Brief notes to accompany Post's *Recursively enumerable sets of positive integers and their decision problems.*

By 1944 a few more 'unsolvable problems' had been found to add to the results of Church and Turing from 1936. But these problems are quite similar, and the question Post poses in this paper is: are these the only kind of unsolvable problems or are there other, different kinds? This became known as Post's problem.[1]

Put another way, Post wanted to know if some problems were 'more unsolvable' than others. In particular, he believed that the unsolvable problems found to date were of the hardest kind (he called them "creative"), and so his paper is a search for unsolvable problems that are simpler in some way than the creative ones, but not so simple as to be recursive.

Bases and normal systems

In the introduction, Post introduces what he calls a **normal system**, which is one of his models of computation. These are a special form of what is known as a Post canonical system, and are closely related to other string rewriting systems such as tag systems (Post, 1943), Thue systems (Thue, 1914) and, of course, formal grammars (Chomsky, 1956).

None of the rest of the paper depends on this model of computation so, for example, you can regard the enumeration of bases on page 287

$$O : B_1, B_2, B_3, \cdots$$

equivalently as an enumeration of Turing machines[2], or Java programs etc. Note that this enumeration O is worked out from their description (e.g. the rules, the list of transitions, the source code), not from running them.

1 Core Definitions

In this section I've listed the main definitions from Post's paper (I've added later definitions of *productive* and *immune* since they seemed helpful). In all these definitions, 'sets' should be taken to mean 'sets of natural numbers', i.e. they are always subsets of $\mathbb{N}$.

Reducibility

- A set A is **m-reducible** to a set B if there is a recursive function f such that for any x, $x \in A \Leftrightarrow f(x) \in B$. Post calls this "many-to-one reducible".

Notation: in this case we write $A \leq_m B$. In the special case where f is one-to-one, we may write $A \leq_1 B$.

When both $A \leq_m B$ and $B \leq_m A$ we write $A \equiv_m B$. Since $\equiv_m$ is an equivalence relation, it partitions the set of subsets of $\mathbb{N}$ into equivalence classes:

- An m-**degree** is a collection of sets each of which are m-equivalent; that is, for any two sets A and B in an m-degree we have $A \equiv_m B$.

[1] Not to be confused with the Post *correspondence* problem, which was just another problem proved unsolvable by Post in 1946.

[2] In more modern texts it is usual to denote this sequence as $\varphi_1, \varphi_2, \varphi_3, \ldots$

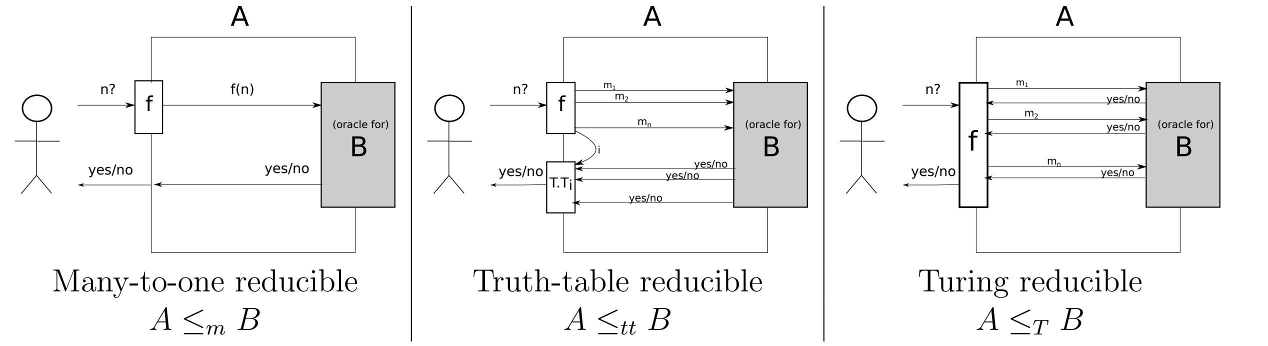

Figure 1: An illustration of the kinds of reduction. In all cases the (decision problem for the) set A is being reduced to the (decision problem for the) set B, using the recursive function f.

Reduction types: Given some set B, Post discusses four different ways of building a machine that reduces another decision problem to that of B. These are:

- m-reducible: generate a question for B, give B's answer directly to user. (§4)
- truth-table-reducible: generate a number of questions for B and generate a truth table, feed B's answers through truth table, give result to user. (§6)
- bounded-truth-table-reducible: as for truth table, but the maximum size of the truth table (and thus of the number of questions) is hard-wired in advance into the reduction.
- Turing-reducible: can interrogate B, processing B's answers, asking different questions for different scenarios, as many (finite) times as you want. (§11)

Whenever you talk of reductions (or degrees, or completeness, see below), you should always specify the type of reducibility. Notation: $A \leq_m B$, $A \leq_{btt} B$, $A \leq_{tt} B$, $A \leq_T B$ (not used by Post, but you may see it elsewhere).

Note that an *m-degree* can different from (contain fewer sets than) a *btt-degree*; similarly for the others.

Completeness

- The **complete set** K is just the 'Halting' set: i.e. the indices of bases (or Turing Machines) that halt when given their own index as input (§3).
- A set is categorised as m-**complete** if it is in the same m-degree as the complete set K.

  For each kind of reduction we get a version of completeness; note that a *btt-complete* set need not be *m-complete*, etc.
- A set S is **productive** if there is a recursive function f such that, for every r.e. set α with basis B_ν, we have $\alpha \subseteq S \Rightarrow f(\nu) \in (S - \alpha)$.
- A set S is **creative** if S is recursively enumerable and $\overline{S}$ is productive. (§3)
- It turns out that a set is creative iff it is m-complete. (Proved by Myhill in 1955)

Post's search for *in*complete recursively enumerable sets

- A set S is **immune** if S is infinite and S does not have a r.e. subset.
- A set S is **simple** if S is recursively enumerable and $\overline{S}$ is immune. (§5)
- Simple sets are not 1-complete, m-complete (§5) or btt-complete (§7) but are tt-complete (§8).
- A set S is **hyperimmune** if S is infinite and there is no disjoint set of sequences which have all members intersecting it.
- A set S is **hypersimple** if S is recursively enumerable and $\overline{S}$ is hyperimmune.
- Hypersimple sets are not tt-complete (§10). Unfortunately, they are T-complete - this was left open by Post at the end of §10, but proved by Dekker in 1954.

2 The sequel

Post's paper provoked a flurry of results, particularly in the early 1950s, culminating in 1956 with separate solutions by Richard M. Friedberg[3] in the U.S.A. and Albert Muchnik in the U.S.S.R.:

- **Friedberg-Muchnik Theorem** (1956/7): "solution to Post's problem"
  There exist recursively enumerable sets A and B such that neither $A \leq_T B$ nor $B \leq_T A$.

In fact, there are (countably) infinitely many incomparable degrees of such sets. This implies, in particular, the existence of r.e. T-degrees in-between the recursive sets and the creative sets.

Related definitions

In case you're reading about this material elsewhere, this section defines some related concepts on Turing degrees that will be useful to know.

Degree Theory

- If A is any set, we define **the jump of** A, written A' to be the set of Turing machines that halt (on their own index) when using A as an oracle - this is a kind of *relativised* Halting problem. This set is r.e. in A but not recursive in A. (This is also called the *Turing* jump.)
- Notation: Often the sequence of Turing machines is denoted $\varphi_1, \varphi_2, \ldots$, where φ_x is thus the x^{th} Turing machine, or the Turing machine with index x (or Gödel number x).

  The x^{th} Turing machine with access to some oracle A is denoted φ_x^A.

  Thus we can define the jump of a set A more formally as: $A' = \{x \mid \varphi_x^A(x) \text{ halts}\}$.

[3] Friedberg, Richard (1957). Two Recursively Enumerable Sets of Incomparable Degrees of Unsolvability. *Proceedings of the National Academy of Sciences of the United States of America* 43(2): 236-238.

- By overloading the notation, we can apply the jump operators to Turing degrees; the jump of a T-degree X is X', defined as the T-degree of the jump of the individual sets.
- If we take the empty set $\emptyset$, then its Turing degree is just the set of recursive sets and is typically denoted $\mathbf{0}$. The set $\emptyset'$ is then the Halting problem (called K by Post); the degree $\mathbf{0}'$ is thus the Turing degree of K.

  Thus Post's problem can be re-stated as finding an r.e. degree strictly between $\mathbf{0}$ and $\mathbf{0}'$.
- If you use the notation from the arithmetical hierarchy then the recursive, r.e. and co-r.e. sets become Δ_0, Σ_1 and Π_1 respectively, and the Turing jump corresponds to adding an extra quantifier.

Toward Friedberg-Muchnik

In 1954 Kleene and Post[4] published a paper with a result similar to Friedberg-Muchnik:

- There exist sets A and B such that neither $A \leq_T B$ nor $B \leq_T A$.

Unfortunately their A and B are not r.e. sets, so this isn't a solution to Post's problem. However, their proof provides a stepping-stone towards the one by Friedberg and Muchnik.

The idea is to build A by going through the sets φ_n^B for $n = 1, 2, 3 \ldots$, and diagonalising - i.e. adding an element to A to make it different from φ_n^B at that point.

The tricky bit is that you also have to build B in the same way (diagonalising each φ_n^A), and at the same time, so the build processes must be run in parallel. Thus when you diagonalise you only have a *partially-built* version of the set to diagonalise against, and you need to be careful to make sure that your assumptions will hold true for the final version too.

We define two infinite sequences f and g by building up a series of finite approximations in stages. Each sequence consists only of 0s and 1s, and thus defines a subset of $\mathbb{N}$.

- Stage 0: Initially, let f_0 and g_0 be empty.
- Stage $2n+1$: Let $m = 2n$ and let f_m be of length $x - 1$.
  Find the smallest finite extension $\tilde{g}$ of g such that $\varphi_n^{\tilde{g}}(x)$ is defined.
  Note that the oracle accesses are limited to querying only the finite sequence $\tilde{g}$.
  - If so, set $f_{m+1} = f_m \cup \{x \mapsto 0\}$ and $g_{m+1} = \tilde{g}$.
  - Otherwise, set $f_{m+1} = f_m \cup \{x \mapsto 1\}$ and let $g_{m+1} = g_m$.
- Stage $2n+2$: Let $m = 2n+1$; this is now as for stage $2n+1$, but with f and g swapped.

Finally, define $f = \bigcup_{i \to \infty} f_i$ and $A = \{x \mid f(x) = 1\}$, and similarly $g = \bigcup_{i \to \infty} g_i$ and $B = \{x \mid g(x) = 1\}$. $\square$

This proof can be seen as satisfying an infinite sequence of **requirements** of the form

- R_{2n}: $\exists x\, A(x) \neq \varphi_n^B(x)$
- R_{2n+1}: $\exists x\, B(x) \neq \varphi_n^A(x)$

In each case the x we find is called the **witness** to the satisfaction of the requirement.

Note the proof assumes that we can decide whether or not $\varphi_n^{\tilde{g}}(x)$ is defined (halts) - i.e. that we have access to an oracle for the Halting problem. In effect this places our two constructed sets A and B beyond the r.e. sets and into Δ_2.

[4] Kleene, S. C., and Emil L. Post (1954). The Upper Semi-Lattice of Degrees of Recursive Unsolvability. *The Annals of Mathematics* 59(3). Second Series: 379-407.

3 Books (in the library)

Many of the textbooks in the library on computability just cover the material you would have done in CS370, and stop there. For the material in Post's paper and beyond you need to look at the more specialised texts.

Probably the gentlest of these is is [SV03]; this is not too mathematical, and has modest pace and scope. It's also nice and short: they get as far as Friedberg-Muchnik, but not much further.

The monumental [Odi89] takes the scenic route, and has a whole section on Post's problem (section III), and much more detail, discussion and exercises. It's also kind of chatty, and has lots of digressions. There's even a volume II delving into complexity and covering yet more advanced material. Much the same ground is covered in chapters 6-10 of [Rog87], though this is an older work and perhaps the notation (or maybe just the typography) is a little harder to follow.

Another standard is [Soa87]. It dives straight in and gets through most of what we've done by page 80, though there are some more advanced topics interwoven even with this. Similarly [Coo03] goes through Post's work, but with an eye on how it fits in to the bigger picture.

Apparently there's a forthcoming new updated edition of [Soa87] in two volumes, but it's been "to appear" since about 2012. The author's papers on the history of computability are interesting, and closely related to the move from the terminology of "recursive" (back) to "computable" in the 1990s.

Most recently, [DH10] and [Nie12] both cover the results of Post's paper in the first 30 pages or so, so these are good if you want a quick overview, or maybe a starting point for your postgraduate research...

References

[Coo03] S. Barry Cooper. *Computability Theory*. Chapman and Hall/CRC, November 2003.

[DH10] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Theory and Applications of Computability. Springer, November 2010.

[Nie12] André Nies. *Computability and Randomness*. Number 51 in Oxford Logic Guides. OUP Oxford, Oxford; New York, May 2012.

[Odi89] Piergiorgio Odifreddi. *Classical recursion theory: the theory of functions and sets of natural numbers*. Number 125 in Studies in Logic and the Foundations of Mathematics. North-Holland, 1989.

[Rog87] Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. The MIT Press, April 1987.

[Soa87] Robert I. Soare. *Recursively enumerable sets and degrees: a study of computable functions and computably generated sets*. Springer-Verlag, 1987.

[SV03] Axexander Shen and Nikolai K. Vereshchagin. *Computable Functions*. Number 19 in Student Mathematical Library. American Mathematical Society, February 2003.