Cone avoidance 3

Prerequisites: Chapter 2

The appellation *first-jump control*[1] encompasses the set of techniques to build a set G while controlling its $\Sigma^0_1(G)$ properties. An immediate application is the construction of sets of low degree whenever the process is Δ^0_2. With the development of reverse mathematics, the subject gained a whole lot of interest, as being the main tool to prove separations over RCA_0. We shall see a variety of preservation properties (cone avoidance, PA avoidance, ...) motivated by specific subsystems of second-order arithmetic, such as ACA_0 and WKL_0. Nowadays, these techniques are part of the mandatory toolbox of a researcher in reverse mathematics.

1: The name might be confusing at first, since the technique is about computation and not jump computation. Actually, by deciding $\Sigma^0_1(G)$ properties, the first-jump control determines what the jump G' *is*, not what it *computes*. Moreover, since the predicate $\Phi^G_e(x)\downarrow$ is $\Sigma^0_1(G)$, the first-jump control enables to decide G-computation.

The general setting is the following: One wants to build a set G satisfying some structural properties (being a path through a tree, being homogeneous for a coloring, or more generally being a solution to an instance of a mathematical problem), while preserving some computational weakness properties (not computing a fixed set, not being of PA degree, being of low degree). There is a tension between the computational strength induced by the structural properties, and the desired computational weakness. As it turns out, all these proofs have a common denominator: the design of a so-called forcing question with good definitional properties. The study of the relation between the forcing question and iterated jump control constitutes the main subject of this textbook.

The first weakness property that we shall consider is called *cone avoidance*. Proofs of cone avoidance are good examples of the use of the forcing question, and they do not require to make the whole construction effective, as in proofs of lowness.

3.1 Context and motivation

Consider a mathematical problem P, formulated in term of *instances* and *solutions*.[2] The computability-theoretic study of P consists in identifying, given a (computable) instance X of P, the computational power of computing a solution to X. For this, one proves lower bounds, of the form "There exists a (computable) instance of P such that every solution is computationally strong" and upper bounds of the form "Every (computable) instance of P admits a computationally weak solution".

2: For example, *weak König's lemma* is the problem whose instances are infinite binary trees, and whose solutions are infinite paths

One of the first questions to ask about the strength of a problem is its ability to *encode* a Turing degree. More precisely, given a set C, is there a computable instance of P such that every solution computes C? This question is about the *computational strength* of P. One can ask the same question with no computable restriction to the instance of P. It is then about the *combinatorial strength* of P. The notion of cone avoidance is a strong negative answer to the first question.

Definition 3.1.1. A problem P admits *cone avoidance* if for every set Z and every non-Z-computable set C, every Z-computable instance X of P admits a solution Y such that C is not $Z \oplus Y$-computable. $\diamond$

It might be simpler to think of its unrelativized version, where $Z = \emptyset$. Every known natural problem which satisfies the unrelativized version also satisfies the general statement. However, one can create artificial problems which do not.

Informally, if a problem admits cone avoidance, then it is not able to encode any non-computable Turing degree. If one drops the restriction by replacing "every Z-computable instance X of P" with "every instance X of P", one obtains the notion of *strong cone avoidance*.

A proof of cone avoidance of a problem P is an interesting statement in its own right, but it also has useful consequences in reverse mathematics. Recall that ACA_0 is the base system RCA_0 augmented with the comprehension axiom for arithmetical formulas with parameters. Since the halting set $\emptyset'$ is Σ^0_1-definable, every ω-model of ACA_0 contains the halting set. [3]

On the other hand, if a Π^1_2 problem P admits cone avoidance[4], then it admits an ω-model which avoids the halting set, hence is not a model of ACA_0.

3: By the same argument, every ω-model of ACA_0 is closed under the Turing jump. Actually, there exists a smallest ω-model of ACA_0 whose second-order part is exactly the arithmetical sets.

4: A problem P is Π^1_2 if if the relations $X \in \operatorname{dom} \mathsf{P}$ and $Y \in \mathsf{P}(X)$ are both arithmetically definable. Then, $\mathcal{M} \models \mathcal{P}$ if

$$\mathcal{M} \models \forall X \in \operatorname{dom} \mathsf{P}\ \exists Y \in \mathsf{P}(X)$$

Proposition 3.1.2. Fix a non-computable set C. Let P be a Π^1_2 problem which admits cone avoidance. There exists an ω-model of $\mathsf{RCA}_0 + \mathsf{P}$ which does not contain C. ★

Proof. Recall that an ω-model is fully characterized by its second-order part, and that it satisfies RCA_0 iff its second-order part is a Turing ideal. Also recall that $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ is Cantor's pairing function.

We are going to define a sequence of sets $Z_0 \leq_T Z_1 \leq_T \dots$ such that for all $n \in \mathbb{N}$,

(1) if $n = \langle e, s \rangle$ and $\Phi_e^{Z_s}$ is a P-instance X, then Z_{n+1} computes a solution to X;
(2) $C \not\leq_T Z_n$.

$Z_0 = \emptyset$. Suppose we have defined Z_n and say $n = \langle e, s \rangle$. If $\Phi_e^{Z_s}$ is not a P-instance, then let $Z_{n+1} = Z_n$. Otherwise, by cone avoidance of P relativized to Z_n, there is a solution Y to $\Phi_e^{Z_s}$ such that $C \not\leq_T Z_n \oplus Y$. Let $Z_{n+1} = Z_n \oplus Y$.

Let $\mathcal{I} = \{X \in 2^{\mathbb{N}} : \exists n\ X \leq_T Z_n\}$. By construction, the class $\mathcal{I}$ is a Turing ideal. Moreover, by (1), every P-instance $X \in \mathcal{I}$ admits a solution in $\mathcal{I}$. Last, by (2), $C \notin \mathcal{I}$. ■

3.2 First examples

Before starting the development of an abstract framework to prove cone avoidance, let us start with a few basic proofs, in order to see some emerging patterns.

The most basic example of cone avoidance is Cohen genericity. Indeed, this notion of forcing enjoys very nice computability-theoretic features: the partial order is computable, with a computable domain. Recall that Cohen forcing is the notion of forcing whose conditions are finite strings, partially ordered by the suffix relation.

Theorem 3.2.1
Let C be a non-computable set. For every sufficiently Cohen generic set G, $C \not\leq_T G$.

Proof. It suffices to prove the following lemma, where $\Phi_e^G \neq C$ is a shorthand for $\exists x \Phi_e^G(x)\uparrow \vee \exists x \Phi_e^G(x)\downarrow \neq C(x)$.

Lemma 3.2.2. For every condition $\sigma \in 2^{<\mathbb{N}}$ and every Turing index $e \in \mathbb{N}$, there is an extension $\tau \geq \sigma$ forcing $\Phi_e^G \neq C$. ⋆

Proof. Fix a condition σ. Consider the following set[5]

$$U = \{(x, v) \in \mathbb{N} \times 2 : \exists \tau \geq \sigma \; \Phi_e^\tau(x)\downarrow = v\}$$

Note that the set U is Σ_1^0. There are three cases:[6]

- Case 1: $(x, 1 - C(x)) \in U$ for some $x \in \mathbb{N}$. Let $\tau \geq \sigma$ witness $(x, 1 - C(x)) \in U$, that is, let $\tau \geq \sigma$ be such that $\Phi_e^\tau(x)\downarrow = 1 - C(x)$. Then τ forces $\Phi_e^G \neq C$.
- Case 2: $(x, C(x)) \notin U$ for some $x \in \mathbb{N}$. We claim that σ already forces $\Phi_e^G \neq C$. Indeed, if for some $Z \in [\sigma]$, $\Phi_e^Z = C$, then by the use property, these is some $\tau \preceq Z$ such that $\Phi_e^\tau(x)\downarrow = C(x)$, and by choosing τ long enough, it would witness $(x, C(x)) \in U$, contradiction.
- Case 3: None of Case 1 and Case 2 holds. Then U is a Σ_1^0 graph of the characteristic function of C, hence C is computable. This contradicts our hypothesis.[7] ■

5: In other words, U is a set of pairs (input/-value) such that one can find an extension forcing $\Phi_e^G(x)$ to halt and output v. This set will be recurrent in the proofs of cone avoidance, with the 3-case analysis pattern.

6: The idea is the following: the set U claims to be a nice (Σ_1^0) description of a set C which is hard to describe (not computable). Thus, either U gives only partial information about C (Case 2) or it gives some wrong information (Case 1).

7: We assume here that the functional Φ_e is $\{0, 1\}$-valued.

We are now ready to prove Theorem 3.2.1. Given $e \in \mathbb{N}$, let $\mathcal{D}_e$ be the set of all conditions τ forcing $\Phi_e^G \neq C$. It follows from Lemma 3.2.2 that every $\mathcal{D}_e$ is dense, hence every $\{\mathcal{D}_e : e \in \mathbb{N}\}$-generic set G satisfies $C \not\leq_T G$. ■

Theorem 3.2.1 can be used to prove the existence of incomparable Turing degrees, as shows the following exercise:

Exercise 3.2.3.

1. Fix a set C. Show that for every sufficiently Cohen generic set G, C does not compute G.
2. Use Theorem 3.2.1 and the previous question to deduce the existence of incomparable Turing degrees. ⋆

The following example shows that every set A admits a Δ_2^0 description which avoids a cone. It is a fundamental bridge between computational weaknesses and combinatorial weaknesses of theorems, as we shall see later.

Theorem 3.2.4
Fix a set A and a non-computable set C. There exists a set G such that $G' \geq_T A$ and $G \not\geq_T C$.

Proof. By Shoenfield's limit lemma [8], $G' \geq_T A$ iff there is a G-computable function $f : \mathbb{N}^2 \to 2$ such that for every $x \in \mathbb{N}$, $\lim_y f(x, y)$ exists and equals $A(x)$. We are therefore going to build directly the function f by forcing, and let G be the graph of f. The forcing conditions are pairs (g, n), such that

- $g \subseteq \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ is a partial function[8] with two parameters whose domain is finite, representing an initial segment of the function f that we are building.
- m is an integer "locking" the m first columns of f to the m first bits of A, meaning that from now on, when we extend the domain of g with a new pair (x, y), if $x < m$ then $g(x, y) = A(x)$.

8: The notation $f \subseteq A \to B$ is used for partial functions from A to B.

In other words the first m columns of the function f have already reached their limit behavior, which is $A{\upharpoonright}_m$. The *interpretation* $[g, m]$ of a condition (g, m) is the class of all partial or total functions $h \subseteq \mathbb{N}^2 \to 2$ such that

(1) $g \subseteq h$, i.e. $\operatorname{dom} g \subseteq \operatorname{dom} h$ and for all $(x, y) \in \operatorname{dom} g$, $g(x, y) = h(x, y)$;
(2) for all $(x, y) \in \operatorname{dom} h \setminus \operatorname{dom} g$, if $x < m$, then $h(x, y) = A(x)$.

Note that set of conditions is computable, but unlike Cohen forcing, the partial order is not. Thankfully, for a fixed condition (g, n), the set of all conditions extending (g, n) is computable. Indeed, it suffices to "hard code" the initial segment $A{\upharpoonright}_n$ in the algorithm, which is a finite piece of information.

A condition (h, n) *extends* (g, m) (denoted $(h, n) \leq (g, m)$) if $n \geq m$ and $h \in [g, m]$. Every filter $\mathcal{F}$ for this notion of forcing induces a function $f_{\mathcal{F}} = \bigcup\{g : (g, n) \in \mathcal{F}\}$. In particular, $f_{\mathcal{F}} \in \bigcap\{[g, n] : (g, n) \in \mathcal{F}\}$. Moreover, if $\mathcal{F}$ is sufficiently generic, then $f_{\mathcal{F}}$ is total, and $\lim_x f_{\mathcal{F}}(x, y) = A(x)$.

Lemma 3.2.5. For every condition (g, n) and every Turing index $e \in \mathbb{N}$, there is an extension $(h, n) \leq (g, n)$ forcing $\Phi_e^f \neq C$. ★

Proof. Fix a condition (g, n). Consider the following set

This is the second appearance of the set U of all pairs (input/value) such that one can find an extension forcing $\Phi_e^f(x)$ to halt and output v.

$$U = \{(x, v) \in \mathbb{N} \times 2 : \exists h \in [g, n]\ \Phi_e^h(x){\downarrow}= v\}$$

Note that the set U is Σ_1^0 since by the use property, the existential quantifier is first-order. There are three cases:

We have the same 3-case analysis as in the proof Lemma 3.2.2, and which is characteristic of proofs of cone avoidance.

- Case 1: $(x, 1 - C(x)) \in U$ for some $x \in \mathbb{N}$. Let $h \in [g, n]$ witness $(x, 1-C(x)) \in U$, that is, let $h \in [g, n]$ be such that $\Phi_e^h(x){\downarrow}= 1-C(x)$. Then (h, n) forces $\Phi_e^f \neq C$.
- Case 2: $(x, C(x)) \notin U$ for some $x \in \mathbb{N}$. We claim that (g, n) already forces $\Phi_e^f \neq C$. Indeed, if for some $f \in [g, n]$, $\Phi_e^f = C$, then by the use property, these is some finite $h \subseteq f$ such that $\Phi_e^h(x){\downarrow}= C(x)$, and by choosing $\operatorname{dom} h \supseteq \operatorname{dom} g$, it would witness $(x, C(x)) \in U$, contradiction.
- Case 3: None of Case 1 and Case 2 holds. Then U is a Σ_1^0 graph of the characteristic function of C, hence C is computable. This contradicts our hypothesis. ■

We are now ready to prove Theorem 3.2.4. Let $\mathcal{F}$ be a sufficiently generic filter for this notion of forcing, and let $f = f_{\mathcal{F}}$. The set of conditions (g, n) such that $x \in \operatorname{dom} g$ is dense, thus f is total. Moreover, for every $k \in \mathbb{N}$, the set of conditions (g, n) such that $n \geq k$ is also dense, so for every $x \in \mathbb{N}$, $\lim_y f(x, y) = A(x)$. Last, by Lemma 3.2.5, $f \not\geq_T C$. This completes the proof of Theorem 3.2.4. ■

Recall that a set G is of *high* degree if $G' \geq_T \emptyset''$. It follows from Theorem 3.2.4 that if C is a non-computable set, there exists a set G of high degree such that $C \not\leq_T G$.

Our last example is the famous *cone avoidance* Π_1^0 *basis theorem*. It says that if every path of an infinite computable binary tree computes a single set, then this set is computable. This will be our first example of the use of an over-approximation because the natural formula does not have the desired complexity.

Theorem 3.2.6 (Jockusch and Soare [9])
Fix a non-computable set C and a non-empty Π^0_1 class $\mathcal{P} \subseteq 2^{\mathbb{N}}$. There exists a member $G \in \mathcal{P}$ such that $G \not\geq_T C$.

Proof. Jockusch-Soare forcing is the notion of forcing whose conditions are infinite computable binary trees $T \subseteq 2^{<\mathbb{N}}$, partially ordered by the subset relation. The *interpretation* $[T]$ of a tree T is the class of its paths. Every sufficiently filter $\mathcal{F}$ for this notion of forcing induces a path $G_{\mathcal{F}}$ which is the unique element of $\bigcap\{[T] : T \in \mathcal{F}\}$.

Lemma 3.2.7. For every condition T and every Turing index $e \in \mathbb{N}$, there is an extension $S \subseteq T$ forcing $\Phi^G_e \neq C$. ⋆

A natural first attempt would be to define $\mathcal{U}$ as the set

$$\{(x, v) : \exists\sigma \text{ extendible in } T\ \Phi^\sigma_e(x)\downarrow= v\}$$

However, being extendible is a Π^0_1 predicate, hence U would be Σ^0_2. The third case would then yield that C is $\emptyset'$-computable, which does not contradict our hypothesis.

Proof. Fix a condition T. Consider the following set

$$U = \{(x, v) \in \mathbb{N} \times 2 : \exists \ell \in \mathbb{N} \forall \sigma \in 2^\ell \cap T\ \Phi^\sigma_e(x)\downarrow= v\}$$

The over-approximation is the following: at every length, at least one node must be extendible in T, so it suffices to ask the property to hold for every nodes of a given length.

Note that the set U is Σ^0_1. There are three cases:

- Case 1: $(x, 1 - C(x)) \in U$ for some $x \in \mathbb{N}$. We claim that T already forces $\Phi^G_e \neq C$. Indeed, for every $G \in [T]$, letting $\sigma = G\upharpoonright_\ell$, where ℓ witnesses $(x, 1 - C(x)) \in U$, we have $\sigma \in 2^\ell \cap T$, hence $\Phi^\sigma_e(x)\downarrow= 1 - C(x)$. By the use property, $\Phi^G_e(x)\downarrow= 1 - C(x)$
- Case 2: $(x, C(x)) \notin U$ for some $x \in \mathbb{N}$. Let

  $$S = \{\sigma \in T : \forall s < |\sigma|\ \Phi^\sigma_e(x)[s]\uparrow \vee \Phi^\sigma_e(x)[s]\downarrow\neq C(x)\}$$

  Since $(x, C(x)) \notin U$, S contains a string of every length. Moreover, S is closed under prefix, so it is an infinite binary subtree of T. Again, by the use property, S forces $\Phi^G_e \neq C$.
- Case 3: None of Case 1 and Case 2 holds. Then U is a Σ^0_1 graph of the characteristic function of C, hence C is computable. This contradicts our hypothesis. ■

We still have the same 3-case analysis as in the proof Lemma 3.2.2, but the situation is slightly different: instead of taking a proper extension in Case 1 and already forcing the property in Case 2, the situation is inverted.

We are now ready to prove Theorem 3.2.6. Let $\mathcal{F}$ be a sufficiently generic filter for this notion of forcing, and let $G = G_{\mathcal{F}}$. By Lemma 3.2.7, $G \not\geq_T C$. This completes the proof of Theorem 3.2.6. ■

Exercise 3.2.8. A *(computable) Mathias condition* is a pair (σ, X) where $\sigma \in 2^{<\mathbb{N}}$ and $X \subseteq \mathbb{N}$ is an infinite (computable) set with $|\sigma| < \min X$. The *interpretation* $[\sigma, X]$ of a (computable) Mathias condition is the class $\{Y \in 2^{\mathbb{N}} : \sigma \subseteq Y \subseteq \sigma \cup X\}$, identifying σ with the finite set $\{n < |\sigma| : \sigma(n) = 1\}$. Intuitively, σ is the initial segment of the set that we construct, and X is an infinite reservoir which restricts the futur elements of the set.

A condition (τ, Y) *extends* a condition (σ, X) if $\tau \succeq \sigma$, $Y \subseteq X$ and $\tau \setminus \sigma \subseteq X$. Every filter $\mathcal{F}$ for this notion of forcing induces a set $G_{\mathcal{F}} = \bigcup\{\sigma : (\sigma, X) \in \mathcal{F}\}$.

Prove that if C is a non-computable set, then for every sufficiently generic filter $\mathcal{F}$, $C \not\leq_T G_{\mathcal{F}}$. ⋆

3.3 Forcing question

One can easily see an emerging pattern in all the previous proofs of cone avoidance. In every case, given a condition p, one defines a set U of pairs

(x, v) such that such that there is an extension forcing $\Phi_e^G(x)\downarrow= v$. Moreover, for every pair (x, v) outside U, there is an extension forcing the opposite. This motivates the following definition:

Definition 3.3.1. Given a notion of forcing $(\mathbb{P}, \leq)$ and a family of formulas Γ, a *forcing question* is a relation $?\vdash : \mathbb{P} \times \Gamma$ such that, for every $p \in \mathbb{P}$ and $\varphi(G) \in \Gamma$,

1. If $p \mathrel{?\vdash} \varphi(G)$, then there is an extension $q \leq p$ forcing $\varphi(G)$;
2. If $p \mathrel{?\nvdash} \varphi(G)$, then there is an extension $q \leq p$ forcing $\neg\varphi(G)$. ◊

One can see a forcing question as a completion of the forcing relation. Intuitively, given a formula $\varphi(G) \in \Gamma$, one can divide the conditions in $\mathbb{P}$ into three categories: the ones which force $\varphi(G)$, those which force $\neg\varphi(G)$, and the ones which do not decide $\varphi(G)$. A forcing question has no degree of freedom when considering conditions of the first two categories: it must give the appropriate answer. On the other hand, a condition belonging to the third category has extensions forcing $\varphi(G)$ and other extensions forcing $\neg\varphi(G)$. A forcing question draws a dividing line within this category.

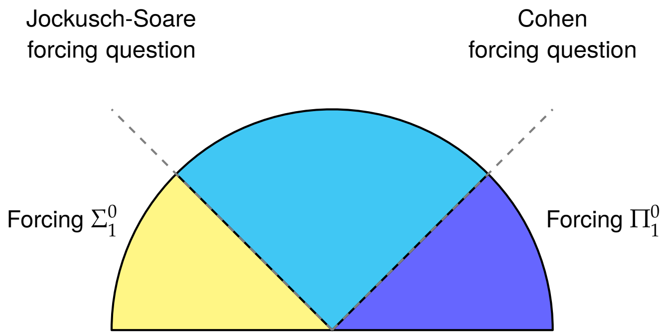

Figure 3.1: The yellow part and the dark blue part represent the conditions forcing a fixed Σ_1^0 and its negation, respectively. The light blue part represent the conditions of the third category. In the proof of Theorem 3.2.6, the dividing line is at the left-most position, while for Cohen forcing, the dividing line is at the opposite position.

Exercise 3.3.2. Show that a relation $?\vdash : \mathbb{P} \times \Gamma$ is a forcing question for Γ iff it satisfies the following properties:

1. If p forces $\varphi(G)$, then $p \mathrel{?\vdash} \varphi(G)$;
2. If p forces $\neg\varphi(G)$, then $p \mathrel{?\nvdash} \varphi(G)$. ★

In each cone avoidance proof, one then considers the following set:

$$U = \{(x, v) \in \mathbb{N} \times 2 : p \mathrel{?\vdash} \Phi_e^G(x)\downarrow= v\}$$

By definition of a forcing question, the two first cases can be handled abstractly. On the other hand, the contradiction of the third case lies on the complexity of the set U. This is our last ingredient of the proof.

Definition 3.3.3. Given a notion of forcing $(\mathbb{P}, \leq)$ and a family of formulas Γ, a forcing question is Γ-*preserving* if for every $p \in \mathbb{P}$ and every formula $\varphi(G, x) \in \Gamma$, the relation $p \mathrel{?\vdash} \varphi(G, x)$ is in Γ uniformly in x. ◊

We are now ready to prove our abstract theorem of cone avoidance.

Theorem 3.3.4
Let $(\mathbb{P}, \leq)$ be a notion of forcing with a Σ_1^0-preserving forcing question.

For every non-computable set C and every sufficiently generic filter $\mathcal{F}$, $C \not\leq_T G_{\mathcal{F}}$.

Proof. It suffices to prove the following lemma:

Lemma 3.3.5. For every condition $p \in \mathbb{P}$ and every Turing index $e \in \mathbb{N}$, there is an extension $q \leq p$ forcing $\Phi_e^G \neq C$. ⋆

Proof. Consider the following set

$$U = \{(x, v) \in \mathbb{N} \times 2 : p \mathrel{?\!\vdash} \Phi_e^G(x)\downarrow = v\}$$

Since the forcing question is Σ_1^0-preserving, the set U is Σ_1^0. There are three cases:

- Case 1: $(x, 1-C(x)) \in U$ for some $x \in \mathbb{N}$. By Property (1) of the forcing question, there is an extension $q \leq p$ forcing $\Phi_e^G(x)\downarrow = 1 - C(x)$.
- Case 2: $(x, C(x)) \notin U$ for some $x \in \mathbb{N}$. By Property (2) of the forcing question, there is an extension $q \leq p$ forcing $\Phi_e^G(x)\uparrow$ or $\Phi_e^G(x)\downarrow \neq C(x)$.
- Case 3: None of Case 1 and Case 2 holds. Then U is a Σ_1^0 graph of the characteristic function of C, hence C is computable. This contradicts our hypothesis. ■

We are now ready to prove Theorem 3.3.4. Given $e \in \mathbb{N}$, let $\mathcal{D}_e$ be the set of all conditions $q \in \mathbb{P}$ forcing $\Phi_e^G \neq C$.. It follows from Lemma 3.3.5 that every $\mathcal{D}_e$ is dense, hence every sufficiently generic filter $\mathcal{F}$ is $\{\mathcal{D}_e : e \in \mathbb{N}\}$-generic, so $C \not\leq_T G_{\mathcal{F}}$. This completes the proof of Theorem 3.3.4. ■

By the abstract theorem above, the question whether a problem admits cone avoidance is reduced to the question whether one can construct solutions using a notion of forcing which admits a forcing question with the right definitional property.

We can revisit the previous proofs in terms of forcing questions.

Exercise 3.3.6. Given a string $\sigma \in 2^{<\mathbb{N}}$ and a Σ_1^0 formula $\varphi(G)$, define $\sigma \mathrel{?\!\vdash} \varphi(G)$ to hold if there is some $\tau \succeq \sigma$ such that $\varphi(\tau)$ holds. Prove that the relation is a Σ_1^0-preserving forcing question for Cohen forcing. ⋆

Exercise 3.3.7. Given a computable infinite binary tree $T \subseteq 2^{<\mathbb{N}}$ and a Σ_1^0 formula $\varphi(G)$, define $T \mathrel{?\!\vdash} \varphi(G)$ to hold if there is some level $\ell \in \mathbb{N}$ such that $\varphi(\sigma)$ holds for every node σ at level ℓ in T. Prove that the relation is a Σ_1^0-preserving forcing question for Jockusch-Soare forcing. ⋆

The notion of forcing question is more useful as a unifying terminology than as a formal notion. We shall see in the next section a disjunctive notion of forcing building two generic sets simultaneously. Although the concept of forcing question will need some adaptation to the current setting, the similarity of terminology will help emphasize the common features with the previous proofs of cone avoidance.

3.4 Seetapun's theorem

In short, Seetapun's theorem states that Ramsey's theorem for pairs admits cone avoidance. It is one of the most celebrated theorems of reverse mathematics. Given a set $X \subseteq \mathbb{N}$, we let $[X]^n$ denote the set of all n-element subsets of X.[9] A set $H \subseteq \mathbb{N}$ is *homogeneous* for a coloring $f : [\mathbb{N}]^n \to k$ if f is monochromatic on $[H]^n$. Ramsey's theorem for n-tuples and k colors is the problem RT^n_k whose instances are colorings $f : [\mathbb{N}]^n \to k$ and whose solutions are infinite f-homogeneous sets.[10]

9: We shall often identify $[X]^n$ with the set of increasing ordered n-tuples, and write $f(x_0, \dots, x_{n-1})$ rather than $f(\{x_0, \dots, x_{n-1}\})$, assuming $x_0 < \dots < x_{n-1}$.

10: Ramsey's theorem is formulated in terms of colorings of $[\mathbb{N}]^n$. However, it is a set-theoretic statement, and it still holds when replacing $\mathbb{N}$ with any infinite set. One can prove prove this stronger statement as a blackbox: Given an infinite set $X \subseteq \mathbb{N}$ and a coloring $f : [X]^n \to k$, define the coloring $g : [\mathbb{N}]^n \to k$ by $g(F) = f(\iota[F])$, where $\iota : \mathbb{N} \to X$ is the canonical bijection. For any infinite g-homogeneous set $H \subseteq \mathbb{N}$, the set $\iota[H]$ is an infinite f-homogeneous subset of X.

When using the stronger statement, one must take into account the computational strength of the set X, as the f-homogeneous set is $H \oplus X$-computable.

In particular, RT^1_k is the infinite pigeonhole principle[11], while the statement RT^2_k states that if the edges of an infinite clique is k-colored, then there is an infinite subset of vertices whose induced subgraph is monochromatic. The question whether Ramsey's theorem for pairs implies ACA_0 over RCA_0 was open for a decade, before Seetapun [10] answered it negatively by proving that RT^2_2 admits cone avoidance. Since then, the original proof was simplified [11] and extended to other preservation properties [12]. We will present the simplified version and leave the original one as an exercise.

11: It might be useful to consider sets $A \in 2^{\mathbb{N}}$ as instances of RT^1_2. A solution to A is then an infinite subset $H \subseteq A$ or $H \subseteq \overline{A}$.

The modern version of Seetapun's theorem is divided into two steps, based on the decomposition of Ramsey's theorem for pairs into the cohesiveness and the pigeonhole principles. An infinite set $C \subseteq \mathbb{N}$ is *cohesive* for a sequence of sets $\vec{R} = R_0, R_1, \dots$ if for every $n \in \mathbb{N}$, $C \subseteq^* R_n$ or $C \subseteq^* \overline{R_n}$, where $\subseteq^*$ means "included up to finite changes". The *cohesiveness principle* is the problem COH whose instances are infinite sequences of sets, and whose solutions are infinite cohesive sets.

We start with a proof of Ramsey's theorem for pairs using the cohesiveness principle and the pigeonhole principle, with no computability-theoretic consideration.

Theorem 3.4.1 (Ramsey)
Every coloring $f : [\mathbb{N}]^2 \to 2$ admits an infinite f-homogeneous set.

Proof. The proof is divided into three steps.

Cohesive step: Let $\vec{R} = R_0, R_1, \dots$ be the sequence of sets defined for every $x \in \mathbb{N}$ by $R_x = \{y \in \mathbb{N} : f(x, y) = 1\}$. By COH, there is an infinite $\vec{R}$-cohesive set $X \subseteq \mathbb{N}$. In particular, for every $x \in X$, $\lim_{y \in X} f(x, y)$ exists.

From a computability-theoretic perspective, the sequence $\vec{R}$ is f-computable, the coloring $\hat{f}$ is $\Delta^0_2(f \oplus X)$, and the set H is $f \oplus X \oplus Y$-computable.

Pigeonhole step: Let $\hat{f} : X \to 2$ be the limit coloring of f, that is, $\hat{f}(x) = \lim_{y \in X} f(x, y)$. By RT^1_2, there is an infinite $\hat{f}$-homogeneous set $Y \subseteq X$ for some color $i < 2$.

Post-processing: Since for every $x \in Y$, $\lim_{y \in Y} f(x, y) = i$, one can thin out the set Y to obtain an infinite f-homogeneous subset $H \subseteq Y$. ■

Seetapun's theorem will therefore be proven by combining cone avoidance of the cohesiveness principle and strong cone avoidance of the pigeonhole principle. There exists a simple proof of cone avoidance of COH using computable Mathias forcing.

Theorem 3.4.2
Let C be a non-computable set. For every uniformly computable sequence of sets $R_0, R_1, \ldots$, there is an infinite $\vec{R}$-cohesive set G such that $C \not\leq_T G$.

Proof. Recall the notion of computable Mathias forcing[12] from Exercise 3.2.8. Given a condition (σ, X) and a Σ^0_1 formula $\varphi(G)$, one can define a Σ^0_1-preserving forcing question $(\sigma, X) \mathrel{?\vdash} \varphi(G)$ which holds if there is some $\rho \subseteq X$ such that $\varphi(\sigma \cup \rho)$ holds. Thus, for every sufficiently generic filter $\mathcal{F}$, $C \not\leq_T G_{\mathcal{F}}$. We now show that $G_{\mathcal{F}}$ is $\vec{R}$-cohesive.

12: One could have used a variant of Mathias forcing where conditions are pairs (σ, X) such that $C \not\leq_T X$. In general, one requires the reservoirs to satisfy the desired property of the theorem.

Given some $n \in \mathbb{N}$, let $\mathcal{D}_n$ be the set of all conditions (σ, X) such that either $X \subseteq R_n$, or $X \subseteq \overline{R}_n$. The set $\mathcal{D}_n$ is dense, since given a computable Mathias condition (σ, X), either $X \cap R_n$ is infinite, or $X \cap \overline{R}_n$ is infinite (say the former case holds), in which case $(\sigma, X \cap R_n) \in \mathcal{D}_n$. Thus, if $\mathcal{F}$ is $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$-generic, then $G_{\mathcal{F}}$ is $\vec{R}$-cohesive. ■

Actually, the exact computational strength of the cohesiveness principle is well-understood: given a uniformly computable sequence of sets $\vec{R} = R_0, R_1, \ldots$, and $\sigma \in 2^{<\mathbb{N}}$, one can define the set R_σ as follows:

$$R_\sigma = \bigcap_{\sigma(n)=0} \overline{R}_n \bigcap_{\sigma(n)=1} R_n$$

Then, let $\mathscr{C}(\vec{R})$ be the $\Pi^0_1(\emptyset')$ class of all $P \in 2^{\mathbb{N}}$ such that for every $\sigma \prec P$, R_σ is infinite.

The natural proof of COH consists in deciding which one of R_0 or $\overline{R}_0$ is infinite (say R_0), then picking an element $x_0 \in R_0$, then deciding which one of $R_0 \cap R_1$ or $R_0 \cap \overline{R}_1$ is infinite (say $R_0 \cap \overline{R}_1$), then picking an element $x_1 \in R_0 \cap \overline{R}_1$, and so on. The class $\mathscr{C}(\vec{R})$ represents the collection of all "valid" decisions, that is, choices which will not yield a finite set.

Exercise 3.4.3 (Jockusch and Stephan [13]).

1. Fix a uniformly computable sequence of sets $\vec{R} = R_0, R_1, \ldots$. Show that the degrees of the $\vec{R}$-cohesive sets are exactly the degrees whose jump computes a a member of $\mathscr{C}(\vec{R})$.
2. Show that for every $\Pi^0_1(\emptyset')$ class $\mathscr{P} \subseteq 2^{\mathbb{N}}$, there exists a uniformly computable sequence of sets $\vec{R} = R_0, R_1, \ldots$ such that $\mathscr{C}(\vec{R}) = \mathscr{P}$. ⋆

It follows from Exercise 3.4.3 that the computability-theoretic study of COH is inherited from the study of Π^0_1 classes. In particular, since there exists a universal Π^0_1 class whose members are of PA degree, there exists a maximally difficult sequence of uniformly computable sets $\vec{R} = R_0, R_1, \ldots$ such that the jump of every $\vec{R}$-cohesive set is of PA degree over $\emptyset'$.

Exercise 3.4.4. Combine Exercise 3.4.3 and Theorem 3.2.4 to give an alternative proof of Theorem 3.4.2. ⋆

Exercise 3.4.5 (Patey [14]). Use Exercise 3.4.3 to prove that if a computable instance of COH admits a solution of low degree, then it admits a computable solution. ⋆

The last component of our proof of Seetapun's theorem is strong cone avoidance of the pigeonhole principle.[13]

13: The proof of Ramsey's theorem involves only Δ^0_2 instances of the pigeonhole principle. Thus, at first sight, it seems too strong to consider arbitrary instances. However, by Theorem 3.2.4, every instance of RT^1_2 is Δ^0_2 relative to a cone avoiding degree, so considering arbitrary instances or Δ^0_2 instances is equivalent.

Theorem 3.4.6 (Dzhafarov and Jockusch [11])
Let C be a non-computable set. For every set A, there is an infinite subset $H \subseteq A$ or $H \subseteq \overline{A}$ such that $C \not\leq_T H$.

Proof. Fix C and A. The first difficulty of this theorem is the disjunctive nature of the statement. One does not know in advance what side of A is more suitable to build an infinite subset. This is why we are going to build two sets G_0, G_1 simultaneously, with $G_0 \subseteq A$ and $G_1 \subseteq \overline{A}$. For simplicity, let $A_0 = A$ and $A_1 = \overline{A}$.

The two sets will be constructed through a variant of Mathias forcing, whose *conditions* are triples (σ_0, σ_1, X) where

1. (σ_i, X) is a Mathias condition for each $i < 2$;
2. $\sigma_i \subseteq A_i$;
3. $C \not\leq_T X$.

There is an easy way to see that at least one of the two initial segments is extendible into an infinite solution: Given a condition (σ_0, σ_1, X), there is some $i < 2$ such that $X \cap A_i$ is infinite. Thus, $\sigma_i \cup (X \cap A_i)$ is an infinite subset of A_i.

One must really think of a condition as a pair of Mathias conditions which share a same reservoir. The *interpretation* $[\sigma_0, \sigma_1, X]$ of a condition (σ_0, σ_1, X) is the class

$$[\sigma_0, \sigma_1, X] = \{(G_0, G_1) : \forall i < 2\ \sigma_i \preceq G_i \subseteq \sigma_i \cup X\}$$

Note that throughout the proof, the only manipulations of the reservoir are finite truncation and splitting based on a Π^0_1 class of 2-colorings. Thus, the whole argument would work by fixing a Scott ideal $\mathcal{M}$ such that $C \notin \mathcal{M}$ and requiring $X \in \mathcal{M}$.

A condition (τ_0, τ_1, Y) *extends* (σ_0, σ_1, X) if (τ_i, Y) Mathias extends (σ_i, X) for each $i < 2$. Any filter $\mathcal{F}$ induces two sets $G_{\mathcal{F},0}$ and $G_{\mathcal{F},1}$ defined by $G_{\mathcal{F},i} = \bigcup\{\sigma_i : (\sigma_0, \sigma_1, X) \in \mathcal{F}\}$. Note that $(G_{\mathcal{F},0}, G_{\mathcal{F},1}) \in \bigcap\{[\sigma_0, \sigma_1, X] : (\sigma_0, \sigma_1, X) \in \mathcal{F}\}$.

The goal is therefore to build two infinite sets G_0, G_1, satisfying the following requirements for every $e_0, e_1 \in \mathbb{N}$: [14]

$$\mathcal{R}_{e_0,e_1} : \Phi_{e_0}^{G_0} \neq C \vee \Phi_{e_1}^{G_1} \neq C$$

14: One could use Posner's trick, saying that if G_0 and G_1 both compute C, then there is a single Turing functional Φ_e such that $\Phi_e^{G_0} = \Phi_e^{G_1} = C$. Then, the requirement becomes $\mathcal{R}_e : \Phi_e^{G_0} \neq C \vee \Phi_e^{G_1} \neq C$.

If every requirement is satisfied, then an easy *pairing argument*[15] shows that either $C \not\leq_T G_0$, or $C \not\leq_T G_1$. However, in general, it is not possible to ensure that G_0 and G_1 are both infinite. For example, A could be finite or co-finite. Thankfully, in any of these cases, there is a simple computable solution. More generally, we make the following assumption:

15: A *pairing argument* says that if for every $(a, b) \in \mathbb{N}^2$, either $a \in A$ or $b \in B$, then either $A = \mathbb{N}$ or $B = \mathbb{N}$.

$$\text{There is no infinite set } H \subseteq A \text{ or } H \subseteq \overline{A} \text{ such that } C \not\leq_T H. \tag{H1}$$

Under this assumption, one can prove that if $\mathcal{F}$ is sufficiently generic, then both $G_{\mathcal{F},0}$ and $G_{\mathcal{F},1}$ are infinite.

Lemma 3.4.7. Suppose (H1). Let $p = (\sigma_0, \sigma_1, X)$ be a condition and $i < 2$. There is an extension (τ_0, τ_1, Y) of p and some $n > |\sigma_i|$ such that $n \in \tau_i$. ★

Proof. If $X \cap A^i$ is empty, then $X \subseteq A^{1-i}$, but $C \not\leq_T X$, which contradicts (H1). Thus, there is $n \in X \cap A^i$. Let $\tau_i = \sigma_i \cup \{n\}$, and $\tau_{1-i} = \sigma_{1-i}$. Then, $(\tau_0, \tau_1, X \setminus \{0, \dots, n-1\})$ is an extension of p such that $n \in \tau_i$. ■

We will now prove the core lemma.

Lemma 3.4.8. Let $p = (\sigma_0, \sigma_1, X)$ be a condition, and $e_0, e_1 \in \mathbb{N}$. There is an extension (τ_0, τ_1, Y) of p forcing $\mathcal{R}_{e_0,e_1}$. ★

Proof. Consider the following set[16]

$$U = \{(x, v) \in \mathbb{N} \times 2 : \forall Z_0 \sqcup Z_1 = X\ \exists i < 2\ \exists \rho \subseteq Z_i\ \Phi_{e_i}^{\sigma_i \cup \rho}(x)\downarrow = v\}$$

16: The naïve set to consider would be $U = \{(x, v) : \exists i < 2\ \exists \rho \subseteq X \cap A_i\ \Phi_{e_i}^{\sigma_i \cup \rho}(x)\downarrow = v\}$. It would yield valid forcing question, but with a bad definitional complexity: the set U is $\Sigma^0_1(X \oplus A)$. The third case would yield that $C \leq_T X \oplus A$, which is not a contradiction.

One must get rid of the set A which is arbitrary complex. For this, we use an over-approximation by considering *all* instances of RT^1_2. Since the class of all instances of RT^1_2 is efffectively closed in Cantor space, hence effectively compact, this over-approximation yields a $\Sigma^0_1(X)$ set.

At first sight, this set seems computationally very strong, as it contains a universal second-order quantification. However, by a compactness argument[17],

17: Consider the tree of finite 2-partitions of initial segments of $\mathbb{N}$.

the set can be equivalently defined as

$$\{(x,v) \in \mathbb{N} \times 2 : \exists \ell \in \mathbb{N} \forall Z_0 \sqcup Z_1 = X{\upharpoonright}_\ell \ \exists i < 2 \ \exists \rho \subseteq Z_i \ \Phi_{e_i}^{\sigma_i \cup \rho}(x){\downarrow}= v\}$$

Thus, the set U is $\Sigma^0_1(X)$. There are three cases:

- Case 1: $(x, 1 - C(x)) \in U$ for some $x \in \mathbb{N}$. Letting $Z_0 = A_0 \cap X$ and $Z_1 = A_1 \cap X$, there is some $i < 2$ and some $\rho \subseteq Z_i$ such that $\Phi_{e_i}^{\sigma_i \cup \rho}(x) \downarrow= 1 - C(x)$. Letting $\tau_i = \sigma_i \cup \rho$ and $\tau_{1-i} = \sigma_{1-i}$, the condition $(\tau_0, \tau_1, X \setminus \{0, \dots, \max \rho\})$ is an extension of p forcing $\Phi_{e_i}^{G_i}(x) \downarrow\neq C(x)$.
- Case 2: $(x, C(x)) \notin U$ for some $x \in \mathbb{N}$. Consider the class $\mathcal{P}$ of all sets $B \in 2^{\mathbb{N}}$ such that, letting $B_0 = B$ and $B_1 = \overline{B}$, for every $i < 2$, and every $\rho \subseteq X \cap B_i$, $\Phi_{e_i}^{\sigma_i \cup \rho}(x){\uparrow}$ or $\Phi_{e_i}^{\sigma_i \cup \rho}(x){\downarrow}\neq C(x)$. The class $\mathcal{P}$ is $\Pi^0_1(X)$, so by the cone avoidance basis theorem (Theorem 3.2.6), there is some $B \in \mathcal{P}$ such that $C \not\leq_T X \oplus B$. Since X is infinite, there is some $i < 2$ such that $X \cap B_i$ is infinite. The condition $(\sigma_0, \sigma_1, X \cap B_i)$ is an extension of p forcing $\Phi_{e_i}^{G_i}(x){\uparrow} \vee \Phi_{e_i}^{G_i}(x){\downarrow}\neq C(x)$.
- Case 3: None of Case 1 and Case 2 holds. Then U is a $\Sigma^0_1(X)$ graph of the characteristic function of C, hence C is X-computable. This contradicts our hypothesis. ∎

Because of the use of an over-approximation, in Case 2, the instance B of RT^1_2 witnessing the negation has nothing to do with the original instance A. The instance B is chosen so that every solution to it will satisfy the Π^0_1 fact. By committing to be simultaneously a solution to A and B, one can create a solution to A which forces the Π^0_1 fact. This ability to be simultaneously a solution to multiple instances is a feature of Ramsey-type statements.

We are now ready to prove Theorem 3.4.6. Let $\mathcal{F}$ be a sufficiently generic filter for this notion of forcing, and for each $i < 2$, let $G_i = G_{\mathcal{F},i}$. By Lemma 3.4.7, both sets are infinite. Moreover, by Lemma 3.4.8, either $C \not\leq_T G_0$ or $C \not\leq_T G_1$. Letting H be this set, it satisfies the statement of Theorem 3.4.6. ∎

One can formulate the proof of Theorem 3.4.6 in terms of forcing question, with the appropriate disjunctive definition.

Definition 3.4.9. Given a disjunctive notion of forcing $(\mathbb{P}, \leq)$ and a family of formulas Γ, a *forcing question* is a relation $?{\vdash} : \mathbb{P} \times \Gamma$ such that, for every $p \in \mathbb{P}$ and every pair of formulas $\varphi_0(G), \varphi_1(G) \in \Gamma$,

1. If $p \mathrel{?{\vdash}} \varphi_0(G_0) \vee \varphi_1(G_1)$, then there is an extension $q \leq p$ forcing $\varphi_i(G_i)$ for some $i < 2$;
2. If $p \mathrel{?{\nvdash}} \varphi_0(G_0) \vee \varphi_1(G_1)$, then there is an extension $q \leq p$ forcing $\neg\varphi_i(G_i)$ for some $i < 2$. ◊

Note that if $p \mathrel{?{\nvdash}} \varphi_0(G_0) \vee \varphi_1(G_1)$, one does not force $\neg\varphi_0(G_0) \wedge \neg\varphi_1(G_1)$, but their disjunction.

Exercise 3.4.10. Fix a non-computable set C, a set A, and consider the notion of forcing of Theorem 3.4.6. Given a condition $p = (\sigma_0, \sigma_1, X)$ and two Σ^0_1 formulas $\varphi_0(G), \varphi_1(G)$, define $p \mathrel{?{\vdash}} \varphi_0(G_0) \vee \varphi_1(G_1)$ to hold if for every 2-partition $Z_0 \sqcup Z_1 = X$, there is some $i < 2$ and a finite set $\rho \subseteq Z_i$ such that $\varphi(\sigma_i \cup \rho)$ holds.

1. Show that the relation $p \mathrel{?{\vdash}} \varphi_0(G_0) \vee \varphi_1(G_1)$ is $\Sigma^0_1(X)$.
2. Prove that it is a forcing question in the sense of Definition 3.4.9. ⋆

We now have all the necessary ingredients to prove Seetapun's theorem.

Theorem 3.4.11 (Seetapun [10])
Let C be a non-computable set. For every computable coloring $f : [\mathbb{N}]^2 \to \mathbb{N}$, there is an infinite f-homogeneous set H such that $C \not\leq_T H$.

Proof. The proof follows the one of Theorem 3.4.1, using cone avoidance of COH (Theorem 3.4.2) and strong cone avoidance of RT^1_2 (Theorem 3.4.6).

Fix C and f. Let $\vec{R} = R_0, R_1, \dots$ be the computable sequence of sets defined for every $x \in \mathbb{N}$ by $R_x = \{y \in \mathbb{N} : f(x,y) = 1\}$. By Theorem 3.4.2, there is an infinite $\vec{R}$-cohesive set $X \subseteq \mathbb{N}$ such that $C \not\leq_T X$. In particular, for every $x \in X$, $\lim_{y \in X} f(x,y)$ exists. Let $\hat{f} : X \to 2$ be the limit coloring of f, that is, $\hat{f}(x) = \lim_{y \in X} f(x,y)$. By Theorem 3.4.6, there is an infinite $\hat{f}$-homogeneous set $Y \subseteq X$ for some color $i < 2$ such that $C \not\leq_T Y \oplus X$. Since for every $x \in Y$, $\lim_{y \in Y} f(x,y) = i$, one can thin out the set Y to obtain an infinite f-homogeneous subset $H \subseteq Y$. ∎

The original proof of Seetapun's theorem [10] was more direct, using a notion of forcing to build homogeneous sets for colorings of pairs. We leave it as an exercise.

Exercise 3.4.12 (Seetapun and Slaman [10]). Fix a computable coloring $f : [\mathbb{N}]^2 \to 2$ and a non-computable set C. Consider the notion of forcing whose conditions[18] are 3-tuples (σ_0, σ_1, X) such that for every $i < 2$,

1. (σ_i, X) is a Mathias condition ;
2. For every $x \in X$, $\sigma_i \cup \{x\}$ is f-homogeneous for color i ;
3. $C \not\leq_T X$.

The extension relation is the same as in the proof of Theorem 3.4.6. Given a condition $p = (\sigma_0, \sigma_1, X)$ and two Σ^0_1 formulas $\varphi_0(G)$ and $\varphi_1(G)$, let $p \mathrel{?{\vdash}} \varphi_0(G_0) \vee \varphi_1(G_1)$ iff for every 2-partition $Z_0 \sqcup Z_1 = X$, there is some $i < 2$ and a finite f-homogeneous set $\rho \subseteq Z_i$ for color i such that $\varphi_i(\sigma_i \cup \rho)$ holds.[19][20]

1. Prove that the relation $p \mathrel{?{\vdash}} \varphi_0(G_0) \vee \varphi_1(G_1)$ is $\Sigma^0_1(X)$.
2. Show that it is a forcing question in the sense of Definition 3.4.9.
3. Prove Seetapun's theorem using this notion of forcing. ★

18: One can apply the same trick as in Theorem 3.4.6 to see that one of the initial segments is extendible. Given a condition (σ_0, σ_1, X), apply Ramsey's theorem for pairs to $f{\restriction}[X]^2$ to obtain an infinite f-homogeneous subset $H \subseteq X$ for some color $i < 2$. The properties of the condition are designed to ensure that $\sigma_i \cup H$ is f-homogeneous.

19: Notice the strong similarity of this forcing question with the one in Theorem 3.4.6. The only difference is that one requires ρ to be f-homogeneous as well.

20: If the coloring f is stable, that is, $\lim_y f(x,y)$ always exists, then the interpretation of the 2-partition $Z_0 \sqcup Z_1 = X$ is clear: it is the limit coloring of f. This forcing question might be more confusing in the general case, since f has no limit behavior. This is where compactness comes into play: find a bound to quantify over finite 2-partitions, then "stabilize" the behavior of f over this finite initial segment, by thinning out the remaining reservoir. This limit behavior induces a 2-partition of the initial segment.

It is sometimes useful to think of instances of COH as countably many instances of RT^1_2, where a solution is an infinite set which is simultaneously homogeneous for all instances of RT^1_2, up to finite changes. With this intuition in mind, one can strengthen Theorem 3.4.2 to prove that it holds even when considering arbitrary instances of COH.

Exercise 3.4.13 (Wang [15]). Fix a non-computable set C and an arbitrary countable sequence $\vec{R} = R_0, R_1, \dots$ of sets, with no effectiveness restriction whatsoever. Consider the variant of Mathias forcing, whose conditions[21] are pairs (σ, X) where $C \not\leq_T X$.

1. Use Theorem 3.4.6 to show that the set $\mathcal{D}_n = \{(\sigma, X) : X \subseteq R_n \vee X \subseteq \overline{R_n}\}$ is dense.
2. Deduce the existence of an infinite $\vec{R}$-cohesive set G such that $C \not\leq_T G$. ★

21: Note that contrary to the proof of cone avoidance of COH, one needs to use Mathias conditions (σ, X) where $C \not\leq_T X$ instead of computable Mathias conditions.

Cone avoidance fails when considering computable colorings of 3-tuples. The reason is that one can create computable coloring $f : [\mathbb{N}]^3 \to 2$ such that every infinite homogeneous set H is so sparse, that its principal function p_H is very fast-growing, and dominates the modulus of $\emptyset'$. Recall that the *principal function* p_X of an infinite set $X = \{x_0 < x_1 < \dots\}$ is defined by $p_X(n) = x_n$.

Exercise 3.4.14 (Jockusch [16]).

1. Show that for every function $g : \mathbb{N} \to \mathbb{N}$, there is a g-computable coloring $f : [\mathbb{N}]^2 \to 2$ such that for every infinite f-homogeneous set H, the principal function p_H dominates g.
2. Show that for every $\emptyset'$-computable coloring $f : [\mathbb{N}]^2 \to 2$, there is a computable coloring $h : [\mathbb{N}]^3 \to 2$ such that every infinite h-homogeneous set is f-homogeneous.
3. Deduce the existence of a computable coloring $h : [\mathbb{N}]^3 \to 2$ such that every infinite h-homogeneous set computes $\emptyset'$. ★

One can actually go one step further, and construct a computable coloring $f : [\mathbb{N}]^3 \to 2$ such that every infinite homogeneous set is of PA degree over $\emptyset'$.

Exercise 3.4.15 (Hirschfeldt and Jockusch [17]).
A set $P \subseteq \mathbb{N}$ is *pre-homogeneous* for a coloring $f : [\mathbb{N}]^{n+1} \to 2$ if for every $F \in [P]^n$ and every $x, y \in P$ with $\max F < x, y$, then $f(F \cup \{x\}) = f(F \cup \{y\})$. Construct a computable coloring $f : [\mathbb{N}]^3 \to 2$ such that every infinite pre-homogeneous set is of PA degree over $\emptyset'$. ★

3.5 Preserving definitions

The existence of a notion of forcing with a Σ^0_1-preserving forcing question enables to prove abstractly some stronger weakness properties, such as preservation of one non-Σ^0_1 definition. Some sets such as $\emptyset'$ can be used to “simplify” the definition of other sets in the arithmetic hierarchy. For example, any Σ^0_2 set is $\Sigma^0_1(\emptyset')$. The notion of preservation of 1 non-Σ^0_1-definition reflects the unability of a problem to simplify the description of a non-Σ^0_1 set to make it Σ^0_1 relative to a solution.

Definition 3.5.1. A problem P admits *preservation of 1 non-Σ^0_1 definition* if for every set Z and every non-$\Sigma^0_1(Z)$ set C, every Z-computable instance X of P admits a solution Y such that C is not $\Sigma^0_1(Z \oplus Y)$. ◊

Thanks to Post’s theorem, preservation of 1 non-Σ^0_1 definition implies cone avoidance:

Exercise 3.5.2. Prove that if a problem P admits preservation of 1 non-Σ^0_1 definition, then it admits cone avoidance. ★

The proof of Theorem 3.3.4 can be strengthened to prove an abstract theorem about preservation of 1 non-Σ^0_1 definition.[22]

22: The proof of preservation of non-Σ^0_1 definitions is simpler and arguably more natural than the one of cone avoidance. This naturality comes from the fact that, in some sense, Σ^0_1 sets are more natural than computable ones, as they form a syntactic family and thus have a better behavior.

Theorem 3.5.3
Let $(\mathbb{P}, \leq)$ be a notion of forcing with a Σ^0_1-preserving forcing question. For every non-Σ^0_1 set C and every sufficiently generic filter $\mathcal{F}$, C is not $\Sigma^0_1(G_\mathcal{F})$.

Proof. It suffices to prove the following lemma:

Lemma 3.5.4. For every condition $p \in \mathbb{P}$ and every Turing index e, there is an extension $q \leq p$ forcing $C \neq W_e^G$. ★

Proof. Consider the following set

$$U = \{x \in \mathbb{N} : p \mathrel{?\vdash} x \in W_e^G\}$$

Since the forcing question is Σ^0_1-preserving, the set U is Σ^0_1. There are three cases:

- Case 1: there is some $x \in U \setminus C$. By Property (1) of the forcing question, there is an extension $q \leq p$ forcing $x \in W_e^G$.
- Case 2: there is some $x \in C \setminus U$. By Property (2) of the forcing question, there is an extension $q \leq p$ forcing $x \notin W_e^G$.
- Case 3: $U = C$. Then C is Σ^0_1, contradiction.

In the first two cases, the extension q forces $W_e^G \neq C$. ■

We are now ready to prove Theorem 3.5.3. Given $e \in \mathbb{N}$, let $\mathscr{D}_e$ be the set of all conditions $q \in \mathbb{P}$ forcing $W_e^G \neq C$. It follows from Lemma 3.5.4 that every $\mathscr{D}_e$ is dense, hence every sufficiently generic filter $\mathcal{F}$ is $\{\mathscr{D}_e : e \in \mathbb{N}\}$-generic, so C is not $\Sigma^0_1(G_{\mathcal{F}})$. This completes the proof of Theorem 3.5.3. ■

It follows from Theorem 3.5.3 that the proofs of cone avoidance for Cohen genericity and Π^0_1 classes have a straightforward adaptation to prove preservation of 1 non-Σ^0_1 definition. We leave these adaptations as an exercise:

Exercise 3.5.5. Let C be a non-Σ^0_1 set. Prove that for every sufficiently Cohen generic set G, C is not $\Sigma^0_1(G)$. ★

Exercise 3.5.6. Let C be a non-Σ^0_1 set. Prove that for every non-empty Π^0_1 class $\mathscr{P} \subseteq 2^{\mathbb{N}}$, there is a member $G \in \mathscr{P}$ such that C is not $\Sigma^0_1(G)$. ★

It is natural to wonder whether some problems admit cone avoidance but not preservation of 1 non-Σ^0_1 definition. Actually, this happens not to be the case, thanks to the relativized formulation of both notions.[23]

23: The proof of Exercise 3.5.2 also holds when considering non-relativized versions of cone avoidance of preservation of 1 non-Σ^0_1 definitions. On the other hand, the reverse direction uses a different set Z. One can construct artificial problems which admit non-relativized cone avoidance but not non-relativized preservation of 1 non-definition.

Theorem 3.5.7 (Downey et al. [18])
Let C be a non-Σ^0_1 set. There is a set Z and a set $D \not\leq_T Z$ such that for every set G such that C is $\Sigma^0_1(G \oplus Z)$, $D \leq_T G \oplus Z$.

The proof of Theorem 3.5.7 is quite technical and outside the scope of this book.

Corollary 3.5.8 (Downey et al. [18])
A problem P admits preservation of 1 non-Σ^0_1 definition iff it admits cone avoidance.[24]

24: Given the simplicity of the forward direction, the technicality of the reciprocal, and the naturality of the proof of preservation of 1 non-Σ^0_1 definition using a Σ^0_1-preserving forcing question, it is preferable to directly prove preservation of 1 non-Σ^0_1 definition when the result is needed.

Proof. The forward direction is Exercise 3.5.2. Let us prove reciprocal. Suppose P admits cone avoidance. Fix a set Z and a non-$\Sigma^0_1(Z)$ set C and let $X \leq_T Z$ be an instance of P. By Theorem 3.5.7 relativized to Z, there is a set Z_1 and a set $D \not\leq_T Z \oplus Z_1$ such that for every set G such that C is $\Sigma^0_1(G \oplus Z \oplus Z_1)$, $D \leq_T G \oplus Z \oplus Z_1$. By cone avoidance of P relativized to $Z \oplus Z_1$, there is a solution Y to X such that $D \not\leq_T Y \oplus Z \oplus Z_1$. By choice of Z_1 and D, it follows that C is not $\Sigma^0_1(Y \oplus Z \oplus Z_1)$. In particular, C is not $\Sigma^0_1(Y \oplus Z)$. ■

3.6 Preserving hyperimmunities

There exists a well-known duality between computing sets and computing fast-growing functions. The simplest example is the correspondence between the halting set $\emptyset'$, and the halting time function $\mu_{\emptyset'} : \mathbb{N} \to \mathbb{N}$ which to e associates the smallest time t such that $\Phi_e(e)[t]\downarrow$, if it exists, and equals 0 otherwise. The function μ is $\emptyset'$-computable, and every function dominating $\mu_{\emptyset'}$ computes $\emptyset'$. More generally, a function $f : \mathbb{N} \to \mathbb{N}$ is a *modulus* of a set X if every function dominating f computes X. If furthermore f is X-computable, then it is a *self-modulus*. By Solovay [19], the sets admitting a modulus are exactly the Δ^1_1 sets, or equivalently the hyperarithmetic sets. On the other hand, there exist Δ^0_3 sets with no self-modulus.

Proposition 3.6.1 (Martin and Miller [20]). Every Δ^0_2 set admits a self-modulus. ★

Proof. Let A be a Δ^0_2 set, with Δ^0_2 approximation $A_0, A_1, \dots$ The *computation function* $c_A : \mathbb{N} \to \mathbb{N}$ maps x to the smaller integer $n \geq x$ such that $A_n \restriction_x = A \restriction_x$. Let f be a function dominating c_A. Let $h(x)$ be the largest $y \leq x$ such that for all $x \leq t \leq f(x)$, $A_t \restriction_y = A_{f(x)} \restriction_y$. The function h is total f-computable. Moreover, h tends towards $+\infty$, because the approximation of A being Δ^0_2, it will stabilize on increasingly larger initial segments. Finally, as $x \leq c_A(x) \leq f(x)$, then if $h(x) = y$, $A_x \restriction_y = A_{c_A(x)} \restriction_y = A \restriction_y$. Then, to decide if $n \in A$, it suffices to find an integer x such that $h(x) > n$, then test if $n \in A_x$. This procedure is f-computable. ■

Recall that a function $f : \mathbb{N} \to \mathbb{N}$ is *hyperimmune* if it is not dominated by any computable function. In particular, if a function f is a modulus of a non-computable set C, then it is hyperimmune. Moreover, if it is a self-modulus, then avoiding the cone above C is equivalent to preserving the hyperimmunity of the function f. This motivates the following definition:

> **Definition 3.6.2.** A problem P admits *preservation of 1 hyperimmunity* if for every set Z and every Z-hyperimmune function f, every Z-computable instance X of P admits a solution Y such that f is $Z \oplus Y$-hyperimmune. ◇

At first sight, the sole existence of a Σ^0_1-preserving forcing question does not seem to be sufficient to prove preservation of 1 hyperimmunity. One furthermore needs the forcing question to satisfy some kind of compactness as follows:

> **Definition 3.6.3.** Given a notion of forcing $(\mathbb{P}, \leq)$, a forcing question is Σ^0_n-*compact* if for every $p \in \mathbb{P}$ and every Σ^0_n formula $\varphi(G, x)$, if $p \mathrel{?\vdash} \exists x \varphi(G, x)$ holds, then there is a finite set $F \subseteq \mathbb{N}$ such that $p \mathrel{?\vdash} \exists x \in F\ \varphi(G, x)$. ◇

All the forcing questions seen in this chapter are Σ^0_1-compact. Thanks to this compactness property, one can prove preservation of 1 hyperimmunity.

Theorem 3.6.4
Let $(\mathbb{P}, \leq)$ be a notion of forcing with a Σ^0_1-compact, Σ^0_1-preserving forcing question. For every hyperimmune function $f : \mathbb{N} \to \mathbb{N}$ and every sufficiently generic filter $\mathcal{F}$, f is $G_{\mathcal{F}}$-hyperimmune.

Proof. It suffices to prove the following lemma:

Lemma 3.6.5. For every condition $p \in \mathbb{P}$ and every Turing index e, there is an extension $q \leq p$ forcing Φ_e^G not to dominate f.[25] ★

25: By this, we mean forcing either Φ_e^G to be partial, or $\Phi_e^G(x) < f(x)$ for some $x \in \mathbb{N}$.

Proof. Suppose first that $p \mathrel{?}\nVdash \exists v \Phi_e^G(x) \downarrow = v$ for some $x \in \mathbb{N}$. Then by Property (2) of the forcing question, there is an extension $q \leq p$ forcing $\Phi_e^G(x)\uparrow$, and we are done. Suppose now that for every $x \in \mathbb{N}$, $p \mathrel{?}\vdash \exists v \Phi_e^G(x) \downarrow = v$. By Σ_1^0-compactness of the forcing question, for every $x \in \mathbb{N}$, there is a finite set $F_x \subseteq \mathbb{N}$ such that $p \mathrel{?}\vdash \exists v \in F_x\ \Phi_e^G(x)\downarrow = v$. Let $h : \mathbb{N} \to \mathbb{N}$ be the function which on input x, looks for some finite set F_x such that $p \mathrel{?}\vdash \exists v \in F_x\ \Phi_e^G(x)\downarrow = v$ and outputs $\max F_x$. Such a function is total by hypothesis, and computable by Σ_1^0-preservation of the forcing question. Since f is hyperimmune, $h(x) < f(x)$ for some $x \in \mathbb{N}$. By Property (1) of the forcing question, there is an extension $q \leq p$ forcing $\exists v \in F_x \Phi_e^G(x)\downarrow = v$. Since $f(x) > \max F_x$, q forces $\Phi_e^G(x)\downarrow < f(x)$. ■

We are now ready to prove Theorem 3.6.4. Given $e \in \mathbb{N}$, let $\mathcal{D}_e$ be the set of all conditions $q \in \mathbb{P}$ forcing Φ_e^G not to dominate f. It follows from Lemma 3.5.4 that every $\mathcal{D}_e$ is dense, hence every sufficiently generic filter $\mathcal{F}$ is $\{\mathcal{D}_e : e \in \mathbb{N}\}$-generic, so f is $G_{\mathcal{F}}$-hyperimmune. This completes the proof of Theorem 3.6.4. ■

Contrary to preservation of 1 non-Σ_1^0 definition, there is no immediate link between preservation of 1 hyperimmunity and cone avoidance. Furthermore, preservation of 1 hyperimmunity seems to require an extra property which may not always be satisfied. However, the two notions turn out again to be equivalent in their relativized form. Recall Theorem 3.2.4 which informally says that every set can become Δ_2^0 while avoiding a cone.

Theorem 3.6.6 (Downey et al. [18])
If a problem P *admits preservation of 1 hyperimmunity, then it admits cone avoidance.*

Proof. Fix a set Z, a set $C \not\leq_T Z$ and an instance $X \leq_T Z$ of P. By Theorem 3.2.4, there is a set Z_1 such that $C \not\leq_T Z \oplus Z_1$ and $C \leq_T (Z \oplus Z_1)'$. By Proposition 3.6.1 relative to $Z \oplus Z_1$, there is a $C \oplus Z \oplus Z_1$-computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every function g dominating f, $C \leq_T g \oplus Z \oplus Z_1$. In particular, f is $Z \oplus Z_1$-hyperimmune. Since P admits preservation of 1 hyperimmunity, there is a solution Y to X such that f is $Y \oplus Z \oplus Z_1$-hyperimmune. It follows that $C \not\leq_T Y \oplus Z \oplus Z_1$. ■

The reverse direction also holds, using the following theorem which says that every non-decreasing hyperimmune function is a modulus of some set in a relativized setting.

Theorem 3.6.7 (Downey et al. [18])
Fix a non-decreasing hyperimmune function $f : \mathbb{N} \to \mathbb{N}$. There is a set Z and a set $C \not\leq_T Z \oplus G$ such that f is a Z-modulus for C.

Here again, the proof of Theorem 3.6.7 is out of the scope of this book.

Corollary 3.6.8 (Downey et al. [18])
A problem P *admits preservation of 1 hyperimmunity iff it admits cone avoidance.*

PROOF. The forward direction is Theorem 3.6.6. Let us prove reciprocal. Suppose P admits cone avoidance. Fix a set Z, a Z-hyperimmune function $f : \mathbb{N} \to \mathbb{N}$, and let $X \leq_T Z$ be an instance of P. By Theorem 3.6.7 relativized to Z, there is a set Z_1 and a set $C \not\leq_T Z \oplus Z_1$ such that f is a Z-modulus for C. By cone avoidance of P relativized to $Z \oplus Z_1$, there is a solution Y to X such that $C \not\leq_T Y \oplus Z \oplus Z_1$. By choice of Z_1 and C, it follows that f is $Y \oplus Z \oplus Z_1$-hyperimmune. In particular, f is not $Y \oplus Z$-hyperimmune. ■