Turing Reducibility

Let $A \subseteq \mathbb{N}$. We will think about computations that can use A as an oracle, that is at any stage of the computation we can ask the oracle if $n \in A$. We let $P_0, P_1, \ldots$ be a nice listing of all programs that allow oracle queries. Let ϕ_n^A be the partial function computed by program P_n using oracle A.

We say that B is *Turing reducible* to A if there is an e such that $\phi_e^A = \chi_B$, the characteristic function of A. We write $B \leq_T A$.

If $A \leq_T B$ and $B \leq_T A$ we say write $A \equiv_T B$. The equivalence class of A is called the *Turing degree* of A.

Exercise 0.1 a) Show that $\neg A \leq_T A$

b) Show that if $A \leq_T B$ and $B \leq_T C$, then $A \leq_T C$.

c) Show that if $A \leq_m B$, then $A \leq_T B$.

d) Show that if A is recursive and $B \leq_T A$ then B is recursive. Conclude that the $\emptyset <_T H$, where H is the halting problem.

Exercise 0.2 For $A, B \subseteq \mathbb{N}$ let $A \oplus B = \{2n : n \in A\} \cup \{2n+1 : n \in B\}$.

a) Show that $A, B \leq_T A \oplus B$.

b) Show that if $A, B \leq_T C$ then $A \oplus B \leq_T C$. Thus $A \oplus B$ is a least upper bound for A and B.

Exercise 0.3 a) Let $A' = \{e : \phi_e(e) \downarrow\}$. We call A' the *jump* of A. Show that $A <_T A'$.

b) Show that if $A \leq_T B$, then $A' \leq_T B'$. In particular, if $A \equiv_T B$, then $A' \equiv_T B'$.

Lemma 0.4 (Use Principle) *If $\phi_e^A(n) = l$, there is k such that if $m \geq k$ and $A \cap \{0, \ldots, m\} = B \cap \{0, \ldots, m\}$, then $\phi_e^B(n) = l$.*

Proof The computation of $\phi_e^A(n)$ halts after s steps. It will make finitely many queries to the oracle. Let m be the largest number queried. In the computation of $\phi_e^B(n)$ will make the same queries and get the same answers so the computation will be the same. $\square$

It will be useful in the proofs below to define computations with oracles $\sigma \in 2^{<\omega}$. In a computation with oracle $\sigma = \langle a_0, \ldots, a_m \rangle$ we say that $\phi_e^\sigma(n) \downarrow$ if the computation only make oracle queries about numbers $i \leq m$ answering yes if and only if $a_i = 1$.

Note that deciding if $\phi_e^\sigma(n) \downarrow$ in s steps is recursive. Also if $\phi_e^\sigma(n) = l$, $f \in 2^\omega$ and $\sigma \subset f$, then $\phi_e^f(n) = l$.

We will give two proofs that there are incomparable Turing degrees. The first proof will be relatively straightforward. The second will be more complicated but will show that there Turing incomparable recursively enumberable sets.

Theorem 0.5 (Kleene–Post) *There are $A, B \leq_T H$ such that $A \not\leq_T B$ and $B \not\leq_T A$. In particular, $\emptyset <_T A, B <_T H$.*

Proof We build $\sigma_0 \subseteq \sigma_1 \subseteq \dots$ and $\tau_0 \subseteq \tau_1 \subseteq \dots$ such that $\bigcup \sigma_n = \chi_A$ and $\bigcup \tau_n = \chi_B$. Our entire construction will be computable using H as an oracle. We do our construction to meet the requirements $R_1, R_2, \dots,$ where:

- R_{2e+1} is the requirement to make sure $\chi_A \neq \phi_e^B$ and
- R_{2e+2} is the requirement to make sure $\chi_B \neq \phi_e^A$.

In this construction we will be able to meet requirement R_i at stage i.

Let $\sigma_0 = \tau_0 = \emptyset$

<u>stage $s+1 = 2e+1$</u> We try to make sure that $\chi_A \neq \phi_e^B$.

Let $\sigma_s = \langle a_0, \dots, a_{m-1} \rangle$.

case 1: There is no $\tau \supseteq \tau_s$ such that $\phi_e^\tau(m)$ converges

In this case we let $\sigma_{s+1} = \sigma_s\widehat{\ }0$ and $\tau_{s+1} = \tau_s$. We have made sure that ϕ_e^B is not even total.

case 2: There is $\tau \supseteq \tau_s$ such that $\phi_e^\tau(m)$ converges.

Choose τ a sequence with minimal code such that $\phi_e^\tau(m)$ converges with output i. Let $\tau_{s+1} = \tau$ and let $\sigma_{s+1} = \sigma_s\widehat{\ }j$ where $j = 0, 1$ and $j \neq i$. We have made sure that $\phi_e^B(n)$ disagrees with χ_A.

Note that with the oracle for the Halting Problem H we can decide which case we are and find τ_{s+1}. In either case, we have made sure that $\phi_e^B(n)$ disagrees with the characteristic function of A. We can think of this as having to satisfy two types of requirements:

<u>stage $s+1 = 2e+2$</u> We try to make sure that $B \neq \phi_e^A$. is

This is similar to case when $s+1$ is odd changing the roles of A and B.

At the end of the construction we have insure $A \neq \phi_e^B$ and $B \neq \phi_e^A$. Thus $A \not\leq_T B$ and $B \not\leq_T A$ $\square$

Post's Problem: Is there a recursively enumerable set A such that $\emptyset <_T A <_T H$.

Theorem 0.6 (Friedberg–Muchnik) *There are recursively enumerable A and B such that $A \not\leq_T B$ and $B \not\leq_T A$. Note that we must have $\emptyset <_T A, B <_T H$.*

Proof The proof is a finite injury priority argument.

We will do a recursive construction where at various stages we will enumerate elements into A and B. At any stage s of our construction we will have finite sets A_s and B_s. As in the previous constructions we have to eventually meet the same requirements $R_1, R_2, \ldots$ from the previous construction. The difference is that we may not be able to tell when we've met R_i.

Basic Strategy Here is the idea on how we will try to meet R_{2e+1} and make sure that $\chi_A \neq \phi_e^B$.

Pick a number n bigger than all the numbers we've considered so far. For the moment we will commit to keeping n out of A. At every further stage s of the construction we will run the computation $\phi_e^{B_s}(n)$ for s steps. If it doesn't halt or halts and outputs something other than 0, then we don't have to do anything as ϕ_e^B will end up being a partial function or $\phi_e^B(n) \neq 0 = \chi_A(n)$. If it halts and $\phi_e^{B_s}(n) \downarrow= 0$ and k is the largest element for which we make an oracle query, we then set up a restraint saying that no number less than or equal to k will ever be allowed into B. This will insure $\phi_e^B(n) = \phi_e^{B_s}(n)$. If, in addition, $\phi_e^{B_s}(n) = 0$, we will then change our mind and add n to A_{s+1}. This will make sure that $\phi_e^B(n) \neq \chi_A(n)$.

We will do similar things to try to meet the requirements $R_{2i+2} : \phi_i^A \neq B$. The problem is that these tasks may interfere with each other. For example, suppose in the description of the Basic Strategy that we decide we have to add n to A_{s+1}, but there is a requirement that is restraining us from adding n to A because it wants us to preserve some computation. How do we settle these conflicts between requirements? We priortize them! We say that R_i has priority over R_j if $i < j$. In the situation above if the requirement that wants to put n into A has higher piority we let it and the other requirement has to start over. But if the requirement that is restraining n has higher priority, the first requirement will have to start over. The trick is we need to make sure that each of the requirements is eventually satisfied.

Requirement R_n will pick x_n a witness that the requirement is met. For example, if $n = 2e + 1$, then, at the end of the construction, we will have $\chi_A(x_n) \neq \phi_e^B(x_n)$. Fix $\pi\colon \mathbb{N}^2 \to \mathbb{N}$ a recursive bijection. The witness x_n will be chosen from the numbers $\pi(n, 0), \pi(n, 1), \ldots$. Let $x_n(s)$ be our quess for the witness x_n. We must arrange things so that $x_n(s) = x_n$ for all sufficiently large s.

At some points the requirement R_n may want to keep numbers out of the one of the sets. For example, if $n = 2e + 1$ requirement n may want to preserve a computation by keeping numbers out of B. We will define a function $r_n(s)$ such that requirement $n = 2e + 1$ wants to keep all numbers $\leq r_n(s)$ our of A or B.

stage 0

Let $A_0 = B_0 = \emptyset$. Set $x_n(0) = \pi(n, 0)$ and $r_n(0) = -1$ for all n.

stage $s + 1$

Check for requirements that require attention.

We say that R_n *requires attention*, for $n = 2e+1$ if $\phi_e^{B_s}(x_n(s)) \downarrow= 0$ in at most s-steps and $r(n, s) = -1$. Simiarly, if $n = 2e+2$, R_n requires attention if $\phi_e^{A_s}(x_n(s) \downarrow= 0$ and $r(n, s) = -1$.

If no R_n requires attention for $n \leq s$, we let $x_i(s + 1) = x_i(s)$ and $r_i(s + 1) = r_i(s)$ for all i. Otherwise let R_n be the requirement of highest priority that requires attention. Without loss of generality assume $n = 2e+1$.

We take the following actions:

• for $i < n$ let $x_i(s + 1) = x_i(s)$ and $r_i(s + 1) = r_i(s)$, i.e., do nothing about requirements of higher priority;

• enumerate $x_n(s)$ into A i.e., we let $A_{s+1} = A_s \cup \{x_n(s)\}$, $B_{s+1} = B_s$ and $x_n(s + 1) = x_n$;

• let $r_n(s + 1) = K$ where K is maximal such that the computation of $\phi_e^{B_s}(x_n(s))$ makes an oracle query about K;

• for $j > n$ we say R_j is *injured* and everything R_j has done is wiped out, i.e., we set $r_j(s + 1) = -1$ and $x_j(s + 1) = \pi(j, k)$ where k is minimal such that $x_j(s + 1) > x_j(s)$ and $x_j(s + 1) > r(i, s + 1)$ for all $i < j$ (i.e., we pick the next possible witness above all restraints that we have set of higher priority).

claim Each requirement R_n requires attention only finitely often and each requirement is eventually met.

Note that R_1 is never injured. Suppose $n = 2e + 1$ (the other case is similar). Let s be the last stage where some R_i of higher priority requires attention. At that stage we will have set $r_n(s+1) = -1$. We will be keeping $x_n(s + 1)$ out of A and hope that $\phi_e^B(x_n(s + 1)) \neq 0$. If R_n never requires attention, this will happen and the requirement will be met. If not, let $t > s$

be the first stage where R_n requires attention. At this stage we will put $x_n(t)$ into A and restrain elements from entering B so that $\phi_e^B(x_n(t)) = 0$. As R_n will never be injured again, we will preserve this restraint. In either case $\phi_e^B \neq \chi_A$. $\square$

There has been a lot of work on the structure of the Turing degrees of recursively enumerable set–see for example R. Soare's *Recursively Enumerable Sets and Degrees*. Interestingly, there are no "natural" examples of recursively enumerable sets of intermediate degree.