Chapter 12

Relative Computability and Turing Reduction

Many-one reducibility is an interesting relation because it preserves recursiveness, r.e.-ness and co-r.e.-ness. These features are bought at a price. For example, the following properties are strange:

Exercise 12.1 *Prove that*

1. there are exactly 3 recursive M-degrees.
2. K does not M-reduce $\overline{K}$.

These curiosities reflect the fact that M-reduction amounts to a very restrictive way to use a given decision procedure as a subroutine; namely, one may only compose another routine inside of it:

$$Q(x) = P(f(x))$$

Thus, M-reduction does not allow for composition outside:

$$\overline{K}(x) = \overline{sg}(K(x))$$

12.1 Relative Computability

Such considerations suggest a notion of reducibility based on an unrestricted notion of subroutine. This is usually done with Turing machines that are allowed to ask arbitrary questions about a given set A and to use the results in their computations as they please. We can do essentially the same thing in Kleene's klean formulation of the theory by simply adding the characteristic function of A to the stock of basic functions and assigning it all indices of length 6.

$$
\begin{aligned}
&k = 0 \Rightarrow \phi^k_{A,x} = x \\
&k > 0 \wedge \\
&\quad lh(x) = 0 \Rightarrow \phi^k_{A,x} = C(o, p^1_k) \\
&\quad lh(x) = 1 \Rightarrow \phi^k_{A,x} = C(s, p^1_k) \\
&\quad lh(x) = 2 \Rightarrow \phi^k_{A,x} = p_k^{\min((x)_0, k)} \\
&\quad lh(x) = 3 \Rightarrow \phi^k_{A,x} = C(\phi^{lh((x)_1)}_{(x)_0}, \phi^k_{((x)_1)_0}, \ldots, \phi^k_{((x)_1)_{lh((x)_1) \dot{-} 1}}) \\
&\quad lh(x) = 4 \Rightarrow \phi^k_{A,x} = R(\phi^{k \dot{-} 1}_{(x)_0}, \phi^{k+1}_{(x)_1}) \\
&\quad lh(x) = 5 \Rightarrow \phi^k_{A,x} = M(\phi^{k+1}_{(x)_0}) \\
&\quad lh(x) = 6 \Rightarrow \phi^k_{A,x} = C(\chi_A, p^1_k) \\
&\quad lh(x) > 6 \Rightarrow \phi^k_{A,x} = C(o, p^1_k)
\end{aligned}
$$

Now define

$$\psi \text{ is partial recursive in } A \iff \exists n, k \ (\psi = \phi^k_{A,n})$$

$$W_{A,n} = dom(\phi^1_{A,n})$$
$$E_{A,n} = dom(\phi^1_{A,n})$$

$$B \text{ is recursive in } A \iff$$
$$\text{the characteristic function of } B \text{ is partial recursive in } A$$

$$B \text{ is r.e. in } A \iff$$
$$B = \emptyset \ \vee \ \exists \text{ total } f \text{ partial recursive in } A \text{ such that } B = rng(f)$$

Proposition 12.1 *All our earlier results relativize to A in the natural way.*

Exercise 12.2 *Convince yourself of this by stating and sketching the proof of the A-relativized universal theorem. Let U_A denote the A-relativized Kleene universal predicate. Important: we will need the property that the "resource bound" also bounds all the inputs for which values are queried in the computation.*

12.2 Turing Reduction

We may now define a more powerful kind of reduction relation called Turing reducibility (because it is usually defined using Turing machines as described above). Turing equivalence and degrees can be defined as before.

$$A \leq B \iff A \text{ is recursive in } B$$

Proposition 12.2

1. $\leq$ is a pre-order
2. $\leq_M \subset \ \leq$
3. $A \leq \overline{A}$
4. B is recursive, then $B \leq A$
5. recursiveness in A is preserved downward under $\leq$
6. r.e.-ness and co-r.e.-ness in A are *not* preserved downward under $\leq$

Exercise 12.3 *Prove it. Note that the inclusion in 2 is proper!*

The last clause of the proposition shows that Turing reduction is also somewhat strange, for we can no longer rely on Turing computability to preserve one-sided notions of success. Can you think of a notion of reduction that does not restrict the use of subroutines but that would preserve r.e.-ness?

The usual mathematical habits yield Turing-equivalence and Turing-degrees, by proposition 12.2.1:

$$A \equiv B \iff (A \leq B \wedge B \leq A)$$

$$[A] = \{B \subseteq \mathbf{N} : A \equiv B\}$$

$$[A] \leq [B] \iff A \leq B$$

Turing degrees (or simply "degrees") are conventionally denoted by boldfaced letters: $\mathbf{a}, \mathbf{b}, \mathbf{c} \ldots$
Now define:

$$\begin{aligned} \mathbf{a} \text{ is a } \Gamma\text{-degree} &\iff \mathbf{a} \cap \Gamma \neq \emptyset \\ \mathbf{a} \text{ is } \Gamma\text{-hard} &\iff \forall \mathbf{b}(\mathbf{b} \leq \mathbf{a}) \\ \mathbf{a} \text{ is } \Gamma\text{-complete} &\iff (\mathbf{a} \cap \Gamma \neq \emptyset \ \wedge \ \forall \mathbf{b}(\mathbf{b} \leq \mathbf{a})) \end{aligned}$$

Important: an r.e. degree is also a co-r.e. degree and not every element of an r.e. degree is r.e.

Corollary 12.3

1. There is a unique recursive degree. Call it $\mathbf{0}$.
2. $[A]_M \subseteq [A]$.
3. $[K]$ is r.e.-complete.

Proof 1 comes from proposition 12.2.4.
2 and 3 come from proposition 12.2.2.⊣

12.3 The Jump Operation

The co-halting problem is just the counter-diagonal of the table

$$T[x, y] = W_x(y)$$

That is,

$$\begin{aligned} \overline{K}(x) &= \overline{sg}(T[x, x]) \\ &= \overline{sg}(W_x(x)) \end{aligned}$$

We could do the same thing for the table of sets r.e. in A:

$$\overline{K}(x) = \overline{sg}(W_{A,x}(x))$$

Then we know (by exactly the same argument) that $\overline{K}_A(x)$ is not r.e. in A, so K is not recursive in A. The usual notation is:

$$A' = K_A = \{x | \phi_{A,x}(x)\downarrow\}$$

$$\mathbf{a}' = [A'], \text{ for some } A \in \mathbf{a}$$

In our new notation, we have:

$$\mathbf{0}' = [K]$$

The following records that jumps are essentially like the halting problem.

Proposition 12.4

1. $\overline{A}'$ is not r.e. in A.
2. A' is not recursive in A.
3. A' is r.e. in A.
4. A' is A-r.e. M-complete.
5. A is recursive $\Rightarrow A' \equiv K$.
6. $A \leq B \Rightarrow A' \leq B'$.

Proof. 1, 2. Already shown.
3. Using the A-universal construction, define A-partial recursive

$$\phi^1_{A,z}(x) \approx (\mu y)\ U_A(x, (y)_0, (y)_1, \langle x \rangle)$$

Then $A' = W_{A,z}$.
4. Relativize the proof that K is r.e.-complete (with respect to M-reduction) and apply proposition 12.2.2.
5. Replace each occurrence of the characteristic function of A in the derivation tree with a program that computes it.

6. Suppose $A \leq B$.
So let $\phi_{B,z} = \chi_A$.
By the argument for part 3, A' is r.e. in A in virtue of

$$\psi(x) \approx (\mu y)\ U_A(x, (y)_0, (y)_1, \langle x \rangle)$$

In the derivation tree for ψ, we may replace each occurrence of χ_A with $\phi_{B,z}$.
Hence, A' is B-r.e.
By part 4, B' is B-r.e. complete.
So $A' \leq B'$. $\dashv$

12.4 Incomplete Degrees (Post's Problem)

Say that **a** is an incomplete r.e. degree $\iff$

- **a** is r.e.
- **a** is not r.e. complete
- **a** is not recursive

Simple sets determine incomplete r.e. M-degrees as we have seen, but Turing degrees are coarser (since Turing reducibility is more lenient) so it doesn't follow that simple sets determine incomplete r.e. degrees. The worry is justified.

Proposition 12.5 *Every r.e. degree (e.g., the complete one) contains a simple set.*

Exercise 12.4 *Prove it. Big hint:*
Suppose B is r.e., but not recursive.
Construct a total recursive, bijective f such that $B = rng(f)$.
Define

$$A(x) \iff \exists_{y>x}(f(y) < f(x))$$

A is r.e. by the projection theorem. Show:

1. *$A \equiv B$*
2. *A is simple*

More hints: To establish simplicity, assume that the required properties are false and then use f to decide A.

Thus, a new technique is required to construct an incomplete r.e. Turing degree.

It suffices to find two non-recursive r.e. degrees such that neither is Turing reducible to the other, for an r.e.-complete degree is comparable with every r.e. degree.

The construction of such a degree was called Post's problem. The solution was discovered independently by Friedburg and Muchnik. The proof is called

a "priority argument". The priority technique was applied to a number of problems in degree theory, leading to a renaissance in recursion theory in the 1960's and 1970's.

Proposition 12.6 ***(Friedburg-Muchnik 1956)*** *There exist r.e. degrees* $\mathbf{a}, \mathbf{b}$ *such that neither* $\mathbf{a} \leq \mathbf{b}$ *nor* $\mathbf{b} \leq \mathbf{a}$.

Idea. (following Soare's theorem 2.1): Here's the intuitive idea. An even requirement is a statement of form:

$$even(z) \wedge \chi_A \neq \phi_{B,\frac{z}{2}}$$

An odd requirement is a statement of form:

$$odd(z) \wedge \chi_B \neq \phi_{A,\frac{z-1}{2}}$$

Clearly, satisfying all the requirements would make A and B Turing irreducible to one another.

Even requirement z can be satisfied by adding to A a witness x on which $\phi_{B,\frac{z}{2}}$ halts with 0 or never adding x to A if $\phi_{B,\frac{z}{2}}$ halts with 0. Odd requirement z can be satisfied by adding to B a witness x on which $\phi_{A,\frac{z-1}{2}}$ halts with 0 or never adding x to A if $\phi_{A,\frac{z-1}{2}}$ halts with 0.

We can therefore use halting with 0 as an effective sign to add the witness to the appropriate set and rest smugly until the condition arises.

The trouble is that requirements we thought were "passively" satisfied may halt with 0 later. Such a requirement is said to require *attention*. Then we have to add the witness to the appropriate set to keep the requirement satisfied.

But this may screw up other computations witnessing other requirements, since these computations depend on the sets in question. We say that these requirements are injured.

To make progress, we have to protect more and more witnessing computations from injury. We do this by enumerating the requirements (*a priori* ranking) and protect higher priority witnessing computations from lower priority requirements by requiring that witnesses for fresh requirements be selected beyond a *protective wall* we build around the members of the sets queried in computations of higher priority.

Each requirement is injured only by requirements of higher priority. There are only finitely many of these. So each requirement is injured only finitely often. In the limit, every requirement is satisfied.

Since we never had to decide which way the requirements are satisfied, the sets constructed by adding witnesses are r.e.

This is called a *priority* or *finite injury* argument.

Proof: we need to construct r.e. sets A, B such that both are r.e. and neither is Turing-reducible to the other. We construct A, B in stages:

$$A(x) \iff \exists n \ (A_n(x))$$

$$B(x) \iff \exists n \ (B_n(x))$$

Each stage results from possibly adding something new.

$$A_n(x) \iff \exists_{m \le n}\ (\textit{add-to-}A(m,x))$$

$$B_n(x) \iff \exists_{m \le n}\ (\textit{add-to-}B(m,x))$$

We begin the construction by searching for the highest priority requirement that needs attention, if any.

$\textit{attend-to}(n) =$
$\quad \mu z \le n$
$\quad\quad (\textit{even}(z) \wedge \textit{wall}(n, \frac{z}{2}) = 0 \wedge U_{A_n}(\frac{z}{2}, n, 0, \langle \textit{try}(\frac{z}{2}, n)\rangle))$
$\quad\quad \vee(\textit{odd}(z) \wedge \textit{wall}(n, \frac{z-1}{2}) = 0 \wedge U_{A_n}(\frac{z-1}{2}, n, 0, \langle \textit{try}(\frac{z-1}{2}, n)\rangle)))$
$\quad\quad$ *(Seek the highest priority requirement whose protective wall is retracted and whose program has halted in the required number of steps with output 0.)*
$\quad$ if there is one
$\quad n+1$ otherwise.

$\textit{wall}(0, z) = 0$
$\textit{wall}(n+1, z) =$
$\quad \textit{wall}(n, z)$ if $z < \textit{attend-to}(n) \ \vee\ \textit{attend-to}(n) > n$
$\quad$ *(keep protecting higher-priority witnesses; do nothing if nothing is attended to)*
$\quad n+1$ if $z = \textit{attend-to}(n)$
$\quad$ *(establish wall protecting currently attended to requirement)*
$\quad 0$ if $z > \textit{attend-to}(n)$
$\quad$ *(drop protection on injured requirements indicating that they may again require attention)*

Keep old witnesses unless a higher priority requirement gets attention. Then increment lower priority witnesses to a safe place.

$\textit{try}(0, z) = \langle 0, z\rangle$
$\textit{try}(n+1, z) =$
$\quad \textit{try}(n, z)$ if $z \le \textit{attend-to}(n) \vee \textit{attend-to}(n) > n$
$\quad$ *(keep high priority witnesses)*
$\quad (\mu w)\ (w, 1) = z \wedge \neg A_n(w) \wedge \neg B_n(w) \wedge \forall k < z(\textit{wall}(n, z) < z$ otherwise.
$\quad$ *(injured witnesses should be selected so as not to disturb higher priority requirements).*

$$\textit{add-to-}A(n,x) \iff \textit{even}(\textit{attend-to}(n)) \wedge \textit{attend-to}(n) \le n \wedge x = \textit{try}(n, \textit{attend-to}(n))$$

$$\textit{add-to-}B(n,x) \iff \textit{odd}(\textit{attend-to}(n)) \wedge \textit{attend-to}(n) \le n \wedge x = \textit{try}(n, \textit{attend-to}(n))$$

Fact a all the component functions are total recursive, so by the projection theorem A and B are r.e.

Fact b every requirement is injured only finitely often.

Fact c after it stops being injured, each requirement is eventually satisfied.

Hence, A and B are Turing-incomparable. $\dashv$

12.5 A Taste of Degree Theory

Define:

$$A \text{ is low} \iff A' \equiv K$$

Proposition 12.7 *A is low $\Rightarrow$ A is not r.e.-complete.*

Proof: suppose A is r.e.-complete.
So $A \equiv K$.
So $A' \equiv K'$.
K' is not reducible to K by proposition 12.4.2 above.
So A' is not reducible to K.
So A is not low. $\dashv$

Define the *join* operation on sets and degrees as follows:

$$(A + B)(x) = (A((x)_0) \wedge (x)_1 = 0) \vee (B((x)_0) \wedge (x)_1 = 1)$$

$$\mathbf{a} \cup \mathbf{b} = [A + B], \text{ for } \mathbf{a}(A), \mathbf{b}(B)$$

Proposition 12.8 *The meet operation is the greatest lower bound operation corresponding to the Turing reducibility order.*
Thus, Turing degrees form an upper-semilattice under join*.*

Proof: in fact, A, B are both M-reducible to $A + B$, by means of the M-reductions $(x)_0$ and $(x)_1$.
For minimality, suppose that S also Turing-reduces both A and B.
Then $A + B$ is reducible to S as well:
for a given x, $(x)_1$ tells you whether to feed $(x)_0$ through the reduction of A or the reduction of B. $\dashv$

The importance of the low degrees is that they generate all the r.e. degrees when we close under join.

Proposition 12.9 ***(Sacks' splitting theorem 1963)***
For each r.e. $\mathbf{a} > 0$*, there are incomparable, low r.e. degrees* $\mathbf{b}$*,* $\mathbf{c}$ *such that* $\mathbf{a} = \mathbf{b} \cup \mathbf{c}$.

The r.e. degrees have many entertaining properties including the following:

Lachlan 1966 There is a minimal pair of r.e. degree above 0.

Sacks 1964 The r.e. degrees are densely ordered.

Shoenfield Every countable partial order can be embedded in the r.e. degrees in a manner that preserves sups.
(e.g., each countable ordinal is embeddable.)

Yates 1966 Every incomplete r.e. degree is incomparable with some other r.e. degree.