Computability and Randomness

Andre Nies

March 22, 2022

Contents

1 The complexity of sets

1.1 The basic concepts

1.1.1 Partial computable functions

Given expression α, β,

$$\alpha \simeq \beta$$

means that either both expressions are undefined, or they are defined with the same value

The function $\Xi(e,x) \simeq \Phi_e(x)$. A Turing program computing Ξ is called a **universal Turing program**

Theorem 1.1 (Parameter Theorem)**.** *For each partial computable function Θ in two variables there is a computable strictly increasing function q s.t.*

$$\forall e \forall x \Phi_{q(e)}(x) \simeq \Theta(e,x)$$

An index for q can be obtained effectively from an index for Θ

Lemma 1.2 (Padding Lemma)**.** *For each e and each m, one may effectively obtain $e' > m$ s.t. the Turing program $P_{e'}$ behaves exactly like P_e*

Theorem 1.3 (Recursion Theorem)**.** *Let $g : \mathbb{N} \to \mathbb{N}$ be computable. Then there is an e s.t. $\Phi_{g(e)} = \Phi_e$. We say that e is a* ***fixed point*** *for g*

Proof. There is q s.t. $\Phi_{q(e)}(x) \simeq \Phi_{g(\Phi_e(e))}(x)$ for all e, x. Choose an i s.t. $q = \Phi_i$, then

$$\Phi_{q(i)} = \Phi_{\Phi_i(i)} = \Phi_{g(\Phi_i(i))}$$

□

Theorem 1.4 (Recursion Theorem with Parameters)**.** *Let $g : \mathbb{N}^2 \to \mathbb{N}$ be computable. Then there is a computable function f, which can be obtained effectively from g, s.t. $\Phi_{g(f(n),n)} = \Phi_{f(n)}$ for each n*

Proof. There is g_n s.t. $g_n(f(n)) = g(f(n), n)$. Then let $f(n)$ be the fixed point of $\Phi_{g'(x)}$ □

Exercise 1.1.1. Extend the Recursion Theorem by showing that computable function g has infinitely many fixed points. Conclude that the function f in Theorem 1.4 can be chosen one-one

Proof. There is infinite many i s.t. $q = \Phi_i$ □

1.1.2 Computably enumerable sets

Definition 1.5. $A \subseteq \mathbb{N}$ is **computably enumerable** (**c.e.**) if A is the domain of some partial computable function

Let

$$W_e = \text{dom}(\Phi_e)$$

Then $(W_e)_{e \in \mathbb{N}}$ is an effective listing of all c.e. sets. A sequence of sets $(S_e)_{e \in \mathbb{N}}$ s.t. $\{\langle e, x \rangle : x \in S_e\}$ is c.e. is called **uniformly computably enumerable**

A is called **computable** if its characteristic function is computable; otherwise A is called **incomputable**

Proposition 1.6. *A is computable $\Leftrightarrow$ A and $\mathbb{N} - A$ are c.e.*

We may obtain a c.e. incomputable set denoted $\emptyset'$ by a direct diagonalization. We define $\emptyset'$ in such a way that $\mathbb{N} - \emptyset'$ differs from W_e at e: let

$$\emptyset' = \{e : e \in W_e\}$$

The set $\emptyset'$ is called the **halting problem**, since $e \in \emptyset'$ iff program P_e^1 halts on input e

Proposition 1.7. *The set $\emptyset'$ is c.e. but not computable*

Proof. $\emptyset'$ is c.e. since $\emptyset' = \mathrm{dom}(J)$, where J is the partial computable function given by $J(e) \simeq \Phi_e(e)$. If $\emptyset'$ is computable then there is e s.t. $\mathbb{N} - \emptyset' = W_e$. Then $e \in \emptyset' \leftrightarrow e \in W_e \leftrightarrow e \in \emptyset'$, a contradiction ☐

The sequence $(W_e)_{e\in\mathbb{N}}$ is universal for uniformly c.e. sequences

Corollary 1.8. *For each uniformly c.e. sequence $(A_e)_{e\in\mathbb{N}}$ there is a computable function q s.t. $A_e = W_{q(e)}$ for each e*

Proof. Define the partial computable function Θ by $\Theta(e, x) \simeq 0$ iff $x \in A_e$, and $\Theta(e, x)$ is undefined otherwise. Then the function q obtained by the Parameter Theorem is as required. ☐

Exercise 1.1.2. Suppose $(\hat{W}_e)_{e\in\mathbb{N}}$ is a further universal uniformly c.e. sequence. Assume that $(\hat{W}_e)_{e\in\mathbb{N}}$ also has the padding property, one may effectively obtain $e' > m$ s.t. $\hat{W}'_e = \hat{W}_e$. Show that there is a computable permutation π of $\mathbb{N}$ s.t. $\hat{W}_e = W_{\pi(e)}$ for each e

Proof. there is q s.t. $W_{q(e)} = \hat{W}_e$, there is p s.t. $\hat{W}_{p(e)} = W_e$ Find

1. $q'(e) > e$
2. q' is 1-1
3. $W_{q'(e)} = \hat{W}_e$

padding $\pi(m, e) > m$. $W_{\pi(e,q(e))} = W_{q(e)}$
similar to cantor-bernstein
or back-and-forth ☐

1.1.3 Indices and approximations

Definition 1.9. We write
$$\Phi_{e,s}(x) = y$$
if $e, x, y < s$ and the computation of program P_e on input x yields y in at most s computation steps. Let $W_{e,s} = \mathrm{dom}(\Phi_{e,s})$

At stage s we have complete information about $\Phi_{e,s}$ and $W_{e,s}$. To state this more formally, we need to specify an effective listing $D_0, D_1, ...$ of the finite subsets of $\mathbb{N}$

Definition 1.10. Let $D_0 = \emptyset$. If $n > 0$ has the form $2^{x_1} + \dots + 2^{x_r}$, where $x_1 < \dots < x_r$, then let $D_n = \{x_1, \dots, x_r\}$. We say that n is a **strong index** for D_n. For instance, $D_5 = \{0, 2\}$

There is a computable function f s.t. $f(e, s)$ is a strong index for $W_{e,s}$. We think of a computable enumeration of a set A as an effective listing $a_0, a_1, \dots$ of the elements of A in some order. To include the case that A is finite, we formalize this via an effective union of finite sets (A_s). We view A_s as the set of elements enumerated by the end of stage s. At certain stages we may decide not to enumerate any element

Definition 1.11. A **computable enumeration** of a set A is an effective sequence $(A_s)_{s\in\mathbb{N}}$ of (strong indices for) finite sets s.t. $A_s \subseteq A_{s+1}$ for each s and $A = \bigcup_s A_s$

Each c.e. set W_e has the computable enumeration $(W_{e,s})_{s\in\mathbb{N}}$. Conversely, if A has a computable enumeration then A is c.e., for $A = \mathrm{dom}(\Phi)$ where Φ is the partial computable function given by the following procedure: at stage s we let $\Phi(x) = 0$ if $x \in A_s$. An **index for a c.e. set** A is a number e s.t. $A = W_e$

Proposition 1.12. *For each computable function Φ, $\mathrm{ran}(\Phi)$ is c.e.*

Proof. Suppose $\Phi = \Phi_e$ and we enumerate $A = \mathrm{ran}(\Phi)$. Since we have complete information about Φ_s at stage s, we can compute from s a strong index for $A_s = \mathrm{ran}(\Phi_s)$. Then $(A_s)_{s\in\mathbb{N}}$ is the required computable enumeration of A $\square$

Exercise 1.1.3.

Exercise 1.1.4.

Proof. find a subsequence with increasing required steps $\square$

1.2 Relative computational complexity of sets

Definition 1.13. X is **many-one reducible** to Y, denoted $X \leq_m Y$, if there is a computable function f s.t. $n \in X \leftrightarrow f(n) \in Y$ for all n

If X is computable, $Y \neq \emptyset$, and $Y \neq \mathbb{N}$, then $X \leq_m Y$: choose $y_0 \in Y$ and $y_1 \notin Y$. Let $f(n) = y_0$ if $n \in X$ and $f(n) = y_1$ otherwise. Then $X \leq_m Y$ via f.

For each set Y the class $\{X : X \leq_m Y\}$ is countable. In particular, there is no greatest many-one degree

Proposition 1.14. *A is c.e. $\Leftrightarrow A \leq_m \emptyset'$*

An index for the many-one reduction as a computable function can be obtained effectively from a c.e. index for A, and conversely

Proof. $\Rightarrow$: We claim that there is a computable function g s.t.

$$W_{g(e,n)} = \begin{cases} \{e\} & n \in A \\ \emptyset \end{cases}$$

For let $\Theta(e, n, x)$ converge if $x = e$ and $n \in A$. Then there is a computable function g s.t. $\forall e, n, x[\Theta(e, n, x) \simeq \Phi_{g(e,n)}(x)]$. By Theorem 1.4, there is a computable function h s.t. $W_{g(h(n),n)} = W_{h(n)}$ for each n. Then

$$n \in A \Rightarrow W_{h(n)} = \{h(n)\} \Rightarrow h(n) \in \emptyset'$$
$$n \notin A \Rightarrow W_{h(n)} = \emptyset \Rightarrow h(n) \notin \emptyset$$

$\Leftarrow$: If $A \leq_m \emptyset'$ via h, then $A = \mathrm{dom}(\Psi)$ where $\Psi(x) \simeq J(h(x))$ (recall that $J(e) \simeq \Phi_e(e)$) ☐

Definition 1.15. A c.e. set C is called r**-complete** if $A \leq_r C$ for each c.e. set A

we say that $X \leq_1 Y$ if $X \leq_m Y$ via a one-one function f

Exercise 1.2.1. The set $\emptyset'$ is 1-complete

Exercise 1.2.2. $X \equiv_1 Y \Leftrightarrow$ there is a computable permutation p of $\mathbb{N}$ s.t. $Y = p(X)$

Our intuitive understanding of "Y is at least as complex as X" is: X can be computed with the help of Y. To formalize more general ways of relative computation, we extend the machine model by a one-way infinite "oracle" tape which holds all the answers to oracle questions of the form "is k in Y".

We write $\Phi_e^Y(n)\downarrow$ if the program P_e halts when the oracle is Y and the input is n; we write $\Phi_e(Y; n)$ or $\Phi_e^Y(n)$ for this output. The Φ_e are called **Turing functionals**. And we let $W_e^Y = \mathrm{dom}(\Phi_e^Y)$. W_e is a **c.e. operator**

Definition 1.16. A total function $f : \mathbb{N} \to \mathbb{N}$ is called **Turing reducible** to Y, or **computable relative to** Y, or **computable in** Y, if there is an e s.t. $f = \Phi_e^Y$. We denote this by $f \leq_T Y$. We also say that Y **computes** f. For a set A, we write $A \leq_T Y$ if the characteristic function of A is Turing reducible to Y

For a total functions g, $f \leq_T g$ means that f is Turing reducible to the **graph** of g, that is, to $\{\langle n, g(n)\rangle : n \in \mathbb{N}\}$

Exercise 1.2.3. $\leq_m$ and $\leq_T$ are preorderings of the subsets of $\mathbb{N}$

A set A is **c.e. relative to** Y if $A = W_e^Y$ for some e. We view $\Phi_e$as $\Phi_e^{\emptyset}$

Proposition 1.17. *A is computable in $Y \Leftrightarrow A$ and $\mathbb{N} - A$ are c.e. in Y*

Definition 1.18. We write $J^Y(e) \simeq \Phi_e^Y(e)$. The set $Y' = \text{dom}(J^Y)$ is the **Turing jump** of Y. The map $Y \to Y'$ is called the **jump operator**

Theorem 1.19. *For each computable binary function g there is a computable function f s.t. $\Phi^Y_{g(f(n),n)} = \Phi^Y_{f(n)}$*

Proposition 1.20. *A is c.e. in Y iff $A \leq_m Y'$*

Proposition 1.21. *For each Y, the set Y' is c.e. relative to Y. Also, $Y \leq_m Y'$ and $Y' \not\leq_T Y$, and therefore $Y <_T Y'$*

Proof. Y' is c.e. in Y since $Y' = \text{dom}(J^Y)$. As Y is c.e. relative to itself, by Proposition 1.20 $Y \leq_m Y'$. If $Y' \leq_T T$ then there is e s.t. $\mathbb{N} - Y' = W_e^Y$. Then $e \in Y' \leftrightarrow e \in W_e^Y \leftrightarrow e \notin Y'$ □

Definition 1.22. We define $Y^{(n)}$ inductively by $Y^{(0)} = Y$ and $Y^{(n+1)} = (Y^{(n)})'$.

Proposition 1.23. *For each Y, Z we have $Y \leq_T Z \Leftrightarrow Y' \leq_m Z'$*

Proof. $\Rightarrow$: Y' is c.e. in Y and hence c.e. in Z. Therefore $Y' \leq_m Z'$ by Proposition 1.20

$\Leftarrow$:By Proposition 1.17, Y and $\mathbb{N}-Y$ are c.e. in Y. So $Y, \mathbb{N}-Y \leq_m Y' \leq_m Z'$, whence both Y and $\mathbb{N} - Y$ are c.e. in Z. Hence $Y \leq_T Z$ □

Fact 1.24. *From a Turing functional $\Phi = \Phi_e$ one can effectively obtain a computable strictly increasing function p, called a **reduction function** for Φ, s.t. $\forall Y \forall x \Phi^Y(x) \simeq J^Y(p(x))$*

Proof. Let $\Theta^Y(x,y) \simeq \Phi^Y(x)$, by the oracle version of the Parameter Theorem there is a computable strictly increasing function p s.t. $\forall Y \forall y \Phi^Y_{p(x)}(y) \simeq \Theta^Y(x,y) \simeq \Phi^Y(x)$. Letting $y = p(x)$ we obtain $J^Y(p(x)) = \Phi^Y_{p(x)}(p(x)) = \Phi^Y(x)$ □

We identify $\sigma \in \{0,1\}^*$ with $n \in \mathbb{N}$ s.t. the binary representation of $n+1$ is 1σ. For instance, 000 is 7

Definition 1.25. We write $\Phi^Y_{e,s}(x) = y$ if $e, x, y < s$ and the computation of program P_e on input x yields y in at most s computation steps, with all oracle queries less than s.

The **use principle** is the fact that a terminating oracle computation only asks finitely many oracle questions. Hence $(\Phi^Y_{e,s})_{s\in\mathbb{N}}$ approximates Φ^Y_e

Definition 1.26. The **use** of $\Phi^Y_e(x)$, denoted $\mathsf{use}\,\Phi^Y_e(x)$, is defined if $\Phi^Y_e(x)\downarrow$, where its value is 1+the largest oracle query asked during this computation.

We write

$$\Phi^\sigma_e(x) = y$$

if $\Phi^F_e(x)$ yields the output y, where $F = \{i < |\sigma| : \sigma(i) = 1\}$, and the use is at most $|\sigma|$. Then for each set Y

$$\Phi^Y_e(x) = y \leftrightarrow \Phi^{Y\upharpoonright u}_e(x) = y$$

where $u = \mathsf{use}\,\Phi^Y_e(x)$

If a Turing functional Φ_e is given then $\lambda Yx.\,\mathsf{use}\,\Phi^Y_e$ is also a Turing functional (namely there is i s.t. $\Phi^Y_i(x) \simeq \mathsf{use}\,\Phi^Y_e(x)$ for each Y and x). Thus if Y is an oracle s.t. $f = \Phi^Y_e$ is total, the function $\mathsf{use}\,\Phi^Y_e$ is computable in Y.

Definition 1.27. A function $f : \mathbb{N} \to \mathbb{N}$ is **weak truth-table** reducible to Y, denoted $f \le_{wtt} Y$, if there is a Turing functional Φ_e and a computable bound r s.t. $f = \Phi^Y_e$ and $\forall n\,\mathsf{use}\,\Phi^Y_e(n) \le r(n)$.

Definition 1.28. A function $f : \mathbb{N} \to \mathbb{N}$ is **truth-table** reducible to Y, denoted $f \le_{tt} Y$, if there is a Turing functional Φ_e and a computable bound r s.t. $f = \Phi^Y_e$ $f = \Phi^Y_e$ and Φ^Z_e is total for each oracle Z (we call such a Φ_e a truth table reduction).

Proposition 1.29. *1. $X \le_{tt} Y \Leftrightarrow$ there is a computable function g s.t. for each n,*

$$n \in X \Leftrightarrow \bigvee_{\sigma\in D_{g(n)}} [\sigma \preceq Y]$$

2. $X \le_{tt} Y$ implies $X \le_{wtt} Y$

Proof. 1. $\Rightarrow$: Suppose $X \le_{tt} Y$ via a truth-table reduction $\Phi = \Phi_e$. The tree $T_n = \{\sigma : \Phi^\sigma_{|\sigma|}(n)\uparrow\}$ is finite for each n, for otherwise it has an infinite path Z by Kőnig's Lemma and $\Phi^Z(n)\uparrow$. Given n one can compute a strong index $\tilde{g}(n)$ for the finite set of minimal string σ s.t. $\Phi^\sigma_{|\sigma|}(n)\downarrow$. Hence one can compute a strong index $g(n)$ for the set of all minimal strings σ s.t. $\Phi^\sigma_{|\sigma|}(n)\downarrow= 1$. Then $D_{g(n)}$ is as required

$\Leftarrow$:Consider the following procedure relative to an oracle Z: on input n, first compute $D_{g(n)}$. If $\sigma \preceq Z$ for some $\sigma \in D_{g(n)}$, output 1, otherwise output 0

2. For each Z use $\Phi_e^Z(n)$ is bounded by $\max\{|\sigma| : \sigma \in D_{g(n)}\}$

$\square$

Proposition 1.30. *$f \leq_{tt} A \Leftrightarrow$ there is a Turing functional Φ and a computable function t s.t. $f = \Phi^A$ and the number of steps needed to compute $\Phi^A(n)$ is bounded by $t(n)$*

Proof. $\Leftarrow$: Let $\tilde{\Phi}$ be the Turing functional s.t. $\tilde{\Phi}^Z(n) = \Phi^Z_{t(n)}(n)$ if the latter is defined and $\tilde{\Phi}^Z(n) = 0$ otherwise $\square$

Definition 1.31. The **effective disjoint union** of sets A and B is

$$A \oplus B = \{2n : n \in A\} \cup \{2n+1 : n \in B\}$$

Exercise 1.2.4.

1. $A, B \leq_m A \oplus B$
2. Let $\leq_r$ be one of the reducibilities above. Then for any set X

$$A, B \leq_r X \Leftrightarrow A \oplus B \leq_r X$$

Exercise 1.2.5. Let $C = A_0 \cup A_1$ where A_0, A_1 are c.e. and $A_0 \cap A_1 = \emptyset$. Then $C \equiv_{wtt} A_0 \oplus A_1$

Proof. Since A_0, A_1 are c.e., for each $n \in C$, we can determine if $n \in A_0 \cup A_1$ in finite steps. $\square$

Exercise 1.2.6. Show that $\exists Z f \leq_{tt} Z \Leftrightarrow$ there is a computable h s.t. $\forall n f(n) \leq h(n)$

Proof. Trivial $\square$

1.3 Descriptive complexity of sets

In a computable enumeration $(Z_s)_{s\in\mathbb{N}}$ of a set Z, for each x, $Z_s(x)$ can change at most once, namely from 0 to 1. Which sets Z are described if we allow an arbitrary finite number of changes

Definition 1.32. We say that a set Z is Δ^0_2 if there is a computable sequence of strong indices $(Z_s)_{s\in\mathbb{N}}$ s.t. $Z_s \subseteq [0, s)$ and $Z(x) = \lim_s Z_s(x)$. We say that $(Z_s)_{s\in\mathbb{N}}$ is a **computable approximation** of Z

Given an expression E that is approximated during stages s,

$$E[s]$$

denotes its value at the **end of** stage s. For instance, given a Δ^0_2 set Z with a computable approximation, instead of $\Phi^{Z_s}_{e,s}(x)$ we simply write $\Phi^Z_e(x)[s]$. We say that the expression E is **stable at** s if $E[t] = E[s]$for all $t \geq s$

Lemma 1.33 (Shoenfield Limit Lemma)**.** *Z is $\Delta^0_2 \Leftrightarrow Z \leq_T \emptyset'$. The equivalence is uniform*

Proof. $\Leftarrow$: Fix a Turing functional Φ_e s.t. $Z = \Phi^{\emptyset'}_e$. Since $\emptyset'$ is c.e., let $\langle \emptyset'_s \rangle_{s \in \mathbb{N}}$ be a computable enumeration of $\emptyset'$. Define

$$Z_s(x) = \begin{cases} 1 & \Phi^{\emptyset'_s \downarrow}_{e,s} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Let $u = \text{use}\, \Phi^{\emptyset'}_e(x)$, there is s s.t. $\emptyset'_s \upharpoonright u = \emptyset' \upharpoonright u$, thus there is $s \geq t$ s.t. $Z_s(x) = Z(x)$

Then the required approximation is given by $Z_s = \{x < s : \Phi^{\emptyset'}_e(x)[s] = 1\}$

$\Rightarrow$: We define a c.e. set C s.t. $Z \leq_T C$. This is sufficient because $C \leq_m \emptyset'$ by Proposition 1.14. The set C is called the **change set** because it records the changes of the computable approximation. If $Z_s(x) \neq Z_{s+1}(x)$ we put $\langle x, i \rangle$ into C_{s+1}, where i is the least s.t. $\langle x, i \rangle \notin C_s$. To show that $Z \leq_T C$, on input x, using the oracle C compute the least i s.t. $\langle x, i \rangle \notin C$. If i is even then $Z(y) = Z_0(y)$, otherwise $Z(y) = 1 - Z_0(y)$

We have obtained C and the Turing reduction of Z to C effectively from the computable approximation of Z. Proposition 1.14 is also effective $\square$

If $Z = \Phi^{\emptyset'}_e$ we say that e is a Δ^0_2**-index** for Z. A number e is a Δ^0_2 index only if $\Phi^{\emptyset'}_e$ is total

Definition 1.34.

1. We say that a set Z is ω**-c.e.** if there is a computable approximation $(Z_s)_{s \in \mathbb{N}}$ of Z and a computable function b s.t.

$$b(x) \geq \#\{s > x : Z_x(x) \neq Z_{s-1}(x)\} \text{ for each } x$$

2. If $Z_s(s-1) = 0$ for each $s > 0$ and $b(x)$ can be chosen constant of value n, then we say Z is n-c.e.

Thus Z is 1-c.e. iff Z is c.e., and Z is 2-c.e. iff $Z = A - B$ for c.e. sets A, B

Proposition 1.35. *Z is ω-c.e. $\Leftrightarrow Z \leq_{wtt} \emptyset' \Leftrightarrow Z \leq_{tt} \emptyset'$*
The equivalence are effective

Corollary 1.36. *A is $\Delta^0_2 \Leftrightarrow A$ is both Σ^0_2 and Π^0_2*

Proof.

$$\begin{aligned} A \in \Delta^0_2 &\Leftrightarrow A \leq_T \emptyset' \\ &\Leftrightarrow A \text{ and } \mathbb{N} - A \text{ are c.e. in } \emptyset' \\ &\Leftrightarrow A \in \Sigma^0_2 \cap \Pi^0_2 \end{aligned}$$

last iff from Theorem 1.39 □

Definition 1.37. Let $A \subseteq \mathbb{N}$ and $n \geq 1$

1. A is Σ^0_nif $x \in A \leftrightarrow \exists y_1 \forall y_2 \dots Q y_n R(x, y_1, \dots, y_n)$, where R is a symbol for a computable relation
2. A is Π^0_n if $\mathbb{N} - A$ is Σ^0_n
3. A is **arithmetical** if A is Σ^0_n for some n

Fact 1.38. *A is $\Sigma^0_1 \Leftrightarrow A$ is c.e.. The equivalence is uniform*

Proof. $\Rightarrow$: Suppose $x \in A \leftrightarrow \exists y R(x, y)$ for computable R. Let Φ be the partial computable function given by the Turing program that on input x looks for a witness y s.t. $R(x, y)$, and halts when such a witness is found. Then $A = \mathrm{dom}(\Phi)$

$\Leftarrow$: Suppose $A = \mathrm{dom}(\Phi)$ for a partial computable function Φ. Let R be the computable relation given by $R(x, s) \leftrightarrow \Phi(x)[s] \downarrow$. Then $x \in A \leftrightarrow \exists s R(x, s)$, so A is Σ^0_1 □

A Σ^0_n set C is Σ^0_n**-complete** if $A \leq_m C$ for each Σ^0_n set A

Theorem 1.39. *Let $n \geq 1$*

1. *A is $\Sigma^0_n \Leftrightarrow A$ is c.e. relative to $\emptyset^{(n-1)}$*
2. *$\emptyset^{(n)}$ is Σ^0_n-complete*

Proof. Induction on n. 1.38 and 1.14. Now let $n > 1$

1. First suppose A is Σ^0_n for some computable relation R. Then the set

$$B = \{\langle x, y_1\rangle : \forall y_2 \dots Q y_n R(x, y_1, \dots, y_n)\}$$

is Π^0_{n-1} and A is c.e. relative to B. By (2) for $n-1$ we have $B \leq_m \mathbb{N} - \emptyset^{(n-1)}$. So A is c.e. relative to $\emptyset^{(n-1)}$

Now suppose A is c.e. relative to $\emptyset^{(n-1)}$. Then there is a Turing functional Φ s.t. $A = \mathrm{dom}(\Phi^{\emptyset^{(n-1)}})$. By the use principle

$$x \in A \Leftrightarrow \exists \eta, s \ulcorner \Phi^\eta_s(x) \downarrow \wedge \forall i < |\eta| \; \eta(i) = 1 \leftrightarrow i \in \emptyset^{(n-1)\urcorner}$$

The innermost part can be put into Σ^0_n-form, so A is Σ^0_n.

2. Follows by Proposition 1.20 where $Y = \emptyset^{(n-1)}$

□

Proposition 1.40. *Let $n \geq 1$. Then A is $\Delta^0_n \Leftrightarrow A \leq_T \emptyset^{(n-1)}$*

Proof. By Theorem 1.39, A is $\Delta^0_n \Leftrightarrow A$ and $\mathbb{N} - A$ are c.e. in $\emptyset^{(n-1)}$. By Proposition 1.17, this condition is equivalent to $A \leq_T \emptyset^{(n-1)}$ □

Proposition 1.41. *Z is $\Sigma^0_2 \Leftrightarrow$ there is a computable sequence of strong indices $(Z_s)_{s \in \mathbb{N}}$ s.t. $Z_s \subseteq [0, s)$ and $x \in Z \leftrightarrow \exists s \forall t \geq s \; Z_t(x) = 1$. The equivalence is uniform*

Proof. $\Rightarrow$: By Theorem 1.39, there is a Turing functional Φ s.t. $Z = \mathrm{dom}(\Phi^{\emptyset'})$. Now let $Z_s = \{x < s : \Phi^{\emptyset'}(x)[s] \downarrow\}$

$\Leftarrow$: □

Definition 1.42. The **index set** of a class S of c.e. sets is the set $\{i : W_i \in S\}$

Exercise 1.3.1. $\emptyset'$ is not an index set

Proof. We can find g s.t. $W_{g(n)} = \{n\}$. Thus there is e s.t. $W_{g(e)} = W_e = \{e\}$. By padding lemma, we have $W_i = W_e$ but $i \notin \emptyset'$ □

Exercise 1.3.2.

1. $\{e : W_e \neq \emptyset\}$ is Σ^0_1-complete
2. The set $\{e : W_e \text{ finite }\}$ is Σ^0_2-complete.
3. The set $\mathrm{Tot} = \{e : \mathrm{dom}(\Phi_e) = \mathbb{N}\} = \{e : W_e = \mathbb{N}\}$ is Π^0_2-complete
4. Both $\{e : W_e \text{ cofinite}\}$ and $\mathrm{Cop} = \{e : W_e \text{ computable}\}$ are Σ^0_3-complete

Proof.
1. Given e, $\Phi_{f(n)}$ doesn't converge in $\mathbb{N} - \{e\}$. And converges on e is $\Phi_e(e) \downarrow$. Thus $\emptyset' \leq_m \{e : W_e \neq \emptyset\}$

2. Let $\mathrm{Fin} = \{e : W_e \text{ finite}\}$. Then $x \in \mathrm{Fin} \Leftrightarrow \exists s \forall t \geq s (W_{e,s} = W_{e,t})$

3. $e \in \mathrm{Tot} \Leftrightarrow \forall n \exists s \Phi_{e,s}(n) \downarrow$

   For any A in Π_2^0, $x \in A \Leftrightarrow \forall y \exists z R(x, y, z)$

   We could define

$$\Phi_{q(x)}(u) = \begin{cases} 0 & \forall y \leq u \exists z R(x, y, z) \\ \uparrow \end{cases}$$

   Then $x \in A \Leftrightarrow W_{q(x)} = \omega \Leftrightarrow q(x) \in \mathrm{Tot}$

   $x \in \overline{A} \Leftrightarrow W_{q(x)}$ is finite

4. $e \in \mathrm{Cof} \Leftrightarrow \exists z \forall n \geq z \exists s \Phi_{e,s}(n) \downarrow$, thus $\mathrm{Cof} \in \Sigma_3^0$

□

Exercise 1.3.3. Let $X \subseteq \mathbb{N}$

1. Each relation $R \leq_T X$ is first-order definable in the structure $(\mathbb{N}, +, \cdot, X)$

2. The index set $\{e : W_e \leq_T X\}$ is $\Sigma_3^0(X)$

Proof. □