursive and r.e. relations are both closed under Cartesian product $\times$.

Exercise 7.6 *Prove it. Hint: For a Boolean algebra it suffices to show closure under $\wedge$, $\neg$. The conjunction case requires dovetailing in the r.e. case. And doesn't the $\exists$ case sound familiar?*

Exercise 7.7 *(optional) You already know how to show that the recursive sets are not closed under $\exists$ and the r.e. sets are not closed under $\forall$. Try showing these facts using only results on the table. Hint: express the halting problem and its complement using quantifiers over the Kleene predicate.*