

Computability theoretic forcing*

How to prove theorems about Turing reducibility the hard way.

Dino Rossegger

2026-10-12

1 Reducibility, hierarchies, and complete sets

The concept of relative computability goes back to Turing's 1939 paper and is usually modeled using oracle Turing machines. Informally a function $f : \omega \rightarrow \omega$ is **computable relative to** $A \subseteq \omega$ if there is an algorithm for computing $f(n)$ where the algorithm has access to elementary information about A , i.e., it may ask questions of the form "is $k \in A$?" during execution (and will also receive answers to these questions).

More formally, an oracle Turing machine Γ is a Turing machine with an additional read-only "oracle tape". The machine has an independent oracle reading head which it may move along the oracle tape and read the bit under it. For an infinite binary string $x \in 2^\omega$, we write Γ^x for the oracle Turing machine with x written on its oracle tape.

A function $f : \omega \rightarrow \omega$ is computable relative to $A \subseteq \omega$ if there is (a program for) an oracle Turing machine Γ such that for every n , $f(n) = \Gamma^A(n)$ (strictly speaking we put the characteristic function χ_A of A on the oracle tape and not A . To ease notation we identify A with its characteristic function when appropriate). A set $A \subseteq \omega$ is computable relative to $B \subseteq \omega$ if its characteristic function χ_A is computable relative to B .

Any computation $\Gamma^A(n)$ must terminate after a finite amount of steps and thus can only read a finite amount of the oracle tape. So there must be some $k \in \omega$ so that if $B \upharpoonright k = A \upharpoonright k$, then $\Gamma^B(n) = \Gamma^A(n)$. The least such k is called the **use** $u(\Gamma, A, n)$ of $\Gamma^A(n)$. Motivated by this we also allow finite strings $\sigma \in 2^{<\omega}$ on the oracle tape. We use the convention that if $\Gamma^\sigma(n)$ tries to move past σ on the oracle tape then $\Gamma^\sigma(n) \uparrow$ (i.e., is undefined) and that if $\Gamma^\sigma(n) \downarrow$, then it does so within $|\sigma|$ steps, i.e. $\Gamma^\sigma(n) \downarrow$ iff $\Gamma^\sigma_{|\sigma|}(n) \downarrow$.

The fact alluded to above that if $\Gamma^\sigma(n) \downarrow$ and $A \succ \sigma$, then $\Gamma^A(n) = \Gamma^\sigma(n)$ is called the **use principle**. The use principle is an important property of oracle computations. We will now gather some more facts which follow from easy adaptations of the proofs of similar facts for Turing machines.

Proposition 1.1.

1. There is a **universal oracle Turing machine**, i.e., a Turing machine Φ such that for each Turing machine Γ , there is an $e \in \omega$ such that for all n and $X \subseteq \omega$ $\Phi^X(e, n) = \Gamma^X(n)$. We will write Φ_e for $\Phi^-(e, -)$ and refer to $(\Phi_e)_{e \in \omega}$ as **standard enumeration** of Turing machines.
2. (The s-m-n theorem) There is an injective computable function $s : \omega^2 \rightarrow \omega$ so that for all x, y, z and all $X \subseteq \omega$ $\Phi_{s(x,y)}^X(z) = \Phi_x^X(y, z)$.

*Notes accompanying the lecture "Topics in computability theory" taught in winter 2026. These notes are continuously updated during winter term 2026; feedback and corrections are welcome.

3. (The padding lemma) *There is an injective computable function $p : \omega^2 \rightarrow \omega$ so that for every n and every i $\Phi_n^X = \Phi_{p(n,i)}^X$.*
4. (The recursion theorem) *Let $f : \omega \rightarrow \omega$ be a computable function. Then there is some e such that for all X , $\Phi_e^X = \Phi_f(e)^X$.*

1.1 Turing reducibility and the Turing degrees

Definition 1.1. Let $(\Phi_e)_{e \in \omega}$ be an enumeration of the oracle Turing machines. We say that A is computable relative to B , or, A is **Turing reducible to B** , $A \leq_T B$, if there is an e such that $\Phi_e^B = A$. A and B are **Turing equivalent**, $A \equiv_T B$, if $A \leq_T B$ and $B \leq_T A$.

The **Turing degree** of A is

$$\deg_T(A) = \mathbf{a} = \{B : B \equiv_T A\}.$$

The set of Turing degrees is denoted by $\mathcal{D}_T$. It is partially ordered by the relation

$$\mathbf{a} \leq \mathbf{b} \iff \exists A \in \mathbf{a} \exists B \in \mathbf{b} \ A \leq_T B$$

Clearly there is a least Turing degree consisting of the computable sets, the degree of $\emptyset$, denoted by $\mathbf{0}$.

There are many other reducibilities between sets studied in computability theory such as arithmetic, hyperarithmetic, enumeration, (weak) truth table reducibility, etc. Two very important ones that will be useful to us are the following.

Definition 1.2. Let $A, B \subseteq \omega$, then

1. A is **many-one reducible to B** , $A \leq_m B$, if there is a computable function $f : \omega \rightarrow \omega$ so that $n \in A$ iff $f(n) \in B$. The induced many-one equivalence is denoted by $\equiv_m$.
2. A is **one-one reducible to B** , $A \leq_1 B$, if there is a computable injective function $f : \omega \rightarrow \omega$ so that $n \in A$ iff $f(n) \in B$. The induced one-one equivalence is denoted by $\equiv_1$.

It is not hard to see that $\leq_1$ and $\leq_m$ are finer than $\leq_T$, i.e. $A \leq_m B$ implies $A \leq_T B$. One-one equivalence is often denoted as the *recursive isomorphism* relation as Myhill's isomorphism theorem shows that $A \equiv_1 B$ if and only if there is a computable bijection $f : \omega \rightarrow \omega$ such that $n \in A$ iff $f(n) \in B$.

The **computable join** of $A, B \subseteq \omega$ is

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

Proposition 1.2. *The Turing degrees $(\mathcal{D}_T, \leq)$ form an upper semilattice with*

$$\mathbf{a} \vee \mathbf{b} = \deg_T(A \oplus B).$$

Proof. Fix $A, B \subseteq \omega$ of degree $\mathbf{a}$, respectively $\mathbf{b}$. The functions $n \mapsto 2n$ and $n \mapsto 2n + 1$ are computable and thus $A, B \leq_m A \oplus B$, so $\deg_T(A \oplus B) \geq \mathbf{a}, \mathbf{b}$. Suppose $\mathbf{c} \geq \mathbf{a}, \mathbf{b}$. Then for any $C \in \mathbf{c}$, $C \geq_T A, B$ and thus also $C \geq_T A \oplus B$. To see this suppose that $\Phi_{e_0}^C = A$ and $\Phi_{e_1}^C = B$ and consider the function $f : 2^\omega \times \omega \rightarrow 2$

$$f(X, n) = \begin{cases} \Phi_{e_0}^X(k/2) & \text{if } n = 2k \\ \Phi_{e_1}^X(k/2) & \text{if } n = 2k + 1 \end{cases}$$

This function is clearly computable with an index depending only on e_1, e_2 and thus there is an oracle Turing machine Φ such that $\Phi^X = f(X, -)$. To be more precise, consider the program running program e_0 on even inputs and program e_1 on odd inputs and appeal to the Church-Turing thesis. $\square$

1.1.1 The Turing jump

The most important function between sets in computability theory is the **Turing jump**

$$J: 2^\omega \rightarrow 2^\omega \\ A \mapsto A' = \{n : \Phi_n^A(n) \downarrow\}.$$

Theorem 1.3 (Properness of the Turing jump). *For all $X \subseteq \omega$, $X <_T X'$.*

Proof. We first show that $X' \geq_T X$. For each n , construct a program $\Phi_{f(n)}^X$ which halts on any input m if and only if $n \in X$. Then $X \leq_1 X'$ via the computable function f , as $n \in X$ if and only if $\Phi_{f(n)}^X(f(n)) \downarrow$ if and only if $f(n) \in X'$.

That $X \not\geq_T X'$ is the straight-forward relativization of the fact that the halting problem is not computable. We give it here to illustrate how many facts about computable functions relativize.

Suppose towards a contradiction that $X' \leq_T X$, then the partial function

$$f(n) = \begin{cases} 1 & n \notin X' \\ \uparrow & n \in X' \end{cases}$$

is partial computable from X , i.e., there is an index e such that $\Phi_e^X = f$. But then

$$\Phi_e^X(e) \downarrow \iff e \notin X' \iff \Phi_e^X(e) \uparrow,$$

a clear contradiction. $\square$

Next we see that the Turing jump is degree invariant in a strong sense and thus can be viewed as a function on the Turing degrees.

Theorem 1.4. *For all $A, B \subseteq \omega$, $A \leq_T B \iff A' \leq_1 B'$.*

Proof. Suppose that $A \leq_T B$, via Φ_e and for each n construct a program $\Phi_{f(n)}^B$ such that for any m

$$\Phi_{f(n)}^B(m) \downarrow \iff \Phi_n^{\Phi_e^B}(n) \downarrow.$$

Then $n \in A'$ iff $\Phi_{f(n)}^B(f(n)) \downarrow$ iff $f(n) \in B'$, showing that $A' \leq_1 B'$.

On the other hand assume that $A' \leq_1 B'$ via f . Note that relativizing the completeness of $\emptyset'$, if $C \leq_1 B'$, then C is c.e. in B . Hence, A is c.e. in B . To finish we need to show that $\omega \setminus A$ is c.e. in B . To see this consider g such that for all m $\Phi_{g(n)}^A(m) \downarrow$ if and only if $n \notin A$. Then $n \notin A$ if and only if $g(n) \in A'$ if and only if $f(g(n)) \in B'$. As both g and f are computable, $\omega \setminus A$ is c.e. in B . $\square$

1.2 The arithmetical hierarchy

Definition 1.3 (The relativized arithmetical hierarchy). Let $X \subseteq \omega$ and $A \subseteq \omega$. Then for all $n \in \omega$

$$1. \Delta_0^0(X) = \Sigma_0^0(X) = \Pi_0^0(X) = \{A \subseteq \omega : A \leq_T X\}$$

2. A is $\Sigma_n^0(X)$ if there is an X -computable predicate R such that

$$x \in A \iff (\exists \vec{u}_1)(\forall \vec{u}_2)(\exists \vec{u}_3) \cdots (Q_n \vec{u}_n) R(x, \vec{u}_1, \dots, \vec{u}_n),$$

where $Q_n = \exists$ if n is odd and $Q_n = \forall$ otherwise.

3. A is $\Pi_n^0(X)$ if there is an X -computable predicate R such that

$$x \in A \iff (\forall \vec{u}_1)(\exists \vec{u}_2)(\forall \vec{u}_3) \cdots (Q_n \vec{u}_n) R(x, \vec{u}_1, \dots, \vec{u}_n),$$

where $Q_n = \exists$ if n is even and $Q_n = \forall$ otherwise.

4. A is $\Delta_n^0(X)$ if and only if $A \in \Sigma_n^0(X) \cap \Pi_n^0(X)$

We write Γ_n^0 instead of $\Gamma_n^0(\emptyset)$ for $\Gamma \in \{\Sigma, \Pi, \Delta\}$.

The term arithmetical hierarchy comes from the fact that the Σ_n^0 sets coincide with the sets definable in $(\mathcal{N}, 0, 1, +, \cdot)$ via $\exists_n$ -formulas.

Proposition 1.5. *A set $A \subseteq \omega$ is Π_n^0 if and only if $\omega \setminus A$ is Σ_n^0 .*

We will now connect the arithmetic hierarchy with Turing reducibility and the Turing jump. Towards this note that we can iterate the Turing jump. Given $X \subseteq \omega$, let $X^{(0)} = X$ and for $n \in \omega$ let the $(n+1)$ -th **Turing jump** of X be $X^{(n+1)} = (X^{(n)})'$.

Lemma 1.6. *For every $A, X \subseteq \omega$, $A \in \Sigma_1^0(X)$ iff A is c.e. in X if and only if $A \leq_m X'$.*

Proof. Proofs before, mutatis mutandis. □

Theorem 1.7 (Post's theorem). *For every $A, X \subseteq \omega$ and $n > 0$, $A \in \Sigma_n^0(X)$ iff A is c.e. in $X^{(n-1)}$ iff $A \leq_m X^{(n)}$.*

Proof. The proof is by induction with the base case covered by Lemma 1.6. That A is c.e. in $X^{(n-1)}$ iff $A \leq_m X^{(n)}$ follows by replacing X in Lemma 1.6 with $X^{(n-1)}$.

Suppose by induction that the theorem holds for n , and suppose that A is $\Sigma_{n+1}^0(X)$ defined by the computable relation R . Then the set $P = \{(n, \vec{u}_1) : \forall \vec{u}_2 \dots Q_n \vec{u}_n R(-)\}$ is $\Pi_n^0(X)$ and note that $A \in \Sigma_1^0(P)$. As $P \in \Pi_n^0(X)$, $\omega \setminus P$ is $\Sigma_n^0(X)$ and by the induction hypothesis, $\omega \setminus P \leq_m X^{(n)}$. In particular, $P \leq_T X^{(n)}$, and as A is c.e. in P , A is also c.e. in $X^{(n)}$.

On the other hand, suppose that A is c.e. in $X^{(n)}$. Then by the base case A is $\Sigma_1^0(X^{(n)})$, i.e. $x \in A \iff \exists u R(x, u)$ for R $X^{(n)}$ -computable. Clearly $X^{(n)} \leq_m X^{(n)}$ and so by hypothesis $X^{(n)}$ is $\Sigma_n^0(X)$. Let Φ_e^X be such that $\Phi_e^X = R$ and let σ_k be the k th finite binary string in the lexicographic ordering. We claim that

$$x \in A \iff \exists u \exists k \Phi_e^{\sigma_k}(x, u) \downarrow = 1 \wedge (\forall m < |\sigma_k|)(\sigma_k(m) = 1 \leftrightarrow m \in X^{(n)}).$$

To see this recall the use principle. $\Phi_e^{\sigma_k}(x, u) \downarrow = 1$ iff there is a $\sigma_k \prec X^{(n)}$ such that $\Phi_e^{\sigma_k}(x, u) \downarrow = 1$. $(\forall m < |\sigma_k|) \sigma_k(m) = 1 \leftrightarrow m \in X^{(n)}$ is a $\Sigma_{n+1}^0(X)$ -definition, as $\Sigma_{n+1}^0(X) \supset \Sigma_n^0(X) \cup \Pi_n^0(X)$. Thus, the whole definition is $\Sigma_{n+1}^0(X)$. □

Corollary 1.8. *For $X, A \subseteq \omega$, A is $\Delta_n^0(X)$ if and only if $A \leq_T X^{(n-1)}$.*

Corollary 1.9. *For every $X \subseteq \omega$ and $n \in \omega$, there is $X^{(n)} \in \Sigma_n^0(X) \setminus \Pi_n^0(X)$.*

Definition 1.4. Say that X is Γ_n^0 -**complete** for $\Gamma \in \{\Sigma, \Pi\}$ if every $Y \in \Gamma$, $Y \leq_m X$.

By Post's theorem we obtain that $\emptyset^{(n)}$ is Σ_n^0 -complete.

1.3 Examples of complete sets

Recall that $(W_e)_{e \in \omega}$ is the canonical enumeration of c.e. sets, i.e., $W_e = \{n : \varphi_e(n) \downarrow\}$.

Proposition 1.10. *The set $K_1 = \{e : W_e \neq \emptyset\}$ is Σ_1^0 -complete.*

Theorem 1.11. *The set $\text{Fin} = \{e : W_e \text{ is finite}\}$ is Σ_2^0 -complete and the set $\text{Tot} = \{e : W_e = \omega\}$ is Π_2^0 -complete.*

Proof. Suppose that $S \in \Sigma_2^0$, then $n \in S$ iff $\exists k \forall m R(n, k, m)$. Define $W_{f(n)}$ via a c.e. approximation as follows. For every s , let k_s be the least k such that $\forall m < s R(n, k, m)$ if such k exists and $k_s = s$ otherwise. Suppose we have defined $W_{f(n), s-1}$ and let j_{s-1} be the least element not in $W_{f(n), s-1}$. If $k_s > k_{s-1}$ enumerate j_{s-1} into $W_{f(n), s}$, otherwise $W_{f(n), s} = W_{f(n), s-1}$. We then let $W_{f(n)} = \lim_s W_{f(n), s-1}$. Note that every $W_{f(n), s}$ is computable, uniformly in s and that $W_{f(n), s} \supseteq W_{f(n), s-1}$ is increasing. Thus $m \in W_{f(n)}$ iff $\exists s m \in W_{f(n), s}$ and hence $W_{f(n)}$ is Σ_1^0 , so c.e.

If $n \in S$, then $\lim_s k_s \downarrow$ and thus $W_{f(n)}$ is an initial segment of ω . On the other hand if $n \notin S$, then $\lim_s k_s \uparrow$ and furthermore for every m , there is s with $k_s = m$. Thus, $m \in W_{f(n)}$ and $W_{f(n)} = \omega$. $\square$

1.4 Limit lemma

In the proof of Theorem 1.11 we have built a c.e. set as a limit of computable sets. Let us introduce the notion of limit computability more formally. A set A is **limit computable** in X if there is a sequence of finite sets $(F_i)_{i \in \omega}$, computable uniformly in X (i.e., there is an X -computable function f such that for all i, j $f(i, j) = \chi_{F_i}(j)$) such that $\chi_A = \lim_i \chi_{F_i}(j)$.

Theorem 1.12 (Limit lemma). *A set A is $\Delta_2^0(X)$ (equivalently $A \leq_T X'$) if and only if there is a uniformly X -computable sequence of finite sets $(F_i)_{i \in \omega}$ such that for all n , $A(n) = \lim_i F_i(n)$.*

Proof. Suppose that $A \leq_T X'$, say $A = \Phi_e^{X'}$. Let X'_s be the stage s approximation of X' , i.e., the finite set such that $n \in X'_s$ if and only if $\Phi_n^{X \cap \{0, \dots, s\}}(n) \downarrow$. Clearly the sets X'_s are computable uniformly in X . Now let $A_s = \Phi_e^{X'_s}$, note that A_s is finite and computable in X , uniformly in s , as X'_s is a finite oracle (note that for this assumption to hold we not only need that X'_s is finite but also that $X'_s \subseteq \{0, \dots, s\}$). Now, if $\Phi_e^{X'} = i \downarrow$, then there is a shortest $\sigma \prec X'$ such that $\Phi_e^\sigma(i) \downarrow$. Furthermore, as X' is c.e., there is a stage s such that for all $t > s$, $X'_t \succeq \sigma$. Thus, for all such t , $A_t(n) = A(n)$.

On the other hand suppose that $A = \lim_s A_s$ with $(A_s)_{s \in \omega}$ uniformly computable in X . Then

$$\begin{aligned} n \in A &\iff \exists m (\forall k > m) n \in A_k \\ n \notin A &\iff \exists m (\forall k > m) n \notin A_k \end{aligned}$$

The first equation shows that A is $\Sigma_2^0(X)$ and the second equation shows that A is $\Pi_2^0(X)$. Thus, A is $\Delta_2^0(X)$. $\square$