Logic & Computability : Synopsis & Exercises
Session 10, April 17, 2002

4.6. Proposition. *A partial function f is computable if and only if the set $\Gamma_f = \{(x, y) \mid f(x)\downarrow$ and $f(x) = y\}$ is r.e.*

Proof. Suppose f is computable. We construct a partial computable g s.t. $\operatorname{dom} g = \Gamma_f$. To compute $g(x, y)$ start computing $f(x)$. As soon as the latter compuatation produces an answer, and the answer is in fact y, put $g(x, y) = 0$. Keep on looping idly if the answer is different from y.

Suppose Γ_f is r.e.: $\Gamma_f = \operatorname{dom} g$, where g is partial computable.. The algorithm to compute $f(x)$ starts computing $g(x, 0), g(x, 1), \ldots$ in parallel. As soon as one of these computations $g(x, y)$ converges, put $f(x) = y$. ∎

4.7. Proposition. *Let f be a total computable function.*
(a) *If X is decidable then $f^{-1}[X]$ is decidable and $f[X]$ is r.e.*
(b) *If X is r.e. then both $f^{-1}[X]$ and $f[X]$ are r.e.*

Proof. Let $X = \operatorname{dom} g$ with g partial computable. If $X = \varnothing$ then clearly both clauses hold trivially. So assume $X \neq \varnothing$ and hence by 4.3 there is a total computable h with $X = \operatorname{rng} h$.

(b). $f^{-1}[X] = \operatorname{dom} g \circ f$ and $f[X] = \operatorname{rng} f \circ h$.

(a). We have $x \in f^{-1}[X]$ iff $f(x) \in X$, so $f^{-1}[X]$ is decidable. The second part follows from (b). ∎

In Section 2 we have seen that each Turing machine M can be represented by a string σ containing information about its working alphabet, states, and the transition function. (We have also seen that one can restrict the size of the working alphabet to about six symbols and still be able to compute any function computed by a Turing machine working in any finite alphabet.) It is clearly decidable whether a string represents a valid description of a Turing machine. If we fix a computable bijection between natural numbers and strings, then expressions like 'the eth Turing machine' make sense. Furthermore, the function $e \mapsto$ (the string describing the eth Turing machine) is computable.

4.8. Definition. Let φ_e denote the (partial) function computed by the eth Turing machine. The number e is called the *index* of the function φ_e. We shall think of φ_e as 'the eth partial computable function'. In view of the above discussion, the function $U(e, x) = \varphi_e(x)$ is computable. It is often called the *universal* partial computable function because for each partial computable $f(x)$ there exists an e s.t. $f(x) = U(e, x)$ for all x.

The set $W_e = \operatorname{dom} \varphi_e$ is known as the eth r.e. set. The (r.e.) set $K = \{(x, y) \mid x \in W_y\} = \operatorname{dom} U$ is then the *universal* r.e. set. We also define the r.e. set $K_0 = \{x \mid x \in W_x\}$. e is the *index* of W_e.

4.9. Proposition. *K_0 is not decidable.*

Proof. Suppose it was. Then we could define a total computable function

$$f(x) = \begin{cases} \varphi_x(x) + 1 & \text{if } x \in K_0 \\ 0 & \text{otherwise} \end{cases}.$$

There must then exist a number e s.t. $f = \varphi_e$. Since f is total, $\varphi_e(e)$ is defined so $e \in K_0$.

However, $\varphi_e(e) = f(e) = \varphi_e(e) + 1$. The contradiction shows that K_0 is not decidable. ∎

Similarly to the above proof, one can show that there exists no total recursive function with the universality property for total recursive functions, i.e., no total recursive $f(x, y)$ s.t. for each total recursive $g(y)$ there is an e with $f(e, y) = g(y)$ for all y: One simply considers the function $g(y) = f(y, y) + 1$ and derives a similar contradiction.

4.B. m-reducibility and m-completeness

4.10. Definition. Let $f : \mathbb{N} \to \mathbb{N}$ be a total recursive function and X, Y sets of natural numbers. f is an *m-reduction* from X to Y, or equivalently, f *m-reduces* X to Y, written $f : X \leq_{\mathrm{m}} Y$ if $x \in X \Leftrightarrow f(x) \in Y$ for all x (equivalently, $X = f^{-1}[Y]$). Write $X \leq_{\mathrm{m}} Y$ if a reduction from X to Y exists. Write $X \equiv_{\mathrm{m}} Y$ if $X \leq_{\mathrm{m}} Y$ and $Y \leq_{\mathrm{m}} X$.

A *1-reduction* from X to Y is a reduction that is an injective (as well as total recursive) function $\mathbb{N} \to \mathbb{N}$. In this case we write $f : X \leq_1 Y$. The set X is *1-reducible* to Y, written $X \leq_1 Y$, if a 1-reduction from X to Y exists. $X \equiv_1 Y$ means $X \leq_1 Y \wedge Y \leq_1 X$.

Recall that $X \oplus Y = \{2x \mid x \in X\} \cup \{2x + 1 \mid x \in Y\}$.

4.11. Lemma. (a) *If $X \leq_1 Y$ then $X \leq_{\mathrm{m}} Y$.*
(b) *Both $\leq_{\mathrm{m}}$ and $\leq_1$ are transitive.*
(c) *$X, Y \leq_1 X \oplus Y$.*
(d) *If $X, Y \leq_{\mathrm{m}} Z$ then $X \oplus Y \leq_{\mathrm{m}} Z$.*
(e) *$X \leq_{\mathrm{m}} Y$ iff $\overline{X} \leq_{\mathrm{m}} \overline{Y}$. Similarly for $\leq_1$.*

Proof. (a) and (e) follow at once from the definitions. (b)–(d) are handled identically to Classroom Exercise 2 and Homework Exercise 3 from Session 3. ∎

4.12. Lemma. *Suppose $X \leq_{\mathrm{m}} Y$. If Y is r.e. then so is X. If Y is decidable then so is X.*

Proof. This is merely a restatement of 4.7. ∎

4.13. Examples. (a) *If X is decidable and $Y \notin \{\varnothing, \mathbb{N}\}$ then $X \leq_{\mathrm{m}} Y$.*
(b) $\overline{K}_0 \not\leq_{\mathrm{m}} K_0$.

Proof. (a). Since $Y \notin \{\varnothing, \mathbb{N}\}$, we may fix two numbers $a \in Y$ and $b \notin Y$. The function $f(x) = \begin{cases} a & \text{if } x \in X \\ b & \text{if } x \notin X \end{cases}$ is total computable.

(b). If $\overline{K_0} \leq_{\mathrm{m}} K_0$ then by 4.12 $\overline{K_0}$ is r.e. By 4.5 K_0 is then decidable. We know however that this is not the case by 4.9. ∎

Similarly to the enumeration $(\varphi_e)_{e \in \omega}$ of partial recursive functions in one variable, one constructs enumerations $(\varphi_e^{(n)})_{e \in \omega}$ of functions is n arguments. One may think of Turing machines that require n inputs, or alternatively, one views the input of a Turing machine as the code for some n-tuple of numbers using the (iterated, if necessary) pairing function like the one from Homework Exercise 1.

4.14. Theorem (The Parameter Theorem, aka the *s-m-n* Theorem). *For each m and n there is a total recursive function s_n^m s.t.* $\varphi^{(n)}_{s_n^m(x, y_1, \ldots, y_m)}(z_1, \ldots, z_n) =$

$\varphi_x^{(m+n)}(y_1, \ldots, y_m, z_1, \ldots, z_n)$ *for all* $x, \vec{y}, \vec{z}$.

Proof. Here is the algorithm computing $s_n^m(x, \vec{y})$: Given x and $\vec{y}$, construct the Turing machine M that, given $\vec{z}$, prefixes this tuple with $\vec{y}$ and then behaves like $\varphi_x^{(m+n)}$ on the resulting input $(\vec{y}, \vec{z})$. $s_n^m(x, \vec{y})$ is then the number of M in the standard enumeration of $(n+m)$-argument Turing machines. ∎

4.15. Theorem (The Fixed Point Theorem). *To every total recursive* $f(x, \vec{y})$ *there is a total recursive function* $n(\vec{y})$ *s.t.* $\varphi_{n(\vec{y})} = \varphi_{f(n(\vec{y}), \vec{y})}$. *In other words,* $n(\vec{y})$ *and* $f(n(\vec{y}), \vec{y})$ *index the same function.*

Proof. Consider the partial recursive function

$$g(x, \vec{y}, \vec{z}) = \begin{cases} \varphi_{\varphi_x(x,\vec{y})}(\vec{z}) & \text{if } \varphi_x(x, \vec{y})\downarrow \\ \uparrow & \text{oterwise} \end{cases}.$$

By the Parameter Theorem, there is a total recursive function $d(x, \vec{y})$ s.t. $\varphi_{d(x,y)}(\vec{z}) = g(x, \vec{y}, \vec{z})$ for all $x, \vec{y}, \vec{z}$. Let v be an index of $f(d(x, \vec{y}), \vec{y})$, that is, $\varphi_v(x, \vec{y}) = f(d(x, \vec{y}), \vec{y})$. We have $\varphi_{d(v,\vec{y})}(\vec{z}) = f(v, \vec{y}, \vec{z}) = \varphi_{\varphi_v(v,\vec{y})}(\vec{z}) = \varphi_{f(d(v,\vec{y}),\vec{y})}(\vec{z})$. The conclusion of the theorem is therefore satisfied by putting $n(\vec{y}) = d(v, \vec{y})$. ∎

4.16. Corollary. *To every total recursive* $f(x)$ *there is an index* n *s.t.* $\varphi_n = \varphi_{f(n)}$.

Proof. Since there are no parameters $\vec{y}$, $n = d(v)$ is just a number. ∎

4.17. Definition. An r.e. set X is *m-complete* if $Y \leq_{\mathrm{m}} X$ for any r.e. set Y. Similarly for 1- in place of m-.

The sets X and Y are *(recursively) isomorphic* if there is a total recursive permutation $f : \mathbb{N} \to \mathbb{N}$ s.t. $f : X \leq_{\mathrm{m}} Y$.

4.18. Lemma. (a) *Any 1-complete set is m-complete.*

(b) *Any m-complete set is undecidable.*

(c) K *and* K_0 *are m-complete.*

Proof. (a) follows at from 4.11(a).

(b). Let X be m-complete. Then $K_0 \leq_{\mathrm{m}} X$ since K_0 is r.e. If X were decidable then by 4.12 so would be K_0 which it is not.

(c). Let X be an r.e. set. Then $X = W_e$ for some e. Then the function $x \mapsto (e, x)$ reduces X to K. Thus K is m- (in fact, also 1-) complete.

Consider the partial recursive function $f(x, y) = \begin{cases} 0 & \text{if } x \in X \\ \uparrow & \text{otherwise} \end{cases}$. By the Parameter Theorem, fix a total recursive $h(x)$ s.t. $\varphi_{h(x)}(y) = f(x, y)$. Observe that we have $W_{h(x)} = \begin{cases} \omega & \text{if } x \in X \\ \varnothing & \text{otherwise} \end{cases}$. Therefore $x \in X \Leftrightarrow h(x) \in W_{h(x)} \Leftrightarrow h(x) \in K_0$. Thus K_0 is also m-complete. ∎

4.19. Proposition. *An r.e. set is m-complete if and only if it is 1-complete.*

Proof. The (if) part is 4.18(a). We focus on (only if). Let A be an m-complete set and let f be an m-reduction from K_0 to A.

Fix a coding of finite sets of natural numbers by natural numbers. (For example, let

$\sum_{i\in D} 2^i$ be the code of D.) This coding allows us to consider (partial) recursive functions taking finite sets (of natural numbers) as arguments.

C l a i m . There is a total recursive function g s.t. for any non-empty finite set D we have

$$\begin{aligned} D \subseteq A \quad &\Rightarrow \quad g(D) \in A - D, \quad \text{and} \\ D \subseteq \overline{A} \quad &\Rightarrow \quad g(D) \in \overline{A} - D. \end{aligned}$$

Define

$$W_{h(D)} = \begin{cases} f^{-1}[D] & \text{if } D \cap A = \varnothing \\ \omega & \text{if } D \cap A \neq \varnothing \end{cases}$$

with h total recursive. (Observe that both h and g view D as a s ingle input while $f^{-1}[D] = \{x \mid f(x) \in D\}$ treats D as a set.) In finer detail, we define a total recursive function h s.t.

$$\varphi_{h(D)} = q(D, x) = \begin{cases} 0 & \text{if } f(x) \in D \text{ or } D \cap A \neq \varnothing \\ \uparrow & \text{otherwise} \end{cases}.$$

Further, put

$$g(D) \begin{cases} = f(h(D)) & \text{if } f(h(D)) \notin D \\ \in A - D & \text{if } f(h(D)) \in D \end{cases}.$$

If $A = \operatorname{rng} k$ with k total recursive, then the second line is shorthand for $g(D) = k$(the minimal n s.t. $k(n) \notin D$). Observe that g is total and $g(D) \notin D$ in all cases. Suppose now that $D \neq \varnothing$.

If $D \subseteq A$ then $W_{h(D)} = \omega$, hence $h(D) \in W_{h(D)}$, so that $h(D) \in K_0$. This implies $f(h(D)) \in A$ because $f : K_0 \leq_{\mathrm{m}} A$, and hence $g(D) \in A$ by the definition of $g(D)$.

If $D \subseteq \overline{A}$ then $W_{h(D)} = f^{-1}[D]$. Suppose $h(D) \in f^{-1}[D] = W_{h(D)}$. Then $h(D) \in K_0$ so that $f(h(D)) \in A$. We would also have $f(h(D)) \in D$ implying $f(h(D)) \in A \cup D \neq \varnothing$. This contradicts the assumption $D \subseteq \overline{A}$. Thus $h(D) \notin f^{-1}[D] = W_{h(D)}$. Therefore $h(D) \notin K_0$ and hence $f(h(D)) \notin A$. On the other hand, $f(h(D)) \notin D$ implies $g(D) = f(h(D))$ by the definition of g. Together this entails $g(D) \notin A$ as required.

The proof of the Claim is thus complete.

Next, define the function t by putting $t(i, m) = g(\{m\} \cup \{t(j, m) \mid j < i\})$. By the Claim we have using induction on i that $m \in A \Leftrightarrow t(i, m) \in A$, and that $T(i, m) \neq t(j, m)$ whenever $i \neq j$.

Finally, let X be an r.e. set and d an m-reduction of X to A. We construct a 1-reduction $d' : X \leq_1 A$:

$$d'(x) = t(\text{the minimal } i \text{ s.t. } t(i, d(x)) \notin \{d(y) \mid y < x\}).$$

That this function is the required 1-reduction is easily seen from the properties of t that we have just stated. ∎

4.20. THEOREM. *If $X \equiv_1 Y$ then X and Y are isomorphic.*

PROOF. Suppose $f : X \leq_1 Y$ and $g : Y \leq_1 X$. We are going to define a recursive permutation of $\mathbb{N}$ by stages. At Stage i we have a partial 1–1 function h_i defined on i inputs, and our task is to extend it to a function h_{i+1} by defining it on a single additional input. At even stages we take care that $h(x)$ is defined for all x. At odd stages we take care that no y is left without pre-image. As is usual with back-and-forth constructions, it is sufficient to describe a single even stage of the construction, for the odd stages are entirely symmetric.

So let a be the minimal number such that $a \notin \operatorname{dom} h_i$. We are looking for a number b s.t. $h_i \cup \{(a,b)\}$ is 1–1 and $a \in X \Leftrightarrow b \in Y$. Consider $f(a)$ Since $f : X \leq_1 Y$, we have $a \in X \Leftrightarrow f(a) \in Y$. So if $f(a) \notin \operatorname{rng} h_i$, we can put $b = f(a)$. Otherwise, consider $f(h_i^{-1}(f(a)))$. We have $a \in X \Leftrightarrow f(a) \in Y \Leftrightarrow h_i^{-1}(f(a)) \in X \Leftrightarrow f(h_i^{-1}(f(a))) \in Y$. However, this number may again fall in $\operatorname{rng} h_i$. Generally, we choose the least n with $f \circ (h_i^{-1} \circ f)^n(a) \notin \operatorname{rng} h_i$ and put $b = f \circ (h_i^{-1} \circ f)^n(a)$. We need to see that such n exists. If there is no such n then, since $\operatorname{rng} h_i$ is finite, there are $k < j$ s.t. $f \circ (h_i^{-1} \circ f)^k(a) = f \circ (h_i^{-1} \circ f)^j(a)$. Let j be the minimal such. Since f is 1–1, $(h_i^{-1} \circ f)^k(a) = (h_i^{-1} \circ f)^j(a)$. We cannot have $k = 0$ for $a \notin \operatorname{dom} h_i$. Therefore since h_i^{-1} is also 1–1 we have $f \circ (h_i^{-1} \circ f)^{k-1}(a) = f \circ (h_i^{-1} \circ f)^{j-1}(a)$ which contradicts the minimality of j. Thus all $f \circ (h_i^{-1} \circ f)^n(a)$ are distinct for all n for which it is defined, and hence for some n this number is not in $\operatorname{rng} h_i$. This completes the proof of the theorem. ∎

4.21. Corollary. *All m-complete sets are pairwise isomorphic.*

Proof. Follows at once from the two preceding results. ∎

4.22. Definition. An r.e. set S is *simple* if $\overline{S}$ is infinite but contains no infinite r.e. subset.

4.23. Theorem. *No simple r.e. set can be either decidable or m-complete.*

Proof. Suppose S were decidable. Then $\overline{S}$ is also decidable and hence r.e. Thus S is not simple.

If S were m-complete then by 4.19 it would be 1-complete as well. Consider the r.e. set $K \oplus \varnothing$. Let $f : K \oplus \varnothing \leq_1 S$. We have $O = \{2x+1 \mid x \in \mathbb{N}\} \subseteq \overline{K \oplus \varnothing}$. Hence $f[O]$ is an infinite r.e. subset of $\overline{S}$ because f is 1–1. Thus S cannot be m-complete. ∎

4.24. Definition. Let $x \in \mathbb{N}$. Define $K(x)$ to be the minimal e s.t. $\varphi_e(0) = x$. $K(x)$ is the *Kolmogorov complexity* of x. Call x *random* if $x \leq K(x)$.

4.25. Theorem. *The set N of non-random numbers is simple.*

Proof. First, the set $N = \{x \mid K(x) < x\}$ is r.e.: $x \in N$ if and only if $x \in \{\varphi_0(0), \ldots, \varphi_{x-1}(0)\}$.

Let us next see that $\overline{N}$ is infinite: Fix n and let x be the minimal number not in $\{\varphi_0(0), \ldots, \varphi_n(0)\}$. Then $x \leq n+1 \leq K(x)$. Thus x is random so $x \notin N$. Therefore for an arbitrary n there are random numbers x with $k(x) > n$. Hence there are infinitely many random numbers.

Finaly, suppose W_j were an infinite r.e. set of random numbers. We have $W_j = \operatorname{rng} \varphi_k$ for some total φ_k. By the Parameter Theorem, find a total recursive function $t(s,m)$ s.t. $\varphi_s(m) = \varphi_{t(s,m)}(0)$ for all s, m. Clearly, $K(\varphi_s(m)) \leq t(s,m)$. Define

$$\varphi_{g(e)}(m) = \varphi_k(\text{the minimal } n \text{ s.t. } \varphi_k(n) > t(e,0)).$$

Since $\operatorname{rng} \varphi_k = W_e$ is infinite, g is total. By the Fixed Point Theorem, choose an e s.t. e and $g(e)$ are indices of the same function. We have $\varphi_e(0) = \varphi_k(\text{the minimal } n \text{ s.t. } \varphi_k(n) >$

$t(e, 0)) > t(e, 0) \geq K(\varphi_e(0))$ which implies that $\varphi_e(0)$ is non-random. But $\varphi_e(0) \in \operatorname{rng} \varphi_k = W_j$, and we have assumed that all elements of W_j are random. The contradiction proves that N is simple. ∎

4.26. Corollary. *There exist r.e. sets that are neither decidable nor m-complete.* ∎

4.C. Index Sets

4.27. Theorem (Rice's Theorem). *Let $\mathcal{C}$ be a class of partial recursive functions. Consider $C = \{e \mid \varphi_e \in \mathcal{C}\}$, the index set*[30] *of $\mathcal{C}$. Then C is undecidable unless $\mathcal{C}$ is the empty class or the class of all partial recursive functions. In fact, one has $K \leq_{\mathrm{m}} C$ or $\overline{K} \leq_{\mathrm{m}} C$.*

Proof. Without loss of generality we may assume that ε, the nowhere defined function, is an element of $\mathcal{C}$ (otherwise replace $\mathcal{C}$ by $\overline{\mathcal{C}}$). Let $\varphi_a \in \overline{\mathcal{C}}$. Define

$$\varphi_{f(x)} = \begin{cases} \varphi_a & \text{if } x \in K_0 \\ \uparrow & \text{otherwise} \end{cases}.$$

Then $x \in K_0$ implies $\varphi_{f(x)} = \varphi_a \in \mathcal{C}$ implies $f(x) \in C$. Similarly, $x \notin K_0$ implies $\varphi_{f(x)} = \varepsilon \notin \mathcal{C}$ implies $f(x) \notin C$. Therefore $f : K \leq_{\mathrm{m}} C$, thus C is undecidable. ∎

4.28. Example. $K_0, \overline{K_0} \leq_{\mathrm{m}} \mathrm{FIN}$, *where* $\mathrm{FIN} = \{e \mid W_e$ *is finite*$\}$.

Proof. Put

$$\varphi_{f(x)}(y) = \begin{cases} 0 & \text{if } x \in K_0 \\ \uparrow & \text{if } x \notin K_0 \end{cases}.$$

Then $W_f(x) = \omega$ if $x \in K_0$ and $W_f(x) = \varnothing$ otherwise. Thus $f : K_0 \leq_{\mathrm{m}} \mathrm{FIN}$.

Put

$$\varphi_{f(x)}(y) = \begin{cases} \uparrow & \text{if } x \in K_0 \text{ is established in } \leq y \text{ steps} \\ 0 & \text{if } x \in K_0 \text{ is not established in } \leq y \text{ steps} \end{cases}.$$

Then $W_{f(x)} = \omega$ if $x \notin K_0$, and $W_{f(x)} = \{0, \ldots, n\}$ if $x \in K_0$, where $\varphi_x(x)$ converges in $n+1$ steps. Thus $f : \overline{K_0} \leq_{\mathrm{m}} \mathrm{FIN}$. ∎

Classroom Exercises

1. Let f be a partial computable function and X a decidable set. Need $f^{-1}[X] = \{x \mid f(x)$ is defined and $f(x) \in X\}$ be decidable?
2. Let f be a one-to-one partial computable function. Show that f^{-1} is computable.
3. Suppose that for all but finitely many x we have $x \in A \Leftrightarrow x \in B$. Show that if A is decidable then so is B. Show that if A is r.e. then so is B.
4. Construct a non-decidable set A s.t. $A \leq_{\mathrm{m}} \overline{A}$ and $\overline{A} \leq_{\mathrm{m}} A$.
5. Let X nad Y be simple r.e. sets. Show that $X \cap Y$ is also simple.

[30] In particular, once e and d are indices of the same partial recursive function then they are either both in C, or both outside C.

Homework Exercises

1. Show that the function $f(x, y) = \frac{1}{2}(x + y + 1)(x + y) + x$ is a computable bijection between $\mathbb{N}$ and $\mathbb{N}^2$.
2. Construct an example of a set X s.t. neither X nor $\overline{X}$ are r.e.
3. An r.e. set is finite iff its every r.e. subset is decidable.
4. Let $\mathrm{INF} = \{e \mid W_e \text{ is infinite}\}$ and $\mathrm{TOT} = \{e \mid W_e = \omega\}$. Show $\mathrm{INF} \leq_{\mathrm{m}} \mathrm{TOT}$ and $\mathrm{TOT} \leq_{\mathrm{m}} \mathrm{INF}$.
5. Suppose the theory T in a finite language is axiomatized by an r.e. collection of axioms. Show that if T is complete then (the set of theorems of) T is decidable.
6. Let f be a total recursive function. Show that there are infinitely many n s.t. $\varphi_n = \varphi_{f(n)}$.

Extra Exercises

1. The group of computable permutations of ω is not finitely generated.
2. The set of theorems of predicate calculus in the language consisting of (equality and) a single unary predicate symbol is decidable.