Supplementary Lecture J

Beyond Undecidability

Oracle Machines and Relative Computation

We know that virtually all interesting questions about Turing machines—whether a given TM halts on a given input, whether a given TM accepts a finite set, and so on—are undecidable. But are all these questions equally hard? For example, suppose by some magic we were given the power to decide the halting problem. Could we somehow use that power to decide if a given TM accepts a finite set? In other words, *relative to the halting problem*, is finiteness decidable?

Questions about relative computability can be formalized and studied using *oracle Turing machines*. Intuitively, an oracle TM is a TM equipped with an *oracle*, a set B to which the TM may pose membership questions and always receive correct answers after a finite time. The interesting thing about this definition is that it makes sense even if B is not recursive.

Formally, an *oracle Turing machine* is a TM that in addition to its ordinary read/write tape is equipped with a special one-way-infinite read-only input tape on which some infinite string is written. The extra tape is called the *oracle tape*, and the string written on it is called the *oracle*. The machine can move its oracle tape head one cell in either direction in each step and make decisions based on the symbols written on the oracle tape. Other than that, it behaves exactly like an ordinary Turing machine.

D. C. Kozen, *Automata and Computability*

We usually think of the oracle as a specification of a set of strings. If the oracle is an infinite string over $\{0,1\}$, then we can regard it as the characteristic function of a set $B \subseteq \mathbb{N}$, where the nth bit of the oracle string is 1 iff $n \in B$. In that way we can study computation relative to the set B.

There is nothing mysterious about oracle TMs. They operate exactly like ordinary TMs, the only difference being the oracle. Ordinary TMs are equivalent to oracle TMs with the null oracle $\varnothing$, whose characteristic function is $00000\cdots$; for such machines, the oracle gives no extra information that the TM doesn't already have.

For $A, B \subseteq \Sigma^*$, we say that A is *recursively enumerable (r.e.) in* B if there is an oracle TM M with oracle B such that $A = L(M)$. In addition, if M is total (i.e., halts on all inputs), we write $A \leq_T B$ and say that A is *recursive in* B or that A *Turing reduces to* B.

For example, the halting problem is recursive in the membership problem, since halting is decidable in the presence of an oracle for membership. Here's how: given a TM M and input x, first ask the oracle whether M accepts x. If the answer is yes, then M certainly halts on x. If the answer is no, switch accept and reject states of M to get the machine M', then ask the oracle whether M' accepts x. If the answer is yes, then M rejects x, therefore halts on x. If the answer is still no, then M neither accepts or rejects x, therefore loops on x. In all cases we can say definitively after a finite time whether M halts on x.

Likewise, the membership problem is recursive in the halting problem, since we can determine membership in the presence of an oracle for halting. Given a TM M and input x, modify M so as never to reject by making the reject state r into a nonreject state. You can add a new dummy inaccessible reject state if you like. Call this modified machine M'. Now on any input, M' accepts iff it halts, and $L(M) = L(M')$, so we can determine whether M accepts x by asking the oracle whether the modified machine M' halts on x.

It is not hard to show that the relation $\leq_T$ is transitive; that is, if A is recursive in B and B is recursive in C, then A is recursive in C. Moreover, the relation $\leq_m$ refines $\leq_T$; in other words, if $A \leq_m B$, then $A \leq_T B$ (Miscellaneous Exercise 141).

The relation $\leq_T$ is strictly coarser than $\leq_m$, since $\sim\text{HP} \not\leq_m \text{HP}$ but $\sim\text{HP} \leq_T \text{HP}$. In fact, any set A Turing reduces to its complement, since with an oracle for A, on input x one can simply ask the oracle whether $x \in A$, accepting if not and rejecting if so.

The Arithmetic Hierarchy

Once we have the notion of relative computation, we can define a hierarchy of classes as follows. Fix the alphabet $\{0,1\}$ and identify strings in $\{0,1\}^*$ with the natural numbers according to the one-to-one correspondence (28.1). Define

$$\begin{aligned}
\Sigma_1^0 &\stackrel{\text{def}}{=} \{\text{r.e. sets}\},\\
\Delta_1^0 &\stackrel{\text{def}}{=} \{\text{recursive sets}\},\\
\Sigma_{n+1}^0 &\stackrel{\text{def}}{=} \{\text{sets r.e. in some } B \in \Sigma_n^0\},\\
\Delta_{n+1}^0 &\stackrel{\text{def}}{=} \{\text{sets recursive in some } B \in \Sigma_n^0\},\\
\Pi_n^0 &\stackrel{\text{def}}{=} \{\text{complements of sets in } \Sigma_n^0\}.
\end{aligned}$$

Thus Π_1^0 is the class of co-r.e. sets. The classes Σ_n^0, Π_n^0, and Δ_n^0 comprise what is known as the *arithmetic hierarchy.*

Here is perhaps a more revealing characterization of the arithmetic hierarchy in terms of alternation of quantifiers. Recall from Exercise 1 of Homework 11 that a set A is r.e. iff there exists a decidable binary predicate R such that

$$A = \{x \mid \exists y\ R(x,y)\}. \tag{J.1}$$

For example,

$$\begin{aligned}
\text{HP} &= \{M\#x \mid \exists t\ M \text{ halts on } x \text{ in } t \text{ steps}\},\\
\text{MP} &= \{M\#x \mid \exists t\ M \text{ accepts } x \text{ in } t \text{ steps}\}.
\end{aligned}$$

Note that the predicate "M halts on x" is not decidable, but the predicate "M halts on x in t steps" is, since we can just simulate M on input x with a universal machine for t steps and see if it halts within that time. Alternatively,

$$\begin{aligned}
\text{HP} &= \{M\#x \mid \exists v\ v \text{ is a halting computation history of } M \text{ on } x\},\\
\text{MP} &= \{M\#x \mid \exists v\ v \text{ is an accepting computation history of } M \text{ on } x\}.
\end{aligned}$$

Thus the class Σ_1^0 is the family of all sets that can be expressed in the form (J.1).

Similarly, it follows from elementary logic that Π_1^0, the family of co-r.e. sets, is the class of all sets A for which there exists a decidable binary predicate R such that

$$A = \{x \mid \forall y\ R(x,y)\}. \tag{J.2}$$

We argued in Lecture 29 that a set is recursive iff it is both r.e. and co-r.e. In terms of our new notation,

$$\Delta_1^0 = \Sigma_1^0 \cap \Pi_1^0.$$

These results are special cases of the following theorem.

Theorem J.1 *(i) A set A is in Σ_n^0 iff there exists a decidable $(n+1)$-ary predicate R such that*

$$A = \{x \mid \exists y_1\ \forall y_2\ \exists y_3\ \ldots\ \mathrm{Q}y_n\ R(x, y_1, \ldots, y_n)\},$$

where $\mathrm{Q} = \exists$ if n is odd, $\forall$ if n is even.

(ii) A set A is in Π_n^0 iff there exists a decidable $(n+1)$-ary predicate R such that

$$A = \{x \mid \forall y_1\ \exists y_2\ \forall y_3\ \ldots\ \mathrm{Q}y_n\ R(x, y_1, \ldots, y_n)\},$$

where $\mathrm{Q} = \forall$ if n is odd, $\exists$ if n is even.

(iii) $\Delta_n^0 = \Sigma_n^0 \cap \Pi_n^0$.

Proof. Miscellaneous Exercise 137. □

Example J.2 The set EMPTY $\stackrel{\text{def}}{=} \{M \mid L(M) = \varnothing\}$ is in Π_1^0, since

$$\text{EMPTY} = \{M \mid \forall x\ \forall t\ M \text{ does not accept } x \text{ in } t \text{ steps}\}.$$

The two universal quantifiers $\forall x\ \forall t$ can be combined into one using the computable one-to-one pairing function $\mathbb{N}^2 \to \mathbb{N}$ given by

$$(i, j) \mapsto \binom{i+j+1}{2} + i. \tag{J.3}$$

i \ j	0	1	2	3	4	5
0	0	1	3	6	10	15
1	2	4	7	11	16	
2	5	8	12	17		
3	9	13	18			
4	14	19		⋱		
5	20					

□

Example J.3 The set TOTAL $\stackrel{\text{def}}{=} \{M \mid M \text{ is total}\}$ is in Π_2^0, since

$$\text{TOTAL} = \{M \mid \forall x\ \exists t\ M \text{ halts on } x \text{ in } t \text{ steps}\}.$$

□

Example J.4 The set $\text{FIN} \stackrel{\text{def}}{=} \{M \mid L(M) \text{ is finite}\}$ is in Σ_2^0, since

$$\begin{aligned}\text{FIN} &= \{M \mid \exists n\ \forall x \text{ if } |x| > n \text{ then } x \notin L(M)\} \\ &= \{M \mid \exists n\ \forall x\ \forall t\ |x| \leq n \text{ or } M \text{ does not accept } x \text{ in } t \text{ steps}\}.\end{aligned}$$

Again, the two universal quantifiers $\forall x\ \forall t$ can be combined into one using (J.3). □

Example J.5 A set is *cofinite* if its complement is finite. The set

$$\text{COF} \stackrel{\text{def}}{=} \{M \mid L(M) \text{ is cofinite}\}$$

is in Σ_3^0, since

$$\begin{aligned}\text{COF} &= \{M \mid \exists n\ \forall x \text{ if } |x| > n \text{ then } x \in L(M)\} \\ &= \{M \mid \exists n\ \forall x\ \exists t\ |x| \leq n \text{ or } M \text{ accepts } x \text{ in } t \text{ steps}\}.\end{aligned}$$ □

Figure J.1 depicts the inclusions among the few lowest levels of the hierarchy. Each level of the hierarchy is strictly contained in the next; that is, $\Sigma_n^0 \cup \Pi_n^0 \subseteq \Delta_{n+1}^0$, but $\Sigma_n^0 \cup \Pi_n^0 \neq \Delta_{n+1}^0$. We have shown that there exist r.e. sets that are not co-r.e. (HP, for example) and co-r.e. sets that are not r.e. ($\sim$HP, for example). Thus Σ_1^0 and Π_1^0 are incomparable with respect to set inclusion. One can show in the same way that Σ_n^0 and Π_n^0 are incomparable with respect to set inclusion for any n (Miscellaneous Exercise 138).

Completeness

The membership problem $\text{MP} \stackrel{\text{def}}{=} \{M\#x \mid M \text{ accepts } x\}$ is not only undecidable but is in a sense a "hardest" r.e. set, since every other r.e. set $\leq_{\text{m}}$-reduces to it: for any Turing machine M, the map

$$x \mapsto M\#x \tag{J.4}$$

is a trivially computable map reducing $L(M)$ to MP.

We say that a set is *r.e.-hard* if every r.e. set $\leq_{\text{m}}$-reduces to it. In other words, the set B is *r.e.-hard* if for all r.e. sets A, $A \leq_{\text{m}} B$. As just observed, the membership problem MP is r.e.-hard. So is any other problem to which the membership problem $\leq_{\text{m}}$-reduces (e.g., the halting problem HP), because the relation $\leq_{\text{m}}$ is transitive.

A set B is said to be *r.e.-complete* if it is both an r.e. set and r.e.-hard. For example, both MP and HP are r.e.-complete.

More generally, if $\mathcal{C}$ is a class of sets, we say that a set B is $\leq_{\text{m}}$*-hard for* $\mathcal{C}$ (or just $\mathcal{C}$-hard) if $A \leq_{\text{m}} B$ for all $A \in \mathcal{C}$. We say that B is $\leq_{\text{m}}$*-complete for* $\mathcal{C}$ (or just $\mathcal{C}$-complete) if B is $\leq_{\text{m}}$-hard for $\mathcal{C}$ and $B \in \mathcal{C}$.

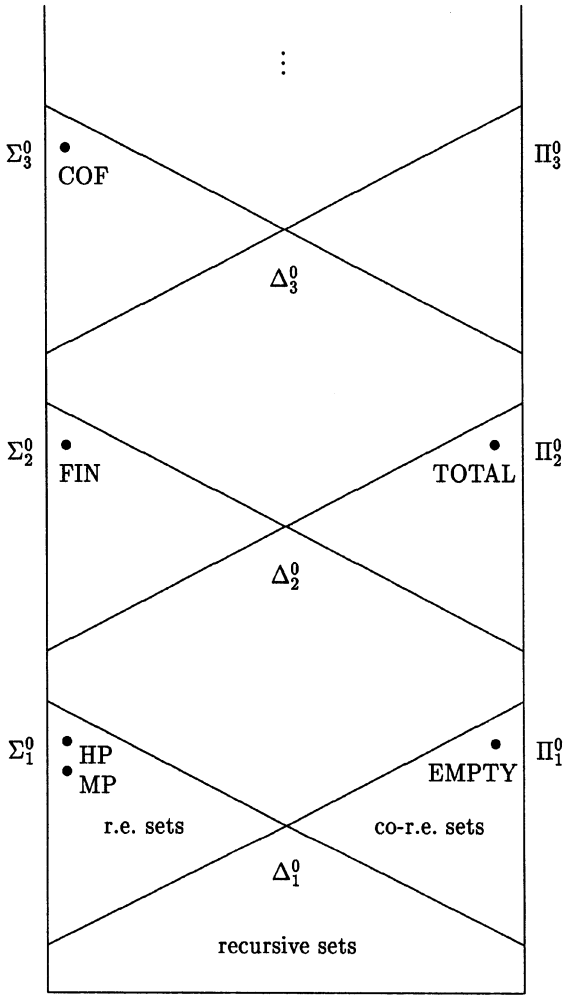

Figure J.1. The Arithmetic Hierarchy

One can prove a theorem corresponding to Theorem 33.3 that says that if $A \leq_m B$ and $B \in \Sigma_n^0$, then $A \in \Sigma_n^0$, and if $A \leq_m B$ and $B \in \Delta_n^0$, then $A \in \Delta_n^0$. Since we know that the hierarchy is strict (each level is properly contained in the next), if B is $\leq_m$-complete for Σ_n^0, then $B \notin \Pi_n^0$ (or Δ_n^0 or Σ_{n-1}^0).

It turns out that each of the problems mentioned above is $\leq_m$-complete for the level of the hierarchy in which it naturally falls:

(i) HP is $\leq_m$-complete for Σ_1^0,

(ii) MP is $\leq_m$-complete for Σ_1^0,

(iii) EMPTY is $\leq_m$-complete for Π_1^0,

(iv) TOTAL is $\leq_m$-complete for Π_2^0,

(v) FIN is $\leq_m$-complete for Σ_2^0, and

(vi) COF is $\leq_m$-complete for Σ_3^0.

Since the hierarchy is strict, none of these problems is contained in any class lower in the hierarchy or $\leq_T$-reduces to any problem complete for any class lower in the hierarchy. If it did, then the hierarchy would collapse at that level. For example, EMPTY does not reduce to HP and COF does not reduce to FIN.

We prove (v); the others we leave as exercises (Miscellaneous Exercise 142). We have already argued that FIN $\in \Sigma_2^0$. To show that it is $\leq_m$-hard for Σ_2^0, we need to show that any set in Σ_2^0 reduces to it. We use the characterization of Theorem J.1. Let

$$A = \{x \mid \exists y\ \forall z\ R(x, y, z)\}$$

be an arbitrary set in Σ_2^0, where $R(x, y, z)$ is a decidable ternary predicate. Let M be a total machine that decides R. We need to construct a machine N effectively from a given x such that $N \in$ FIN iff $x \in A$; in other words, N accepts a finite set iff $\exists y\ \forall z\ R(x, y, z)$. Let N on input w

(i) write down all strings y of length at most $|w|$;

(ii) for each such y, try to find a z such that $\neg R(x, y, z)$ (i.e., such that M rejects $x\#y\#z$), and accept if all these trials are successful. The machine N has x and a description of M hard-wired in its finite control.

In step (ii), for each y of length at most $|w|$, N can just enumerate strings z in some order and run M on $x\#y\#z$ until some z is found causing M to reject. Since M is total, N need not worry about timesharing. If no such z is ever found, N just goes on forever. Surely such an N can be built effectively from M and x.

Now if $x \in A$, then there exists y such that for all z, $R(x, y, z)$ (i.e., for all z, M accepts $x\#y\#z$); thus step (ii) fails whenever $|w| \geq |y|$. In this case N accepts a finite set. On the other hand, if $x \notin A$, then for all y there exists a z such that $\neg R(x, y, z)$, and these are all found in step (ii). In this case, N accepts Σ^*.

We have shown that the machine N accepts a finite set iff $x \in A$, therefore the map $x \mapsto N$ constitutes a $\leq_{\mathrm{m}}$-reduction from A to FIN. Since A was an arbitrary element of Σ_2^0, FIN is $\leq_{\mathrm{m}}$-hard for Σ_2^0.

The Analytic Hierarchy and Π_1^1

The arithmetic hierarchy is defined in terms of *first-order quantification*, or quantification over natural numbers or strings. But it doesn't stop there: if we consider *second-order quantification*—quantification over functions and relations—we get the so-called *analytic hierarchy* consisting of classes Σ_n^1, Π_n^1, Δ_n^1. The entire arithmetic hierarchy is strictly contained in Δ_1^1, the lowest class in the analytic hierarchy. Elements of Δ_1^1 are called *hyperarithmetic sets*.

A remarkable theorem due to Kleene says that the sets of natural numbers definable by one universal second-order quantifier (i.e., the Π_1^1 sets) are exactly the sets definable by first-order induction.

The class Π_1^1 also has natural complete problems. For example, suppose we are given a recursive binary relation $\prec$ on $\mathbb{N}$; that is, a recursive subset of $\mathbb{N}^2$. A natural question to ask is whether the relation is *well founded*; that is, whether there exists no infinite descending chain

$$n_0 \succ n_1 \succ n_2 \succ \cdots.$$

This decision problem is $\leq_{\mathrm{m}}$-complete for Π_1^1.

These results are getting a bit beyond our scope, so we'll stop here.

Historical Notes

Oracle Turing machines were first defined by Turing [121]. The arithmetic and analytic hierarchies were studied by Kleene [68]; see also Rogers [106], Shoenfield [115], Kleene [69], and Soare [116].

Modern-day complexity theory has its roots in the theory of recursive functions and effective computability. The $\leq_{\mathrm{T}}$- and $\leq_{\mathrm{m}}$-reducibility relations, the concepts of completeness and hardness, and the arithmetic hierarchy all have their subrecursive counterparts; see Karp [64], Cook [28], and Stockmeyer [118]. For an introduction to complexity theory, see Hartmanis and Stearns [57], Garey and Johnson [40], or Papadimitriou [97].