Splitting theorems and strong reductions

Wang, Shaoyi

2019

Wang, S. (2019). Splitting theorems and strong reductions. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/106851>

<https://doi.org/10.32657/10220/49671>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

SPLITTING THEOREMS AND STRONG REDUCTIONS

WANG SHAOYI

**SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES
2019**

SPLITTING THEOREMS AND STRONG REDUCTIONS

WANG SHAOYI

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

A thesis submitted to the Nanyang Technological
University in partial fulfilment of the requirement for the
degree of Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

22/01/2019
.....

Date

Wang Shaoyi / Wang Shaoyi
.....

Author Name

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it of sufficient grammatical clarity to be examined. To the best of my knowledge, the thesis is free of plagiarism and the research and writing are those of the candidate's except as knowledge in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethic policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

22/01/2019
Date

Wu Guohua

Supervisor Name

Authorship Attribution Statement

This thesis does not contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

22/01/2019
.....

Date

Wang Shaoyi / Wang Shaoyi
.....

Author Name

Abstract

This thesis contains the following parts:

In Chapter 1, we briefly review the background of recursion theory, with stress on weak-truth-table degrees and d.r.e. degrees.

In Chapter 2, we use splitting to show that the d.r.e. *wtt*-degrees are dense. This improves Wu and Yamaleev's work and gives a contrast to the nondensity theorem of d.r.e. Turing degrees.

In Chapter 3, we present a set-splitting, and also degree-splitting of an non-recursive r.e. set, into two properly d.r.e. sets, with some new ideas provided.

In Chapter 4, we generalize a result of Diamondstone and construct an r.e. set which is not cuppable by any superlow d.r.e. set.

Acknowledgements

First of all, I owe my deepest gratitude to my supervisor Prof. Wu Guohua, for his invaluable encouragement and guidance during my PhD studies. This thesis would not have been possible without his consistent and inspiring instructions. From his supervision, I learn the mathematical knowledge and the method of how to do mathematical research, both of which will help me a lot for many years. His guidance helped me in all the time of research.

Moreover, I would like to thank my fellow group members who shared with me the experience and idea in research and helped me a lot in my daily life. Thanks for accompanying me these years, for always being there when I am in need.

Last but not least, I want to thank my parents. They provide unending encouragement, support and patience, which have accompanied me throughout my studies. I owe them everything and wish I could show them just how much I love and appreciate them.

Wang Shaoyi

Jan 2019

Contents

Abstract	1
Acknowledgements	2
Table of Contents	3
1 Introduction	5
1.1 Recursive functions and sets	5
1.2 Reduction, Turing jump, r.e. sets and degrees	7
1.3 Weak-truth-table degrees	11
1.4 D.r.e. sets and degrees	15
1.5 Thesis contents	17
2 Density in d.r.e. weak-truth-table degrees	21
2.1 Setting up	21
2.2 Satisfying the P -requirement	22
2.3 Satisfying one N -requirement	24
2.4 Construction	29

2.5	Verification	31
3	Set-splittings of r.e. sets	35
3.1	Construction of a properly d.r.e. set	35
3.2	Construction of B_0 and B_1	37
3.3	Construction	41
3.4	Verification	44
4	A generalization of a theorem of Diamondstone	47
4.1	Setting up	48
4.2	Basic strategy	50
4.3	Priority restraints	54
4.4	Extension to d.r.e. case	63
4.5	Construction	71
4.6	Verification	71
	Bibliography	75

Chapter 1

Introduction

1.1 Recursive functions and sets

The study of computability theory starts from the question whether a given function $f : \mathbb{N} \rightarrow \mathbb{N}$ can be computed through a “mechanical”, or “effective” process (in standard words, such kind of process is referred to as an “algorithm”). By “mechanical”, we mean that there is a collection of preinstalled rules of operations, the process can proceed all following these rules for any given natural number n as the input, further, it terminates and correctly outputs $f(n)$ if and only if f is defined at n .

During the time of 1930s, the beginning period of this research area, several definitions related to the concept of “computation” were raised separately, including the inductive definition of partial recursive functions (Kleene), λ -calculus (Church), and Turing machine (Turing). Each of these definitions presents a class of functions, which turn out to be equivalent, e.g., any function that satisfies the recursive property under Kleene’s definition is also Turing

computable. Thus, the definition of “recursiveness” is robust. Even for modern computers today, with highly improved performance, the ability of computing has not been stronger than the computation models in those early days.

Using Gödel’s numbering on Turing machines, we can have a total list of all partial recursive functions: $\varphi_1, \varphi_2, \dots$.

As a set of natural numbers can be identified using its characteristic function, the property of computability can then be naturally transferred to sets. Intuitively speaking, a set $A \subseteq \mathbb{N}$ is *recursive* if there is some algorithm deciding whether $n \in A$ or not for each n .

A set $A \subseteq \mathbb{N}$ is *recursively enumerable*, or r.e., if A is either empty, or the members of A can be effectively enumerated. More precisely speaking, for a nonempty A , there is a recursive function f , such that $A = \{f(0), f(1), \dots\}$. Different from the case of a recursive set, here we have an algorithm to find $n \in A$ if it is true, but this algorithm cannot make the identification if n is not in A . This is why we also call an r.e. set semi-decidable. An equivalent condition of A being recursive is that both A and its complement, $\overline{A}$, are recursively enumerable.

A well-known fact about the r.e. sets is that they are domains of partial recursive functions. We use W_i to denote the domain of φ_i . Thus, we have an effective list of all r.e. sets: $W_1, W_2, \dots$.

Another fact of r.e. sets is that they are Σ_1 sets. Turing’s seminal work proved that not all r.e. sets are recursive, by providing the halting problem as an example. The halting set K is defined as $K = \{e \mid \varphi_e(e) \downarrow\}$.

The existence of nonrecursive sets leads to another issue in computability theory: how can we differentiate the levels of nonrecursiveness of sets. For this purpose, we need some concepts to measure them.

1.2 Reduction, Turing jump, r.e. sets and degrees

The concept of reduction originates from the idea of computing a set with the knowledge of another set. This idea can be achieved through an extension on the Turing machine, by adding some external information, called the oracle, to the original model. The oracle encodes the information of another set B , like a “black box”. During the computing process, the Turing machine can refer to the oracle and get definite results for questions like “is $n \in B$ ”, and work with the results in the computation. With such modification, the Turing machine becomes an oracle Turing machine. The notion of “recursive” are generalized to the notion “recursive in B ” correspondingly.

For sets A and B of natural numbers, we say that A is Turing reducible to B , denoted as $A \leq_T B$, if A is computable through an oracle Turing machine with B as the oracle. It is easy to see that the relation of Turing reducibility is reflexive ($A \leq_T A$) and transitive ($A \leq_T B, B \leq_T C \Rightarrow A \leq_T C$).

Instead of Turing functions, we call the instructions of computation with oracles *Turing functionals*, using upper Greeks to represent. Similarly, we can have an effective list of them: $\Phi_1, \Phi_2, \dots$.

We can also define jump operator using oracles. We say B is the *Turing*

jump of A if $B = \{e \mid \Phi_e^A(e) \downarrow\}$, written as $B = A'$. Clearly, $A \leq_T A'$. We can iteratively define $A'' = (A')'$, A''' and so on.

Through the relation of reduction, we can induce an equivalence relation on the power set of natural numbers. Say that two sets A and B are *Turing equivalent*, denoted as $A \equiv_T B$, if both $A \leq_T B$ and $B \leq_T A$. This equivalence relation gives rise to the corresponding equivalence classes, where each class is called a *Turing degree*. Reduction and jump can be naturally transferred to Turing degrees. We use $\mathbf{0}$ to denote the degree of recursive sets. The structure of Turing degrees makes a partial order under Turing reduction, with $\mathbf{0}$ as the least element.

A Turing degree is an r.e. degree if it contains an r.e. set. The degree of K , also easy to see as $\mathbf{0}'$, is r.e., as K itself is r.e.. It turns out that $\mathbf{0}'$ is the greatest r.e. degree, as all r.e. sets are Turing reducible to K .

There has been extensive research on the structure of Turing degrees in the 1950s and 1960s, when people proved several fundamental theorems on this structure. It is easy to see that Turing degrees make an upper-semi lattice, and the Turing jump is an increasing mapping on Turing degrees. Spector proved the existence of exact pairs of infinite increasing sequences of Turing degrees, showing that Turing degrees do not form a lattice. Spector also proved the existence of minimal degrees below $\mathbf{0}''$, and hence this structure is not dense. Sacks proved that such minimal degrees can be below $\mathbf{0}'$ and later Yates showed that minimal degrees can exist below any nonzero degree containing r.e. sets. The techniques developed in this period are finite extension methods, infinite

extension methods, forcing on trees, etc.

Post asked whether there exists some other r.e. degrees, strictly between $\mathbf{0}$ and $\mathbf{0}'$. He himself put much effort towards this question, and raised many ideas in the attempt to construct a set with such degree. The concepts he proposed are simple sets, hypersimple sets and hyperhypersimple sets, but none of them succeeded as these sets can be Turing complete, i.e. can have degree $\mathbf{0}'$. However, Post's attempts gave insight to many other degree structures, as we will see soon, as simple sets cannot be m -complete, and hypersimple sets cannot be truth-table complete.

Post's problem was eventually solved by Friedberg [21] and Muchnik [31] separately, and both of them used a new technique, now called finite injury argument. They proved that there are two incomparable r.e. degrees. Sacks proved in [37] that any nonzero r.e. degree is the joint of two smaller r.e. degrees, where Sacks introduced the Sacks preservation strategy to ensure that some requirement will eventually be satisfied, even we cannot foresee how many times we need to put on restraints to preserve it. This method, together with the Sacks coding strategy, enables Sacks to prove that the r.e. degrees are dense in [38]. This work shows the start of infinite injury argument. After this, Lachlan [26] and Yates [43] proved the existence of minimal pairs in r.e. degrees, by very different methods.

Lachlan [27] proved in 1975 that Sacks splitting theorem and Sacks density theorem cannot be combined, by showing the existence of two r.e. degrees $\mathbf{a} > \mathbf{b}$ such that if $\mathbf{a} = (\mathbf{c}_0 \vee \mathbf{b}) \vee (\mathbf{c}_1 \vee \mathbf{b})$, then either $\mathbf{a} = \mathbf{c}_0 \vee \mathbf{b}$ or $\mathbf{a} = \mathbf{c}_1 \vee \mathbf{b}$. This

work is called Lachlan's nonsplitting theorem, and the method developed in this work was called $0'''$ argument, which was also called the monster method in 1980s. Harrington [24] improved this nonsplitting by showing that $\mathbf{a}$ can be $\mathbf{0}'$. That is, there exists an r.e. degree $\mathbf{b}$ such that $\mathbf{0}'$ cannot be split nontrivially above $\mathbf{b}$. Using this monster method, Lachlan [28] proved in 1979 the existence of nonzero r.e. degrees bounding no minimal pairs.

Besides Turing reduction, several other types of reduction have been defined to observe the relative computability between sets under various conditions and requirements, including:

- Many-one reduction: $A \leq_m B$, if there is a computable function f such that for each x , $x \in A$ if and only if $f(x) \in B$. In fact, a set A is r.e. if and only if $A \leq_m K$.
- One-one reduction: $A \leq_1 B$, if the function f in many-one reduction is a one-to-one function.
- Truth-table reduction: $A \leq_{tt} B$, if there is a computable function f such that for each x , $x \in A$ if and only if $B \models \sigma_{f(x)}$. Here $\{\sigma_i\}_{i \in \omega}$ is a total list of truth tables.
- Weak-truth-table reduction: $A \leq_{wtt} B$, if A is Turing reducible to B , with a computable function f which bounds the use of the reduction. The use function is defined as: for the reduction $A = \Phi^B$, $\phi(x)$ is the largest number n in B that is referred to while computing $\Phi^B(x)$.

These reductions are called strong reductions. The implication relation of them is:

$$A \leq_1 B \Rightarrow A \leq_m B \Rightarrow A \leq_{tt} B \Rightarrow A \leq_{wtt} B \Rightarrow A \leq_T B.$$

The definition of Turing degrees can be adapted to other notions of reduction in the same way. Let r be one of the reductions, the equivalent classes induced from r -reducibility make the structure of r -degrees, and we can examine as above the structure of r -degrees, r.e. r -degrees, etc. In this thesis, we are mainly focused on the wtt -degrees.

1.3 Weak-truth-table degrees

Weak-truth-table reduction, wtt for short, was first introduced by Friedberg and Rogers. As mentioned above, weak-truth-table reduction is essentially similar to Turing reduction, but with its use function bounded by a recursive function.

As wtt reduction is stronger than Turing reduction, two sets of the same wtt degree are obviously of the same Turing degree, and a Turing degree may contain more than one wtt -degrees. An r.e. Turing degree that contains exactly one wtt degree is called a *contiguous* degree. Obviously, $\mathbf{0}_T$ is contiguous. The existence of nonrecursive contiguous degree was shown by Ladner and Sasso [29]. Their result implies that every nonrecursive r.e. degree bounds a contiguous degree.

Theorem 1 (Ladner and Sasso [29]). *For every nonrecursive r.e. Turing degree*

$\mathbf{b}$, there exists a nonrecursive contiguous r.e. Turing degree $\mathbf{a} < \mathbf{b}$.

Ladner and Sasso also proved in [29] that noncontiguous degrees exist below every nonzero r.e. degree $\mathbf{b}$. In [39], Stob proved the existence of r.e. contiguous degree as a top of a minimal pair.

Theorem 2 (Stob [39]). *There is a minimal pair of r.e. degrees with a contiguous supremum.*

Stob pointed out that Theorem 2 has many consequences of continuity of cupping and capping.

The construction of nonrecursive contiguous degrees was studied by Ambos-Spies with great details, where, based on these details, he was able to construct various contiguous degrees:

Theorem 3 (Ambos-Spies [1]). *Given r.e. degrees $\mathbf{b}$ and $\mathbf{c}$ such that $\mathbf{b} \not\leq \mathbf{c}$ and $\mathbf{c}$ is low, there is an r.e. contiguous degree $\mathbf{a}$ such that $\mathbf{a} < \mathbf{b}$ but $\mathbf{a} \not\leq \mathbf{c}$.*

Theorem 4 (Ambos-Spies [1]). *There is a noncappable r.e. contiguous degree $\mathbf{a}$, i.e., for every r.e. degree $\mathbf{b} > \mathbf{0}$, there is an r.e. degree $\mathbf{c}$ such that $\mathbf{0} < \mathbf{c} \leq \mathbf{a}, \mathbf{b}$.*

Nonetheless, the existence of contiguous degrees among r.e. Turing degrees is limited. Robinson [34] showed that the r.e. Turing degrees $\mathbf{a}$ with $\mathbf{a}' = \mathbf{0}''$ cannot be contiguous. As pointed out by Cohen in his PhD thesis [6], an r.e. contiguous degree $\mathbf{a}$ must be low_2 ($\mathbf{a}'' = \mathbf{0}''$). Ambos-Spies and Fejer [2] proved the following nowhere dense result:

Theorem 5 (Ambos-Spies and Fejer [2]). *The r.e. contiguous degrees are nowhere dense in the low r.e. degrees. If $\mathbf{a} < \mathbf{b}$ are low r.e. degrees, then there are r.e. degrees $\mathbf{c}, \mathbf{d}$, such that $\mathbf{a} < \mathbf{c} < \mathbf{d} < \mathbf{b}$ and there is no r.e. contiguous degree in the interval $[\mathbf{c}, \mathbf{d}]$.*

In contrast to Lachlan's nonsplitting theorem, Ladner and Sasso proved in [29] that splitting and density can be combined in r.e. *wtt*-degrees.

Theorem 6 (Ladner and Sasso [29]). *Given two r.e. wtt-degrees $\mathbf{b} < \mathbf{a}$, there are incomparable r.e. wtt-degrees $\mathbf{c}_0$ and $\mathbf{c}_1$ such that $\mathbf{b} < \mathbf{c}_0, \mathbf{c}_1 < \mathbf{a}$, and $\mathbf{c}_0 \vee \mathbf{c}_1 = \mathbf{a}$.*

This result implies that if an r.e. Turing degree is not contiguous, then it contains infinitely many r.e. *wtt*-degrees. The main reason for this difference is due to the recursive bounds of the use in the *wtt*-reductions. Generally, proofs for *wtt*-reductions are simpler than for Turing reduction. For those properties that require infinite injury construction for Turing degrees, it is possibly sufficient to use finite injury to handle for *wtt*-degrees.

On the other hand, *wtt*-degrees can have properties which are not true for Turing degrees.

Theorem 7 (Ladner and Sasso [29]). *Every nonzero r.e. wtt-degree $\mathbf{a}$ has the anti-cupping property. That is, there is some $\mathbf{b} < \mathbf{a}$ such that for any $\mathbf{c} < \mathbf{a}$, $\mathbf{b} \vee \mathbf{c} < \mathbf{a}$.*

Harrington proved that this property is not true for Turing degrees.

Based on Ladner and Sasso's work of combining splitting and density, Downey proved that the splitting can have infimum in [15]:

Theorem 8. *Given r.e. wtt-degrees $\mathbf{b} < \mathbf{a}$, there exist r.e. wtt-degrees $\mathbf{c}_0, \mathbf{c}_1$ such that $\mathbf{b} < \mathbf{c}_0, \mathbf{c}_1 < \mathbf{a}$, $\mathbf{c}_0 \vee \mathbf{c}_1 = \mathbf{a}$ and $\mathbf{c}_0 \wedge \mathbf{c}_1$ exists.*

This shows that we can embed a lattice into the interval $[\mathbf{b}, \mathbf{a}]$ preserving the top. Downey also proved that any incomplete r.e. wtt-degree is the bottom of a lattice.

Theorem 9 (Downey [15]). *Given an r.e. wtt-degree $\mathbf{b} \neq \mathbf{0}'$, there exists an r.e. wtt-degree $\mathbf{a} > \mathbf{b}$ such that the interval $[\mathbf{b}, \mathbf{a}]$ is a lattice.*

Actually, lattices are dense in the r.e. wtt-degrees:

Theorem 10 (Downey [15]). *Given r.e. wtt-degree $\mathbf{b} < \mathbf{a}$, there are r.e. wtt-degrees $\mathbf{c}, \mathbf{d}$ with $\mathbf{b} \leq \mathbf{c} < \mathbf{d} \leq \mathbf{a}$ such that the interval $[\mathbf{c}, \mathbf{d}]$ is a lattice.*

However, it turns out that the whole structure of r.e. wtt-degrees is not a lattice, as we have the following result regarding pairs of wtt-degrees without infimum:

Theorem 11 (Downey [15]). *Given an r.e. wtt-degree $\mathbf{a} \neq \mathbf{0}'$, there are r.e. wtt-degrees $\mathbf{b}, \mathbf{c}$ above $\mathbf{a}$ such that $\mathbf{b} \vee \mathbf{c} = \mathbf{0}'$ and $\mathbf{b} \wedge \mathbf{c}$ does not exist.*

Downey also pointed out that some initial segment of r.e. wtt-degrees may not form a lattice, by showing the existence of an r.e. wtt-degree $\mathbf{a}$ such that any non-trivial initial segment of r.e. wtt-degrees below $\mathbf{a}$ is a lattice.

Perhaps the following property is the most famous difference between the r.e. *wtt*-degrees and the r.e. Turing degrees: the r.e. *wtt*-degrees form a distributive upper semi-lattice, while the upper semi-lattice of r.e. Turing degrees is not distributive.

1.4 D.r.e. sets and degrees

A set D is d.r.e., if D is the difference of two r.e. sets A_0 and A_1 with $A_0 \supseteq A_1$. Another way of defining d.r.e. sets is through recursive approximations. For a set A , suppose that there is a recursive 0-1 valued function $f(x, s)$ such that for any x :

1. $f(x, 0) = 0$;
2. $A(x) = \lim_s f(x, s)$;
3. $|\{s + 1 : f(x, s + 1) \neq f(x, s)\}| \leq 2$.

We can generalize this definition by replacing 2 with any natural number n , and we have n -r.e. sets. Obviously, r.e. sets are 1-r.e. sets. We can also replace n by a recursive function g , and we get ω -r.e. sets.

A Turing degree is a d.r.e. degree if it contains a d.r.e. set. A d.r.e. degree is called properly d.r.e., if it contains no r.e. set.

We have seen many structural properties of r.e. degrees. In the study of the structure of d.r.e. degrees, we ask whether the same structural properties are true for d.r.e. degrees. Arslanov proved that every nonzero d.r.e. degree is

cuppable in the d.r.e. degrees, providing the first structural difference between d.r.e. degrees and r.e. degrees. Downey proved later that the diamond lattice can be embedded into the d.r.e. degrees preserving 0 and 1, providing another difference between these two structures. A fundamental difference between these two structures is that the d.r.e. degrees are not dense, proved by Cooper, Harrington, Lachlan, Lempp, and Soare [11]:

Theorem 12. *There exists a maximal d.r.e. degree. That is, there exists a d.r.e. degree $\mathbf{d} < \mathbf{0}'$ such that there is no d.r.e. degree between $\mathbf{d}$ and $\mathbf{0}'$.*

In fact, as pointed out by the authors, with some modification, it can be proved that there is an incomplete d.r.e. degree $\mathbf{d}$ such that there is no ω -r.e. degree between $\mathbf{d}$ and $\mathbf{0}'$.

Even though d.r.e. degrees are not dense, several weak density properties are proved for d.r.e. cases. For example, Cooper [9] proved that an analogue of Sacks splitting theorem is true in d.r.e. degrees.

Theorem 13. *For any nonzero d.r.e. degree $\mathbf{d} > \mathbf{0}$, there are incomparable d.r.e. degrees $\mathbf{b}$ and $\mathbf{c}$ below $\mathbf{d}$ with $\mathbf{b} \vee \mathbf{c} = \mathbf{d}$.*

The proof involves some idea of showing that if the splitting cannot be done, then the reason of leading to such a failure is that $\mathbf{d}$ is actually r.e.. Cooper introduced the injury set for each strategy, which is r.e. Thus, when we cannot split $\mathbf{d}$ in this way, we can deduce that $\mathbf{d}$ is r.e., and we can apply Sacks splitting theorem to obtain a splitting of $\mathbf{d}$ in another way. Thus, the proof is

not uniform. Cooper and Li proved that this uniformity is necessary for the proof.

While the d.r.e. degrees are not dense, Cooper proved that locally, low_2 d.r.e. degrees are dense.

Theorem 14. *For any nonzero low_2 d.r.e. degree $\mathbf{b} < \mathbf{d}$, there is a d.r.e. degree $\mathbf{a}$ between $\mathbf{b}$ and $\mathbf{d}$.*

Cooper, Lempp and Watson [13] also proved that between any two r.e. degrees, there is a properly d.r.e. degree.

Theorem 15. *For any r.e. degrees $\mathbf{b} < \mathbf{a}$, there is a properly d.r.e. degree $\mathbf{d}$ with $\mathbf{b} < \mathbf{d} < \mathbf{a}$.*

In a recent work, Wu and Yamaleev initiated the study of the structure of d.r.e. *wtt*-degrees in [42], and proved that no maximal element exists in the d.r.e. *wtt*-degree:

Theorem 16. *Given a d.r.e. *wtt*-degree $\mathbf{d} < \mathbf{0}'$, there is a d.r.e. *wtt* degree $\mathbf{c}$ such that $\mathbf{d} < \mathbf{c} < \mathbf{0}'$.*

Again, the construction of set C with the proposed degree is a finite injury argument, with its use bounded by a recursive function.

1.5 Thesis contents

In Chapter 2, we improve Wu and Yamaleev's result by showing that the d.r.e. *wtt*-degrees are dense.

Theorem 17. *Given two d.r.e. wtt-degrees $\mathbf{c} < \mathbf{d}$, there is a d.r.e. wtt-degree strictly between $\mathbf{c}$ and $\mathbf{d}$.*

In some sense, the construction is similar to Ladner and Sasso's construction, but in our construction, we need the notion of Lachlan sets, which provide information about when a number is removed from the d.r.e. set. It is the first time to use Lachlan sets in the study of degree structure, and we believe that such an idea will have more applications to other structural properties.

From the Sacks splitting theorem and Cooper, Lempp and Watson's weak density theorem, we can deduce that any nonzero r.e. degree is the join of two properly d.r.e. degrees. However, we cannot get a set-splitting, like the one in Sacks splitting.

In Chapter 3, we give a set-splitting, and also degree-splitting of a nonrecursive r.e. set, into two properly d.r.e. sets.

Theorem 18. *Given a nonrecursive r.e. set A , there are d.r.e. sets B_0 and B_1 such that $B_0 \cap B_1 = \emptyset$, $B_0 \cup B_1 = A$ and both B_0 and B_1 have properly d.r.e. degree.*

In the construction, a few new ideas of accommodating the construction of properly d.r.e. degrees with splitting of A is provided, and we show the idea of making strategies work in a consistent way. This idea can actually be combined with the well-known Sacks splitting strategy to make the construction easy to understand.

As to contiguous degrees, Wu and Yamaleev constructed a contiguous degree which is properly d.r.e.

In Chapter 4, we generalize a result of Diamondstone [14] and construct an r.e. set which is not cuppable by any superlow d.r.e. set.

Theorem 19. *There is a promptly simple set A such that no superlow d.r.e. set can cup A to $\mathbf{0}'$.*

As a matter of fact, the existence of a promptly simple set that is not cuppable by any superlow Δ_2^0 set can be inferred from Greenberg and Nies' work. Their work observes the strongly jump traceable property of r.e. sets, and the proposed promptly simple set has this property. However, our generalization work here construct the promptly simple set in a direct way.

Theorem 17 is joint work with G. Wu and M. Yamaleev. Theorem 18 is joint work with G. Wu. Theorem 19 is joint work with G. Wu and H. Wu. Our notation follows Soare's book [36], or Downey and Hirschfeldt's book [17].

Chapter 2

Density in d.r.e. weak-truth-table degrees

In this chapter, we prove Theorem 17, which says that the d.r.e. *wtt*-degrees are dense.

2.1 Setting up

Given $\mathbf{c} < \mathbf{d}$ as required, we take d.r.e. sets C, D such that $C \in \mathbf{c}$, $D \in \mathbf{d}$, and $C \cap D = \emptyset$. Let $\{C_s : s \in \omega\}$ and $\{D_s : s \in \omega\}$ be recursive approximations of C and D , respectively, such that exactly one number enters or leaves $C \cup D$ at each stage.

In particular, we make an extra assumption that $C \oplus L(D) <_{wtt} D$, where $L(D)$ is the Lachlan set of D with respect to the approximation $\{D_s : s \in \omega\}$, i.e.,

$$L(D) = \{s : \exists x[(x \in D_s - D_{s-1}) \text{ and } x \notin D]\}.$$

It is easy to see that $L(D)$ is r.e.. Moreover if D is a d.r.e. set of a properly

d.r.e. degree then $L(D) <_{wtt} D$ (note also that if D is r.e. then $L(D)$ is empty). It is possible that $C \oplus L(D) \equiv_{wtt} D$ (even though $C <_{wtt} D$), in this case, we can choose $C \oplus L(D)$ instead. Consider the Lachlan set of $C \oplus L(D)$:

$$L(C \oplus L(D)) \equiv_{wtt} L(C) \oplus L(L(D)) \equiv_{wtt} L(C) \leq_{wtt} C <_{wtt} D \equiv_{wtt} C \oplus L(D).$$

This implies that we can always assume $C \oplus L(D) <_{wtt} D$.

To prove the theorem, we split D into two d.r.e. sets, D_0 and D_1 , such that D cannot be wtt -reducible to either $D_0 \oplus C$ or $D_1 \oplus C$. As a consequence, $D_0 \oplus C$ and $D_1 \oplus C$ are not wtt -reducible to each other, and hence above $\mathbf{c}$.

D_0 and D_1 are constructed to meet the following requirements:

$$P : D_0 \cup D_1 = D, D_0 \cap D_1 = \emptyset;$$

$$N_{e,i} : \text{If } D = \Phi_e^{D_i \oplus C} \text{ and the use function } \phi_e \text{ is bounded by } \psi_e, \text{ then } D \leq_{wtt} L(D) \oplus C.$$

Here $\{\langle \Phi_e, \psi_e \rangle : s \in \omega\}$ is an effective list of pairs $\langle \Phi_e, \psi_e \rangle$, where Φ_e is a partial recursive functional, with use function ϕ_e , and ψ_e is a partial recursive function.

2.2 Satisfying the P -requirement

In this section, we first review basic ideas of splitting techniques in various theorems, like Sacks splitting theorem for r.e. sets, and Cooper splitting theorem for d.r.e. sets, and Ladner and Sasso's splitting, for wtt -reduction. Our splitting has features of all these techniques, tailored for the wtt -reduction between d.r.e. sets.

We first have a quick look at Sacks splitting and Cooper splitting, where the reduction involved is Turing reduction. Note that Sacks splitting is a set-splitting and also a degree splitting, while Cooper splitting is just a degree-splitting, not a set-splitting. Ladner-Sasso's splitting is also a set-splitting and a wtt -degree splitting.

In Sacks splitting and Ladner-Sasso's splitting, the sets involved are r.e., and we construct two disjoint r.e. sets D_0 and D_1 whose union is D . That is, to split D , when a number x enters D , it is immediately enumerated into one of D_0 and D_1 , but not both. In this case, once a number x is enumerated into D_i , it will be kept in it forever, and because of this, we have $D \equiv_{wtt} D_0 \oplus D_1$ easily.

In Cooper splitting, the given set D is d.r.e., and we are constructing d.r.e. sets D_0 and D_1 such that $D_0 \oplus D_1 \equiv_T D$, thus D_0 and D_1 form a degree-splitting of D . Contrary to the previous two splitting settings, Cooper's splitting technique doesn't provide a set-splitting.. In this splitting, when a number x enters D , we will put a number y_j , not x itself, chosen from a trace-block of x into one of D_0 or D_1 , D_0 say (y_j is called the current trace of x). We may later remove y_j out of D_0 (due to the injury from a strategy with higher priority), and we put another trace y_{j+1} (also from the trace-block of x) into D_1 , for the coding of $D(x)$ in $D_0 \oplus D_1$. This kind of changes of tracing numbers (and swapping from one side to the other) can happen at most finitely often, as we know in advance the number of strategies with higher priority, thus we know how to set the size of the trace block. When this x leaves D , we will remove the

current trace of x out of $D_0 \cup D_1$ immediately. Due to such kind of removing we cannot directly restrain D_0 and D_1 , however it allows us to relocate traces of other elements to avoid injuries of lower priority strategies. Moreover if such situation happens infinitely often disrespecting restraints of some strategy then this strategy wins through a divergence outcome.

In our construction, we are dealing with the *wtt*-reduction, and the construction will be similar to Ladner-Sasso's construction. We will not have the divergence outcome as in Cooper splitting, and for x , we do not need to have a trace-block — once x enters D at stage s , we enumerate x into one of D_0 and D_1 (not both) at this stage, and when x leaves D , we just remove it from $D_0 \cup D_1$. The crucial point here is that when x leaves D , s , the stage when x enters D , enters $L(D)$, the Lachlan set of D , and this enumeration into $L(D)$ keeps the consistency between N -strategies. We will give more details in the next section.

2.3 Satisfying one N -requirement

The core of Sacks splitting is the technique of Sacks preservation, where the main point is to preserve the agreement between D and $\Phi_e^{D_i}$, to force a contradiction, as D is nonrecursive. This technique was developed by Robinson [35] to show that r.e. sets split above low r.e. sets. Lachlan's nonsplitting theorem says that Sacks splitting cannot be combined with Sacks density, and the main obstacle is that one strategy can impose restraints with limit infinite. Ladner-Sasso's splitting theorem says that splitting and density can be combined for

r.e. wtt -degrees. That is, $D \not\equiv_{wtt} D_i \oplus C$ can be guaranteed by applying Sacks preservation, via threatening $D \not\leq_{wtt} C$. In particular, to make $D \neq \Phi_e^{D_i \oplus C}$, where Φ_e is a wtt -reduction, we will construct a wtt -reduction $\Delta_{e,i}$ such that once we see D and $\Phi_e^{D_i \oplus C}$ agree on a certain length ℓ , we will preserve the D_i -side, and define $\Delta_{e,i}^C$ up to ℓ . Thus, for $n \leq \ell$, either $\Phi_e^{D_i \oplus C}(n)$ is preserved, and $\Delta_{e,i}^C(n)$ is kept, or C changes below the use $\varphi_e(n)$, and this change undefines $\Delta_{e,i}^C(n)$. As for a fixed n , $\Phi_e^{D_i \oplus C}(n)$ converges at most finitely many times, and each change of the computation $\Phi_e^{D_i \oplus C}(n)$ is caused by a C -change, $\Delta_{e,i}^C(n)$ can be defined at most finitely many times, and each time when it is defined, we define it as $D(n)$, so if $\Delta_{e,i}^C(n) \neq D(n)$, we will have $D(n) \neq \Phi_e^{D_i \oplus C}(n)$.

We are now ready to describe how to satisfy one N -requirement, like $N_{e,i}$, where C and D are d.r.e. sets. At a stage s , we define the length of agreement of D and $\Phi_e^{D_i \oplus C}$ as:

$$\ell(e, i, s) = \max\{x < s : \forall y < x [\Phi_e^{D_i, s \oplus C}(y)[s] \downarrow = D_s(y)]\},$$

and define the restriction function as:

$$r(e, i, s) = \max\{\phi(x) : x \leq \ell(e, i, s)\}.$$

As usual, we say that a stage s is expansionary for some $N_{e,i}$ if $\ell(e, i, s) > \ell(e, i, t)$ for all $t < s$. As in Sacks preservation strategy, if $x \leq r(e, i, s)$ enters D at this stage, then we enumerate x into D_{1-i} , and we are expecting to preserve the length of agreement $\ell(e, i, s)$ by ensuring that no enumerations into the D_i -part can change the computations, like the splitting of Ladner-Sasso. However, in our construction, both C and D are d.r.e. sets, n can enter C or D first and

then leave at a later stage. Due to this, we cannot guarantee a disagreement between $D(y)$ and $\Phi_e^{D_i \oplus C}(y)$ if y enters D at a stage s with

$$D(y)[s] = 1 \neq 0 = \Phi_e^{D_i \oplus C}(y)[s],$$

as when y leaves D at a stage $s' > s$, we will come back to

$$D(y)[s'] = 0 = \Phi_e^{D_i \oplus C}(y)[s].$$

Therefore, for x between y and $\ell(e, i, s)$, we also need to protect $\Phi_e^{D_i \oplus C}(x)$ by preventing a new element from being enumerated into D_i (We definitely cannot prevent the removal of elements from D_i), and because of this, we will keep the restraint we had at stage s forever, even when we see y enters D at stage s . This is a feature different from those in Sacks splitting and Ladner-Sasso splitting.

We comment here that the original Sacks splitting and Ladner-Sasso splitting still work if we do the preservation in this way, as the restraints imposed in this way will be finite in the whole construction. However, it cannot be done in Cooper splitting, as otherwise, one strategy will impose restraints tending to infinity.

In our construction, this idea is crucial, as we are expecting that all changes caused by enumerations into C or D may change back later, as noted and used in Wu and Yamaleev's paper [42]. We will refer to this idea "once forever". That is, once a restraint $r(e, i, s)$ is set at a stage s , no number less than $r(e, i, s)$ can be put in D_i at a later stage (unless this strategy was initialized), and we will show that, by the assumption that $L(D) \oplus C <_{wtt} D$, this strategy will impose only finitely many restraints in the whole construction.

Recall that we are constructing a partial recursive functional $\Delta_{e,i}$, such that if there are infinitely many expansionary stages between D and $\Phi_e^{D_i \oplus C}$, where the use function ϕ_e is bounded by ψ_e , then $\Delta_{e,i}^{L(D) \oplus C}$ will be defined infinitely often, and at any expansionary stage, $\Delta_{e,i}^{L(D) \oplus C}$ computes D correctly at arguments in the current domain.

As above, for $n < \ell(e, i, s)$, if $\Delta_{e,i}^{L(D) \oplus C}(n)[s]$ is not defined, we define

$$\Delta_{e,i}^{L(D) \oplus C}(n)[s] = D_s(n),$$

with the C -part of the use as $\psi_e(n)$, and the $L(D)$ -part of the use as the latest stage $t < s$ at which some number $y \in D_i[s]$ less than $\psi_e(n)$ enters D_i . Once we define it, the C -part and the $L(D)$ -part of the use of Δ will not be changed in the remainder of the construction, unless the strategy is initialized. Due to this, the reductions are weak-truth-table reductions.

Note that $L(D)$ is r.e., once a number t is enumerated into $L(D)$, t will be in $L(D)$ forever, which means that $\Delta_{e,i}^{L(D) \oplus C}(n)$ never comes back to $\Delta_{e,i}^{L(D) \oplus C}(n)[s]$, and when it is redefined later, t is already in $L(D)$. This just says that if the computation $\Phi_e^{D_i \oplus C}(n)$ changes due to some number less than $\psi_e(n)$ leaves D_i (and hence leaves D), then $\Delta_{e,i}^{L(D) \oplus C}(n)$ will get undefined, and the new definition will be different from the one at stage s . Also note that when $\Delta_{e,i}^{L(D) \oplus C}(n)$ is defined at stage s , a restraint is put on to prevent the enumeration of any number less than $\psi_e(n)$ into D_i . By this restraint, $\Delta_{e,i}^{L(D) \oplus C}(n)$ does not recover to a previous definition. Thus, $\Delta_{e,i}^{L(D) \oplus C}(n)$ can be undefined in this way by the change of D_i at most $\psi_e(n)$ many times, and as no number less than $\psi_e(n)$ can be enumerated into D_i (by the “once forever” rule), once it is defined again at

stage $s' > s$, $D_i[s'] \subseteq D_i[s]$, and hence we can still keep the use of the $L(D)$ -part the same as before.

Now consider the C -part of the use. By the discussion above, we only need to consider definitions of $\Delta_{e,i}^{L(D) \oplus C}(n)$ between two $L(D)$ -changes, if any. In this case, $\Delta_{e,i}^{L(D) \oplus C}(n)$ changes due to C -changes. In particular, these changes are below $\psi_e(n)$.

There are two cases.

Case 1. C changes and the computation $\Phi_e^{D_i \oplus C}(n)$ does not recover to a previous one.

Then $\Delta_{e,i}^{L(D) \oplus C}(n)$ is redefined again at a later stage.

Case 2. C -changes and $\Phi_e^{D_i \oplus C}(n)$ is recovered to a previous one.

In this case, $\Delta_{e,i}^{L(D) \oplus C}(n)$ is also recovered to a previous definition.

In both cases, if $D(n) = \Phi_e^{D_i \oplus C}(n)$, then we can have $D(n) = \Delta_{e,i}^{L(D) \oplus C}(n)$, either by redefining $\Delta_{e,i}^{L(D) \oplus C}(n)$ and letting it be $D(n)$, or because of $D_{s'}(n) = \Phi_e^{D_i \oplus C}(n)[s'] = \Phi_e^{D_i \oplus C}(n)[s] = D_s(n)$, and

$$\Delta_{e,i}^{L(D) \oplus C}(n)[s'] = \Delta_{e,i}^{L(D) \oplus C}(n)[s] = \Phi_e^{D_i \oplus C}(n)[s] = D_s(n) = D_{s'}(n).$$

Therefore, once $\Delta_{e,i}^{L(D) \oplus C}(n)$ is defined, then the use will be kept the same, and if for all n , $D(n) = \Phi_e^{D_i \oplus C}(n)$, then $\Delta_{e,i}^{L(D) \oplus C}(n)$ is defined, where the $L(D)$ -part and the C -part of the use are fixed, and computes $D(n)$ correctly. This shows that D is *wtt*-reducible to $L(D) \oplus C$, which is impossible, by our

assumption. Therefore, in the construction, after a certain stage, we will not have more agreement between D and $\Phi_e^{D_i \oplus C}$, and the requirement $N_{e,i}$ will be satisfied.

2.4 Construction

We are ready to construct a splitting D_0, D_1 , of D satisfying the requirements. Also we assume that C has changes at odd stages and D has changes at even stages.

We list all requirements as follows:

$$P < N_{0,0} < N_{0,1} < N_{1,0} < N_{1,1} < \dots < N_{e,0} < N_{e,1} < \dots .$$

P is a global requirement and has the highest priority. We will take care of it at every stage. For $N_{e_1,j_1} < N_{e_2,j_2}$, N_{e_1,j_1} has higher priority. Say that a requirement $N_{e,j}$ *requires attention* at a stage s , if the change of $C \cup D$ at number x at stage s changes the computation in $N_{e,j}$, or s is an expansionary stage for $N_{e,j}$.

Stage $s = 0$. Define $D_0 = D_1 = \emptyset$, undefine all constructed functionals, go to Stage 1.

Stage $s + 1$.

Step 1: Check whether stage $s + 1$ is an expansionary stage for some requirement among $N_{0,0}, N_{0,1}, \dots, N_{s,0}, N_{s,1}$.

If Yes, let $N_{e,i}$ be the requirement with the highest priority among them. Update the definition of $\Delta_{e,i}^{L(D) \oplus C}$ with the C -part of the use of $\delta_{e,i}(n)$ defined

as $\psi_e(n)$ and the $L(D)$ -part of the use $\delta_{e,i}$ defined as the last stage $t < s + 1$ such that some number $y < \psi_e(n)$ enters D_i and still in D_i . Initialize all the requirements with lower priority and proceed to step 2.

If No, do nothing and proceed to step 2.

Step 2: Suppose that x enters or leaves $C \cup D$ at stage $s + 1$.

s odd: If x enters D at this stage, then find the requirement with the highest priority, $N_{e,i}$ say, whose restraint is bigger than x , and enumerate x into D_{1-i} .

If x leaves D at this stage, then remove x from $D_0 \cup D_1$. Let $N_{e,i}$ be the requirement with the highest priority that is affected by the removal of D .

In both cases, initialize all the requirements with priority lower than $N_{e,i}$.

s even: If x enters or leaves C at this stage, then find the requirement with the highest priority, $N_{e,i}$ say, whose definition of $\Delta_{e,i}^{D_i \oplus C}(n)$ is affected by the change of C at x , if any, and initialize all the requirements with priority lower than $N_{e,i}$. If there is no such a requirement, do nothing. In both cases, go to the next stage.

□

This completes the construction of D_0 and D_1 .

2.5 Verification

Obviously, the constructed D_0 and D_1 form a splitting of D . In this section, we verify that D_0 and D_1 satisfies all the N -requirements. Thus, the wtt -degrees of $D_0 \oplus C$ and $D_1 \oplus C$ form a wtt -degree splitting of $\deg_{wtt}(D)$ above $\deg_{wtt}(C)$.

Lemma 1. *For each e, i ,*

- (1) $N_{e,i}$ can be initialized at most finitely many times;
- (2) $N_{e,i}$ can be affected by the approximations of C and D at most finitely many times and consequently, $D \neq \Phi_e^{D_i \oplus C}$, so that $N_{e,i}$ is satisfied;
- (3) $r(e, i) = \lim_{s \rightarrow \infty} r(e, i, s)$ exists and is finite, and $N_{e,i}$ acts and initializes requirements with lower priority at most finitely many times.

Proof. We prove by induction on $\langle e, i \rangle$. Fix $\langle e, i \rangle$ and assume (1), (2), (3) hold for all $\langle k, j \rangle < \langle e, i \rangle$.

According to the construction, $N_{e,i}$ is initialized only by strategies of higher priority, i.e., $N_{k,j}$, with $\langle k, j \rangle < \langle e, i \rangle$. By the induction hypothesis (3), each such $N_{k,j}$ initializes $N_{e,i}$ at most finitely many times, and hence, $N_{e,i}$ is initialized finitely many times. (1) is true for $N_{e,i}$.

Let t be the least stage such that $N_{\langle e, i \rangle}$ cannot be initialized after this.

Assume for contradiction that $N_{e,i}$ is affected by approximations of C and D infinitely many times. As for a fixed n , for computation $\Phi_e^{D_i \oplus C}(n)$, the use $\phi_e(n)$ is bounded by $\psi_e(n)$, the effect of approximations of C and D on it is

finite. So our assumption implies that $D = \Phi_e^{D_i \oplus C}$. We show below that under this assumption, D is *wtt*-reducible to $L(D_i) \oplus C$, via $\Delta_{e,i}$ defined by us.

Fix p , and assume that after stage $t' > t$, further approximations of C and D do not affect computations $\Phi_e^{D_i \oplus C}(q)$, and also $\Delta_{e,i}^{L(D_i) \oplus C}(q)$, where $q < p$, and $\Delta_{e,i}^{L(D) \oplus C}(q) = D(q)$. By our assumption $D = \Phi_e^{D_i \oplus C}$, there is a stage $t'' > t$ such that further approximations of C and D do not affect the computation $\Phi_e^{D_i \oplus C}(p)$, and $\Phi_e^{D_i \oplus C}(p) = \Phi_e^{D_i \oplus C}(p)[t'] = D_{t'}(p) = D(p)$.

We now show that at a stage $s > t$, if we are defining $\Delta_{e,i}^{L(D) \oplus C}(p)$ at this stage for the first time, we have $\Phi_e^{D_i \oplus C}(p)[s] = D_s(p)$, and we define $\Delta_{e,i}^{L(D) \oplus C}(p)[s] = D_s(p)$, with the C -part use as $\psi_e(p)$ and the $L(D)$ -part use as the biggest stage at which some number less than $\psi_e(p)$ in $D_{i,s}$ enters D . This use will be fixed since this stage, and because of this, $\Delta_{e,i}$ is a *wtt*-reduction. That is, if at a stage $s' > s$, $\Delta_{e,i}^{L(D) \oplus C}(p)$ is undefined, then we make the definition, and keep the use unchanged. It is obvious for the C -part use, as $\psi_e(p)$ is fixed, and the C -part use in the computation of $\Phi_e^{D_i \oplus C}(p)$ is below $\psi_e(p)$. For the $L(D)$ -part, note that from stage s onwards, no number less than $\psi_e(p)$ is allowed to be enumerated into D_i , and hence for any computation $\Phi_e^{D_i \oplus C}(p)$ we see after stage s , numbers in D_i involved in the computation are all in $D_{i,s}$, and hence when we redefine $\Delta_{e,i}^{L(D) \oplus C}(p)$, the $L(D)$ -part use is always less than the one we defined at stage s . Thus, $\Delta_{e,i}^{L(D) \oplus C}(p)$ is defined, with use fixed since stage s .

Now we show that from stage s , if $\Delta_{e,i}^{L(D) \oplus C}(p)$ has definition at stage s' , or is defined at this stage, then $\Phi_e^{D_i \oplus C}(p)[s'] = D_{s'}(p)$, we have $\Delta_{e,i}^{L(D) \oplus C}(p)[s'] = D_{s'}(p)$. It is true because for any stage s'' with $s \leq s'' < s'$, if computations

$\Phi_e^{D_i \oplus C}(p)[s']$ and $\Phi_e^{D_i \oplus C}(p)[s'']$ are different, then either C changes below $\psi_e(p)$ or some numbers less than $\psi_e(p)$ leave D_i , and in the latter case, $L(D)$ changes below the use. In both cases, $\Delta_{e,i}^{L(D) \oplus C}(p)[s'] = D_{s'}(p)$.

This shows that for each p , $\Delta_{e,i}^{L(D) \oplus C}(p)$ is defined, $\Delta_{e,i}^{L(D) \oplus C} = D$ and the reduction $\Delta_{e,i}$ is a *wtt*-reduction, contradicting our assumption that $D >_{wtt} L(D) \oplus C$.

Therefore, $N_{e,i}$ can be affected by the approximations of C and D at most finitely many times, and we have $D \neq \Phi_e^{D_i \oplus C}$. (2) holds for $N_{e,i}$. As a consequence, $\Delta_{e,i}^{L(D) \oplus C}$ can have the definition extended at most finitely many times.

For (3), note that $N_{e,i}$ sets a restraint at a stage s only when $\Delta_{e,i}^{L(D) \oplus C}(n)$ is defined at this stage for some n , or approximations of C and D affect computations involved. By (2), we know that such actions can happen at most finitely many times. Hence we can set restraint for $N_{e,i}$ finitely many times, and hence has a finite limit. $N_{e,i}$ initializes strategies with lower priority only when $N_{e,i}$ is affected during the construction, and can take only finitely many such initializations during the whole construction. (3) holds for $N_{e,i}$.

This completes the proof of Lemma 1. □

Chapter 3

Set-splittings of r.e. sets

This chapter is devoted to the proof of Theorem 18. Note that a properly d.r.e. degree splitting of a nonzero r.e. degree follows directly from Sacks splitting theorem and the weak density of properly d.r.e. degrees among r.e. degrees. The splitting of A in Theorem 18 is not just a degree splitting, also a set splitting. The construction of such a splitting involves a few new ideas, one of which is to show how we can swap a number from one-side, B_0 , to the other side B_1 . This kind of idea does not appear in the literature, and we will see this soon, after a briefing of construction of properly d.r.e. sets.

3.1 Construction of a properly d.r.e. set

In this section, we give a brief introduction of the construction of a properly d.r.e. set B . This was originally given by Cooper.

To make B properly d.r.e., we need to make B not Turing equivalent to any

r.e. set. That is, B will be constructed to satisfy the following requirements:

$$N_i : B \neq \Phi_i^{W_i} \vee W_i \neq \Psi_i^B,$$

where $\langle W_i, \Phi_i, \Psi_i \rangle_{i \in \omega}$ is an effective list of $\langle W, \Phi, \Psi \rangle$, with W an r.e. set, Φ and Ψ partial recursive functionals.

Here is the basic idea of satisfying one single requirement N_i . We omit the subscript i for simplicity.

1. Choose a number x as a witness, and wait for

$$\Phi^W(x) = 0 \quad \text{and} \quad W \upharpoonright (\phi(x) + 1) = \Psi^B \upharpoonright (\phi(x) + 1);$$

2. We see at stage s_1 the agreement between Φ^W and B bigger than x and the agreement between W and Ψ^B bigger than $\phi(x)[s_1]$. Then at this stage, enumerate x into B , restrain $B \upharpoonright (\psi \circ \phi(x)[s_1] + 1)$;

3. Wait for

$$\Phi^W(x) = 1 \quad \text{and} \quad W \upharpoonright (\phi(x) + 1) = \Psi^B \upharpoonright (\phi(x) + 1);$$

4. We see at stage s_2 the agreement between Φ^W and B bigger than x and the agreement between W and Ψ^B bigger than $\phi(x)[s_2]$.

Then at this stage, remove x from B , restrain $B \upharpoonright (\psi \circ \phi(x)[s_2] + 1)$.

□

We have these outcomes:

- (1) Keep waiting at step 1. Then, $B(x) = 0$, and either $\Phi^W(x) \neq 0$, and we have $\Phi^W(x) \neq B(x)$, or $\Phi^W(x) = 0$, and W and Ψ^B do not agree on $\phi(x)$. N_i is satisfied.
- (2) Keep waiting at step 3. Then $B(x) = 1$, and either $\Phi^W(x) \neq 1$, and we have $\Phi^W(x) \neq B(x)$, or $\Phi^W(x) = 1$, and W and Ψ^B do not agree on $\phi(x)$. N_i is satisfied.
- (3) We reach step 4, and x is removed from B . As $B \upharpoonright (\psi \circ \phi(x)[s_1] + 1)$ has been restored, $\Psi^B(\phi(x))$ returns to 0, we then have W and Ψ^B do not agree on this point, so N_i is satisfied.

3.2 Construction of B_0 and B_1

Let A be a nonrecursive r.e. set, with recursive approximation $\{A_s : s \in \omega\}$. We assume $A_0 = \emptyset$, $|A_{s+1} \setminus A_s| = 1$ at each stage s . B_0 and B_1 are constructed to meet the following requirements:

$$P: B_0 \cup B_1 = A, B_0 \cap B_1 = \emptyset;$$

$$Q: A \equiv_T B_0 \oplus B_1;$$

$$N_{i,0} : B_0 \neq \Phi_i^{W_i} \vee W_i \neq \Psi_i^{B_0};$$

$$N_{i,1} : B_1 \neq \Phi_i^{W_i} \vee W_i \neq \Psi_i^{B_1}.$$

As above, $\langle W_i, \Phi_i, \Psi_i \rangle_{i \in \omega}$ is an effective list of triples $\langle W, \Phi, \Psi \rangle$, W an r.e. set, Φ and Ψ partial recursive functionals.

To satisfy the P requirement, we need to make sure that each number in A will appear in B_0 or B_1 , but not both. This ensures that $A \leq_T B_0 \oplus B_1$. For $B_0 \leq_T A$, and a number x , to decide whether x is in B_0 or not, we first check whether x is in A . If x is not in A , then x is not in B_0 . If x is in A , we find a stage s at which x enters A , and then find a stage $t \geq s$ such that $A \upharpoonright x = A_t \upharpoonright x$. Check whether x is in B_0 by the end of stage t . Then $B_0(x) = B_{0,t}(x)$. This gives $B_0 \leq_T A$. The same idea applies to show that $B_1 \leq_T A$.

To satisfy an $N_{i,j}$ requirement, we need to modify the basic strategy given above to make it compatible with the splitting of A . Here we cannot choose witness x arbitrarily. As x will be enumerated into B , x has to enter A first. Instead of choosing a witness at the beginning of the strategy, we wait till a number x enters A , and x can be used as a witness in the following sense: x is less than ℓ , the length of agreement between B_j and $\Phi_i^{W_i}$, and $\Psi_i^{B_j}$ and W_i agree on $[0, \phi_i(x) + 1]$. As A is nonrecursive, such an x should appear at a certain stage, and then the strategy takes this x and continue.

Another issue is the consistency between N strategies. That is, in our construction, it is possible that to make B_j properly d.r.e., we need to move witness x out of B_j , and, as x is in A , x should be enumerated into B_{1-j} . This action can injure those strategies with lower priority. However, we cannot allow such an action to injure those strategies with higher priorities. With this in mind, the strategy $N_{i,j}$ sets a threshold $k_{i,j}$ first, and whenever some N -strategy with higher priority acts, $N_{i,j}$ gets initialized, and we will change threshold $k_{i,j}$ to a even bigger number. As those N strategies with higher priority can act

at most finitely often, $k_{i,j}$ can be increased to bigger numbers at most finitely often.

Now suppose that $k_{i,j}$ is defined at stage s . Then after stage s , whenever A changes below $k_{i,j}$, we reset the $N_{i,j}$ strategy by restart it, except that we keep $k_{i,j}$ the same. Again, for a fixed $k_{i,j}$, such process of resetting can happen at most finitely times.

Thus, we can assume that after a stage t big enough, the strategy $N_{i,j}$ cannot be initialized and cannot be reset at a later stage. We argue now that $N_{i,j}$ -strategy does not injure those strategies with higher priority, because the number x we use for $N_{i,j}$ is bigger than $k_{i,j}$, and hence is bigger than the uses in all of the computations involved in the strategies with higher priority. Therefore, the enumeration of x into B_j or B_{1-j} does not change the computations of those strategies with higher priority, and the concern we had is solved.

Below is a modified strategy to meet the requirement $N_{i,j}$:

1 Choose the threshold k as a big number.

Reset this strategy if any number less than k enters A in the remainder of the construction.

2 At the next expansionary stage s , we have the length of agreement ℓ between B_j and $\Phi_i^{W_i}$, and the corresponding agreement between $\Psi_i^{B_j}$ and W_i , we extend the definition of f up to ℓ with $f(n) = A_s(n)$ for all $n < \ell$. Set up a restraint to preserve those computations.

3 Wait for some $x < \ell$ entering A .

- 4 Enumerate x into B_j .
- 5 Wait for the length of agreement between B_j and $\Phi_i^{W_i}$ to be greater than x , and the corresponding agreement between $\Psi_i^{B_j}$ and W_i .
- 6 At stage t , we see such length of agreement, then we say that the $N_{i,j}$ strategy is waiting for a change of A below x to remove x out of B_j .
- 7 At stage $t' > t$, A changes below x , at y , say. Then we remove x from B_j and put it into B_{1-j} .

(Note that in this case, y is also put into B_{1-j} .)

□

The discussion above actually describes only one cycle. What we carry on in the formal constructions is in fact multiple cycles for $N_{i,j}$. Once we come to step 6 and begin to wait for a permission from A , we keep the witness in the current cycle and start a new cycle (go to step 2). When permission from A comes (below one of the witnesses), we may have kept several witnesses. In this case, we can take step 7 for one witness, which suffices to complete the diagonalization. It is easy to see, we can just keep the largest witness among all cycles while waiting for a permission from A .

The outcomes of this strategy are:

- (1) The strategy ends at step 2 from some point on, because there is no further agreement between B_j and $\Phi_i^{W_i}$ or between W_i and $\Psi_i^{B_j}$ on $\phi_i(\ell)$.

Then $B_j(\ell) = 0$, and either $\Phi_i^{W_i}(\ell) \neq 0$, and we have $\Phi_i^{W_i}(\ell) \neq B_j(\ell)$, or $\Phi_i^{W_i}(\ell) = 0$, and W_i and $\Psi_i^{B_j}$ do not agree on $\phi_i(\ell)$. Thus $N_{i,j}$ is satisfied.

- (2) There are infinitely many stages at which step 2 is reached and the strategy always waits at step 3. If so, then A is recursive, which cannot happen.
- (3) There is some x such that the strategy waits at step 5 forever. Then, $B_j(x) = 1$, and either $\Phi_i^{W_i}(x) \neq 1$, and we have $\Phi_i^{W_i}(x) \neq B_j(x)$, or $\Phi_i^{W_i}(x) = 1$, and W_i and $\Psi_i^{B_j}$ do not agree on $\phi_i(x)$. Thus $N_{i,j}$ is satisfied.
- (4) There are infinitely many x being enumerated into B_i at step 5, and the strategy keeps waiting at step 6 for all these numbers. Again, if so, A would be recursive, which cannot be true.
- (5) We reach step 7, and some x is removed from B_j . Then we have W_i and $\Psi_i^{B_j}$ do not agree on $\phi_i(x)$, and $N_{i,j}$ is satisfied.

Thus, only (1), (3) (5) can be outcomes of the $N_{i,j}$ strategy.

3.3 Construction

We are ready to construct a splitting B_0 and B_1 of A satisfying all requirements. The priority of these requirements is given as:

$$P, Q < N_{0,0} < N_{0,1} < \cdots < N_{i,0} < N_{i,1} < \cdots .$$

P and Q are global requirements and we take care of them at all stages in the construction. For N -requirements $N_{i_1,j_1} < N_{i_2,j_2}$, N_{i_1,j_1} has higher priority. For

each $N_{i,j}$, we let $x(i,j)$ be the witness chosen for it, $k(i,j)$ be its threshold, $\ell(i,j)$ be the length of agreement between B_j and $\Phi_i^{W_i}$ and W_i and $\Psi_i^{B_j}$ agree on these ϕ_i -uses, and $r(i,j)$ be the restraint set to preserve computations.

In the construction, we say an N strategy is in Status k , $k = 0, 1, 2, 3$, to indicate which step the strategy is currently at. Status 0 means that the strategy is waiting for an expansionary stage, but not for creating a disagreement (step 1 executed), Status 1 means that the strategy is waiting for an expansionary stage to create a disagreement (step 4 executed), Status 2 means the strategy is waiting for an A -change under some kept witness to complete the diagonalization (step 6 executed), and Status 3 means that the strategy has obtained a disagreement (step 7 executed).

Construction:

Stage 0: Define $B_0 = B_1 = \emptyset$, go to Stage 1.

Stage $s + 1$: Let a_s be the number entering A at stage $s + 1$. Reset all N -strategies with their threshold bigger than a_s . *Note that when a strategy is reset, all strategies with lower priority are initialized.*

(I) Find the $N_{i,j}$ -strategy with the highest priority such that the computation in $N_{i,j}$ -strategy is changed due to the enumeration of a_s .

1. If $N_{i,j}$ is currently in Status 0 or 2, the enumeration of a_s allows the $N_{i,j}$ -strategy to reach step 3, then enumerate a_s into B_j , completing step 3. We let $x(i,j)$ to denote this a_s . The $N_{i,j}$ -strategy switches to Status 1.

2. If the enumeration of a_s allows the $N_{i,j}$ -strategy to reach step 7 (currently, the strategy is in Status 2). Suppose the corresponding follower in $N_{i,j}$ is $z(i, j)$. Remove $z(i, j)$ from B_j and enumerate both $z(i, j)$ and a_s into B_{1-j} . Switch the strategy to Status 3.

If there is no such $N_{i,j}$ -strategy being affected, enumerate a_s into B_0 .

(II) Find the $N_{i,j}$ -strategy with the highest priority such that one of the following happens, and implement acts accordingly:

1. $k_{i,j}$ is not defined at stage $s + 1$; then define it as a big number. Switch the $N_{i,j}$ -strategy to Status 0.
2. $N_{i,j}$ is in Status 0 or 2, and has a bigger length of agreement $\ell(i, j)$ between B_j and $\Phi_i^{W_i}$, and also the corresponding agreement between W_i and $\Phi_i^{B_j}$. Then extend the definition of $f_{i,j}$ up to $\ell(i, j)$ with $f_{i,j}(n) = A_{s+1}(n)$ for those n with $f_{i,j}(n)$ not defined yet. *Thus, the $N_{i,j}$ -strategy is waiting for A to change below $\ell(i, j)$.*
3. $N_{i,j}$ is in Status 1, and the length of agreement $\ell(i, j) > x(i, j)$, then replace $z(i, j)$ with the current $x(i, j)$ if $z(i, j) < x(i, j)$ and let $x(i, j)$ be undefined. The strategy is waiting for A to change below $z(i, j)$. Switch it to Status 2.

□

This completes the construction of B_0 and B_1 .

3.4 Verification

We now verify that the constructed B_0 and B_1 satisfy all the requirements.

In the construction, when a number x enters A , we enumerate it into B_0 or B_1 , but not both; and when we remove x from B_j , we put it into B_{1-j} . After this removal, we will not change x back to B_j . Thus A is the disjoint union of B_0 and B_1 , and hence $A \leq_T B_0 \oplus B_1$. Also note that when we remove x from B_j and enumerate it into B_{1-j} at stage s , we also have $a_s < x$. This gives $B_j, B_{1-j} \leq_T A$. Thus, both requirements P and Q are satisfied.

Lemma 2. *For each $i \in \omega$, $j \in \{0, 1\}$,*

- (1) $N_{i,j}$ is initialized finitely many times;
- (2) $N_{i,j}$ can be reset only finitely many times;
- (3) $N_{i,j}$ is met and can initialize other strategies with lower priority only finitely many times.

Proof. We prove by induction on $\langle i, j \rangle$.

(1) $N_{i,j}$ is initialized once at the beginning of the strategy, and by those strategies with higher priority during the construction. As there are finitely many N requirements with higher priority, and by induction hypothesis, (3) for these strategies are true, $N_{i,j}$ can be initialized at most finitely many times.

(2) Suppose s_0 is the last stage at which $N_{i,j}$ is initialized. Then at the next stage, $N_{i,j}$ chooses its threshold $k(i, j)$ as a big number. After this, A can

change below $k(i, j) + 1$ many times, i.e., $N_{i,j}$ can be reset at most $k(i, j) + 1$ many times. (2) is true for $N_{i,j}$.

(3) If $N_{i,j}$ stays at Status 0 or 2 almost all time, we assert $N_{i,j}$ can have at most finitely many expansionary stages, as $f_{i,j}$ is extended nontrivially at each expansionary stage. If on the contrary, $N_{i,j}$ has infinitely expansionary stages, then $f_{i,j}$ is totally defined and computes A correctly (modulo finitely many), so that A is recursive, which cannot be true. This shows that the expansionary stages of $N_{i,j}$ can be finitely many at most, and we can conclude as in the introduction part that $N_{i,j}$ is met, since either B_j and $\Phi_i^{W_i}$, or W_i and $\Psi_i^{B_j}$, do not agree with each other.

$N_{i,j}$ cannot switch from Status 1 to Status 2 infinitely often, as if so, $N_{i,j}$ requests for A -permission infinitely many times, and gets no such permission (otherwise, it will come to Status 3). This cannot be true, as again, it will show that A is computable.

Notice that Status 1 is first switched from Status 0, however can only be from Status 2 later. This also implies that $N_{i,j}$ cannot switch from Status 2 to Status 1 infinitely often. Otherwise it is going to be an infinite loop switching between Status 1 and 2.

Suppose $N_{i,j}$ stays at Status 1 forever from some stage on. In this case, we have $B_j(x(i, j)) = 1$, and either $\Phi_i^{W_i}(x(i, j)) \neq 1$, or W_i and $\Psi_i^{B_j}$ do not agree with each other on $\phi_i(x(i, j))$. $N_{i,j}$ is met.

If Status 3 is reached, then the diagonalization is realized, as we have W_i and $\Psi_i^{B_j}$ do not agree with each other on the use $\phi_i(z(i, j))$. $N_{i,j}$ is met.

Thus, $N_{i,j}$ is met in all cases.

Note that $N_{i,j}$ only initializes those strategies with lower priority when it proceeds to a further step in the construction. As discussed above, eventually, $N_{i,j}$ either stays at Status 0 or 2, with no further expansionary stages; or waits forever at Status 1 for the agreement to be beyond $x(i, j)$; or reaches Status 3, and again, after this, $N_{i,j}$ stops to proceed to a further step. This shows that $N_{i,j}$ cannot initialize strategies with lower priority after some certain stage, and (3) is true for $N_{i,j}$.

This completes the proof of the lemma. □

Chapter 4

A generalization of a theorem of Diamondstone

In this chapter, we generalize a theorem of Diamondstone about the failure of cupping a promptly simple set via superlow sets, from r.e. to d.r.e..

The observation of low cuppability of promptly simple sets is covered in the work of Ambos-Spies, Jockusch, Soare, and Shore. They showed a series of equivalences on r.e. sets with various properties. E.g., an r.e. set has the low cuppable property if and only if it is Turing equivalent with a promptly simple set. Here an r.e. set A is promptly simple if it has an enumeration $\{A_s\}_{s \in \omega}$, and there is a recursive function p , such that for all $i \in \omega$, if W_i is infinite, then there is some $x \in W_i \cap A$, such that x is enumerated into W_i at some stage s , and into A by stage $p(s)$, with $p(s) \geq s$.

One intention on the condition is to change from low sets ($A' \leq_T \emptyset'$) to superlow sets ($A' \leq_{tt} \emptyset'$). In this stronger reduction, it is required that not only one can approximate the jump set of A , but also bound the number of

mistakes in the approximation process. Diamondstone proved that the equivalence of promptly simple degree and low cuppable degree fails when we try to strengthen low cuppable to superlow cuppable degree. In fact, Diamondstone has constructed a promptly simple set which cannot be cupped to $\mathbf{0}'$ by any superlow r.e. set.

Diamondstone's result can be implied from the work of Greenberg and Nies [23]. In fact, Greenberg and Nies' work gives a further implication: there is a promptly simple r.e. set which is not superlow cuppable by any superlow Δ_2^0 set. However, this implication involves discussion on the strongly jump traceable property. The proposed promptly simple sets have this property.

Our work aims to extend Diamondstone's work in a direct way. In this chapter we will construct a promptly simple set A such that for any superlow d.r.e. set D , $\emptyset' \not\leq_T A \oplus D$. The idea presented here can be applied to the ω -r.e. case.

In the following, we start with the introduction of the previous work done by Diamondstone. We then analyze the consequence if we replace r.e. with d.r.e. in the construction. A revised construction that works for d.r.e. case will be presented after that.

4.1 Setting up

Let A be the promptly simple set to construct, D be any d.r.e. set. Without loss of generality, we may treat A as a subset of odd numbers, and D among subsets of even numbers. With this assumption, $A \oplus D \equiv_T A \cup D$. Thus we

can use $A \cup D$ instead in the following.

On the issue of superlowness, we use an equivalent condition of truth-table reduction: $X \leq_{tt} \emptyset'$ if and only if X is ω -r.e., which says that X has a computable approximation whose number of changes is bounded by a recursive function, i.e., there is a recursive $0-1$ function $g(x, s)$ and a recursive function $h(x)$ such that

$$\lim_{s \rightarrow \infty} g(x, s) = X(x),$$

and for all x ,

$$|\{s : g(x, s+1) \neq g(x, s)\}| \leq h(x).$$

So A is superlow if and only if A' is ω -r.e..

The satisfaction of non-superlow cuppability of A can be obtained by ensuring that for any D , if A and D make a cupping of $\emptyset'$, D cannot be superlow. This gives us the condition: if $\emptyset' = \Theta^{A \cup D}$ and $\lim_{s \rightarrow \infty} g(x, s) = D'(x)$, then there is some x such that either $h(x) \uparrow$ or $|\{s : g(x, s+1) \neq g(x, s)\}| > h(x)$, for any choice of Turing functional Θ , partial recursive functions $g(x, s)$ and $h(x)$.

Notice that the last argument of the condition implies that h is not eligible to be a bounding function for the change of g , as it either fails to be total or fails to bound the number of changes of $g(x, s)$ for some x . This condition should be satisfied for every collection of (D, Θ, g, h) . We work on an effective list of the quadruples in our following consideration.

Since all r.e sets are computable in $\emptyset'$, it can be verified that if we replace $\emptyset' = \Theta^{A \cup D}$ with $C = \Theta^{A \cup D}$ in the condition, for an auxiliary r.e set C particularly constructed, the modified condition also works. In fact, as $C \leq_T \emptyset'$, then if

$\emptyset' = \Theta^{A \cup D}$, we have $C \leq_T A \cup D$ thus satisfying the prerequisite of the new condition. So the meeting of the new condition infers the original one. Similarly, if D' is ω -r.e., so is the domain of every partial function computable in D . If we can show there is a Turing functional Ψ such that $\text{dom} \Psi^D$ is not ω -r.e., it can imply that D' is not, either. As a consequence, to satisfy non-superlow cuppability condition of A , it suffices to satisfy the requirements:

R_k : If $C = \Theta_k^{A \cup D_k}$ and $\lim_{s \rightarrow \infty} g_k(x, s) = \text{dom} \Psi^{D_k}$,

then for some x , either $h_k(x) \uparrow$ or $|\{s : g_k(x, s+1) \neq g_k(x, s)\}| > h_k(x)$,

for an effective list of quadruples (D, Θ, g, h) .

On the other hand, to make A promptly simple, we need to make $(2\mathbb{N}+1) \setminus A$ infinite, and, satisfy for all e

P_e : If $|U_e| = \infty$, then $(\exists x)(\exists s)(x \in (U_{e,s+1} - U_{e,s}) \cap A_s)$.

Here, for convenience, we use $2\mathbb{N} + 1$ to denote the set of odd numbers, and $\{U_e\}$ is an enumeration of r.e. sets on $2\mathbb{N} + 1$. These requirements make A promptly simple through the identity function.

4.2 Basic strategy

First we consider how to satisfy one single R requirement, with restricted conditions.

In this section, we temporarily ignore the P -requirements, treating A as unchanged here. On the other hand, we treat D as an r.e. set, with the assumption that $|D_s \setminus D_{s-1}| = 1$.

Recall that the R requirement is:

R_k : If $C = \Theta_k^{A \cup D_k}$ and $\lim_{s \rightarrow \infty} g_k(x, s) = \text{dom} \Psi^{D_k}$,
 then for some x , either $h_k(x) \uparrow$ or $|\{s : g_k(x, s+1) \neq g_k(x, s)\}| > h_k(x)$.

This requirement can be satisfied as long as one of the following three cases is obtained: (a) a disagreement between C and $\Theta_k^{A \cup D_k}$; or (b) a disagreement between $\lim_{s \rightarrow \infty} g_k(x, s)$ and $\text{dom} \Psi^{D_k}$; or (c) $h_k(x)$ fails to bound the number of changes of $g_k(x, s)$, on some x , including that h is not defined on x .

We omit k in the following part. In the strategy, we should require the agreements $\Theta^{A \cup D} = C$ and $\lim_{s \rightarrow \infty} g(x, s) = \text{dom} \Psi^D$ all the time. We periodically change C by enumerate numbers into it, and we define Ψ^D with D that correctly computes C . Each time we have defined Ψ^D on a chosen witness i , we require $g(i, s) = 1$. If the computation $\Theta^{A \cup D}$ does catch after we make a change on C , there must have been a change on D (since we do not consider any change on A here). Such a change undefines Ψ^D . Each time $\Psi^D(i)$ is undefined in this way, we require $g(i, s) = 0$, and wait for this. We later define $\Psi^D(i)$ again, and wait for $g(i, s) = 1$, and then change C one more time. If the process goes on smoothly (which means that we have those agreements before taking any further step), we have a loop of defining and undefining $\Psi^D(i)$. Correspondingly, the value of $g(i, s)$ changes between 0 and 1, and will finally change more than $h(i)$ times.

This idea works as follows:

- (1) Choose an unchosen witness i , wait for $h(i) \downarrow$. Go to (2) when this

happens.

- (2) Choose a set $S \subset \omega$ with $|S| = h(i) + 1$. Wait for $\Theta^{A \cup D_s} \restriction S = C \restriction S$, and let $u = \max \{\theta(x) : x \in S\}$. Go to (3).
- (3) Define $\Psi^{D_s}(i) = 1$ with use u . Wait for $g(i, s) = 1$.

If this happens without any change on $D_s \restriction u$, go to (4).

If $D_s \restriction u$ changes, wait for $g(i, s) = 0$, and $\Theta^{A \cup D_s} \restriction S = C \restriction S$ again. Still, let $u = \max \{\theta(x) : x \in S\}$. Go to (3).

- (4) Enumerate one element of $S \setminus C$ into C . Wait for $\Theta^{A \cup D_s} \restriction S = C \restriction S$, and $g(i, s) = 0$. Go to (3) if $S \setminus C \neq \emptyset$; otherwise the strategy terminates.

□

This idea is shown in Figure 4.1.

Possible outcomes of this strategy include:

- I. The procedure gets stuck at some waiting step. Check the waiting steps, which include:
 - At (1) we wait for $h(i) \downarrow$. Being stuck here implies that h is not total, so R is satisfied;
 - We may need to wait for $\Theta^{A \cup D_s} \restriction S = C \restriction S$ at (2)(3)(4). If the procedure gets stuck here, we have $\Theta^{A \cup D_s} \restriction S \neq C \restriction S$;

Figure 4.1: Basic R -strategy

- We wait for $g(i, s) = 1$ at (3), and $g(i, s) = 0$ at (4). We have $i \in \text{dom}\Psi^D$ in the former case and $i \notin \text{dom}\Psi^D$ in the latter. So in either case we get the disagreement between $g(i, s)$ and $\text{dom}\Psi^D$, if the proposed thing does not happen.

II. The procedure terminates.

This happens only when all $h(i) + 1$ numbers in S have been enumerated into C and no waiting step gets stuck.

Each time we come to (4) from (3), there is a change of $g(i, s)$ from 0 to 1; and each time (4) finishes, $g(i, s)$ changes from 1 to 0. We need $h(i) + 1$ times of these cases happening, to have all in S enumerated into C . Totally, the number of changes on $g(i, s)$ should be (at least) $2h(i) + 1$,

thus satisfying R .

III. The procedure never gets stuck and never halts.

This case happens if $D_s \upharpoonright u$ changes infinitely many times at (3), before $g(i, s) = 1$ is obtained. The strategy then falls in an infinite loop at (3).

Notice that every time $D_s \upharpoonright u$ changes, we first wait for $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$, and update $u = \max \{\theta(x) : x \in S\}$. The fact that $D_s \upharpoonright u$ can change infinitely many times implies that u is growing to infinity (otherwise the total number of $D_s \upharpoonright u$ change is bounded). This implies that $\Theta^{A \cup D_s}$ is partial, also a disagreement between $\Theta^{A \cup D_s}$ and C .

4.3 Priority restraints

Now we involve P -requirements into the strategy, and construct the promptly simple set A with the non-superlow cuppable property.

Consider a P requirement,

$$P_e: \text{ If } |U_e| = \infty, \text{ then } (\exists x)(\exists s)(x \in (U_{e,s+1} - U_{e,s}) \cap A_s).$$

To satisfy this requirement, we can enumerate a number x into A when it enters U_e . For each e , this action is needed exactly once, which is sufficient to have P_e satisfied permanently.

On the other hand, a promptly simple set is coinfinite, so we also need to make $2\mathbb{N} + 1 \setminus A$ infinite. For P_e , we do the enumeration until $x > 4e$ comes into U_e . Such x must exist if U_e is infinite on odd numbers. The set A constructed

in this way contains at most e numbers among the beginning $2e$ odd numbers, thus is a coinfinite set of odd numbers. This makes A a promptly simple set through the identity function.

The main question here is how to carry R -strategies in the presence of P -strategies, for the consistency of each other.

In the R strategy, we require $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$ before taking any further action. To have and preserve this agreement, we need to preserve the part of $A \upharpoonright u$ in the computation. We set this as an A -restraint, to prevent $x < u$ into A . Such a restraint is given priority k . P requirements of lower priorities (i.e. P_e with $e > k$) are only allowed to choose $x > u$ (as well as $x > 4e$).

There is one problem here. As discussed in the previous section, it may happen that strategy R_k repeats step (3) and keeps updating $D_s \upharpoonright u$. In this case, $u = \max \{\theta(x) : x \in S\}$ is going to get infinity. Correspondingly, the restraint on A , with priority k , increases to infinity in the same way. The result is, for any requirement P_e of a lower priority, it may happen that when any number $x > 4e$ enters U_e , it is no larger than the current restraint, so that cannot be enumerated into A . In this sense, P_e is ‘blocked’.

To deal with this matter, we abandon the fixed priority of A -restraints defined in strategy R_k . As we update the use $u = \max \{\theta(x) : x \in S\}$, the A -restraints newly imposed will have decreasing priorities. The purpose of this is that given any fixed level of priority, the restraint on A of this level cannot increase to infinity (proved later). Larger u appears only in restraints with lower priorities.

Figure 4.2: Priorities of Restraints

This idea is fulfilled by introducing more than one witness, to spread A -restraints imposed with different priorities. We use a sequence $i_0, i_1, \dots, i_n$, where i_0 is the original i .

We can see the correspondence between the witnesses and the priorities of A -restraints from Figure 4.2.

In particular, in case that u (or $D_s \upharpoonright u$, equivalently) changes frequently, we always begin with some trial on the computation $\Psi^{A \cup D_s}$, waiting for it to settle down to equal C on S . We do not impose any restraint during this part of the process. This role is taken by i_n , the last one in the sequence of witnesses. To carry on this idea, after we have chosen set S and obtained $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$, instead of working on i_0 , we immediately choose i_1 , to “test the water”. We define $\Psi^{D_s}(i_1) = 1$ with current u as the use, no restraint imposed, then wait for $g(i_1, s) = 1$. If $D_s \upharpoonright u$ changes first, after $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$ is obtained again, we redo the above: define $\Psi^D(i_1)$ and wait for $g(i_1, s) = 1$. In general,

when we start a cycle with $i_0, i_1, \dots, i_n$, we take this loop on i_n .

Our basic idea of the strategy is to keep enumerating numbers into C . Each of such actions is carried when $\Psi^D(i_0)$ is defined and $g(i_0, s) \downarrow = 1$ is obtained. Returning to this idea, we take a ‘tracing back’ action along the witnesses $i_0, i_1, \dots, i_n$. Let $m < n$. For each $m+1$, if we have defined $\Psi^{D_s}(i_{m+1})$, and $g(i_{m+1}, s) = 1$ is successfully obtained (before any change on $D_s \upharpoonright u$), we move backwards, and take the work on i_m . Define $\Psi^{D_s}(i_m) = 1$, with current u as the use. This time we impose an A -restraint of current u , with priority $k+m$. Wait for $g(i_m, s) = 1$ then. When this is obtained, move the work to i_{m-1} if $m \neq 0$; or, when $m = 0$, take step (4) in the strategy, enumerate a number from S into C .

If $D_s \upharpoonright u$ changes before $g(i_m, s) = 1$, we need to wait until $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$ again (with u updated, and also we need to wait for $g(i_p, s) = 0$, $p = m, \dots, n$). Choose an unchosen number as i_{n+1} , wait for $h(i_{n+1}) \downarrow$. Restart this cycle by taking step (3) on i_{n+1} .

As a generalization, there are two kinds of work brought by newly introduced witnesses. The main point, as mentioned above, by the “level down” on the priority of A -restraints, we avoid repeated granting of restraints with a fixed priority. We verify, in later parts, that the restraint of any certain priority can be set finitely many times, so that it is finite. Additionally, on each of $i_0, i_1, \dots, i_n$, we require the accordance between $g(\cdot, s)$ and $\text{dom} \Psi^D$, keeping an eye on the changes of each $g(i_m, s)$. Once we see the changes of $g(i_m, s)$ exceeding $h(i_m)$ for any i_m , the requirement R_k is satisfied, so that we can

terminate the strategy.

For a requirement P_e , when a number $x > 4e$ enters U_e , if it is not forbidden by any A restraint with priority higher or equal to e , we can enumerate it into A . But this may injure some A restraint with a lower priority. From the point of strategy R_k , this reveals to be that $A \restriction u$ may change, when we have imposed some A restraints of u . This is the case that $A_s \restriction u$ changes before $g(i_m, s) = 1$ or $D_s \restriction u$ changes, for some m . Definitely, the A -restraints imposed get violated. Proper initialization within R_k should be taken when this happens.

It is necessary to figure out exactly which restraints defined in R_k are violated so that we see which part is to be initialized. Suppose currently we have $i_0, i_1, \dots, i_n$, and the change of $A \restriction u$ happens when we are waiting for $g(i_m, s) = 1$. As we take step (3) backwards along the witnesses, and also notice that there has been no change of $D_s \restriction u$ through the movement from i_n to i_m (otherwise we will add an i_{n+1} and restart the cycle on it). Along this way, we have defined A -restraints of priorities from $k + n - 1$ to $k + m$, all of which are u . They get violated when $A_s \restriction u$ changes. They are those we ought to deal with as we initialize R_k . For other formerly imposed restraints, they were imposed before the last enumeration from S into C , there is no general conclusion whether they are, or not, violated by this A change.

From this sense, the initialization looks more like some kind of “semi - initialization”. We get to delete $i_m, \dots, i_n$, since restraints with priorities from $k + n - 1$ to $k + m$ are sure to be violated. If $m = 0$, since i_0 gets deleted, the corresponding set S should also be cancelled. We restart the strategy in

this case. Before anything is carried for the next, we wait for the agreement $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$. As the initialization deletes the last witness in the sequence, which is used for the test of the computation of Θ , we should also add a testing witness accordingly. If $m \neq 0$, we add i_m and take step (3) on it; if $m = 0$, we restart the strategy, choose new i_0 and i_1 .

Also in (4), we enumerate one element of $S \setminus C$ into C and wait for agreement between $\Theta^{A \cup D_s} \upharpoonright S$ and $C \upharpoonright S$. Surely $D_s \upharpoonright u$ should change to make this. But if $A \upharpoonright u$ changes first, then all restraints defined get violated. We do a “total” initialization in this case, restart the strategy from the very beginning, similar as the latter case above.

The modified strategy, is then (this is actually Diamondstone’s strategy with respect to r.e. sets):

- (1) Choose an unchosen number as i_0 . Wait for $h(i_0) \downarrow$. Go to step (2).
- (2) Choose a set $S \subset \omega$ with $|S| = h(i_0) + 1$. Wait for $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$.
Let $u = \max \{\theta(x) : x \in S\}$. Choose i_1 and wait for $h(i_1) \downarrow$.
- (3) Suppose we have $i_0, i_1, \dots, i_n$.

- (i) If we are at i_n . Define $\Psi^{D_s}(i_n) = 1$ with current u as the use. Wait for $g(i_n, s) = 1$.

When this happens without changes on $D_s \upharpoonright u$ or $A_s \upharpoonright u$, take step (3) on i_{n-1} .

If $D_s \upharpoonright u$ changes first, wait for $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$ and update $u = \max \{\theta(x) : x \in S\}$. Restart (3) on i_n .

If $A \restriction u$ changes first, replace i_n with a new number. Wait for $h(i_n) \downarrow$ and $\Theta^{A \cup D_s} \restriction S = C \restriction S$. Update u and take step (3) on i_n .

- (ii) If we are at i_m , $m < n$. Define $\Psi^D(i_m) = 1$ with use u . Impose an A -restraint of u with priority $k + m$. Wait for $g(i_n, s) = 1$.

When this happens without changes on $D_s \restriction u$ or $A \restriction u$, if $m \neq 0$, take step (3) on i_{m-1} ; otherwise go to (4).

If $D_s \restriction u$ changes first, wait for $\Theta^{A \cup D_s} \restriction S = C \restriction S$ and update u . Then wait for $g(i_p, s) = 0$ for $p = m, \dots, n$. Choose a new number as i_{n+1} . Wait for $h(i_{n+1}) \downarrow$. Restart step (3) on i_{n+1} .

If $A_s \restriction u$ changes first, delete $i_m, \dots, i_n$. If $m \neq 0$, choose a new i_m and take (3) on i_m when $h(i_m) \downarrow$ and $\Theta^{A \cup D_s} \restriction S = C \restriction S$; otherwise go to step (1).

- (4) Enumerate one number of $S \setminus C$ into C . Wait for $\Theta^{A \cup D_s} \restriction S = C \restriction S$, and then $g(i_m, s) = 0$ on all existing i_m s. Choose an unchosen number as i_{n+1} and wait for $h(i_{n+1}) \downarrow$. Restart step (3) on i_{n+1} .

If $A \restriction u$ changes first, restart the strategy.

□

Figure 4.3 and Figure 4.4 show this strategy. We call the work of tracing backwards along the witnesses a cycle. There may be more than one cycle between two enumerations of numbers from S into C , due to the possible change on $A \restriction u$ or $D \restriction u$ happening during the tracing.

Some remarks for this strategy:

Figure 4.3: R_k Strategy Figure 4.4: The Cycle with $i_0, \dots, i_n$

- I. Part of this strategy works in a “passive” pattern. The most significant evidence is the instructions of “wait”. Such instructions require us to do nothing for several (unpredictable long) stages until the desired equalities appear. Notice that this suspension is carried within the strategy only: strategies with other indices move on. It is possible that we wait forever, which is also a way of satisfying the requirement.
- II. Equalities are required all the time throughout the strategy. First, we require $\Theta^{A \cup D_s} \restriction S = C \restriction S$. Whenever $A \restriction u$, $D \restriction u$ or $C \restriction S$ change, so that the equality does not hold, we wait until another change retrieves the equality. Another point is, we need $g(\cdot, s)$ correctly reflects $\text{dom} \Psi$, so when we define $\Psi^{D_s}(i_m)$, we wait for $g(i_m, s) = 1$; when $D_s \restriction u$ changes, making $\Psi^{D_s}(i_m)$ undefined, we wait for $g(i_m, s) = 0$.
- III. When we choose a new witness, i_m , we start tracking g on i_m . We count the changes of $g(i_m, s)$. In the process, whenever this counting exceeds $h(i_m)$, we stop the strategy, as the requirement has been satisfied. This should be regarded as a general criterion throughout the strategy.
- IV. Compare (4) of the above with what we have presented previously. Here we have omitted the last instruction: terminate the strategy if $S \setminus C = \emptyset$. As we have discussed, if the strategy proceeds and we can keep enumerating numbers of S into C , $g(i_0, s)$ should keep changing as desired. So the counting of the number of change keeps increasing and will exceed $h(i_0)$ at some step. The size $|S| = h(i_0) + 1$, combined with the general criterion

above, guarantees that the requirement will be satisfied and terminated in this way, if other ways fail.

4.4 Extension to d.r.e. case

Now we extend the issue to the case that D is a d.r.e. set. We investigate the adaptability of the previous strategy for the current case. First of all, if we are stuck in some waiting stage, or trapped in an infinite loop, requirement R_k gets satisfied by the same way before. The main concern here, is to guarantee that $g(i_0, s)$ can still change more than $h(i_0)$ times in the progress.

We assume D has an approximation $\{D_s\}_{s \in \omega}$, with $|(D_s \setminus D_{s-1}) \cup (D_{s-1} \setminus D_s)| = 1$ for each $s > 0$. In d.r.e case, D can change in another way: an enumerated number is extracted from D . By definition, the extracted numbers cannot be back.

A change of this pattern may return $D_s \upharpoonright u$ to a previous status, giving a repeated oracle of Ψ . If this happens, Ψ^{D_s} does not get undefined, but is changed from one computation to another computation. As we require the consistency between $g(\cdot, s)$ and $\text{dom} \Psi^{D_s}$, in this case, $g(\cdot, s)$ is not required to change from 1 to 0.

Check the steps in the strategy where we were requiring $g(i_0, s)$ to change. We come to step (4) with $g(i_0, s) = 1$. At step (4), when one number is enumerated from S into C and the desired $D \upharpoonright u$ change (resulting $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$) happens, it undefines $\Psi^{D_s}(i_0)$, consequently we need to wait for $g(i_0, s) = 0$. On the other hand, the next time we take step (3) on i_0 , as we define $\Psi^{D_s}(i_0)$

again, we need $g(i_0, s)$ to switch to 1. These changes on $g(i_0, s)$ should all happen if the process does move on. The fulfilling of them requires $g(i_0, s)$ to change $2h(i_0) + 1$ times in total, with the size of S as $h(i_0) + 1$. Actually we can terminate the strategy halfway when we see $g(i_0, s)$ changes more than $h(i_0)$ times.

Now in d.r.e. case, it is possible that in step (4), after we have enumerated a number of S into C , a change of this type happens: D_s has changed to a status that $\Psi^{D_s}(i_0)$ has been defined previously, for some use. Ideally, such a change brings the equality $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$. If so, we can proceed allowing $g(i_0, s)$ remain to be 1. Suppose until the next time we come to i_0 backwards through $i_{n+1}, i_n, \dots, i_0$, no new status of D_s appears, so $\Psi^{D_s}(i_0)$ stays in a defined state. Here $g(i_0, s)$ being 1 satisfies our need.

In the corresponding part while D is r.e., $\Psi^{D_s}(i_0)$ gets undefined and then defined, so we need $g(i_0, s)$ to change twice. When D is d.r.e., it is possible that $\Psi^{D_s}(i_0)$ is defined the whole time, so $g(i_0, s)$ being 1 is sufficient. As a result, $g(i_0, s)$ may be required to change not so many as $2h(i_0) + 1$ times to have S all into C .

A vital issue here is to investigate the condition for such a case to happen. Before defining Ψ^{D_s} , we always have $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$. So those D_s , that we use as oracles to define Ψ^{D_s} , all satisfy this computation, joint work with A of that time. Notice that each time we define Ψ^{D_s} , we impose an A restraint at the same time, aiming to protect the computation of Θ . The point is, for a restraint, once imposed, it lasts forever. We try to keep $A \upharpoonright u$ due to the assigned

restraint. As long as $A \upharpoonright u$ is preserved, the computation $\Theta^{A \cup D_s} \upharpoonright S$ with those D_s is identified, equalling to some former $C \upharpoonright S$. Under this circumstance, after we have enumerated some number from S into C (meaning $C \upharpoonright S$ has changed, we use C' to distinguish), if D_s returns to a previous status, we have $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S \neq C' \upharpoonright S$.

However, the restraints may not be effective all the time, as they can be injured by some P_e requirement with higher priorities, making $A \upharpoonright u$ fail to be preserved. The change on $A \upharpoonright u$ gives a new initial segment of A , whose joint work as the oracle with formerly appeared D_s , is not predictable. This makes the mentioned case possible. We explain it through the following example.

We wait for $g(i_0, s) = 1$ as we have defined $\Psi^{D_s}(i_0)$. If $D_s \upharpoonright u$ changes first, we add one witness i to the end of the sequence and again define Ψ along the sequence backwards. Suppose in this way we have defined $\Psi^{D_s}(i_0)$ several times, with oracles $D_{t_1}, D_{t_2}, \dots, D_{t_k}$. Let the current A be A_1 . We have $\Theta^{A_1 \cup D_{t_p}} \upharpoonright S = C \upharpoonright S$, $p = 1, \dots, k$. The next time when we come to some i_m with $m \neq 0$, $A_1 \upharpoonright u$ changes, giving A_2 . We wait until some D_s , say, D_2 , makes $\Theta^{A_2 \cup D_2} \upharpoonright S = C \upharpoonright S$. According to the strategy, we delete i_m and those after, and choose a new i_m . Suppose we get $g(i_0, s) = 1$ after we define $\Psi^{D_2}(i_0)$. The next is to enumerate a number from S into C , so it becomes C' . In the waiting for Θ to compute $C' \upharpoonright S$, D changes back to D_{t_k} . Previously, we have not checked the computation $\Theta^{A_2 \cup D_{t_k}}$. It is possible that $\Theta^{A_2 \cup D_{t_k}} \upharpoonright S = C' \upharpoonright S$. If so, we proceed with $g(i_0, s)$ being 1. Until the next time we come to i_0 backwards along the sequence of witness, if D does not change out of D_{t_p} , $p = 1, \dots, k$,

then $\Psi^{D_s}(i_0)$ is always defined so that we do not require $g(i_0, s)$ to be 0. So the strategy can proceed smoothly to the next time we enumerate one of S into C . This coincides the previous description.

As a consequence, with a change on $A \upharpoonright u$ (necessary), it is possible those D_s we have used in previous definitions are capable to compute future status of C (i.e., after we enumerate new numbers into C). D_s returning to previous status avoids the request on $g(i_0, s)$ to change in the process. The worst case, is that we can enumerate all numbers of S into C with $g(i_0, s)$ changing only once.

For further convenience, here we call the change of $D_s \upharpoonright u$, which does not undefine $\Psi^{D_s}(i_0)$ but instead makes its value turn back to a formerly defined one, to be an *invalid change*. The occuring of invalid changes results repeated definitions of $\Psi^{D_s}(i_0)$, allowing the strategy to proceed with $g(i_0, s)$ staying 1.

The previous discussion implies that any set S with a prescribed size at the beginning cannot guarantee enough changes on $g(i_0, s)$ as required (which is $h(i_0)$). The reason is that the happening of invalid changes cannot be determined at the beginning. However, the happening of invalid changes is conditional. They cannot happen consecutively for infinitely many times.

It is easy to see that an invalid change extracts a number out of $D \upharpoonright u$ and return the oracle of Ψ to a previous status. So at any time the extraction can happen finitely many, if there is no new enumeration. Suppose that y is extracted from D . Thus $\Psi^D(i_0)$ returns to an existing definition made before, which, is the one got undefined as y was enumerated into D . As the extracted

number cannot be back, we can have a one-to-one mapping from each invalid change (or the definition it makes Ψ return), to one former definition of $\Psi^D(i_0)$. In other words, future happening of invalid changes can be no more than previously made definitions. So by counting the definitions we have made at step (3)(which becomes a definite number when we come to step (4)), we can have an upper bound for potential happening of invalid changes.

Still, we should choose S large enough. As we enumerate numbers from S into C , we have $|S| + 1$ different status of $S \upharpoonright C$ and need to obtain the equality $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$ this many times. We let $|S|$ larger than the counting of definitions on $\Psi^D(i_0)$, so that the collection of those repeated $D_s \upharpoonright u$ resulted from invalid changes are not enough to make the equality hold for $|S| + 1$ times. Thus an unmet $D_s \upharpoonright u$ is expected in the process, undefining $\Psi^{D_s}(i_0)$ and consequently, letting us require $g(i_0, s)$ to be 0. This time in d.r.e. case, as we do not finish counting the definitions we have made until we come to step (4), we choose set S with the proposed size at step (4), instead of at the beginning.

Notice that a change on $D_s \upharpoonright u$ may also happen when we come to the following step (3). If the resulting $D_s \upharpoonright u$ undefines any $\Psi^D(i_m)$, we wait until $g(i_m, s) = 0$. (Unless $D_s \upharpoonright u$ changes again during the waiting, we then request $g(i_m, s)$ to change accordingly.) If $\Psi^D(i_0)$ gets undefined in this way, it indeed brings us another change on $g(i_0, s)$. On the other hand, it may also bring a sequence of new definitions on $\Psi^D(i_0)$, during our waiting for $g(i_0, s) = 1$. This, enlarges the collection of potential invalid changes. So we count again the

definitions on $\Psi^D(i_0)$, and update the size of S in the following step (4).

Now that it is meaningless to have a definite size of S at the beginning, we start with a set of size 1.

The strategy for R_k is:

- (1) Choose an unchosen number as i_0 . Start counting the changes of $g(i_0, s)$.
Wait for $h(i_0) \downarrow$. Go to (2).
- (2) Choose a set $S \subset \omega$ on unused numbers with $|S| = 1$. Wait for $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$. Let $u = \max \{\theta(x) : x \in S\}$. Choose i_1 and wait for $h(i_1) \downarrow$.
Set the definition counter to 0.
- (3) Suppose we have $i_0, i_1, \dots, i_n$ and we are now at i_m .

Check whether $\Psi^D(i_m)$ has been defined under current D_s with some use $\tilde{u}$. If not, define $\Psi^D(i_m) = 1$, and let the use $\tilde{u}$ of this computation be the latest obtained u .

- (i) If $m = n$. Wait for $g(i_n, s) = 1$.

When this happens without changes on $D_s \upharpoonright \tilde{u}$ or $A_s \upharpoonright u$, take step (3) on i_{n-1} .

If $D_s \upharpoonright \tilde{u}$ changes first, wait for $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$ and update $u = \max \{\theta(x) : x \in S\}$. Restart (3) on i_n .

If $A \upharpoonright u$ changes first, replace i_n with a new number. Wait for $h(i_n) \downarrow$ and $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$. Update u and take step (3) on i_n .

- (ii) If $m < n$. Impose an A -restraint of u with priority $k + m$. Increase the definition counter by 1 if $m = 0$. Wait for $g(i_n, s) = 1$.

When this happens without changes on $D_s \upharpoonright \tilde{u}$ or $A \upharpoonright u$, if $m \neq 0$, take step (3) on i_{m-1} ; otherwise go to (4).

If $D_s \upharpoonright \tilde{u}$ changes first, wait for $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$ and update u . Choose a new number as i_{n+1} . Wait for $h(i_{n+1}) \downarrow$. Restart step (3) on i_{n+1} .

If $A_s \upharpoonright u$ changes first, delete $i_m, \dots, i_n$. If $m \neq 0$, choose a new i_m and take (3) on i_m when $h(i_m) \downarrow$ and $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$; otherwise go to step (1).

- (4) Suppose the definition counter is currently r . Choose a set $S' \subset \omega$ on unused numbers with $|S'| = r + 1$. Let $S = S \cup S'$, and wait for $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$.

Then enumerate one number of $S \setminus C$ into C . Wait for $\Theta^{A \cup D_s} \upharpoonright S = C \upharpoonright S$. Set the counter to 0. Choose a new i_{n+1} , wait for $h(i_{n+1}) \downarrow$, and restart (3) on i_{n+1} .

If $A_s \upharpoonright u$ changes first during either waiting part, restart the strategy.

□

This strategy can be illustrated in Figure 4.5, where the cycle can still be illustrated using Figure 4.4.

Within this strategy, when $D_s \upharpoonright \tilde{u}$ change happens, if $\Psi^D(i_m)$ is undefined for some i_m , we wait for $g(i_m, s) = 0$. We terminate this strategy when we see

Figure 4.5: R_k Strategy for d.r.e case

some $g(i_m, s)$ changes more times than $h(i_m)$.

4.5 Construction

In the construction, we carry each strategy in a “leap forward” pattern. In one single stage, we are not restricted to one step in a strategy. In fact, we proceed as much as possible, until we are requested to wait. The wait takes unpredictable many stages. We continue when the target of waiting is acquired. While we wait in one strategy, we proceed other strategies as usual.

The whole construction process is:

Stage 0. Let $A = \emptyset$, $C = \emptyset$.

Stage s . Start carrying on strategy R_{s-1} .

For each activating strategy R_k , and each witness i_m that currently chosen for it, if

$$|t < s : g_k(i_m, t + 1) \neq g_k(i_m, t)| > h_k(i_m),$$

terminate this strategy.

When $x > 4e$ enters U_e with requirement P_e not yet satisfied, if there is no restraint of priority $p \leq e$, such that the use bound of the restraint is larger or equal to x , enumerate x into A_s .

4.6 Verification

Lemma 3. *The constructed A is not cuppable by any superlow d.r.e. set.*

Proof. It suffices to show that each requirement R_k is satisfied.

First, strategy R_k cannot restart infinitely many times. We restart the strategy only if some P_e strategy with $e < k$ acts, and each P strategy acts at most once. So we can start the consideration from the last action of step (1) in strategy R_k .

It is clear that if the strategy get stuck in a waiting step, it is satisfied. Also, as we have discussed in Section 4, if the strategy keeps enumerating numbers from S into C , it can be guaranteed that the changing of $g(i_0, s)$ will exceed $h(i_0)$, satisfying the requirement.

All we need to consider is the case that the strategy falls into an infinite loop. So the infinite loop can only be restricted within step (3). We mentioned, in the case that D is r.e., that this outcome implies $\Theta^{A \cup D}$ is partial. In the case that D is d.r.e., an infinite loop in step (3) subjects to infinitely many changes on $D_s \upharpoonright u$ as well. u tends to go to infinity to make this. So this implies the same conclusion, making requirement R_k satisfied. $\square$

Lemma 4. *Any restraint with priority p is given finitely many times.*

Proof. As the restraint with priority p can only be given by strategies R_k with $k \leq p$, it suffices to show that each such strategy can give priority p restraint finitely many times.

Assume the contrary and let R_j be the first strategy that gives infinitely many restraints with priority p . Within strategy R_j , p restraints are given after we have defined $\Psi^D(i_m)$ with $m = p - j$, and have had $g(i_m, s) = 1$ in step (3). Similar as above, we assume P_e strategies with $e < p$ that ever act have acted,

so that i_m cannot be canceled.

The infinite granting of restraint with priority p implies that we have taken step (3) on i_m infinitely many times. If this involves infinite actions of step (4), by the previous lemma, we should have terminated the strategy. So it should be an infinite loop containing only step (3). Slightly different from the infinite loop outcome discussed in the previous lemma, now we require the strategy comes to i_m infinitely many times. To have this, there is an i_p with $p \leq m$, such that the follow thing happens infinitely many times. When we have defined $\Psi^D(i_p)$ and wait for $g(i_p, s)=1$, there is a change on $D_s \upharpoonright u$ so that we add a witness and restart the cycle. As $D_s \upharpoonright u$ changes infinitely many times, we can always have unmet $D \upharpoonright u$ that undefines $\Psi^D(i_{m+1})$, requiring $g(i_{m+1}, s)$ change to 0. But in each cycle, we come to i_m with $g(i_{m+1}, s)$. The infinite happening of this gives a contradiction, as we should terminate the strategy when we see $g(i_{m+1}, s)$ changes more than $h(i_{m+1})$ times. $\square$

Lemma 5. *Each requirement P_e is satisfied. Thus set A constructed is promptly simple.*

Proof. It is shown above that $(2\mathbb{N} + 1) \setminus A$ infinite. Consider requirement P_e . Now that restraints with priority $p \leq e$ are given finitely many times, there is a stage s that no more such restraints are given thereafter. If U_e is infinite, there are numbers larger than all restraints entering U_e after stage s . We can enumerate it into A when we see the first one. Thus P_e can be satisfied. And A is then promptly simple. $\square$

This completes the proof of Theorem 19.

In the case that D is an ω -r.e. set, similarly D may possibly return to a past status, so that Ψ^D does not get undefined. The difference between d.r.e and ω -r.e. is that each status of D can be returned once in d.r.e. case, but more times in ω -r.e. case. So each time while updating set S , suppose we have counted r times of definitions of $\Psi^D(i_0)$, we should add $r[f(i_0) - 1] + 1$ numbers to S , where $f(\cdot)$ is the recursive function that bounds the changes in approximation $\{D_s\}, s \in \omega$.

Bibliography

- [1] K. Ambos-Spies: Contiguous r.e. degrees, Computation and proof theory (Aachen, 1983) Lecture Notes in Math. 1104, Springer, Berlin, 1-37 (1984)
- [2] K. Ambos-Spies and P. A. Fejer: Degree theoretical splitting properties of recursively enumerable sets, Journal of Symbolic Logic 53(4), 1110-1137 (1988)
- [3] K. Ambos-Spies, C. G. Jockusch, Jr., R. A. Shore, and R. I. Soare: An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees, Transactions of the American Mathematical Society 281, 109-128 (1984)
- [4] M. Arslanov, S. B. Cooper and A. Li: There is no low maximal d. c. e. degree, Mathematical Logic Quarterly 46(3), 409-416 (2000)
- [5] R. Blaylock, R. Downey and S. Lempp: Infima in the recursively enumerable weak truth table degrees, Notre Dame Journal of Formal Logic 38(3), 406-418 (1997)

- [6] P. F. Cohen: Weak truth table reducibility and the pointwise ordering of 1-1 recursive function, Ph.D Thesis, University of Illinois Urbana-Champaign (1975)
- [7] S. B. Cooper: Degrees of Unsolvability, Ph.D. Thesis, Leicester University (1971)
- [8] S. B. Cooper: The density of the low_2 n-r.e. degrees, *Archive for Mathematical Logic* 31, 19-24 (1991)
- [9] S. B. Cooper: A splitting theorem for the n-r.e. degrees, *Proceedings of the American Mathematical Society* 115(2), 461-471 (1992)
- [10] S. B. Cooper: *Computability theory*, Chapman & Hall/CRC (2003)
- [11] S. B. Cooper, L Harrington, A. H. Lachlan, S. Lempp and R. I. Soare: The d-r.e. degrees are not dense, *Annals of Pure and Applied Logic* 55, 125-191 (1991)
- [12] S. B. Cooper and A. Li: Turing definability in the Ershov hierarchy, *Journal of the London Mathematical Society* 66(3), 513-528 (2001)
- [13] S. B. Cooper, S. Lempp, P. Watson: Weak density and cupping in the d-r.e. degrees, *Israel Journal of Mathematics* 67(2), 137-152 (1989)
- [14] D. Diamondstone: Promptness does not imply superlow cuppability, *Journal of Symbolic Logic* 74(4), 1264-1272 (2009)

- [15] R. G. Downey: Intervals and sublattices of the r.e. weak truth table degrees part 1: density, *Annals of Mathematical Logic* 41, 1-26 (1989)
- [16] R. G. Downey: Intervals and sublattices of the r.e. weak truth table degrees part 2: nonbounding, *Annals of Mathematical Logic* 44, 153-172 (1989)
- [17] R. G. Downey and D. R. Hirschfeldt: *Algorithmic randomness and complexity*, Springer-Verlag, Berlin (2010)
- [18] R. G. Downey and M. Stob: Structural interactions of the recursively enumerable T- and W-degrees, *Annals of Pure and Applied Logic* 31, 205-236 (1986)
- [19] R. G. Downey and M. Stob: Splitting theorems in recursion theory, *Annals of Pure and Applied Logic* 65, 1-106 (1993)
- [20] P. Fischer: Pairs without infimum in the recursively enumerable weak truth table degrees, *Journal of Symbolic Logic* 51(1), 117-129 (1986)
- [21] R. M. Friedberg: two recursively enumerable sets of incomparable degrees of unsolvability, *Proceedings of the National Academy of Sciences* 43, 236-238 (1963)
- [22] R. M. Friedberg and H. Rogers, Jr.: Reducibility and completeness for sets of integers, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 5, 117-125 (1959)

- [23] N. Greenberg and A. Nies: Benign cost functions and lowness properties, *Journal of Symbolic Logic* 76(1), 289-312 (2011)
- [24] L. Harrington: Understanding Lachlan's monster paper, handwritten notes.
- [25] S. Ishmukhametov: D.r.e. sets, their degrees and index sets, Ph.D. thesis, Novosibirsk, Russia (1986)
- [26] A. H. Lachlan: Lower bounds for pairs of recursively enumerable degrees, *Proceedings of the London Mathematical Society* 16, 537-569 (1966)
- [27] A. H. Lachlan: A recursively enumerable degree which will not split over all lesser ones, *Annals of Mathematical Logic* 9, 307-365 (1975)
- [28] A. H. Lachlan: Bounding minimal pairs, *Journal of Symbolic Logic* 44, 626-642 (1979)
- [29] R. E. Ladner and L. P. Sasso: The weak truth table degrees of recursively enumerable sets, *Annals of Mathematical Logic* 8, 429-448 (1975)
- [30] S. Lempp: Priority arguments in computability theory, model theory, and complexity theory, Lecture notes.
- [31] A. A. Muchnik: On the unsolvability of the problem of reducibility in the theory of algorithms, *Doklady Akademii nauk SSSR* 108, 194-197 (1956)
- [32] P. Odifreddi: *Classical Recursion Theory, Studies in Logic and the Foundations of Mathematics* 125, Elsevier, New York (1989)

- [33] P. Odifreddi: Classical Recursion Theory Volume II, Studies in Logic and the Foundations of Mathematics 143, Elsevier, New York (1999)
- [34] R. W. Robinson: A dichotomy of the recursively enumerable sets, *Zeit. Math. Logik Grundlagen Math.* 14, 339-356 (1968)
- [35] R. W. Robinson: Interpolation and embedding in the recursively enumerable degrees, *Annals of Mathematics* 93(2), 285-314 (1971)
- [36] R. I. Soare: Recursively enumerable sets and degrees, Springer-Verlag, Heidelberg (1987)
- [37] G. E. Sacks: On the degrees less than $0'$, *Annals of Mathematics* 77(2), 211-231 (1963)
- [38] G. E. Sacks: The recursively enumerable sets are dense, *Annals of Mathematics* 80(2), 300-312 (1964)
- [39] M. Stob: Wtt-degrees and T-degrees of r.e. sets, *Journal of Symbolic Logic* 48(4), 921-930 (1983)
- [40] G. Wu: Isolation and lattice embeddings, *Journal of Symbolic Logic* 67(3), 1055-1064 (2002)
- [41] G. Wu, M. M. Yamaleev: Isolation: motivations and applications, *Proceedings of Kazan University (Physics & Mathematics Series)* 52, 204-217 (2012)

- [42] G. Wu and M. M. Yamaleev: There are no maximal d.c.e. wtt-degrees, Lecture Notes in Computer Science 10010, 479-486 (2017)
- [43] C. E. M. Yates: A minimal pair of recursively enumerable degrees, Journal of Symbolic Logic 31, 159-168 (1966)