TURING DEGREES

YUZHOU GU

1. Introduction

Turing degrees, defined by Post [Pos44], measure the degree of undecidability of sets. All recursive sets can be decided by a Turing machine, so they have the same Turing degree. The halting set, which is the set of pairs (Turing machine, input) that halt, is an undecidable set, so it has a Turing degree harder than all recursive sets.

The set of all Turing degrees has a natural structure of a poset, where the partial order is given by Turing reducibility. This poset has many interesting properties. For example, Kleene and Post [KP54] proved, among many other results, that there exist two incomparable Turing degrees. They used a very powerful method called the finite extension method.

An interesting subset of Turing degrees is the set of r.e. degrees, which are degrees that are Turing equivalent to some r.e. sets. Post [Pos44] asked whether there are r.e. degrees that are not recursive nor Turing equivalent to the halting set. Friedberg [Fri57] and Muchnik [Muc56] developed the finite injury priority method and solved Post's problem.

In this expository paper we study the finite extension method and the finite injury priority method, and prove many interesting results using them.

The main references for the finite extension method are Odifreddi [Odi92] and Shore [Sho13]. The main references for the finite injury priority method are Soare [Soa76] and Soare [Soa16].

2. Preliminaries

In this section we give basic definitions and state basic properties of Turing degrees.

The basic objects of study in recursion theory are sets of natural numbers. They can also be interpreted as infinite boolean strings. In the following, we do not make distinction between sets of natural numbers and infinite boolean strings. A set of natural numbers is often just called a set.

Definition 2.1 (Computation)**.** Let A be a boolean string (which can be finite or infinite). Let e, s, x be natural numbers. Let M_e be the Turing machine with oracle A encoded by e.

We say $\Phi^A_{e,s}(x)\downarrow$ if M_e run on input x halts in no more than s steps. We write $\Phi^A_e(x)\downarrow$ for $\exists s(\Phi^A_{e,s}(x)\downarrow)$. We write $\Phi^A_e(x)\uparrow$ if M_e run on input x loops forever.

If $\Phi^A_e(x)\downarrow$, then we write $\phi^A_e(x)$ for the output (a boolean value) of M_e run on input x. If for all x, we have $\Phi^A_e(x)\downarrow$, then we write ϕ^A_e for the infinite boolean string whose x-th entry is $\phi^A_e(x)$.

When A is the empty string, we omit superscripts in the above definitions.

Remark 2.2. Note that $\Phi_e^A \uparrow$ is not the same as $\neg\Phi_e^A \downarrow$. This is because M_e may access some x such that $A(x)$ not defined. In this case M_e is broken and does not halt nor loop forever.

When A is complete, i.e., $A(x)$ is defined for every x, we have $\Phi_e^A \uparrow \Leftrightarrow \neg\Phi_e^A \downarrow$.

Definition 2.3 (Turing reducibility)**.** Let A, B be sets. We say A is Turing reducible to B (denoted as $A \leq_T B$) if there exists e such that $\phi_e^B = A$. We say A is Turing equivalent to B (denoted as $A \equiv_T B$) if $A \leq_T B$ and $B \leq_T A$.

Definition 2.4 (Turing degrees)**.** A Turing degree is an equivalence class under Turing equivalence.

Remark 2.5. As we will see, many natural operations on sets preserve Turing equivalence, so we will not make big distinction between sets and Turing degrees.

Remark 2.6. There are other reductions: many-one reduction, truth-table reduction, etc. Each of them gives rise to a different definition of degrees. In this paper we only study Turing reductions and Turing degrees, so sometimes we omit the word "Turing" without ambiguity.

Let $\mathcal{D}$ denote the poset of Turing degrees with partial order $\leq_T$. Let 0 denote the degree of recursive sets. It is the minimum element of $\mathcal{D}$.

We list some cardinality properties.

Proposition 2.7. *(1) Every Turing degree contains countably many sets.*

(2) There are $2^{\aleph_0}$ different Turing degrees.

(3) For every Turing degree A, there are at most countably many Turing degrees B such that $B \leq_T A$.

Proof. (1) Let A be a set. There are at most countably many B such that $B \equiv_T A$ because each B is ϕ_e^A for some e, and the set of different e's is countable. There are at least countably many such B's because for each $x \in \mathbb{N}$, $A\Delta\{x\} \equiv_T A$ (where Δ is symmetric difference).

(2) There are $2^{\aleph_0}$ sets, and each Turing degree contains countably many sets.

(3) For a set A, there are countably many B such that $B \leq_T A$. □

Definition 2.8 (Join)**.** Let A, B be two sets. Define the join $A \oplus B = \{2x : x \in A\} \cup \{2y+1 : y \in B\}$.

This operation naturally extends to Turing degrees.

Proposition 2.9. *$\mathcal{D}$ is an upper semilattice with join $\oplus$; i.e., for any sets A, B, C, we have $(A \leq_T C \wedge B \leq_T C) \Leftrightarrow A \oplus B \leq_T C$.*

The proof is rather obvious.

Remark 2.10. For a countable list of sets $\{A_i\}_{i\in\mathbb{N}}$, we can also define their join $\bigoplus_{i\in\mathbb{N}} A_i = \{\langle i, x\rangle : x \in A_i\}$ where $\langle -, -\rangle : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ is a recursive bijection.

However, countable join does not extend to Turing degrees; i.e., there exist lists $\{A_i\}_{i\in\mathbb{N}}$, $\{B_i\}_{i\in\mathbb{N}}$ such that $A_i \equiv_T B_i$ for all $i \in \mathbb{N}$, but $\bigoplus_{i\in\mathbb{N}} A_i \not\equiv_T \bigoplus_{i\in\mathbb{N}} B_i$. (If $\{e_i\}_{i\in\mathbb{N}}$ is a list of natural numbers such that $A_i = \phi_{e_i}^{B_i}$, then it is not always true that we can combine them to get a natural number e such that $\bigoplus_{i\in\mathbb{N}} A_i = \phi_e^{\bigoplus_{i\in\mathbb{N}} B_i}$. This is because the list $\{e_i\}_{\in\mathbb{N}}$ may not be uniform, i.e., generated by a single Turing machine.)

Also, in $\mathcal{D}$, the degree of $\bigoplus_{i\in\mathbb{N}} A_i$ is not the join of the degree of A_i for $i \in \mathbb{N}$. Actually, Corollary 3.11 shows that $\mathcal{D}$ does not admit arbitrary countable joins.

There is a special operator on $\mathcal{D}$.

Definition 2.11 (Jump operator)**.** Let A be a set. Define $A' = \{e : \Phi_e^A(e)\downarrow\}$. Then A' is called the jump of A.

This operation naturally extends to Turing degrees.

Remark 2.12. The jump A' of a set A is Turing equivalent to the problem of deciding whether a Turing machine with oracle A halts on a given input. In particular, $0'$ is the degree of the halting problem.

Proposition 2.13. *For any set A, $A <_T A'$; i.e, $A \leq_T A'$ and $A' \not\leq_T A$.*

Proof. That $A \leq_T A'$ is obvious from the previous remark. We only prove that $A' \not\leq_T A$.

Assume $A' \leq_T A$. Then there exists e_0 such that $\phi_{e_0}^A = A'$. So for any e, $\Phi_{e_0}^A(e) = 1 \Leftrightarrow \Phi_e^A(e)\downarrow$. Using e_0, we can construct a Turing machine with oracle A that on input e halts if $\Phi_e^A(e)\uparrow$, and loops forever if $\Phi_e^A(e)\downarrow$. Let e_1 be its index number. Then we have $\Phi_{e_1}^A(e)\downarrow \Leftrightarrow \Phi_e^A(e)\uparrow$. Taking $e = e_1$, we get $\Phi_{e_1}^A(e_1)\downarrow \Leftrightarrow \Phi_{e_1}^A(e_1)\uparrow$, which is a contradiction. $\square$

Definition 2.14. A set A is called low if $A' \leq_T 0'$.

A set A is low if and only if $A' \equiv_T 0'$, because $0' \leq_T A'$ holds for all sets A.

Finally we define r.e. degrees.

Definition 2.15. A Turing degree is called an r.e. degree if it contains an r.e. set.

Remark 2.16. Clearly every r.e. degree A satisfies $A \leq_T 0'$. The converse is not true: there are degrees A such that $A \leq_T 0'$ but A is not r.e.

3. Finite extension method

Assume we have a countable list of requirements that we would like to satisfy. In the finite extension method, we start with the empty string, and in each step, we extend the string we have so that more requirements are satisfied, and previously satisfied requirements are still satisfied.

Let the requirements be $\{R_i\}_{i\in\mathbb{N}}$. We start with the empty string $A_0 = \emptyset$, and at step s, we construct a finite string A_{s+1} that contains A_s, and makes sure that A_{s+1} satisfies R_i for $i \leq s+1$. In the end, we take $A = \bigcup_i A_i$, and this string satisfies all requirements. (In the simplest cases, A_s is a prefix of A_{s+1}. For full generality, we allow a string to have three possible values in each position: 0, 1, or undefined. That A_{s+1} contains A_s means for each position x, if $A_s(x)$ is defined, then $A_{s+1}(x)$ is defined and $A_s(x) = A_{s+1}(x)$. So here strings are actually understood as partial functions.)

We illustrate the finite extension method with a standard example.

Theorem 3.1 (Kleene-Post [KP54])**.** *There exist two incomparable sets A and B, i.e., $A \not\leq_T B$ and $B \not\leq_T A$.*

Proof. We need to satisfy a list of requirements $R_{2e} : \phi_e^A \neq B$, and $R_{2e+1} : \phi_e^B \neq A$. (In the case ϕ_e^A is undefined (i.e., $\Phi_e^A(x)\uparrow$ for some x), the requirement R_{2e} is considered to be automatically satisfied.)

We start with $A_0 = B_0 = \emptyset$.

Step $s = 2e$: We construct A_{s+1} and B_{s+1} from A_s and B_s. Choose some x such that $B_s(x)$ is undefined. If there is a finite extension A_{s+1} of A_s such that $\Phi_e^{A_{s+1}}(x)\downarrow$, then we fix this A_{s+1}, and let B_{s+1} extend B_s by $B_{s+1}(x) = 1-\phi_e^{A_{s+1}}(x)$. Therefore in this case, R_s is satisfied.

If there does not exist such an extension, let $A_{s+1} = A_s$ and $B_{s+1}(x)$ be an arbitrary value (0 or 1). In this case R_s is also satisfied because $\Phi_e^A(x)\uparrow$ for any extension A of A_s.

Step $s = 2e+1$: The construction is similar (with A and B exchanged).

Let $A = \bigcup_s A_s$ and $B = \bigcup_s B_s$. From the analysis above, all requirements are satisfied. So we get the desired sets. □

Remark 3.2. We can strengthen the conditions on A and B.

Note that in step R_{2e}, we only need to answer the question "is there a finite extension A_{2e+1} of A_{2e} such that $\Phi_e^{A_{2e+1}}(x)\downarrow$?", which can be answered with oracle $0'$, the degree of the halting problem. So $A, B \leq_T 0'$.

We can make A and B low (Definition 2.14) by adding additional requirements. This is a technique called forcing the jump, which will be introduced in Theorem 3.12.

We can make A and B r.e. sets. This needs the finite injury priority method, which will be introduced in the next section.

Theorem 3.1 can be easily extended to countably many sets.

Theorem 3.3 (Kleene-Post [KP54])**.** *There exists a sequence of sets $\{A_i\}_{i\in\mathbb{N}}$ such that for all $i \in \mathbb{N}$, we have $A_i \not\leq_T \bigoplus_{j\neq i} A_j$.*

Proof. We need to satisfy countably many requirements $R_{k,e} : \phi_e^{\bigoplus_{i\neq k} A_i} \neq A_k$.

Let $\langle -,-\rangle : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ be a bijection. We start with $A_{i,0} = \emptyset$ for all $i \in \mathbb{N}$.

Step $s = \langle k, e\rangle$: We construct $\{A_{i,s+1}\}_{i\in\mathbb{N}}$ from $\{A_{i,s}\}_{i\in\mathbb{N}}$, and make sure that $\bigoplus_{i\in\mathbb{N}} A_{i,s+1}$ is a finite string.

Choose some x such that $A_{k,s}(x)$ is undefined. If there is a finite extension of $\bigoplus_{i\neq k} A_{i,s}$ such that $\Phi_e^{\bigoplus_{i\neq k} A_{i,s+1}}(x)\downarrow$, then we choose this extension, and let $A_{k,s+1}$ extend $A_{k,s}$ by $A_{k,s+1}(x) = 1-\phi_e^{\bigoplus_{i\neq k} A_{i,s+1}}(x)$. In this case $R_{k,e}$ is satisfied.

If there does not exist such an extension, then let $A_{i,s+1} = A_{i,s}$ for all $i \neq k$, and $A_{k,s+1}(x)$ be an arbitrary value. In this case $R_{k,e}$ is also satisfied because $\phi^{\bigoplus_{i\neq k} A_i}\uparrow$ for any extensions A_i of $A_{i,s}$.

Finally, let $A_i = \bigcup_s A_{i,s}$. From the analysis above, all requirements are satisfied. So we get the desired sets. □

Theorem 3.3 can be seen as an embedding of the countable discrete poset into $\mathcal{D}$. In fact, it can be used to show that every countable poset can be embedded into $\mathcal{D}$. We need the following lemma.

Lemma 3.4 (Mostowski [Mos38])**.** *There exists a recursive countable poset $\mathcal{P}$ such that every countable poset can be embedded into $\mathcal{P}$.*

Proof. We construct a list of finite approximations of $\mathcal{P}$.

Let $\mathcal{P}_0$ be the empty poset. At step i, we construct $\mathcal{P}_{i+1}$ from $\mathcal{P}_i$. Consider every possible order relation between a new element and elements in $\mathcal{P}_i$. (There can be at most $3^{|\mathcal{P}_i|}$ possible relations.) For each of them, we add a new element in $\mathcal{P}_{i+1}$ with this order relation. Finally, we add relations between new elements required by poset axioms.

Clearly $\mathcal{P}_{i+1}$ extends $\mathcal{P}_i$. Let $\mathcal{P} = \bigcup_i \mathcal{P}_i$. Clearly $\mathcal{P}$ is recursive. We prove that every countable poset $\mathcal{Q}$ can be embedded into $\mathcal{P}$.

We list the elements of $\mathcal{Q}$ as $q_1, q_2, \cdots$, and put them into $\mathcal{P}$ one by one. Say $p_i \in \mathcal{P}$ corresponds to q_i. We choose p_i in such a way that $p_i \in \mathcal{P}_i \backslash \mathcal{P}_{i-1}$. In each step, if we have chosen $p_1, \ldots, p_i$, we let $p_{i+1} \in \mathcal{P}_{i+1} \backslash \mathcal{P}_i$ be such that the order relation between p_{i+1} and $p_1, \ldots, p_i$ is exactly the same as the order relation between q_{i+1} and $q_1, \ldots, q_i$. By the construction of $\mathcal{P}_{i+1}$, such p_{i+1} can be found. □

Theorem 3.5 (Sacks [Sac63a]). *Every countable poset can be embedded into $\mathcal{D}$.*

Proof. Let $\mathcal{P}$ be the poset constructed in Lemma 3.4. We only need to prove that $\mathcal{P}$ can be embedded into $\mathcal{D}$. Actually we only need the fact that $\mathcal{P}$ is recursive.

Let $\{A_i\}_{i \in \mathbb{N}}$ be as in Theorem 3.3. Let the elements of $\mathcal{P}$ be $\{p_i\}_{i \in \mathbb{N}}$ and the partial order be $\leq_P$. Define $B_i = \bigoplus_{p_j \leq_P p_i} A_j$.

For $p_j \leq_P p_i$, we have $B_j \leq_T B_i$ by definition of the B_i's (and the fact that $\mathcal{P}$ is recursive). For $p_j \not\leq_P p_i$, we have $B_j \not\leq_T B_i$, because $A_j \leq_T B_j$, $B_i \leq_T \bigoplus_{k \neq j} A_k$, and $A_j \not\leq_T \bigoplus_{k \neq j} A_k$. So $\{B_i\}_{i \in \mathbb{N}}$ defines an embedding of $\mathcal{P}$ into $\mathcal{D}$. □

In the above examples, all strings are initially empty. The following theorem shows that some strings can be fixed in the beginning.

Theorem 3.6 (Kleene-Post [KP54]). *For every nonrecursive set B, there exists a set A that is incomparable with B.*

Proof. We need to satisfy the requirements $R_{2e} : \phi_e^A \neq B$, and $R_{2e+1} : \phi_e^B \neq A$. We start with $A_0 = \emptyset$.

Step $s = 2e+1$: Choose x such that $A_s(x)$ is undefined. If $\Phi_e^B(x)\downarrow$, let A_{s+1} extend A_s by $A_{s+1}(x) = 1 - \phi_e^B(x)$. Otherwise, let $A_{s+1}(x)$ be an arbitrary value. In either case, R_{2e+1} is satisfied.

Step $s = 2e$: Choose some finite extension A_{s+1} of A_s and some x such that either $\Phi_e^{A_{s+1}}(x)\uparrow$ or $\phi_e^{A_{s+1}}(x) \neq B(x)$. Clearly, if such A_{s+1} and x exist, then R_{2e} is satisfied. We only need to prove their existence. Assume such A_{s+1} and x do not exist. This means that for any complete extension A of A_s and any x, $\Phi_e^A(x)\downarrow$ and $\phi_e^A(x) = B(x)$. (Recall that A is complete means that $A(x)$ is defined for every x.) This enables us to construct a Turing machine with a finite oracle A_s that decides B. This contradicts the assumption that B is nonrecursive.

Finally, take $A = \bigcup_s A_s$. By the above analysis, all requirements are satisfied. □

The following theorem states the existence of a minimal pair. Although the requirements look different from previous examples, we can actually use the same argument to deal with them.

Theorem 3.7 (Kleene-Post [KP54]). *For every nonrecursive set B, there exists a nonrecursive set A such that if $C \leq_T A$ and $C \leq_T B$, then $C \equiv_T 0$.*

Proof. We need to satisfy the requirements $R_{2e} : \phi_e \neq A$ and $R_{2\langle e,f\rangle+1} : \phi_e^A = \phi_f^B = C \Rightarrow C \equiv_T 0$. We start with $A_0 = \emptyset$.

Step $s = 2e$: Choose x such that $A_s(x)$ is undefined. If $\Phi_e(x)\downarrow$, let A_{s+1} extend A_s by $A_{s+1}(x) = 1 - \phi_e(x)$. Otherwise, let $A_{s+1}(x)$ be an arbitrary value. In either case, R_{2e} is satisfied.

Step $s = 2\langle e, f\rangle + 1$: If we can choose some finite extension A_{s+1} of A_s and some x such that either $\Phi_e^{A_{s+1}}(x)\uparrow$, or $\Phi_f^B(x)\uparrow$, or $\phi_e^{A_{s+1}}(x) \neq \phi_f^B(x)$, then A_{s+1} satisfies R_s (because such C does not exist). So we only need to consider the case where such A_{s+1} and x do not exist. This means that for any complete extension A of A_s and any x, $\Phi_e^A(x)\downarrow$, $\Phi_f^B(x)\downarrow$, and $\phi_e^A(x) = \phi_f^B(x)$. This enables us to construct a Turing machine with a finite oracle A_s that decides $C = \phi_f^B$. So C is recursive.

Finally, take $A = \bigcup_s A_s$. By the above analysis, all requirements are satisfied. $\square$

The following theorem shows an amazing property of $\mathcal{D}$. Its proof uses a generalization of the finite extension method, called the coinfinite extension method. In the coinfinite extension method, in each step we make sure that the set $\{x : A_s(x) \text{ is undefined}\}$ is infinite and recursive. So A_s itself can have infinitely many positions defined.

Theorem 3.8 (Spector [Spe56]). *For every countable ascending sequence $C_0 \leq_T C_2 \leq_T \cdots$, there exist A, B such that for any D, we have*

$$(D \leq_T A \wedge D \leq_T B) \Leftrightarrow \exists n(D \leq_T C_n).$$

Proof. We need to satisfy the following requirements: $R_{2n} : C_n \leq_T A \wedge C_n \leq_T B$ and $R_{2\langle e,f\rangle+1} : \phi_e^A = \phi_f^B = C \Rightarrow \exists n(C \leq_T C_n)$. We start with $A_0 = B_0 = \emptyset$.

Step $s = 2n$: Choose a recursive, infinite, and coinfinite subset $\{x_i\}_{i\in\mathbb{N}}$ of $\{x : A_s(x) \text{ is undefined}\}$. Let A_{s+1} extend A_s by $A_{s+1}(x_i) = C_n(i)$. We have $C_n \leq_T A_{s+1}$ because $\{x_i\}_{i\in\mathbb{N}}$ is recursive. Construct B_{s+1} similarly.

Step $s = 2\langle e, f\rangle + 1$: If we can find extensions A_{s+1} of A_s, B_{s+1} of B_s, and a natural number x such that

(1) $\{y : A_{s+1}(y) \text{ is defined and } A_s(y) \text{ is undefined}\}$ is finite;
(2) $\{y : B_{s+1}(y) \text{ is defined and } B_s(y) \text{ is undefined}\}$ is finite;
(3) either $\Phi_e^{A_{s+1}}(x)\uparrow$, or $\Phi_f^{B_{s+1}}(x)\uparrow$, or $\phi_e^{A_{s+1}}(x) \neq \phi_f^{B_{s+1}}(x)$,

then they satisfy R_s because such C does not exist.

So we only need to consider the case where such A_{s+1}, B_{s+1}, and x do not exist. This means that for any complete extension A of A_s, complete extension B of B_s and natural number x, we have $\Phi_e^A(x)\downarrow$, $\Phi_f^B(x)\downarrow$, and $\phi_e^A(x) = \phi_f^B(x)$. This enables us to construct a Turing machine with oracle A_s that decides $C = \phi_e^A = \phi_f^B$. (Note that C does not depend on the choice of A and B.) So $C \leq_T A_s$. Choose largest n such that $2n \leq s$. Then $A_s \leq_T \bigoplus_{i\leq n} C_i \equiv_T C_n$. So $C \leq_T C_n$. Thus R_s is satisfied.

Finally, we take $A = \bigcup_s A_s$ and $B = \bigcup_s B_s$. By the above analysis, all requirements are satisfied. $\square$

Corollary 3.9 (Spector [Spe56]). *Let $\mathcal{C} \subseteq \mathcal{D}$ be a countable ideal, i.e., $\mathcal{C}$ is closed under finite joins, and for all $A \in \mathcal{C}$, if $B \leq_T A$, then $B \in \mathcal{C}$. Then there exists A and B such that $\mathcal{C} = \{C : C \leq_T A \wedge C \leq_T B\}$.*

Proof. List the elements of $\mathcal{C}$ as $\{C_i\}_{i\in\mathbb{N}}$. Define $D_i = \bigoplus_{j\leq i} C_j$. Apply Theorem 3.8 to $\{D_i\}_{i\in\mathbb{N}}$. $\square$

Corollary 3.10 (Kleene-Post [KP54]). *The poset $\mathcal{D}$ of Turing degrees is not a lattice.*

Proof. Apply Theorem 3.8 to the sequence defined by $C_0 = 0$ and $C_i = C'_{i-1}$. Then there exist A and B such that $(D \leq_T A \wedge D \leq_T B) \Leftrightarrow \exists n(D \leq_T C_n)$. Assume $\mathcal{D}$ is a lattice. Then there exists E such that $(D \leq_T A \wedge D \leq_T B) \Leftrightarrow D \leq_T E$. So $D \leq_T E \Leftrightarrow \exists n(D \leq_T C_n)$. Taking $D = E$, we get $\exists n(E \leq_T C_n)$. Take m such that $E \leq_T C_m$. By Proposition 2.13, $C_{m+1} \not\leq_T E$. However, $C_{m+1} \leq_T A$ and $C_{m+1} \leq_T B$. This contradicts the fact that E is the meet of A and B. $\square$

Corollary 3.11 (Spector [Spe56])**.** *A sequence of sets $\{C_n\}_{n\in\mathbb{N}}$ has a join in $\mathcal{D}$ iff there exists some m such that $C_i \leq_T \bigoplus_{j\leq m} C_j$ for all i.*

Proof. The if part is trivial. We prove the only if part. Assume the join is E. This means for all set F, $E \leq_T F \Leftrightarrow \forall n(C_n \leq_T F)$. Define $D_i = \bigoplus_{j\leq i} C_j$. Apply Theorem 3.8 to $\{D_i\}_{i\in\mathbb{N}}$. We get sets A and B such that $(F \leq_T A \wedge F \leq_T B) \Leftrightarrow \exists n(F \leq_T D_n)$. Because for all n, $C_n \leq_T D_n \leq_T A$ and $C_n \leq_T D_n \leq_T B$, we have $E \leq_T A$ and $E \leq_T B$. So there exists an m such that $E \leq_T D_m$. Then for all n, $C_n \leq_T E \leq_T D_m$. $\square$

Next we study some properties of the jump operator. First is the existence of nonrecursive low sets.

Theorem 3.12 (Spector [Spe56])**.** *There exists a nonrecursive set A such that $A' \equiv_T 0'$.*

Proof. We only need to construct a set A such that $A \not\equiv_T 0$ and $A' \leq_T 0'$. Recall that $A' \leq_T 0'$ means that there is a Turing machine with oracle $0'$ that decides $\{e : \Phi_e^A(e)\downarrow\}$.

The requirements we have are $R_{2e} : \phi_e \neq A$ and R_{2e+1} : decide whether $\Phi_e^A(e)\downarrow$. It may appear unclear what R_{2e+1} means. It will be clarified by the construction. We start with $A_0 = \emptyset$.

Step $s = 2e$: Choose x such that $A_s(x)$ is undefined. If $\Phi_e(x)\downarrow$, let A_{s+1} extend A_s by $A_{s+1}(x) = 1 - \phi_e(x)$. Otherwise, let $A_{s+1}(x)$ be an arbitrary value. In either case, R_{2e} is satisfied.

Step $s = 2e+1$: If there exists a finite extension A_{s+1} of A_s such that $\Phi_e^{A_{s+1}}(e)\downarrow$, we choose this extension, and know that $\Phi_e^A(e)\downarrow$. Otherwise, we know that $\Phi_e^A(e)\uparrow$.

Finally we let $A = \bigcup_s A_s$. All requirements are satisfied. Because this construction only needs an $0'$ oracle, we get $A' \leq_T 0'$. $\square$

Theorem 3.12 shows that the jump operator is not injective. Another natural question to ask is what the image of the jump operator is. Clearly, any degree C in the image should satisfy $0' \leq_T C$. The following theorem shows that the converse is also true.

Theorem 3.13 (Jump inversion theorem, Spector [Spe56])**.** *For every set C such that $0' \leq_T C$, there exists a set A such that $A' \equiv_T C$.*

Proof. That $A' \equiv_T C$ contains two parts: $A' \leq_T C$ and $C \leq_T A'$. So we will construct A with an oracle for C (which also gives us an oracle for $0'$ because $0' \leq_T C$).

The requirements we have are R_{2e} : decide whether $\Phi_e^A(e)\downarrow$, and R_{2x+1} : encode $C(x)$ into A.

Step $s = 2e$: If there exists a finite extension A_{s+1} of A_s such that $\Phi_e^{A_{s+1}}(e)\downarrow$, we choose this extension, and know that $\Phi_e^A(e)\downarrow$. Otherwise, we know that $\Phi_e^A(e)\uparrow$.

Step $s = 2x + 1$: Let y be the minimum position such that $A_s(y)$ is undefined. Let A_{s+1} extend A_s by $A_{s+1}(y) = C(x)$.

Finally, let $A = \bigcup_s A_s$. The construction only needs an oracle for C, so $A' \leq_T C$. Step $s = 2e$ only needs an oracle for $0'$, so $C \leq_T A \oplus 0' \leq_T A'$. We get the desired set. $\square$

4. Finite injury priority method

The finite extension method has some limits: although the sets constructed are often Turing reducible to $0'$, they are hardly r.e. The finite injury priority method is a way to make the finite extension method r.e. We start with the empty set $A_0 = \emptyset$. In step s, we add some elements to A_s to get A_{s+1} to satisfy some new requirements. (Note that each A_s is a set instead of a finite boolean string.) However, because we cannot prevent elements from entering A_s, we may break some requirements that have been satisfied before. In this case, the broken requirement is called "injured".

In the finite injury priority method, each requirement is given a priority, such that

(1) for each requirement, there can only be finitely many requirements with higher priority;
(2) each requirement can only be injured by requirements with higher priority.

In this way, every requirement is eventually satisfied.

The use function enables us to measure when injuries occur.

Definition 4.1. Let A be a set and e, s, x be natural numbers. Define the use function $u^A_{e,s}(x)$ as

(1) if $\Phi^A_{e,s}(x)\downarrow$, $u^A_{e,s}(x) = \max\{z : A(z) \text{ is accessed during execution}\}$;
(2) otherwise, $u^A_{e,s}(x) = -1$.

At each step s, for each restriction R_e we will define a restriction function $r_{e,s}$ using the use function, and R_e is said to be injured if $A_{s+1} - A_s$ contains some $x \leq r_{e,s}$.

We illustrate the finite injury priority method with an easy example.

Theorem 4.2. *There exists a low simple set.*

Proof. Recall that a set A is simple if A is coinfinite, r.e., and for every infinite r.e. set W, $W \cap A \neq \emptyset$.

Let W_e be the r.e. set generated by the Turing machine with index e. Let $W_{e,s}$ be the set generated by the Turing machine with index e when it has run s steps. Clearly $W_{e,s+1} \supseteq W_{e,s}$ and $W_e = \bigcup_s W_{e,s}$.

The requirements we need to satisfy are R_e : decide whether $\Phi^A_e(e)\downarrow$ and P_e : W_e infinite $\Rightarrow W_e \cap A \neq \emptyset$. The coinfiniteness will be satisfied during the construction.

We first describe the way to construct A, and then prove its correctness. We start with $A_0 = \emptyset$.

Step s: We construct A_{s+1} from A_s. Compute $r_{e,s} = u^{A_s}_{e,s}(e)$ for $e \leq s$. We choose the minimum $i \leq s$ such that $W_{i,s} \cap A_s = \emptyset$ and there exists x such that $x \in W_{i,s}$, $x \geq 2i$, and for all $j \leq i$, we have $x > r_{j,s}$. If there exists such i and x, we let $A_{s+1} = A_s \cup \{x\}$. Otherwise, we let $A_{s+1} = A_s$.

Finally, let $A = \bigcup_s A_s$.

Now we prove the correctness of the construction.

(1) Each R_e is injured a finite number of times.

Proof: Requirement R_e is injured at step s if some element $x \leq r_{e,s}$ is added. By our strategy of adding elements, R_e is injured only when some W_i with $i < e$ is chosen. Each W_i is chosen at most once, so R_e is injured at most e times.

(2) For all e, $\lim_s r_{e,s}$ exists and is finite. Also, R_e is satisfied.

Proof: Choose s_0 so that R_e is not injured at step s for all $s \geq s_0$. If for some $s \geq s_0$, $\Phi_{e,s}^{A_s}(e) \downarrow$, then for all $t \geq s$, we have $\Phi_{e,t}^{A_t}(e) \downarrow$ and $r_{e,t} = r_{e,s}$. Otherwise, for all $s \geq s_0$, we have $r_{e,s} = -1$. So $\Phi_e^A(e) \downarrow$ if and only if $\exists s \geq s_0(\Phi_{e,s}^{A_s}(e) \downarrow)$, and the later can be decided with an oracle $0'$. So R_e is satisfied.

(3) For all e, P_e is satisfied.

Proof: By (2), define $r_e = \lim_s r_{e,s}$. If W_e contains an element x such that $x \geq 2e$ and $x > \max\{r_i : i \leq e\}$, then W_e is chosen at some step, and thus $W_e \cap A \neq \emptyset$. In particular, if W_e is infinite, then $W_e \cap A \neq \emptyset$.

(4) The set A is a low simple set.

Proof: The set A is r.e. because the construction is r.e. The set A is coinfinite because for each W_e we add at most one element larger than or equal to $2e$, and no elements smaller than $2e$. The set A is low by (2). By (3), for every infinite r.e. set W, we have $W \cap A \neq \emptyset$. $\square$

The following theorem is the first result proved using the finite injury priority method. It is slightly more complicated than Theorem 4.2.

Theorem 4.3 (Friedberg [Fri57], Muchnik [Muc56])**.** *There exist two incomparable r.e. sets A and B.*

Proof. We need to satisfy the requirements $R_{2e} : \phi_e^B \neq A$ and $R_{2e+1} : \phi_e^A \neq B$.

We say x is a witness for R_{2e} at step s if $\Phi_{e,s}^{B_s}(x) \downarrow$, and $\phi_{e,s}^{B_s}(x) \neq A_s(x)$. Similarly, we define witnesses for R_{2e+1} at step s.

In each step s, we compute $x_{i,s+1}$ and $r_{i,s+1}$, where $x_{i,s+1}$ is a witness for R_i at step $s+1$, and $r_{i,s+1}$ is the restriction function for step $i+1$. That is, in each step we compute the restriction function for the next step. That $x_{i,s+1} = r_{i,s+1} = -1$ indicates that we do not know a witness yet. We maintain that for all $i \geq s$, we have $x_{i,s} = r_{i,s} = -1$.

We start with $A_0 = B_0 = \emptyset$ and $x_{i,0} = r_{i,0} = -1$ for all i.

Step s: We construct A_{s+1}, B_{s+1}, and compute $x_{i,s+1}$ and $r_{i,s+1}$. We work for each i from 0 to s. In the following, fix $i = 2e$ (the case $i = 2e+1$ is similar).

(a) If $x_{i,s} \neq -1$, let $x_{i,s+1} = x_{i,s}$ and $r_{i,s+1} = r_{i,s}$. Skip step (b) and (c).

(b) Assume $x_{i,s} = -1$. Let x be the minimum number such that $x \notin A_s$ and for all $j < i$, we have $x > r_{j,s}$. If $\neg\Phi_{e,s}^{B_s}(x) \downarrow$, let $x_{i,s+1} = r_{i,s+1} = -1$ and skip step (c).

(c) Now assume $\Phi_{e,s}^{B_s}(x) \downarrow$. Let $x_{i,s+1} = x$ and $r_{i,s+1} = \max\{x_{i,s+1}, u_{e,s}^{B_s}(x_{i,s+1})\}$. If $\phi_{e,s}^{B_s}(x_{i,s+1}) = 1$, let $A_{s+1} = A_s$, $B_{s+1} = B_s$; otherwise, let $A_{s+1} = A_s \cup \{x_{i,s+1}\}$, $B_{s+1} = B_s$. End step s; i.e., we do not work for larger i's.

If there is some i that survives at step (c), then A_{s+1} and B_{s+1} have been defined. Otherwise, let $A_{s+1} = A_s$ and $B_{s+1} = B_s$. Note that for some i's, $x_{i,s+1}$ and $r_{i,s+1}$ may be undefined. We let $x_{i,s+1} = r_{i,s+1} = -1$ for all such i's.

Finally, let $A = \bigcup_s A_s$, $B = \bigcup_s B_s$.

Now we prove the correctness of the construction.

(1) Each R_i is injured a finite number of times.

Proof: Requirement R_i is injured at step s if some $x \leq r_{i,s}$ is added. This only happens when some $j < i$ survives at step (c). Define a number $g_{i,s} = \sum_{j<i, x_{j,s} \neq -1} 2^{-j-1}$. Each time R_i is injured, we have $g_{i,s+1} - g_{i,s} \geq 2^{-i}$. This is because the smallest $j < i$ such that $x_{j,s} \neq x_{j,s+1}$ must have $x_{j,s} = -1$ and $x_{j,s+1} \neq -1$, and then $g_{i,s+1} - g_{i,s} \geq 2^{-j-1} - \sum_{j<k<i} 2^{-k-1} = 2^{-i}$. Clearly, for all s, $0 \leq g_{i,s} \leq 1 - 2^{-i}$. So R_i can be injured at most $2^i - 1$ times.

(1′) For each i, there exists s_0 such that for all $s \geq s_0$, for all $j < i$, j does not survive at step (c).

Proof: Similar to (1).

(2) For all i, $\lim_s r_{i,s}$ and $\lim_s x_{i,s}$ exist and are finite. Also, R_i is satisfied.

Proof: Assume $i = 2e$ (the case $i = 2e + 1$ is similar). Choose s_0 so that R_i is not injured at step s for all $s \geq s_0$. If for some $s \geq s_0$, $x_{i,s} \neq -1$, then for all $t \geq s$, we have $x_{i,t} = x_{i,s}$, $r_{i,t} = r_{i,s}$, $A_t(x_{i,t}) = A_s(x_{i,s})$, $\Phi_{e,t}^{B_t}(x_{i,t}) \downarrow$, and $\phi_{e,t}^{B_t}(x_{i,t}) \neq A_t(x_{i,t})$. So in this case R_i is satisfied.

Now assume for all $s \geq s_0$, we have $x_{i,s} = -1$. We choose s_0 large enough such that it also satisfies the condition in (1′). At step $s \geq s_0$, $\min\{x : x \notin A_s, \forall j < i(x > r_{j,s})\}$ is fixed. Call this number y. That $x_{i,s} = -1$ for all $s \geq s_0$ means that $\neg\Phi_{e,s}^{B_s}(y) \downarrow$ for all $s \geq s_0$. So $\Phi_e^B(y) \uparrow$, and R_i is also satisfied.

(3) The sets A and B are two incomparable r.e. sets.

Proof: They are r.e. because the construction is r.e. They are incomparable because all requirements are satisfied. $\square$

Suppose in some problem, we would like to construct A such that some requirement $\phi_e^A \neq C$ is satisfied. The natural idea is to define the restriction function so that a witness x (that is, $\phi_e^A(x) \neq C(x)$) is always a witness unless the requirement is injured. This is called preserving disagreement. However, sometimes we need also preserve agreement, i.e., preserve x such that $\phi_e^A(x) = C(x)$.

Theorem 4.4 (Sacks [Sac63c]). *For every nonrecursive r.e. set C there exists a simple set A such that $C \not\leq_T A$.*

Proof. We know C is infinite because it is nonrecursive. Let $\{C_s\}_{s\in\mathbb{N}}$ be a recursive enumeration of C, i.e., $|C_s| = s$, $C_s \subseteq C_{s+1}$, $C = \bigcup_s C_s$, and there is a Turing machine that on input C_s outputs C_{s+1}. Recall the notations W_e and $W_{e,s}$ in the proof of Theorem 4.2.

The requirements we need to satisfy are $R_e : \phi_e^A \neq C$ and $P_e : W_e$ infinite $\Rightarrow W_e \cap A \neq \emptyset$.

We start with $A_0 = \emptyset$.

Step s: Compute $l_{e,s} = \max\{x : \forall y < x(\Phi_{e,s}^{A_s}(y) \downarrow \text{ and } \phi_{e,s}^{A_s}(y) = C_s(y))\}$ and $r_{e,s} = \max\{u_{e,s}^{A_s}(x) : x \leq l_{e,s}\}$. (Note that $l_{e,s}$ and $r_{e,s}$ are computable even if $l_{e,s} = \infty$. This is because the possible number of s-step traces of a Turing machine is finite.)

We choose the minimum $i \leq s$ such that $W_{i,s} \cap A_s = \emptyset$ and there exists x such that $x \in W_{i,s}$, $x \geq 2i$, and for all $j \leq i$, we have $x > r_{j,s}$. If there exist such i and x, we let $A_{s+1} = A_s \cup \{x\}$. Otherwise, we let $A_{s+1} = A_s$.

Finally, let $A = \bigcup_s A_s$.

We prove the correctness of the construction.

(1) Each R_e is injured a finite number of times.

Proof: R_e is injured at step s only when some $x \le r_{e,s}$ is added, which occurs only when some W_i ($i < e$) is chosen. Each W_i is chosen at most once, so R_e is injured at most e times.

(2) For all e, $\phi_e^A \neq C$. So R_e is satisfied.

Proof: Assume that $\phi_e^A = C$ for some e. This means that $\lim_s l_{e,s} = \infty$. Choose s_0 so that R_e is not injured at step s for all $s \ge s_0$.

We prove that for $s \ge s_0$, $l_{e,s}$ is non-decreasing. Actually, the only reason for $l_{e,s+1} < l_{e,s}$ is that $C_{s+1} - C_s$ contains an element $x < l_{e,s}$. However, this disagreement will be preserved forever, so $l_{e,t} \le x$ for all $t \ge s$. This contradicts $\lim_s l_{e,s} = \infty$.

Now we build a Turing machine that decides C. On input x, the Turing machine finds the minimum s such that $l_{e,s} > x$ and output $\phi_{e,s}^{A_s}(x)$. Because $l_{e,s}$ never decreases, for all $t \ge s$, we have $\phi_{e,s}^{A_s}(x) = \phi_{e,t}^{A_t}(x) = C_t(x)$. For t large enough, $C_t(x) = C(x)$. So $\phi_{e,s}^{A_s}(x) = C(x)$. Therefore the Turing machine decides C. This contradicts the assumption that C is nonrecursive.

(3) For all e, $\lim_s r_{e,s}$ exists and is finite.

Proof: By (2), choose the minimum x such that $\phi_e^A(x) \neq C(x)$. Choose s_0 such that for all $s \ge s_0$,

(a) R_e is not injured at step s;
(b) $\phi_e^A(y) = \phi_{e,s}^{A_s}(y)$ for $y \le x$;
(c) $C_s(y) = C(y)$ for $y \le x$.

If for some $t \ge s_0$ we have $\Phi_{e,t}^{A_t}(x)\downarrow$, then $\phi_{e,t}^{A_t}(x) = 1 - C(x)$ and $\lim_s r_{e,s} = r_{e,t}$. Otherwise, $\Phi_e^A(x)\uparrow$ and $\lim_s r_{e,s} = r_{e,s_0}$.

(4) For all e, P_e is satisfied.

Proof: By (3), define $r_e = \lim_s r_{e,s}$. If W_e contains an element x such that $x \ge 2e$ and $x > \max\{r_i : i \le e\}$, then W_e is chosen at some step, and thus $W_e \cap A \neq \emptyset$. In particular, if W_e is infinite, then $W_e \cap A \neq \emptyset$.

(5) The set A is a simple set such that $C \not\le_T A$.

Proof: The set A is r.e. because the construction is r.e. The set A is coinfinite because for each W_e we add at most one element larger than or equal to $2e$, and no elements smaller than $2e$. The set A is simple because all requirements P_e are satisfied. We have $C \not\le_T A$ because all requirements R_e are satisfied. $\square$

Remark 4.5. Actually, the set A constructed in the proof is low. Construct a Turing machine M_g that on input e, outputs a natural number $g(e)$ such that $M_{g(e)}$ is a Turing machine with oracle A that on input x, outputs $C(0)$ if $x = 0$ and $\Phi_e^A(e)\downarrow$, and loops forever otherwise.

We have $e \in A' \Leftrightarrow \Phi_e^A(e)\downarrow \Leftrightarrow (\Phi_{g(e)}^A(0)\downarrow \wedge \phi_{g(e)}^A(0) = C(0)) \Leftrightarrow \lim_s l_{g(e),s} > 0$. The last condition can be decided with an oracle $0'$. So $A' \le_T 0'$.

We extend the method of proof of Theorem 4.4 to prove the following theorem.

Theorem 4.6 (Sacks splitting theorem, Sacks [Sac63b])**.** *Let B and C be r.e. sets such that C is nonrecursive. Then there exist two low r.e. sets A_0, A_1 such that (A_0, A_1) is a partition of B and $C \not\le_T A_z$ for $z = 0, 1$.*

Proof. We know C is infinite because it is nonrecursive. If B is finite, then we can let $A_0 = \emptyset$ and $A_1 = B$. So we can assume that B is infinite. Let $\{B_s\}_{s\in\mathbb{N}}$ and $\{C_s\}_{s\in\mathbb{N}}$ be recursive enumerations of B and C, respectively.

The requirements we would like to satisfy are

R_{2e+z} : $\phi_e^{A_z} \neq C$ for $z = 0, 1$;
P_s : if $B_{s+1} - B_s = \{x\}$, then $x \in A_z$ for exactly one z.

Lowness does not need extra requirements because it can be proved using exactly the same method as Remark 4.5.

Start with $A_{z,0} = \emptyset$ for $z = 0, 1$.

Step s: Compute $l_{2e+z,s} = \max\{x : \forall y < x(\Phi_{e,s}^{A_{z,s}}(y)\downarrow \text{ and } \phi_{e,s}^{A_{z,s}} = C_s(y))\}$ and $r_{2e+z,s} = \max\{u_{e,s}^{A_{z,s}}(x) : x \leq l_{2e+z,s}\}$.

Let $B_{s+1} - B_s = \{x\}$. Choose the smallest $2e + z$ such that $x \leq r_{2e+z,s}$. Let $A_{z,s+1} = A_{z,s}$ and $A_{1-z,s+1} = A_{1-z,s} \cup \{x\}$.

Finally, let $A_z = \bigcup_s A_{z,s}$ for $z = 0, 1$.

We prove the correctness of the construction. We prove by induction on $2e + z$ that

(a) R_{2e+z} is injured a finite number of times;
(b) $\phi_e^{A_z} \neq C$;
(c) $\lim_s r_{2e+z,s}$ exists and is finite.

By the induction hypothesis, for $2e' + z' < 2e + z$, the three properties above are satisfied. Choose s_0 such that for all $s \geq s_0$, we have

(A) for all $2e' + z' < 2e + z$, $r_{2e'+z',s} = \lim_s r_{2e'+z',s}$;
(B) if $B_{s+1} - B_s = \{x\}$, then $x > \max\{r_{2e'+z',s} : 2e' + z' < 2e + z\}$.

Property (A) ensures that there exists s_0 such that (B) holds. Property (B) ensures that $2e' + z' < 2e + z$ is not chosen at step s for $s \geq s_0$. Then at each step $s \geq s_0$, if $2e + z$ is chosen, then R_{2e+z} is not injured because $A_{z,s+1} = A_{z,s}$; if $2e' + z' > 2e + z$ is chosen, then R_{2e+z} is not injured because $x > r_{2e+z,s}$ (where $B_{s+1} - B_s = \{x\}$). So R_{2e+z} is never injured at step s for $s \geq s_0$. This proves (a).

The proof for (b) and (c) assuming (a) is exactly the same as part (2) and (3) of proof of Theorem 4.4.

By above discussion, all requirements R_{2e+z} are satisfied. Requirement P_s is satisfied at step $s + 1$. So all requirements are satisfied. The A_z's are r.e. because the construction is r.e. The A_z's are low by the same method as Remark 4.5. $\square$

Let us discuss some applications of Theorem 4.6. We need the following lemma.

Lemma 4.7. *Let B, A_0, A_1 be r.e. sets such that (A_0, A_1) is a partition of B. Then $B \equiv_T A_0 \oplus A_1$.*

Proof. Clearly $B \leq_T A_0 \oplus A_1$. To prove that $A_0 \oplus A_1 \leq_T B$, we construct a Turing machine with oracle B that decides A_0 and A_1. On input x, we first ask oracle if $x \in B$. If not, then $x \notin A_z$ for $z = 0, 1$. Otherwise, we can recursively enumerate A_0 and A_1 at the same time until one of them outputs x. $\square$

Corollary 4.8. *Let B be an r.e. set. Then there exists two incomparable low r.e. sets A_0, A_1 such that (A_0, A_1) is a partition of B.*

Proof. In Theorem 4.6, take $C = B$. We only need to prove incomparability. Without loss of generality, assume $A_0 \leq_T A_1$. Then $C = B \leq_T A_0 \oplus A_1 \leq_T A_1$, which is a contradiction. $\square$

Corollary 4.9. *Low r.e. degrees with finite joins generate all r.e. degrees.*

Proof. Direct from Theorem 4.6 and Lemma 4.7. $\square$

Corollary 4.10. *No r.e. degree is minimal.*

Proof. Direct from Corollary 4.8 and Lemma 4.7. □

References

[Fri57] Richard M. Friedberg. Two recursively enumerable sets of incomparable degrees of unsolvability (solution of Post's problem, 1944). *Proceedings of the National Academy of Sciences*, 43(2):236–238, 1957.

[KP54] Stephen C. Kleene and Emil L. Post. The upper semi-lattice of degrees of recursive unsolvability. *Annals of mathematics*, pages 379–407, 1954.

[Mos38] Andrzej Mostowski. Über gewisse universelle Relationen. *Annales de la Société Polonaise de Mathématique*, 17:117–118, 1938.

[Muc56] Albert A. Muchnik. On the unsolvability of the problem of reducibility in the theory of algorithms. In *Dokl. Akad. Nauk SSSR*, volume 108, pages 194–197, 1956.

[Odi92] Piergiorgio Odifreddi. *Classical recursion theory: The theory of functions and sets of natural numbers*, volume 125 of *Studies in logic and the foundations of mathematics*. Elsevier, 1992.

[Pos44] Emil L. Post. Recursively enumerable sets of positive integers and their decision problems. *Bull. Amer. Math. Soc.*, 50(5):284–316, 1944.

[Sac63a] Gerald E. Sacks. *Degrees of unsolvability*. Number 55. Princeton University Press, 1963.

[Sac63b] Gerald E. Sacks. On the degrees less than $0'$. *Annals of Mathematics*, pages 211–231, 1963.

[Sac63c] Gerald E. Sacks. Recursive enumerability and the jump operator. *Transactions of the American Mathematical Society*, 108(2):223–239, 1963.

[Sho13] Richard A. Shore. Lecture notes on Turing degrees. *Computational Prospects of Infinity II: AII Graduate Summer School*, 2013.

[Soa76] Robert I. Soare. The infinite injury priority method. *The Journal of Symbolic Logic*, 41(02):513–530, 1976.

[Soa16] Robert I. Soare. *Turing Computability: Theory and Applications*. Springer, 2016.

[Spe56] Clifford Spector. On degrees of recursive unsolvability. *Annals of Mathematics*, pages 581–592, 1956.