Σ_2 INDUCTION AND INFINITE INJURY PRIORITY ARGUMENTS, PART III: PROMPT SETS, MINIMAL PAIRS AND SHOENFIELD'S CONJECTURE

C. T. CHONG, LEI QIAN, THEODORE A. SLAMAN, AND YUE YANG

1. Introduction and Preliminaries

Priority constructions are the trademark of theorems on the recursively enumerable Turing degrees. By their combinatorial patterns, they are naturally identified as finite injury, infinite injury, and so forth. Following Chong and Yang [4], [3], we analyze the complexity of infinite injury arguments and pinpoint exactly the position of their degree-theoretic applications within the hierarchy of fragments of Peano arithmetic (cf. Chong and Yang [5] for a discussion of the issues and motivation behind such studies).

Finite injury priority constructions fall essentially into two types: the Friedberg-Mučnik type and the Sacks splitting type. For the former, Chong and Mourad [2] show that even though such constructions cannot be carried out without Σ_1 induction, Σ_1 bounding is a sufficiently strong theory to establish the existence of a pair of incomparable recursively enumerable degrees. For the latter, the results of Mytilinaios [12] and Mourad [11] together imply that the Sacks splitting theorem is equivalent to Σ_1 induction over the base theory of Σ_1 bounding.

Infinite injury constructions, by contrast, are more varied and harder to categorize. Results to-date show that the existence of a high recursively enumerable degree is equivalent to Σ_2 induction over the base theory of Σ_2 bounding [4], and that the Density Theorem is provable under Σ_2 bounding [7] (note that the density theorem fails in all models of Σ_1 bounding in which Σ_1 induction fails, by a result of Mourad [11]). Our intuition suggests that certain Σ_2 properties are necessary for infinite injury arguments to carry through (although, again, there are special models satisfying Σ_1

1991 *Mathematics Subject Classification.* 03D20, 03F30, 03H15.

This joint work was done when Slaman visited National University of Singapore as a visiting professor in May, 1997. Slaman was partially supported by National Science Foundation Grant DMS-9500878.

induction in which virtually every construction in classical recursion theory works). Here we investigate a third type of infinite injury construction, exemplified in the proof that there is a minimal pair of recursively enumerable degrees. Two recursively enumerable degrees are said to form a minimal pair if the only recursively enumerable degree recursive in both of them is the recursive degree. Historically the existence of minimal pairs was proved independently by Yates [16] and Lachlan [10]. The Yates and Lachlan theorem gave a negative solution to Shoenfield's conjecture, that every monomorphism from a finite upper-semi-lattice into the recursively enumerable degrees can always be lifted to embed any larger finite upper-semi-lattice. From the methodological point of view, the construction of a minimal pair of recursively enumerable degrees incorporates a number of features not present in either the construction of an incomplete high recursively enumerable degree or the construction used to prove the Density Theorem. As we shall see, Σ_2 induction is sufficient to establish the existence of a minimal pair, and these two are equivalent over the base theory of Σ_2 bounding.

This paper is organized as follows. After the preliminaries, we investigate in Section 2 the subject of dominating functions in models of Σ_2 bounding. We show that there is a family of total recursive functions indexed by a proper Π_2 cut such that no total recursive function eventually dominates every function in the family. This result is optimal in the sense that there is no bounded Π_2 family of total recursive functions such that any total recursive function is eventually dominated by one in the family. Apart from the intrinsic interest provided by such combinatorial properties, the method used in the proof is later adapted to show that no minimal pairs exist in any model of Σ_2 bounding without Σ_2 induction. In Section 3 we show that minimal pairs exist in every model of Σ_2 induction. In Section 4 we show that there is no minimal pair in any model of Σ_2 bounding in which Σ_2 induction fails. In the final section, we return to the problem of Shoenfield's Conjecture, and show that even with the failure of the minimal pair theorem, the conjecture is still refuted within the theory of Σ_2 bounding. We provide two examples, one involves only meet operator, the other only join operator. We end by posing a number of open problems.

Let us briefly recall the basic definitions and results. More details can be found for example in Mytilinaios [12] or in Chong and Yang [4]. Let P^- be the Peano axioms minus the induction scheme, and let $I\Sigma_n$ and $B\Sigma_n$ denote respectively the induction and the collection schemes for Σ_n formulas. We work on models satisfying $P^- + I\Sigma_0$. By a result of Paris and Kirby [9], for any $n \geq 1$,

$$I\Sigma_n \Rightarrow B\Sigma_n \Rightarrow I\Sigma_{n-1},$$

but not conversely.

We say that a set K is **$\mathcal{M}$-finite** if it has a code in $\mathcal{M}$. A set F is $\mathcal{M}$-finite if and only if there is a one-to-one Σ_0 function from a number a in $\mathcal{M}$ onto F. $I \subset \mathcal{M}$ is said to be a **cut** in $\mathcal{M}$, if I is nonempty, closed downward and closed under the successor function. For our purpose, a cut is always a proper subset of the model under consideration. We say that a model $\mathcal{M}$ is a $B\Sigma_n$ **model** if $\mathcal{M}$ is a model of $P^- + B\Sigma_n$ but not $I\Sigma_n$. In any $B\Sigma_n$ model $\mathcal{M}$, there is a Σ_n cut I and a Σ_n map $f\colon I \to \mathcal{M}$ whose range is unbounded in $\mathcal{M}$. We denote by $<a$ the set $\{x \in \mathcal{M}\colon x < a\}$. A subset A of $\mathcal{M}$ is **bounded** in $\mathcal{M}$ if there is an a in $\mathcal{M}$ such that $A \subseteq <a$. $A \subseteq \mathcal{M}$ is **regular** if for every a in $\mathcal{M}$, $A \cap <a$ is $\mathcal{M}$-finite.

Lemma 1 (H. Friedman)**.** *Suppose that $\mathcal{M}$ is a model of $P^- + I\Sigma_n$ ($n \geq 1$),*

1. *If A is Σ_n in $\mathcal{M}$, then A is regular.*
2. *If f is a partial Σ_n function whose domain is bounded, then the range of f is also bounded.*

As usual, a **Turing functional** is an recursively enumerable set Φ of consistent quadruples, $\langle x, y, P, N\rangle$, where P and N are disjoint $\mathcal{M}$-finite sets and x and y are numbers. We say that $\Phi^A(x) = y$ if there are $\mathcal{M}$-finite sets, P included in A, and N disjoint from A, such that $\langle x, y, P, N\rangle \in \Phi$. We say that B is **(weakly) recursive in** A if for some Turing functional Φ, $\Phi^A = B$. B is **strongly recursive in** A if both

$$\{P\colon P \text{ is } \mathcal{M}\text{-finite and } P \subset B\}$$

and

$$\{N\colon N \text{ is } \mathcal{M}\text{-finite and } N \cap B = \emptyset\}$$

are weakly recursive in A. Groszek and Slaman [8] showed that 'strongly recursive in' is a transitive relation on sets, while weak reducibility is not transitive in general. However, in any model of $P^- + B\Sigma_2$, which is the main object of study in this paper, weak reducibility coincides with strong reducibility for recursively enumerable sets, although not necessarily for Σ_2 sets.

2. Dominating Functions

In this section, we study the problem of dominating functions in $B\Sigma_2$ models. We will consider a bounded family of recursive functions and study the question of domination: Are these functions dominated by a single recursive function? Is every recursive function dominated by one of them? Although these questions are not directly related to infinite injury priority arguments, they provide insights to the intrinsic properties of $B\Sigma_2$ models, and the techniques used in the proofs are applicable to those presented in Section 4.

Let J be any Π_2 cut in $\mathcal{M}$, and suppose that J is defined by

$$j \in J \Leftrightarrow \forall u \exists v \varphi(j, u, v),$$

where $\varphi(j, u, v)$ is a Δ_0 formula. Let a be an upper bound of J in $\mathcal{M}$. We consider a family of partial recursive functions $\{h_i \colon i \leq a\}$ defined uniformly by:

$$h_i(u) = \begin{cases} \text{the least } v \text{ such that } \varphi(i, u, v), & \text{if such } v \text{ exists;} \\ \\ \text{undefined,} & \text{otherwise.} \end{cases}$$

We can make h_i nondecreasing with respect to i and u. That is, if $i' < i$ and $h_i(u)$ is defined, then $h_{i'}(u)$ is defined and $h_{i'}(u) \leq h_i(u)$, and for all i if $u' < u$ and $h_i(u)$ is defined, then $h_i(u')$ is defined and $h_i(u') < h_i(u)$. In fact, just change the above clause to "the least v such that $(\forall i' \leq i)(\forall u' \leq u)(\exists v' < v)\varphi(i', u', v')$". Notice that if $j \in J$ then h_j is total on $\mathcal{M}$. Moreover for points $i \in a - J$, we note that if $f_i(u)$ is undefined then for any $u' > u$ $f_i(u')$ is undefined. This bounded family of uniformly recursive functions offers us some features which do not exist in model of full PA. The following is one example whose proof uses an idea we will return to in the sequel.

Theorem 1. *For any total recursive function g, there is a j in J such that g does not eventually dominate h_j.*

Proof. We prove by contradiction. Suppose that the statement is false. Then there is a recursive function g that eventually dominates all h_j for j in J. Therefore

$$(\forall i \leq a)(\exists n)(\forall t)[h_i(n){\uparrow}\ [t] \vee (h_i(n){\downarrow}\ [t] \wedge (\forall m > n) h_i(m) < g(m))].$$

The first disjunct refers to the i's not in J, while the second refers to those in J. By $B\Sigma_2$, there is an n_0 which bounds all such n. Thus $i \in J$ if and only if $h_i(n_0){\downarrow}$, which implies that J is Σ_1, a contradiction. □

A natural question to ask next is whether something stronger holds, i.e. whether one can have a family of functions such that every total recursive function is eventually dominated by one in the family. The answer is no by the following slightly more general result.

Let $\{f_m \colon m \leq a\}$ be a uniform family of partial recursive functions. Let J be the set

$$J = \{j \leq a \colon f_j \text{ is total}\}$$

which is a Π_2 subset of $\leq a$. Without of loss of generality, we may assume that for any $m \leq a$, the domain of f_m is downward closed. We can also assume that at any stage s there exists at most one pair of numbers m and x such that $m \leq a$ and $f_m(x)$ is defined at stage s.

Theorem 2. *If J is not empty, then there is a total recursive function g which is not eventually dominated by any f_j for $j \in J$.*

Proof. We build a family of partial recursive functions $\{g_n \colon n \leq a\}$ such that at least one of them is total and is not eventually dominated by any of f_j where j is in J. For the rest of the proof, the letters m and n are used for numbers less than or equal to a, and m refers to the function f and n to g.

We need to satisfy the following requirements.

$$R_{\langle m,n,k\rangle} : g_n \text{ is larger than } f_m \text{ at } k \text{ different points,}$$

(provided f_m is total). The strategy to satisfy a single requirement $R_{\langle m,n,k\rangle}$ is as follows. Suppose that for all $l < k$, $R_{\langle m,n,l\rangle}$ is satisfied. Pick a new number x. Stop defining g_n at x until $f_m(x)$ is defined. We call this action as "g_n holds x on f_m for k" or "g_n is assigned on f_m for k". When $f_m(x)$ is defined, we define $g_n(x) = f_m(x) + 1$. Thus the requirement $R_{\langle m,n,k\rangle}$ is satisfied forever. We call it as "g_n releases f_m for k". (g_n can release f_m due to other reasons, when more than one requirements interacts.) In the case when $f_m(x)$ is undefined, g_n may hold x on f_m forever. Consequently, g_n becomes partial.

To motivate the proof, we may view the functions $\{f_m \colon m \leq a\}$ as $(a + 1)$-many columns. At any stage, at any column, there is a number x, which is being held by a unique g_n for some k. Thus we always have a one-to-one correspondence between g_n and f_m. When g_n releases f_m, we arrange some other g to hold f_m according to a given priority list. The main concern is whether a given requirement loses its chance forever because of other higher priority requirements.

Fix a priority list:

$$R_0 < R_1 < \cdots < R_e < \ldots$$

where each index e is viewed as a triple $\langle m, n, k\rangle$.

To simplify matters, we adopt the following conventions. First, at any stage s, if g_n holds x , then for any number $y < x$ not mentioned by the construction $g_n(y)$ will be defined trivially, say equals y. Also we assume that the witness x for $R_{\langle m,n,k\rangle}$ is automatically chosen as the least number at which both g_n and f_m are undefined.

Construction:

Stage 0: Assign g_n to f_n for 0.

Stage s: If there is a triple $\langle m_0, n_0, k_0\rangle$ such that g_{n_0} held an x on f_{m_0} for k_0 at stage $s-1$ and $f_{m_0}(x)$ is defined at stage s. Then define

$$g_{n_0}(x) = f_{m_0}(x) + 1$$

and release f_{m_0}. Cancel $\langle m_0, n_0, k_0\rangle$ from the priority list. Go to switch operation. If no such triple $\langle m_0, n_0, k_0\rangle$ exists, then go to stage $s+1$.

Switch Operation: Given m_0 and n_0 as above.

Consider those n and k so that

- $\langle m_0, n, k\rangle$ is not yet satisfied.
- If g_n is assigned to some f_{m^*} for k^*, then $\langle m_0, n, k\rangle$ has higher priority than $\langle m^*, n, k^*\rangle$.

Choose n_1 and k_1 with these properties so as to maximize the priority of $\langle m_0, n_1, k_1\rangle$. Let f_{m_1} be the function that is assigned to g_{n_1}, if $n_0 \neq n_1$.

Assign g_{n_1} to f_{m_0} for k_1. If $n_0 \neq n_1$, then assign g_{n_0} to f_{m_1} for the least k such that $\langle m_1, n_0, k\rangle$ is not satisfied.

Note, if n_0 is equal to n_1 then g_{n_0} is assigned to f_{m_0} for the next value of k. Otherwise, a new g is chosen for f_{m_0} so as to maximize the priority of the next requirement for f_{m_0} to be attempted.

End of Construction

We now verify that the construction works.

Claim 1. *There is an $n \leq a$ such that g_n is total.*

Proof of Claim 1. We prove by contradiction. Suppose for the contrary that for all $n < a$ g_n is not total. By $I\Sigma_1$ there is a least point x at which g_n is not defined. Thus we have

$$(\forall n \leq a)(\exists x)[x \text{ is the least point at which } g_n \text{ is not defined}].$$

Notice that saying "x is the least point at which g_n is not defined" is a Σ_2 formula. By $B\Sigma_2$, there is a uniform upper bound of x for all $n \leq a$. Call it b. Since J is not empty, there is a total function f_j. At the stage $f_j(b)$ is defined, no g_n can hold f_j below b, in other words, there is a function g_n which is defined up to b. A contradiction.

Claim 2. *Let g_n be a total function (existence shown in Claim 1). Then g_n is not eventually dominated by f_j for any j in J.*

Proof of Claim 2. Let j be given and argue that for all k in $\mathcal{M}$, the requirement $R_{\langle j,n,k\rangle}$ is satisfied.

Suppose that there is a k such that the requirement $R_{\langle j,n,k\rangle}$ is not satisfied. By $I\Sigma_1$, we can pick the least such one. For simplicity let us also use k to denote it.

We say that a requirement $R_{\langle j,n,k\rangle}$ **acts during stage** s if either the construction assigns f_j to g_n for k during stage s (and it was not so assigned during stage $s-1$) or $R_{\langle j,n,k\rangle}$ is satisfied during stage s.

By $I\Sigma_1$ there is a stage s_0 such that for any requirement R_d such that $d < \langle j,n,k\rangle$, R_d does not act after stage s_0. See Mytilinaios [12] for a discussion of finite injury arguments within $I\Sigma_1$.

Suppose that $R_{\langle j,n,k\rangle}$ is not satisfied at stage s_0. Then since g_n is holding some f_i for some l at stage s_0, g_n must hold an f_i for an l at s_0 with $\langle j,n,k\rangle < \langle i,n,l\rangle$, otherwise g_n would not be total. Next since f_j is total, f_j will be released at some stage $t > s_0$. At t, g_n will be switched on f_j for k, to maximize the priority of the next requirement considered for f_j. By the totality of f_j again, $R_{\langle j,n,k\rangle}$ will be satisfied. That establishes Claim 2 and the Theorem. □

Note that $B\Sigma_2$ is necessary for the results above. Without $B\Sigma_2$, the small family of dominating functions might exist. To be more precise, we look at a particular $I\Sigma_1$ model $\mathcal{M}$ not satisfying $B\Sigma_2$, in which there is a $\Delta_2(\mathcal{M})$ function p mapping a Σ_2 cut I one-one onto $\mathcal{M}$ (in fact, the cut I is ω). The model was first constructed by Groszek and Slaman in [8].

Lemma 2. *Let $\mathcal{M}$ be the model of $I\Sigma_1$ above. Let I be a Σ_1 cut and a an upper bound of I. Then there exists a family of partial recursive functions $\{h_n : n \in I\}$ such that any recursive function f is eventually dominated by some h_n.*

Proof. Fix a recursive approximation $p(n,s)$ of $p(n)$. Define uniformly a family of recursive functions $\{g_n : n \leq a\}$ by

$$g_n(s) = \begin{cases} f_{p(n,s)}(s) + 1, & \text{if } f_{p(n,s)}(s) \text{ is defined;} \\ \\ \text{undefined,} & \text{otherwise.} \end{cases}$$

Where $\{f_e : e \in \mathcal{M}\}$ is a fixed list of all partial recursive functions in $\mathcal{M}$. Define

$$h_n(s) = g_n(\mu t \geq s(g_n(t) \text{ is defined.})).$$

Consider an arbitrary $e = p(n)$ in $\mathcal{M}$, let s_0 be the stage such that for all $s > s_0$, $p(n,s) = e$. If f_e is total, then for all $s > s_0$,

$$h_n(s) = g_n(s) = f_e(s) + 1$$

and h_n is total. □

3. $I\Sigma_2$ and Minimal Pairs

In this section, we show that the usual tree construction of minimal pairs can be carried out in any model of $I\Sigma_2$. Since the proof is standard, we only present the skeleton. The key point is to verify that Σ_2 induction is sufficient to prove the existence of the true path.

Theorem 3. *Let $\mathcal{M}$ be a model of $P^- + I\Sigma_2$. Then there exist recursively enumerable sets A and B such that if $C \leq_T A$ and B then C is recursive.*

We construct recursively enumerable sets A and B to satisfy the following requirements for all e in $\mathcal{M}$:

$$P_e^A : A \neq \Phi_e.$$

$$P_e^B : B \neq \Phi_e.$$

$$N_e : \text{If } \Psi_e(A) = \Psi_e(B) = h \text{ total, then } h \text{ is recursive.}$$

The strategy to satisfy P_e^A is to pick an x, wait for $\Phi_e(x) = 0$ and then put x into A. The strategy to satisfy P_e^B is symmetric. The strategy to satisfy N_e is to guarantee that once the length of agreement

$$l(e, s) = \max\{x : (\forall y < x)(\Psi_e(A; y){\downarrow} = \Psi_e(B; y){\downarrow}\ [s])\}$$

reaches a new value then we only allow elements to enter either A or B but not both.

We now proceed to the tree construction. The priority tree T is the full binary tree. Fix a node α on T. If $|\alpha| = 3e$, then $\alpha^\wedge \langle 0\rangle$ corresponds to the Π_2 outcome of N_e, which says that the length of agreement is infinite; $\alpha^\wedge \langle 1\rangle$ corresponds to the Σ_2 outcome of N_e, which says that the length of agreement is finite. If $|\alpha| = 3e + 1$, then $\alpha^\wedge \langle 0\rangle$ is corresponding to the Π_1 outcome of P_e^A, which says that we wait forever for $\Phi_e(x) = 0$; $\alpha^\wedge \langle 1\rangle$ is corresponding to the Σ_1 outcome, which says that we have seen the computation $\Phi_e(x) = 0$ and successfully put x into A. If $|\alpha| = 3e + 2$, then do the same for P_e^B. We assume that 0 is to the left of 1 on tree T.

At stage s, we define a string δ_s of length $\leq s$ by induction. δ_s is called the string **visited at stage** s. Define $\delta_s(0)$ to be the root of the tree T. Suppose $\alpha \subset \delta_s$ and $|\alpha| < s$. For $|\alpha| = 3e + 1$, if there is an $x \in \mathcal{M}^{[e]}$ $\Phi_e(x) = 0[s]$ and $x \in A$, then $\alpha^\wedge \langle 1\rangle \subseteq \delta_s$. Otherwise $\alpha^\wedge \langle 0\rangle \subseteq \delta_s$. Similar definition applies to $|\alpha| = 3e + 2$. For $|\alpha| = 3e$, if $l(e, s) > l(e, t)$ for every $t < s$ such that $\alpha \subseteq \delta_s$, then $\alpha^\wedge \langle 0\rangle \subseteq \delta_s$. Otherwise, let $\alpha^\wedge \langle 1\rangle \subseteq \delta_s$.

Construction:

At stage s, find the $\subseteq$-least $\alpha \subseteq \delta_s$ such that $|\alpha| = 3e + k$, $(k \in \{1, 2\})$ and $\delta_s(|\alpha|) = 0$ for which there exists an x in $\mathcal{M}^{[e]}$ such that $\Phi_e(x) = 0[s]$ and x is bigger than the restraint

$$r(\alpha, s) = \max\{t \colon t < s \text{ and } \delta_t \text{ is to the left or a substring of } \alpha\}.$$

Put the least such x, if any, into A if $k = 1$, or B if $k = 2$. Otherwise, do nothing.
End of Construction

We now verify that the construction works. Let the **true path** Λ be the leftmost path which is visited unboundedly often. First we show that Λ exists. This is the place where we make crucial use of $I\Sigma_2$.

Lemma 3. *For any $e \in \mathcal{M}$, there is a unique α on T of length e such that*
1. *for any s there is a $t > s$ such that $\alpha \subset \delta_t$.*
2. *there is a stage t_0 such that for any β to the left of α and for any $t > t_0$ $\beta \not\subset \delta_t$.*

Proof. Fix e in $\mathcal{M}$, and consider the set of strings

$$\{\sigma \colon |\sigma| \le e \wedge (\forall s)(\exists t > s)(\sigma \subseteq \delta_t)\},$$

which is a nonempty Π_2 bounded set. By $I\Sigma_2$ there exists a leftmost element. Call it α. By definition, (1) is satisfied. To show (2), consider the set

$$X = \{\beta \colon |\beta| \le e \wedge \beta <_L \alpha\}$$

which is $\mathcal{M}$-finite. By the definition of α, we have

$$\forall \beta \in X \exists s \forall t > s \beta \not\subseteq \delta_t.$$

By $B\Sigma_2$ there is a stage t_0 such that for all $t > t_0$, $\beta \not\subseteq \delta_t$. That establishes the lemma. □

Finally we argue that along the true path Λ, every requirement is satisfied.

Lemma 4. *For any e in $\mathcal{M}$, the requirements P_e^A, P_e^B and N_e are satisfied.*

Proof. Let us consider P_e^A (P_e^B is symmetric). Suppose $\alpha \subset \Lambda$ and $|\alpha| = 3e + 1$. If $\alpha^\wedge \langle 1 \rangle \subset \Lambda$ then clearly P_e^A is satisfied. So let us assume that $\alpha^\wedge \langle 0 \rangle \subset \Lambda$. In this case $\mathcal{M}^{[e]} \cap A$ is empty. If $\Phi_e = A$, then we may pick an $x \in \mathcal{M}^{[e]}$, such that $\Phi_e(x) = 0$ and x is larger than any t at which δ_t to the left of α. At any stage s such that $\Phi_e(x) = 0[s]$ and α is visited at stage s, we will put x into A. So P_e^A is satisfied.

For requirement N_e, suppose $\Psi_e(A) = \Psi_e(B) = h$ and α is the string of length $3e$ on the true path Λ. We first observe that $\alpha^\wedge \langle 0 \rangle \subset \Lambda$, because the length of agreement is unbounded. Next we check that h can be computed recursively as follows. Let s_0 be the stage after which no node β to the left of α is visited. To compute $h(p)$, just wait for a stage $s > s_0$, at which α is visited and $l(e, s) > p$. Then the typical argument as in the classical recursion theory shows that $h(p) = \Psi_e(A; p)[s]$. This establishes the lemma and the theorem. □

4. $B\Sigma_2$ and Minimal Pairs

Theorem 4. *Let $\mathcal{M}$ be a $B\Sigma_2$ model. Then there is no nontrivial recursively enumerable minimal pair in $\mathcal{M}$.*

By Ambos-Spies, Jockusch, Shore and Soare [1], the nonprompt recursively enumerable degrees are the halves of minimal pairs. So we shift our attention to prompt sets. We will show that there is no nonprompt set in $\mathcal{M}$. First let us recall the following definition.

Definition 1. *Let f be a total recursive function and W be an recursively enumerable set. We say that an recursively enumerable set A is* **f-prompt for W,** *if*

$$(\exists s)(\exists x)(x \text{ enters } W \text{ at stage } s \text{ and } A \upharpoonright x[s] \neq A \upharpoonright x[f(s)]).$$

We say that A is **f-prompt** *if A is f-prompt for all infinite recursively enumerable set W. We say that A is* **prompt**, *if there is a total recursive function f such that A is f-prompt.*

Lemma 5. *Let $\mathcal{M}$ be a $B\Sigma_2$ model. Then any recursively enumerable set A in $\mathcal{M}$ is either recursive or prompt.*

Proof. First let us fix some notations. Let I be a Σ_2 cut and $f\colon I \to \mathcal{M}$ be a Σ_2 cofinal function. Set $f'(\_,\_)$ to be a recursive approximation of f, defined on $\leq a \times \mathcal{M}$, such that for all i in I, $\lim_s f'(i, s) = f(i)$. Choose a to be an upper bound of I; and let $\{A_s\}_{s \in \mathcal{M}}$ be a fixed recursive enumeration for the recursively enumerable set A such that at any stage, at most one number enters A. We adopt the same assumption for the enumeration of recursively enumerable sets W as well. For the rest of the proof, the letters m and n will refer to numbers less than or equal to a. We build a family of a-many recursive functions $\{g_n \colon n \leq a\}$ (some of which may be partial), such that either

(a) A is recursive; or
(b) A is g_n-prompt for some $n \leq a$.

We attempt to make (b) holds for all $n \leq a$. Thus we try to satisfy for all e in $\mathcal{M}$ and $n \leq a$:

$$R_{e,n}: \ A \text{ is } g_n\text{-prompt for } W_e.$$

Strategy for a single requirement: At stage s, we say that requirement $R_{e,n}$ **requires attention** if A is not yet g_n-prompt and either

- (Condition (1)) There is no restraint on $g_n(t)$ for any t, and there is an x entering W_e at stage s; or
- (Condition (2)) There is a stage $t < s$ at which we put a restraint on $g_n(t)$ because some x entered W_e at stage t, and there is a $y < x$ which enters A at stage s.

When the requirement $R_{e,n}$ requires attention, we take the following actions.

If condition (1) holds, then we add a restraint on $g_n(s)$, i.e. keep $g_n(s)$ undefined until A changes below x, at which time $R_{e,n}$ will require attention again because condition (2) holds. We will refer this action as "g_n holds s for x and W_e".

If condition (2) holds, then we cancel the restraint on $g_n(t)$, and define $g_n(t) = s$. This action will satisfy the requirement $R_{e,n}$ forever.

In any case, if for all $t' \leq t$, $g_n(t')$ is not restrained, and $g_n(t)$ is undefined, then define $g_n(t) = s$.

Strategy for a block of requirements. The main concern for the single strategy is that $R_{e,n}$ may make g_n partial. In fact this argument will break down in models satisfying $I\Sigma_2$ because every g will be partial. The solution is to make a block of requirements hold a single function. Let us look at a block of requirements. Fix a block $B = \{e \colon b_1 \leq e < b_2\}$. We consider requirements $R_{e,n}$ for e in B. In the next few paragraphs, the letters e and d refer numbers in B.

At stage s, if there is no g_n holding a number for any x and W_e, then just proceed as in the single strategy. Suppose that there exists a t less than s and a g_n holds t for some x and W_e. Then we act depending on the following cases.

Case 1. (switch) There are numbers d, $y > x$, and m, such that $R_{d,m}$ requires attention because y enters W_d at stage s, i.e., condition (1) holds.

In this case, we cancel the restraint on $g_n(t)$, and add a restraint on $g_m(s)$. Informally, we have switched the restraint from $g_n(t)$ to $g_m(s)$. Note that m can be equal to n.

Case 2. (win) There is a $y \leq x$ entering A at stage s, such that $R_{n,e}$ requires attention because of condition (2). In this case take the same action as in the single case.

Outcomes of a block of requirements. Before we organize blocks dynamically, let us investigate the final outcomes of a block of requirements. As before, we fix a block B and letter e refers a number in B.

First notice that for any e and n, there is a stage s such that either A is g_n-prompt for W_e at stage s or for all $t > s$, A is not g_n-prompt for W_e at stage t. By $B\Sigma_2$, there is a stage s_0 after which Case 2, the win case, never happens. After stage s_0, if there exist only boundedly many stages at which the switch case happens, then there is a stage $s > s_0$ after which no more actions are taken. In this case, we say that block B has a Σ_2 **outcome.** The global effect is that for some (unique) n, $g_n(t)$ is restrained forever.

On the other hand, if there are unboundedly many switches, then we say that the block B has a Π_2 **outcome.** In this case, we argue that A is recursive:

Claim 3. *If for any stage s there is a $t > s$ such that a switch happens at t, then A is recursive.*

Proof of Claim 3. First observe that under the assumption, there are unboundedly many y's such that each y entering W_e for some e in B causes a switch. (The worry is that there may be cut-many y's which act cofinally many stages.) Otherwise, suppose that all switches are caused by numbers less than x_0. By regularity of recursively enumerable sets under $I\Sigma_1$, the recursively enumerable set $\bigcup_{e\in B} W_e$ restricted to x_0 is $\mathcal{M}$-finite. An easy application of $I\Sigma_1$ shows that the stages at which switches happen are also bounded, contradicting the assumption. Now to recursively decide whether a number x is in A, just wait until a stage t after s_0 at which some $y > x$ causes a switch. Then x is in A if and only if x is in A_t, since otherwise we will see a win case, contradicting the choice of s_0. This establishes Claim 3.

Dynamic arrangement of blocks. Now we organize the blocks in a Σ_2 way. At each stage we have a many blocks $B_{i,s} : i \leq a$. Each $B_{i,s}$ contains numbers e in the interval $[b_{i,s}, b_{i+1,s})$. We may imagine the b_i's as movable markers. Each marker $b_{i,s}$ gets pushed to a new position if either for some $i' \leq i$, $f'(i', s)$ changes or some requirement $R_{e,n}$ in block $B_{i',s}$, where $i' < i$, acts at stage s. More precisely, $b_{i,s}$ is the maximum of the numbers $f'(i, s) + 1$ and the largest stage t at which some requirement belonging to a higher priority block $B_{i',s}$ acts at t. When $b_{i,s}$ changes, we initialize all requirements $R_{e,n} : e \geq b_{i,s}$. Let J denote the set

$$J = \{j\colon (\exists s)(\forall t > s) b_{j,t} = b_{j,s}\}.$$

Observe that J is not empty. For example, 0 is in J because b_0 settles down at $f(0)$. J is downward closed by definition, and J is a subset of I. We now argue along J that the construction works.

Case 1. J has a largest element j_0. In this case, we can argue A is recursive as before. Let b be the final position of the j_0-th marker. $B\Sigma_2$ shows that there is a stage after which no more win cases can happen for requirements $R_{e,n} : e \leq b$. Thus, there must be unboundedly many switches in block $j_0 - 1$. By Claim 3, A is recursive.

Case 2. J is a cut. In this case, we argue that A is prompt for some g_n. As J is a subset of I, J is a proper subset of a. Observe as that each block of requirements holds only one g_n, there is some $n^* \leq a$ such that g_{n^*} is total. We argue this by contradiction. Suppose that for all $n \leq a$ there exists an x such that for all $y \geq x$, $g_n(y)$ is undefined. By $B\Sigma_2$, there is a uniform bound for these x's, which is impossible.

For simplicity, let us use g to denote g_{n^*}. We claim that A is g-prompt.

Claim 4. *A is g-prompt.*

Proof of Claim 4. Suppose W_e is infinite. We show that A is g-prompt for W_e. By the definition of J, e belongs to some permanent block B_j. Fix a stage s after which b_{j+1} never moves. This implies that no action will be taken by any requirement $R_{e,n}$ for $e \in B_j$. On the other hand, after stage s, W_e will require attention. This causes an action, contradicting the choice of s. This ends the proof of Lemma 5. □

Lemma 6. *If both A and B are prompt, then there is a nonrecursive recursively enumerable set C below both A and B.*

Proof. Without of loss of generality, we may assume that both A and B are g-prompt for some total recursive function g (otherwise, just take the maximum). We build an recursively enumerable set C satisfying the nonrecursiveness requirements:

$$P_e : C \neq \Phi_e.$$

To make C recursive in both A and B, we use the permitting method. There is no interference between different strategies.

The strategy to satisfy P_e goes as follows. Wait for a stage s, at which $\Phi_e(x) = 0[s]$ for some $x \in \mathcal{M}^{[e]}$. Then wait until stage $g(g(s))$, and see if both A and B change below x. If they do, then put x into C and satisfy the requirement forever.

By permitting, C is recursive in both A and B, because if $A \upharpoonright x = A_s \upharpoonright x$ then $x \in C$ if and only if $x \in C_s$. The same applies to B. To see that C satisfies P_e, first notice that each requirement only puts at most one number into C, so that if the recursively enumerable set

$$W = \{x \in \mathcal{M}^{[e]} : \Phi_e(x) = 0\}$$

is $\mathcal{M}$-finite, then P_e is satisfied. Suppose W is not $\mathcal{M}$-finite, then by g-promptness, the recursively enumerable set

$$V = \{x \in W : (\exists s < t)(\exists y < x)(\Phi_e(x) = 0[s] \wedge y \in A_t - A_{t-1}\}$$

is not $\mathcal{M}$-finite either. By the g-promptness of B, one of the elements in V will be permitted by B before stage $g(g(s))$, where s is the stage when x enters W, thus $g(s)$ is the stage when x enters V. This ends the proof of Lemma 6, and hence Theorem 4. □

Corollary 1. *Over the base theory $P^- + B\Sigma_2$, the existence of a recursively enumerable minimal pair is equivalent to $I\Sigma_2$.*

The above result on minimal pairs can be generalized to branching degrees in a special $B\Sigma_2$ model. This model was first studied by Mytilinaios and Slaman [13], where they constructed a $B\Sigma_2$ model $\mathcal{M}$ in which every subset of the natural numbers ω is coded on ω. We shall call that model a **saturated** model. The saturated model has other properties. For example,

Lemma 7 (Mytilinaios and Slaman [13])**.** *In a saturated $B\Sigma_2$ model $\mathcal{M}$, every recursively enumerable set is either complete or low.*

By noticing that if A is low, then any Π_1^A formula is equivalent to a Δ_2 one, we have

Corollary 2. *Let A be an incomplete recursively enumerable set in a saturated $B\Sigma_2$ model $\mathcal{M}$. Then $\mathcal{M}$ satisfies $B\Sigma_2^A$.*

Before we relativize the construction in Theorem 4 to any incomplete recursively enumerable set A, we first recall a theorem due to Lachlan.

Theorem 5 (Lachlan [10])**.** *If $\mathbf{a}, \mathbf{b}$ are recursively enumerable degrees and $\mathbf{d}$ is a degree less than or equal to $\mathbf{a}$ and $\mathbf{b}$, then there is a recursively enumerable degree $\mathbf{c}$ such that $\mathbf{d} \leq \mathbf{c}, \mathbf{c} \leq \mathbf{a}$ and $\mathbf{c} \leq \mathbf{b}$.*

The proof is exactly the same as presented in Soare [15]. The only new observation is the reduction showing $D \leq C$ is in fact strong Turing reduction. Other reductions do not matter, as they are all among recursively enumerable sets.

Theorem 6. *Let $\mathcal{M}$ be a saturated $B\Sigma_2$ model. Then there is no branching recursively enumerable degree in $\mathcal{M}$.*

Proof. Relativize the proof for minimal pairs to a potentially branching degree **a**. Then apply Lachlan's Theorem. □

It should be noted that the above proof applies to all low degrees in any $B\Sigma_2$ model.

5. $B\Sigma_2$ and Shoenfield's Conjecture

Historically, after Sacks proved the Density Theorem, Shoenfield made his conjecture which says that the recursively enumerable degrees form a dense structure as an upper semi-lattice.

Shoenfield's Conjecture: Fix any two finite upper semi-lattices with the least and greatest elements (usl) $P \subset Q$. Every usl embedding i of P into the recursively enumerable degrees $\mathcal{R}$ can be extended to an embedding j of Q into $\mathcal{R}$.

If we only require i and j to preserve partial order (not necessarily the join operation), then we have a **weaker form** of Shoenfield's Conjecture.

In classical recursion theory, the failure of Shoenfield's Conjecture was first demonstrated by Lachlan [10] and Yates [16] when they proved the existence of minimal pairs. In the case of a $B\Sigma_2$ model, we have the Density Theorem to hold [7] and yet the minimal pairs do not exist. It is natural to ask whether Shoenfield's Conjecture indeed holds in $B\Sigma_2$ models. We give two examples of the failure of Shoenfield's Conjecture.

5.1. An Example Using Meet Operators.

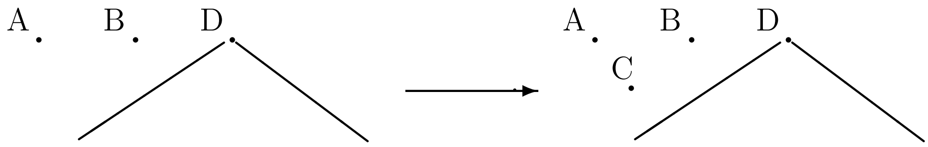
First we use the following example to demonstrate the failure of the weaker version of Shoenfield's Conjecture.

Theorem 7. *Let $\mathcal{M}$ be a model satisfying $I\Sigma_1$. Then there are pairwise incomparable recursively enumerable sets A, B and D such that for any recursively enumerable set C if $C \leq_w A$ and $C \leq_w B$ then $C \leq_w D$.*

Remarks:

(1) If $\mathcal{M}$ satisfies $B\Sigma_2$, then we do not need to distinguish between strong and weak reducibility for recursively enumerable sets. Thus the statement of Theorem 7 can be changed into a theorem about recursively enumerable degrees.

(2) This result refutes the weaker form of Shoenfield's Conjecture, because the following extension of embedding is not possible.



We have three pairs of incomparability requirements:

$$P_e : \Phi_e(A) \neq D, \quad \Phi_e(B) \neq D.$$
$$Q_e : \Psi_e(A) \neq B, \quad \Psi_e(B) \neq A.$$
$$R_e : \Theta_e(D) \neq A, \quad \Theta_e(D) \neq B.$$

We also need a requirement for the meet:

$$S_e : \text{If } \Gamma_e(A) = \Gamma_e(B) = f \text{ and } f \text{ is total, then } \exists \Delta_e(\Delta_e(D) = f).$$

Since we are working under $I\Sigma_1$, we need to make sure that if $\Gamma_e(A) = \Gamma_e(B)$ and total then the length of agreement is unbounded. (Note that it is an easy application of $B\Sigma_2$.) Therefore we add another pair of lowness requirements:

$$N^A_{\langle e,x\rangle} : \text{If } \exists^\infty s\ \Gamma_e(A;x)\downarrow [s], \text{ then } \Gamma_e(A;x)\downarrow.$$
$$N^B_{\langle e,x\rangle} : \text{If } \exists^\infty s\ \Gamma_e(B;x)\downarrow [s], \text{ then } \Gamma_e(B;x)\downarrow,$$

where $\exists^\infty$ stands for "there exist unbounded many". The Turing functionals Φ_e, Ψ_e, Θ_e and Γ_e are given. We will construct the recursively enumerable sets A, B and D and the Turing functional Δ_e.

Description of a Single Strategy In the following discussions, we drop all indices. We will use the letters P, Q, R, S and N to refer to our strategies to satisfy their associated requirements.

The strategy to satisfy N^A is the usual preservation strategy. At stage s, if $\Gamma(A;x)\downarrow [s]$ then preserve A up to the use $\gamma(A;x)[s]$. The strategy for N^B is symmetric.

The strategy to satisfy S is as follows. We enumerate a functional Δ and ensure that if $\Gamma(A)$ and $\Gamma(B)$ are total and equal, then their common value is equal to $\Delta(D)$. In the enumeration of Δ, we measure the length of agreement l between $\Gamma(A)$ and $\Gamma(B)$. If the length of agreement l increases, then define $\Delta(D;y) = \Gamma(A;y)$ for all undefined $y \leq l$ with a use $\delta(D;y)$. The use is determined as follows. The first time that we define $\delta(D;y)$, we give it a value which is big during that stage and say that we set the use for Δ at y. During subsequent stages, not necessarily expansionary, if $\Gamma(A;y)$ and $\Gamma(B;y)$ are both undefined or both defined with a common value which is incompatible with $\Delta(D;y)$, then we enumerate $\delta(D;y)$ into D and reset the use for $\delta(D;y)$ to a new big number. This is essentially the strategy introduced by Fejer [6]. The global effect of S is to put infinitely many numbers into D.

We now look at the incomparability requirements and the possible conflicts with S. They are all based on the Friedberg-Mučnik diagonalization strategy. For simplicity, we only mention one strategy for each pair of requirements, the other one being symmetric.

We begin with the simplest requirement P. The Strategy for $\Phi(A) \neq D$ is as follows. Pick a follower x targeting D. Wait until x is *realized*, i.e. $\Phi(A;x) = 0$, then put x into D and preserve A up to $\varphi(A;x)$. Putting x into D may injure some computations $\Delta(D;y)$ whose $\delta(D;y) > x$. But this conflict is not serious, as we can redefine the value $\Delta(D;y)$. Since the action of P does not change the values of $\Gamma(A;y)$ and $\Gamma(B;y)$, we do not need to change the use $\delta(D;y)$, (Note: Keeping the same use here is consistent with the choice of uses described in S-strategies). The requirement P has a finitary positive effect on D and a finitary negative effect on A or B.

The strategy for Q is similar. Consider the requirement $\Psi(A) \neq B$. Pick a follower x targeting B. Wait until x is realized. Put x in B and preserve A up to $\psi(A;x)$. As the S requirement does not restrain either A or B, putting x in B may injure S. There may exist some y such that $\Delta(D;y)\downarrow = \Gamma(B;y)$, $\Gamma(A;y)\uparrow$ and $x < \gamma(B;y)$. When this happens, Q must put $\delta(D;y)$ in D to destroy the computation $\Delta(D;y)$. The global effect of Q is the also finitary.

The strategy for R is more complicated than the others. Consider the requirement $\Theta(D) \neq B$. R acts as in Q. However, restraining D may cause some problem. When the follower x is enumerated into B, it may injure some computation $\Gamma(B;y)$ as analyzed in Q. Then $\delta(D;y)$ should enter D to correct $\Delta(D;y)$. If $\delta(D;y) < \theta(D;x)$ then R is unable to preserve $\Theta(D;x)$.

Before we modify R, we take a closer look at the conflict between R and a single S requirement. Fix a priority list:

$$P_0 < Q_0 < R_0 < S_0 < N_0^A < N_0^B < P_1 < Q_1 < R_1 < S_1 < N_1^A < \dots$$

Without loss of generality, let us assume that at any stage s, the domains of the functionals $\Gamma(A)$, $\Gamma(B)$ and Δ are downward closed. R picks a follower x_0 targeting B and waits until x_0 is realized, say at stage s_0. If there is no $y \in \operatorname{dom}\Gamma(B) \setminus \operatorname{dom}\Gamma(A)$ such that $x_0 < \gamma(B;y)$ and $\delta(D;y) < \theta(D;x_0)$, then just put x_0 into B and act as in Q. Suppose there is such a y. We will say that y (or the requirement S) **stops x_0 from entering** B. Then

R will not put x_0 into B yet. Instead, R **freezes the setting for** x_0, i.e. R sets the restraints

$$r_B = \max\{\gamma(B; z) \colon z \in \text{dom } \Gamma(B)\},$$
$$r_A = \max\{\gamma(A; z) \colon z \in \text{dom } \Gamma(A)\}$$

on B and A respectively. R also picks a new x_1 targeting B, which is larger than any number we have seen in the construction. Suppose that x_1 is realized at s_1 (otherwise R is satisfied easily). If there is a stage $v \in (s_0, s_1)$ at which the length of agreement l recovers at v, then we can preserve the A-side computation $\Gamma(A; y) = \Delta(D; y)$ instead of B-side. Hence x_0 can enter B at v, and R is satisfied (R needs to set a restraint on A to preserve l, so that no $\delta(D; y) < \theta(D; x_0)$ wants to enter D). If l does not recover at any v, then x_1 can enter B at stage s_1 as there are no new computations $\Delta(D; y)$ being defined, and the existing computations of $\Delta(D; y)$ will not stop x_1 from entering B.

In other words, we may think of R having two substrategies R_0 and R_1. R_0 has higher priority than R_1. R_0 has a follower x_0 which either enters B (in this case, R is satisfied and we do not need R_1); or is held by a finite restraint of S (in this case R_0 shows that S has a finitary outcome and leave the job to R_1). R_1 has a follower x_1 larger than the finite S-restraint. If R_0 fails, then x_1 will be the witness for R.

When there are more than one S strategy, we need to have more substrategies. In general, requirement R_e has at most e many substrategies $R_{e,n}$ for $n \leq e - 1$. $R_{e,m}$ has higher priority than $R_{e,n}$ if $m < n$. Each $R_{e,n}$ has its own follower $x_{e,n}$ and works under the assumption that $R_{e,m}$, $(m < n)$ are held by some S-requirements that have higher priority than R_e.

Construction:

Stage 0: $A_0 = B_0 = D_0 = \emptyset$, $\Delta_e(D_0; x)$ is undefined for all $e, x \in \mathcal{M}$.

Stage $s + 1$: Given A_s, B_s and D_s. For each $e \leq s$, S_e has defined $\Delta_e(D_s; x)[s]$ and R_e has had subrequirements $R_{e,0}, \ldots, R_{e,n_e}$. Each $R_{e,n}$ has a follower $x_{e,n}$. For $n < n_e$ $x_{e,n}$ is realized and x_{e,n_e} is not.

To simplify the description of actions, we adopt some conventions.

(1) At each stage, at most one requirement requires attention (otherwise we just take the least one).
(2) We automatically select followers for requirement P_e, Q_e and R_e (one for each substrategies $R_{e,n}$) such that the new followers are **big**, i.e. larger than any number we have seen in the construction, in particular larger than the restraints. We also assume that R_e starts its first substrategy $R_{e,0}$ automatically.

(3) At the end of each stage, define $\Delta_e(D, y)$ for all y such that $y < l(e, s)$ and $\Delta_e(D; y)[s]$ is not defined. The use $\delta(D; y)$ is selected as in the description of S-strategies.
(4) When a strategy or an R-substrategy acts, all requirements and substrategies of lower priority are initialized, i.e. cancel all followers, restraints and Δ's definitions.
(5) When we set a restraint on A, B or D, we set it large enough to preserve the necessary computations. In fact it is safe to set it to be s, which is the current stage number.

We say that requirement $N^A_{\langle e,x\rangle}$ (or $N^B_{\langle e,x\rangle}$) **requires attention** at stage s if $\Gamma_e(A; x)$ (or $\Gamma_e(B; x)$) is undefined at stage $s - 1$ and it is defined at stage s.

We say that requirement P_e (or Q_e) **requires attention** at stage s if P_e (or Q_e) has a follower x which is realized at stage s.

We say that substrategy $R_{e,n}$ ($n \le e_n$) **requires attention** at stage s if either

Case 1. There is an S_d which stops $x_{e,n}$ from entering A or B at stage $s - 1$ and s is a d-expansionary stage. Or

Case 2. $x_{e,n}$ is realized at stage s.

At stage $s + 1$, if no requirement requires attention, then go to the next stage. Otherwise, we take the following actions.

Suppose that $N^A_{\langle e,x\rangle}$ requires attention. Then restrain A.

Suppose that $P_e : \Phi_e(A) \neq D$ requires attention. Then put the follower x into D, set restraint on A and redefine $\Delta_e(D; y)$ as in the description of P-strategies.

Suppose that $Q_e : \Psi_e(A) \neq B$ requires attention. Then put the follower x into B, set restraint on A and reset $\Delta_e(D; y)$ as in the description of Q-strategies.

Suppose that $R_{e,n}$ which is a substrategy for $R_e : \Theta_e(D) \neq B$ requires attention. We do as follows.

If Case 1 happens, then if no S-requirements stop $x_{e,n}$, then put $x_{e,n}$ in B and preserve D and A; else freeze the setting for $x_{e,n}$.

If Case 2 happens, then x_{e,n_e} is the follower being realized. Check whether or not there is an S-requirement which stops x_{e,n_e} from entering B. If yes, then freeze the setting for x_{e,n_e} and start a new substrategy R_{e,n_e+1}. If no, put x_{e,n_e} into B, and set restraint on D and A.

<u>End of Construction</u>

We now verify within $I\Sigma_1$ that all requirements are satisfied. The following lemmas are the key ingredients in all Friedberg-Mučnik type finite injury arguments.

Lemma 8. *If a requirement acts or is initialized not more than k times for some k in $\mathcal{M}$, then there is a stage s_0 after which it never acts or is initialized.*

Proof. Notice that the function $F\colon <k \to \mathcal{M}$, defined by $F(i) = t$ if the requirement acts the i-th time at stage t, is a partial Σ_1 function. By Friedman's Theorem, the range of F is bounded. Any upper bound s_0 of the range suffices. □

Notice that the number k in Lemma 8 can be replaced by and Σ_1 function $k(e)$, where e is the index of the requirement.

Lemma 9. *There is a total recursive function $f\colon \mathcal{M} \to \mathcal{M}$ such that for each e, the e-th requirement in the priority list acts or is initialized at most $f(e)$ many times.*

Proof. Consider the function $f\colon \mathcal{M} \to \mathcal{M}$ defined by the following recursion:

$$f(0) = 1;$$
$$f(e + 1) = f(e) \cdot (e!).$$

$I\Sigma_1$ shows that f is a total recursive function. We now argue that f is the function we want.

Suppose for the sake of a contradiction that there is a requirement which acts or is initialized more than $f(e)$ many times. By construction, such requirements form a Σ_1 set. By the least number principle, there is a least such e, call it e_0. By Lemma 8, there is a stage s_0 after which no requirement of higher priority acts. If the e_0-th requirement is not of type R, then it can act at most once after s_0. If it is of type R, say it is R_e for some $e < e_0$. Let us count the number of actions of R_e after s_0. The number of actions of R_e is the sum of the number of actions of its substrategies. Notice that at any stage, there are at most e many substrategies $R_{e,n}$. For each substrategy $R_{e,n}$, if $R_{e,m}$ $(m < n)$ stops acting, then $R_{e,n}$ can act at most $e - n$ times (because there are only $e - n$ many 'unfrozen' S requirements). Thus the total number of actions of R_e is less than $e!$ after stage s_0. So the total number of actions of R_e is bounded by

$$f(e_0 - 1) \cdot (e!) < f(e_0 - 1) \cdot (e_0!) = f(e_0)$$

which is a contradiction. □

Lemma 10. *Every requirement is satisfied.*

Proof. Fix an $e \in \mathcal{M}$. For any positive requirement, say R_e, by Lemma 9 and Lemma 8, there is a stage s_0 after which R_e never acts. If R_e is unsatisfied, then R_e will have a substrategy $R_{e,n}$ which has a follower $x_{e,n}$ and $x_{e,n}$ cannot get canceled or enter B after s_0, so $x_{e,n}$ is never realized. Thus R_e is satisfied. The N requirements are satisfied by similar reasons.

For the requirement S_e, let s_0 be the stage after which S_e never gets initialized. Suppose $\Gamma_e(A) = \Gamma_e(B) = f$ which is total. We show that $\Delta_e(D) = f$. Fix a $y \in \mathcal{M}$. There is a stage s_1 after which no requirement with higher priority than $N^A_{\langle e,y+1\rangle}$ and $N^B_{\langle e,y+1\rangle}$ acts. Since $\Gamma_e(A) = \Gamma_e(B)$ is total, for any stage $t > s_1$ $l(e,t) \geq y$. So $\Delta_e(D;y)$ is defined to be the common value and its use $\delta_e(D;y)$ never moves by the convention we have made. This ends the proof of the Lemma and also the proof of the Theorem. □

5.2. **An Example Using Join Operator.** We now give another example of the failure of Shoenfield's Conjecture. We show that in any model satisfying $P^- + I\Sigma_1$, the following weaker version of Slaman-Steel Theorem [14] can be shown.

Theorem 8. *Let* $\mathcal{M}$ *be a model satisfying* $P^- + I\Sigma_1$. *Then there are recursively enumerable sets* A *and* B *such that* $0 < B < A$ *and for any recursively enumerable set* $W < A$, $B \oplus W < A$.

Slaman and Steel proved Theorem 8 without the restriction that W be recursively enumerable . As we shall see later, the set A is low, therefore the weak reducibility for sets below A coincides with the strong ones. Thus we have a corresponding version of Theorem 8 for recursively enumerable degrees.

We build recursively enumerable sets A and B together with Turing functionals Γ and Δ_e for each e in $\mathcal{M}$, such that

$$B = \Gamma(A)$$

and satisfying the following requirements:

$$P_e : B \neq \Phi_e,$$

$$N_e : [W_e = \Psi_e(A) \text{ and } \Theta_e(BW_e) = A] \Rightarrow \Delta_e(W_e) = A,$$

$$R_{\langle e,x\rangle} : \text{If } \exists^\infty s\ \Phi_e(A;x)\downarrow [s], \text{ then } \Phi_e(A;x)\downarrow,$$

where Φ_e, Ψ_e and Θ_e are fixed enumerations of Turing functionals. Since $A \geq_T B$ and B is not recursive, the requirement N_e ensures that $A \not\leq_T B$. The functional Ψ_e is introduced so that we have better control on W_e.

And the lowness requirements R are added for the same purpose as explained in Theorem 7.

Description of Strategies At every stage s, we define $\Gamma(A;x) = B(x)$ with appropriate use $\gamma(A;x)$ for all $x \leq s$. We will keep this equality at any stage $t > s$. Thus whenever a number x enters B at stage t, we must put a number less than or equal to $\gamma(A;x)$ into A and reset $\Gamma(A;x)$. This will make $B = \Gamma(A)$.

The strategy to satisfy $R_{\langle e,x\rangle}$ is the normal preservation strategy. When we see a computation $\Phi_e(A;x)\downarrow[s]$, we restrain A up to the use $\varphi_e(A;x)[s]$.

The strategy for N_e is as follows. We drop the indices in the discussion. We think that $W = \Psi(A)$. At stage s we measure two lengths of agreement

$$l_1(e,s) = \max\{y\colon (\forall z < y)(\Psi(A;z)\downarrow= W(z)[s]\}$$

and

$$l(e,s) = \max\{y\colon (\forall z < y)(\Theta(BW;z)\downarrow= A(z)[s] \wedge \theta(BW;z) < l_1(e,s))\}.$$

When $l(e,s)$ increases, we set $\Delta(W;z) = A(z)$ for all undefined z up to $l(e,s)$ and define the use $\delta(W;z)$ of the computation to be larger than $\theta(BW;z)$. We also keep this equality at any stage $t > s$. The effect of N, when the injury is absent, is to enumerate more and more axioms into Δ.

The first attempt for P_e is the usual Friedberg-Mučnik diagonalization strategy. Pick a follower x not yet in B, wait until x is realized, i.e., $\Phi(x)\downarrow= 0$, then put x into B. However this action conflicts with N. Suppose x is realized and x enters B. To keep $\Gamma(A) = B$, we need to put a number less than or equal to $\gamma(A;x)$ into A to correct $\Gamma(A;x)$. But $\Delta(W;\gamma(A;x))$ may be defined already (if not, then there is no conflict) and W may not change below $\delta(W;\gamma(A;x))$, then we have no chance to reset $\Delta(W;\gamma(A;x))$.

To solve this conflict, we modify the strategy P as follows. Let us consider only one requirement N at this moment. P first picks a pair of numbers (y,x) such that $y \notin A$ is targeting A and $x \notin B$ is targeting B respectively. Unless the requirement P acts, we will ensure that

(1) $x \notin B$ and $y \notin A$.
(2) If $\theta(BW;y)$ is defined then $\theta(BW;y) < x$ and $\theta(BW;y) < \delta(W;y)$.

We need to argue that such pair (y,x) can be found. Initially when we choose x and y, if $\theta(BW;y)$ is defined, then we just pick $x > \theta(BW;y)$, and preserve A up to $\max\{\psi(A;z)\colon z \leq \theta(BW;y)\}$ and preserve B up to $\theta(BW;y)$. Then either $\theta(BW;y)$ never moves (so (2) is satisfies), or W changes below $\theta(BW;y)$. Thus $\Psi(A) \neq W$ forever, so N is satisfied.

Now let us suppose that $\theta(BW;y)$ is undefined, but after x is chosen, $\theta(BW;y)$ is defined and is larger than x. Then we discard the old number x and choose a new number $x' > \theta(BW;y)$ and proceed as before. In any case, we can select the pair (y,x) satisfying (1) and (2).

The action of P is as follows. By the choice of the pair (y,x), when x is realized, $\theta(BW;y)$ is either undefined or it is less than x. Then we can just put y into A and x in B, restrain B up to $\theta(BW;y)$ and redefine $\Gamma(A;x)$. This action will satisfy P. It is also compatible with N. The reason is as follows. At the stage of action,

$$\Theta(BW;y) = 0 \neq 1 = A(y).$$

There are two possible cases:
Case 1 W does not change below $\theta(BW;y)$. Then due to the restraint on B, $\Theta(BW;y) = \Theta(BW;y)[s] = 0$. Therefore, $\Theta(BW) \neq A$.
Case 2 W changes below $\theta(BW;y)$. By condition (2), $\theta(BW;y)$ is less than $\delta(W;y)$. Thus we can redefine $\Delta(W;y)$ too.

Construction:

Fix a priority list

$$N_0 < P_0 < R_0 < N_1 < P_1 < R_1 < \dots$$

Stage 0: Set $A_0 = B_0 = \emptyset$. Set $\Gamma(A;y)[0]$ and $\Delta_e(W;x)[0]$ to be undefined for all $x, y \in \mathcal{M}$.

Stage $s+1$. Given $A_s, B_s, \Gamma(A_s;y)$ for $y < s+1$ and $\Delta_e(W_e;x)$ for all $e, x < s+1$.

We say that a requirement $R_{\langle e,x\rangle}$ **requires attention** if $\Phi_e(A;x)$ is defined at stage s.

We say that a requirement P_e **requires attention** if it is not satisfied yet and one of the following conditions holds:

(A1) P_e has no pair of followers (y,x); or
(A2) P_e has a pair of followers (y,x), and there is an $i \leq e$ such that $\Theta_i(BW_i;y)$ is defined at stage s.
(A3) P_e has a pair of followers (y,x), condition (A2) does not happen, and $\Phi_e(x){\downarrow}= 0$ at stage s.

If no requirement requires attention, then define $\Gamma(A;s) = 0$ with use $s+1$. If there is an e less than or equal to s such that $l(e,s)$ increases, then define $\Delta_e(W_e;z) = A(z)$ with use $\delta_e(W_e;z) = s$, (so in particular, for all $z \leq s$, $\delta_e(W_e,z) > \theta(BW_e;z)$).

If P_e requires attention, then we act based on the conditions.
Case 1. P_e requires attention because (A1) holds.

Then pick a fresh pair (y, x), which means: (i) $y \notin A_s$ and $x \notin B_s$; (ii) for all $i < e$, y and x are larger than the restraint set by P_i on A and B respectively; (iii) x is larger than $\theta_i(BW_i; y)$ if such θ_i is defined at stage s; (iv) (y, x) has not been chosen by P_i for $i \leq e$. Set a restraint on B up to

$$\max\{\theta_i(BW_i; y) \colon i < e \text{ if it is defined}\}$$

and set a restraint on A up to

$$\max\{\psi_i(A; z) \colon i < e, \wedge z \leq \theta_i(BW_i; y).\}.$$

Initialize all unsatisfied requirements N_j and P_j for $j > e$, that is, for N_j, start over the definition of $\Delta_j(W_j)$; for P_j, cancel the pair of followers (y, x). Extend the definition of $\Gamma(A)$ and $\Delta_i(W_i)$ for $i \leq e$ as before.

Case 2. P_e requires attention because (A2) holds.

Then cancel follower x, reselect a fresh x' to replace x, (so in particular, $x' > \theta_i(BW_i; y)$). Set the restraint on A and B. Initialize all unsatisfied N_j and P_j for $j > e$. Extend the definition of $\Gamma(A)$ and $\Delta_i(W_i)$ as in Case 1.

Case 3 P_e requires attention because (A3) holds.

Then put y in A and x in B. Keep the restraint on B up to $\theta_i(BW_i; y)$. Initialize all unsatisfied N_j and P_j for $j > e$. Set $\Gamma(A; x) = 1$ with empty use; set $\Gamma(A; z)[s + 1] = \Gamma(A; z)[s]$ with the same use $\gamma(A; z)[s]$ for $z \neq x$ and $z < s$; set $\Gamma(A; s) = 0$ with use s. Define $\Delta_i(W_i)$ as in Case 1. Declare that P_e is satisfied.

If $R_{\langle e,x\rangle}$ requires attention, then restrain A up to $\varphi_e(A; x)[s]$. Initialize all lower priority requirements.

End of Construction

We now verify that under $P^- + I\Sigma_1$, the constructions works.

Lemma 11. $\Gamma(A) = B$.

Proof. In the construction, whenever we put a number x into B, we also put a number $y \leq \gamma(A; x)$ into A. Thus $\Gamma(A) = B$. □

Lemma 12. *For each $e \in \mathcal{M}$, there is a stage s after which the requirements indexed by e never act or get initialized.*

Proof. Notice that if no requirement with indices less than e acts, then R_e can act at most once and P_e can act at most $e + 2$ times: Once for (A1); e times for (A2) (each N_i $(i < e)$ makes P_e act once); one more time for (A3). The rest of the proof is similar to the ones of Lemma 8 and Lemma 9. □

Lemma 13. *For any e in $\mathcal{M}$, the requirement P_e is satisfied.*

Proof. Fix e, let s_0 be a stage such that after s_0 no requirement P_i $(i \leq e)$ will act. We argue that P_e is satisfied. First P_e has a pair of followers (y, x). If x is in B_{s_0} then P_e is satisfied. Assume that $x \notin B_{s_0}$ and $y \notin A_{s_0}$, then this pair of followers will remain the same since there are no actions of higher priority requirements after s_0. If x is not realized at any stage $t > s_0$ then P_e is satisfied. If x is realized at some stage $t > s_0$ then P_e will act, which is a contradiction. □

Lemma 14. *For any e, x in $\mathcal{M}$, the requirement $R_{\langle e,x\rangle}$ is satisfied. Consequently if $\Phi_e(A)$ is total, then for any $x \in \mathcal{M}$ there is a stage s such that for all $t > s$ $\Phi_e(A) \upharpoonright x[t] = \Phi_e(A) \upharpoonright x[s]$.*

Proof. Let s be a stage after which there is no action by any requirement of higher priority than $R_{\langle e,x\rangle}$. If $\Phi_e(A; x)\downarrow$ after s, then the computation is preserved, which shows that $R_{\langle e,x\rangle}$ is satisfied.

Now assume that $\Phi_e(A)$ is total. The choice of s implies that if $x' < x$ then $R_{\langle e,x'\rangle}$ does not act after stage s. Since each $R_{\langle e,x'\rangle}$ will act to preserve the final computation of $\Phi_e(A; x')$, this computation must exist at stage s and be preserved by $R_{\langle e,x'\rangle}$ during all later stages, which establishes the lemma. □

Lemma 15. *For any e in $\mathcal{M}$, the requirement N_e is satisfied.*

Proof. Fix an e in $\mathcal{M}$. Fix a stage s_0 such that for all $t > s_0$ and for all $i < e$ P_i never acts at stage t. Hence after stage s_0 Δ_e never gets initialized.

Suppose $\Theta_e(BW_e) = A$, we need to show that $\Delta_e(W_e) = A$. Fix y. First we claim that if $\Gamma(A) = B$, $\Psi_e(A) = W_e$ and $\Theta_e(BW_e) = A$, then there is a stage s_1 such that for all $t \geq s_1$, $l(e, t) > y$. The reason is as follows. By Lemma 14, if a set C is weakly recursive in A, then C is also strongly recursive in A. Thus for any set D if $D \leq_T C \leq_T A$ then $D \leq_T A$. Hence if $\Gamma(A) = B$, $\Psi_e(A) = W_e$ then $\Theta_e(BW_e)$ is equal to $\Phi^*(A)$ for some Turing functional Φ^*. By Lemma 14, there is a stage t_1 after which $\Phi^*(A) \upharpoonright (y + 1)$ never changes. Since A is regular, there is a stage t_2 after which $A \upharpoonright (y+1)$ never changes. Thus for any stage $t > s_1 = \max\{t_1, t_2\}$, the length of agreement $l(e, t) > y$. Therefore $\Delta(W_e; y)$ is defined at s_1 and it is equal to $A(y)[s_1]$.

The worry is that after $\Delta(W_e)(y)$ is defined, y enters A. Hence we may assume that y is chosen by a positive requirement P_j for some $j > e$. Let (y, x) be the pair of followers chosen by P_j.

Suppose that at the stage s, y enters A. Then by construction, a restraint up to $\theta_e(BW_e; y)$ is set on B. We argue that either W_e changes below $\theta(BW_e; y)$ or there is a $z \leq y$ such that $\Theta(BW_e; z) \neq A(z)$. Suppose that W_e does not change below $\theta(BW_e; y)$. Then to make $\Theta(BW_e) = A$, B must change below $\theta(BW_e; y)$. In other words, there is a k such that $e < k < j$ and P_k is satisfied after stage s. By $I\Sigma_1$, the set of such k is $\mathcal{M}$-finite, hence has a least element k_0. Let (z, x) be the pair of followers chosen by k_0. At the stage P_{k_0} acts, z enters A and $\Theta(BW_e; z) = 0$. By the choice of k_0, B does not change below $\theta(BW_e, z)$ because of the restraint. By assumption, W_e does not change below $\theta(BW_e; z)$ either. Therefore, $\Theta(BW_e; z) \neq A(z)$.

In any case, N_e is satisfied, which establishes the Lemma and the Theorem. □

We end our paper with a few open questions.

Open Problems

1. Is there a model $\mathcal{M}$ of $I\Sigma_1$ but not $B\Sigma_2$, a Σ_2 cut I in $\mathcal{M}$, and a family of total recursive functions $\{h_i : i \in I\}$ which has the dominating property in $\mathcal{M}$?

   The result we have got in Section 2 does not offer us much insight, as in that particular model, the whole model $\mathcal{M}$ is a Δ_2 rearrangement of the Σ_2 cut ω.
2. Is there a $B\Sigma_2$ model $\mathcal{M}$ such that $\mathcal{M}$ is not saturated and such that there is a branching degree in $\mathcal{M}$?

References

1. K. Ambos-Spies, Carl G. Jockusch, Jr., Richard A. Shore, and Robert I. Soare, *An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees*, Trans. Amer. Math. Soc. **281** (1984), 109–128.
2. C. T. Chong and K. J. Mourad, Σ_n *definable sets without* Σ_n *induction*, Trans. Amer. Math. Soc. **334** (1992), no. 1, 349–363.
3. C. T. Chong and Yue Yang, Σ_2 *induction and infinite injury priority arguments, part II: Tame* Σ_2 *coding and the jump operator*, Ann. Pure Appl. Logic **87** (1997), no. 2, 103–116, Logic Colloquium '95 Haifa.
4. ______, Σ_2 *induction and infinite injury priority arguments, part I: Maximal sets and the jump operator*, J. Symbolic Logic, **63** (1998), no. 3, 797–814.
5. ______, *Recursion theory in weak fragments of Peano arithmetic; a study of cuts*, Proc. Sixth Asian Logic Conference, Beijing 1996 (Singapore), World Scientific, 1998, 47–65.
6. P. A. Fejer, *Branching degrees above low degrees*, Trans. Amer. Math. Soc. **273** (1982), 157–180.
7. Marcia J. Groszek, M. E. Mytilinaios, and Theodore A. Slaman, *The Sacks density theorem and* Σ_2*-bounding*, J. Symbolic Logic **61** (1996), no. 2, 450–467.
8. Marcia J. Groszek and Theodore A. Slaman, *On Turing reducibility*, Preprint, 1994.
9. L. A. Kirby and J. B. Paris, Σ_n*-collection schemas in arithmetic*, Logic Colloquium '77 (Amsterdam), North–Holland Publishing Co., 1978, pp. 199–209.

10. Alistair H. Lachlan, *Lower bounds for pairs of recursively enumerable degrees*, Proc. London Math. Soc. (3) **16** (1966), 537–569.
11. Karim Joseph Mourad, *Recursion theoretic statements equivalent to induction axioms for arithmetic*, Ph.D. thesis, The University of Chicago, 1988.
12. Michael Mytilinaios, *Finite injury and Σ_1-induction*, J. Symbolic Logic **54** (1989), no. 1, 38–49.
13. Michael E. Mytilinaios and Theodore A. Slaman, *Σ_2-collection and the infinite injury priority method*, J. Symbolic Logic **53** (1988), no. 1, 212–221.
14. Theodore A. Slaman and John R. Steel, *Complementation in the Turing degrees*, J. Symbolic Logic **54** (1989), no. 1, 160–176.
15. Robert I. Soare, *Recursively enumerable sets and degrees*, Perspectives in Mathematical Logic, Omega Series, Springer–Verlag, Heidelberg, 1987.
16. C. E. M. Yates, *A minimal pair of recursively enumerable degrees*, J. Symbolic Logic **31** (1966), 159–168.

Department of Mathematics, Faculty of Science, National University of Singapore, Lower Kent Ridge Road, Singapore 119260.

Department of Mathematics, Faculty of Science, National University of Singapore, Lower Kent Ridge Road, Singapore 119260.

Department of Mathematics, University of California Berkeley, Berkeley, CA 94720, USA.

Department of Mathematics, Faculty of Science, National University of Singapore, Lower Kent Ridge Road, Singapore 119260.