Higher jump cone avoidance 11

The conceptual gap from second to iterated jump control is not as significant as from first to second jump control. Indeed, the main difficulty comes from dealing with non-continuous functionals, which already occurs at the Σ^0_2 level. There is therefore often a natural generalization from second to all the levels of the arithmetic hierarchy.

New difficulties arise when trying to control the jump at transfinite levels. The arithmetic hierarchy extends to the hyperarithmetic hierarchy through iterations along computable ordinals. While the arithmetic hierarchy is indexed by integers, which are left unchanged when considering relativization to a generic set, the hyperarithmetic hierarchy is indexed by computable ordinals, which is a relative notion: the generic set might compute more ordinals, and therefore might have more levels in its relative hyperarithmetic hierarchy.

Prerequisites: Chapters 2, 3 and 9

11.1 Context and motivation

The study of iterated jump control at the arithmetic and hyperarithmetic levels has two different motivations, both coming from reverse mathematics.

Arithmetic jump control. At the arithmetic level, arithmetic jump control is an essential tool in the study of Ramsey-type hierarchies. Consider for instance the rainbow Ramsey theorem, which is a particular case of the canonical Ramsey theorem of Erdős and Rado.

> **Definition 11.1.1.** A coloring $f : [\mathbb{N}]^n \to \mathbb{N}$ is *k-bounded* if each color appears at most k times, that is, $|f^{-1}(c)| \leq k$ for every $c \in \mathbb{N}$. A set $H \subseteq \mathbb{N}$ is an *f-rainbow* if f is injective on $[H]^n$. The *rainbow Ramsey theorem* for n-tuples and k-bounds (RRT^n_k) states that every k-bounded coloring $f : [\mathbb{N}]^n \to \mathbb{N}$ admits an infinite f-rainbow. ◊

As for Ramsey's theorem, the rainbow Ramsey theorem forms a hierarchy of statements based on the size n of the tuples. However, while RT^n_2 collapses and is equivalent to ACA_0 for $n \geq 3$, Wang [15] proved that RRT^n_2 is strictly weaker than ACA_0 for every $n \geq 1$. Whether or not the rainbow Ramsey hierarchy is strict remains open.

Csima and Mileti [80] proved that every computable instance of RRT^n_2 admits a Π^0_n solution, while there exists a computable instance of RRT^n_2 with no Σ^0_n solution. The most promising approach to separate RRT^n_2 from RRT^{n+1}_2 is using the natural invariant lying at the Δ^0_n level of the arithmetic hierarchy, namely, low$_n$ness. By Cholak, Jockusch and Slaman [27] and Wang [89], every computable instance of RRT^n_2 admits a low$_n$ solution for $n \in \{2, 3\}$. The general case is likely to be solved using arithmetic jump control.

Hyperarithmetic jump control. The duality between computability and definability is omnipresent in reverse mathematics. The base theory, RCA_0, captures "computable mathematics", and its ω-models admit a nice characterization in terms of Turing ideals. The systems WKL_0 and ACA_0 also admit computability-theoretic formulations, in terms of existence of PA degrees and of the halting set, respectively. On the other hand, the two highest systems of the Big Five, namely, ATR_0 and Π^1_1-CA_0, are better explained in terms of higher recursion theory, stating the existence of every transfinite iterations of the halting set, and the existence of Kleene's $\mathcal{O}$, respectively. Given the importance of arithmetic jump control in the study of the lower systems of reverse mathematics, one can reasonably guess that hyperarithmetic jump control will play some role in the study of principles at the level of ATR_0 and Π^1_1-CA_0.

11.2 First examples

As mentioned, there exists a natural generalization from second jump to arithmetic jump control, using inductive definitions. We illustrate this using Cohen forcing.

Theorem 11.2.1 (Feferman [90])
Fix $n \geq 1$ and let C be a non-Δ^0_n set. For every sufficiently Cohen generic filter $\mathcal{F}$, C is not $\Delta^0_n(G_\mathcal{F})$.

Proof. In order to prove our theorem, we need to define a Σ^0_n-preserving forcing question for Σ^0_n-formulas.

Definition 11.2.2. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition and $\varphi(G) \equiv \exists x \psi(G, x)$ be a Σ^0_n formula for $n \geq 1$.

1. For $n = 1$, let $\sigma \operatorname{?\vdash} \varphi(G)$ hold if there is some $x \in \mathbb{N}$ and some $\tau \geq \sigma$ such that $\psi(\tau, x)$ holds.
2. For $n > 1$, let $\sigma \operatorname{?\vdash} \varphi(G)$ hold if there is some $x \in \mathbb{N}$ and some $\tau \geq \sigma$ such that $\tau \operatorname{?\vdash} \psi(G, x)$.[1] ◊

1: Here, ψ is a Π^0_{n-1}-formula. The notation $\tau \operatorname{?\vdash} \psi(G, x)$ is therefore a shorthand for $\tau \operatorname{?\nvdash} \neg\psi(G, x)$, that is, the forcing question for Π^0_{n-1}-formulas induced by taking the negation of the forcing question for Σ^0_{n-1}-formulas.

A simple induction on the structure of the formulas shows that given a Σ^0_n-formula $\varphi(G)$, the relation $\sigma \operatorname{?\vdash} \varphi(G)$ is Σ^0_n uniformly in its parameters. The following lemma shows that the definition of the forcing question meets a strong version of its specifications.

Lemma 11.2.3. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition and $\varphi(G)$ be a Σ^0_n formula for $n \geq 1$.

1. If $\sigma \operatorname{?\vdash} \varphi(G)$, then there is an extension $\tau \geq \sigma$ forcing $\varphi(G)$.
2. If $\sigma \operatorname{?\nvdash} \varphi(G)$, then σ forces $\neg\varphi(G)$.[2] ⋆

2: This property states that the forcing question for Σ^0_n-formulas is Π^0_n-extremal (see Definition 7.6.5). It follows that sufficiently Cohen generic sets preserve many computational properties.

Proof. We prove simultaneously both items inductively on the structure of the formula $\varphi(G)$. Say $\varphi(G) \equiv \exists \psi(G, x)$ where $\psi(G, x)$ is Π^0_{n-1}.

Base case: $n = 1$.[3] If $\sigma \operatorname{?\vdash} \varphi(G)$, then, letting $\tau \geq \sigma$ and $x \in \mathbb{N}$ witness the definition, for every filter $\mathcal{F}$ containing τ, $G_\mathcal{F} \geq \tau$, hence $\psi(G_\mathcal{F}, x)$ holds, so $\varphi(G_\mathcal{F})$ holds. It follows that τ is an extension of σ forcing $\varphi(G)$. Conversely, if σ does not force $\neg\varphi(G)$, then there is a filter $\mathcal{F}$ containing σ such that $\varphi(G_\mathcal{F})$ holds. Then, by the use property, there is a finite $\tau \prec G_\mathcal{F}$ and some $x \in \mathbb{N}$ such that $\psi(\tau, x)$ holds. Since $\sigma \prec G_\mathcal{F}$, by taking τ long enough, one has $\sigma \prec \tau$, thus $\sigma \operatorname{?\vdash} \varphi(G)$.

3: The base case is a solution to Exercise 3.3.6.

Inductive case: $n > 1$. If $\sigma \operatorname{?\vdash} \varphi(G)$, then there is some $x \in \mathbb{N}$ and some $\tau \geq \sigma$ such that $\tau \operatorname{?\vdash} \psi(G, x)$. By induction hypothesis, there is some $\rho \geq \tau$ forcing $\psi(G, x)$. In particular, ρ is an extension of σ forcing $\varphi(G)$. If $\sigma \operatorname{?\nvdash} \varphi(G)$, then for every $x \in \mathbb{N}$ and every $\tau \geq \sigma$, $\tau \operatorname{?\nvdash} \psi(G, x)$. By induction hypothesis, for every $x \in \mathbb{N}$ and every $\tau \geq \sigma$, there is some $\rho \geq \tau$ forcing $\neg\psi(G, x)$. In other words, for every $x \in \mathbb{N}$, the set of all ρ forcing $\neg\psi(G, x)$ is dense below σ. Thus, for every sufficiently generic filter $\mathcal{F}$ containing σ and for every $x \in \mathbb{N}$, there is some $\rho \in \mathcal{F}$ forcing $\neg\psi(G, x)$, hence $\forall x \neg\psi(G_{\mathcal{F}}, x)$ holds. In other words, σ forces $\neg\varphi(G)$. ■

The following diagonalization lemma is a straightforward generalization of Lemma 3.2.2.

Lemma 11.2.4. For every Cohen condition $\sigma \in 2^{<\mathbb{N}}$ and every Turing index e, there is an extension $\tau \geq \sigma$ forcing $\Phi_e^{G^{(n-1)}} \neq C$. ⋆

Proof. Consider the following set[4]

$$U = \{(x, v) \in \mathbb{N} \times 2 : \sigma \operatorname{?\vdash} \Phi_e^{G^{(n-1)}}(x)\downarrow = v\}$$

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

- Case 1: $(x, 1 - C(x)) \in U$ for some $x \in \mathbb{N}$. By Lemma 11.2.3(1), there is an extension $\tau \geq \sigma$ forcing $\Phi_e^{G^{(n-1)}}(x)\downarrow = 1 - C(x)$.
- Case 2: $(x, C(x)) \notin U$ for some $x \in \mathbb{N}$. By Lemma 11.2.3(2), there is an extension $\tau \geq \sigma$ forcing $\Phi_e^{G^{(n-1)}}(x)\uparrow$ or $\Phi_e^{G^{(n-1)}}(x)\downarrow \neq C(x)$.
- Case 3: None of Case 1 and Case 2 holds. Then U is a Σ_n^0 graph of the characteristic function of C, hence C is Δ_n^0. This contradicts our hypothesis. ■

4: By Post's theorem, the following property is Σ_n^0, although the translation is not straightforward:

$$\Phi_e^{G^{(n-1)}}(x)\downarrow = v$$

We are now ready to prove Theorem 11.2.1. Let $\mathcal{F}$ be a sufficiently generic filter for Cohen forcing, and let $G_{\mathcal{F}} = \bigcup \mathcal{F}$. By genericity of $\mathcal{F}$, $G_{\mathcal{F}}$ is an infinite binary sequence, and by Lemma 11.2.4, $C \not\leq_T G_{\mathcal{F}}^{(n-1)}$, in other words C is not $\Delta_n^0(G)$. This completes the proof of Theorem 11.2.1. ■

Exercise 11.2.5. Let $(\mathbb{P}, \leq)$ be a notion of forcing with a Σ_n^0-preserving forcing question. Show that for every non-Δ_n^0 set C and every sufficiently generic filter $\mathcal{F}$, C is not $\Delta_n^0(G_{\mathcal{F}})$. ⋆

Exercise 11.2.6 (Wang [82]). Let $(\mathbb{P}, \leq)$ be the primitive recursive Jockusch-Soare forcing, that is, $\mathbb{P}$ is the set of all infinite primitive recursive binary trees $T \subseteq 2^{<\mathbb{N}}$, partially ordered by inclusion.

1. Adapt the proof of Theorem 9.4.1 to design a Σ_n^0-preserving forcing question for Σ_n^0-formulas.
2. Deduce that for every non-Δ_n^0 set C and every sufficiently generic $\mathbb{P}$-filter $\mathcal{F}$, C is not $\Delta_n^0(G_{\mathcal{F}})$. ⋆

11.3 Pigeonhole principle

Although the conceptual gap from second-jump to higher jump control is much smaller than from first to second-jump control, the generalization sometimes requires some non-trivial adaptation. The pigeonhole principle is a good example of a statement with a reasonably simple first-jump control (Theorem 3.4.6), with a second-jump control requiring the development of a whole new machinery (Theorem 9.7.1), and whose generalization to higher jump control still contains some subtleties.[5]

5: In order to understand this section, it is mandatory to be completely familiar with the material of Chapter 9.

Theorem 11.3.1 (Monin and Patey [31])
Fix $n \geq 1$ and let C be a non-Δ^0_n set. For every set A, there is an infinite subset $H \subseteq A$ or $H \subseteq \overline{A}$ such that C is not $\Delta^0_n(H)$.

Proof. The case $n = 1$ is Theorem 3.4.6 and the case $n = 2$ is Theorem 9.7.1. We therefore assume that $n \geq 3$, although one could prove all cases simultaneously with more case analysis within the definitions and the proof. Fix C and A. As in the previous cases, we shall construct two sets $G_0 \subseteq A$ and $G_1 \subseteq \overline{A}$ using a disjunctive notion of forcing. For simplicity, let $A_0 = A$ and $A_1 = \overline{A}$.

Hierarchy of Scott ideals. By multiple applications of the low basis theorem (Theorem 4.4.6) and Theorem 4.3.2, there exists a sequence of sets $M_0, \dots, M_{n-2}$ such that for every $s < n-1$,

1. M_s is of low degree over $\emptyset^{(s)}$;
2. M_s is a code for a Scott ideal $\mathcal{M}_s$ containing $\emptyset^{(s)}$.

By the cone avoidance basis theorem (Theorem 3.2.6) relativized to $\emptyset^{(n-1)}$ and Theorem 4.3.2, there is a code M_{n-1} for a Scott ideal $\mathcal{M}_{n-1}$ containing $\emptyset^{(n-1)}$ such that $C \not\leq_T M_{n-1}$. Note that for every $s < n-1$, $M'_s \in \mathcal{M}_{s+1}$.

Hierarchy of partition regular classes. We construct a sequence $D_0, \dots, D_{n-2}$ such that for every $s < n-1$,

1. $\mathcal{U}^{\mathcal{M}_s}_{D_s}$ is an $\mathcal{M}_s$-cohesive large class;
2. $\mathcal{U}^{\mathcal{M}_{s+1}}_{D_{s+1}} \subseteq \langle \mathcal{U}^{\mathcal{M}_s}_{D_s} \rangle$ if $s < n-2$.

First, by Proposition 9.6.25, $\mathcal{M}_1$ contains a set $D_0 \subseteq \mathbb{N}^2$ such that $\mathcal{U}^{\mathcal{M}_0}_{D_0}$ is an $\mathcal{M}_0$-cohesive class. Suppose D_s is defined and belongs to $\mathcal{M}_{s+1}$, with $s < n-2$. By Proposition 9.6.19, there is an $(M'_s \oplus D_s)'$-computable set $E_s \supseteq D_s$ such that $\mathcal{U}^{\mathcal{M}_s}_{E_s}$ is $\mathcal{M}_s$-minimal.[6] In particular, E_s is M'_{s+1}-computable, so $E_s \in \mathcal{M}_{s+2}$. Furthermore, since $M_s \in \mathcal{M}_{s+1}$ and M_{s+1} is a Scott code, there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every $e \in \mathbb{N}$, $f(e)$ is an M_{s+1}-code and e is an M_s-code of the same set. Let $F_{s+1} = \{(a, f(e)) : (a, e) \in E_s\}$. Then $\mathcal{U}^{\mathcal{M}_{s+1}}_{F_{s+1}} = \mathcal{U}^{\mathcal{M}_s}_{E_s}$ and $F_{s+1} \in \mathcal{M}_{s+2}$. By Proposition 9.6.25, $\mathcal{M}_{s+2}$ contains a set $D_{s+1} \supseteq F_{s+1}$ such that $\mathcal{U}^{\mathcal{M}_{s+1}}_{D_{s+1}}$ is $\mathcal{M}_{s+1}$-cohesive. In particular,

6: Note that $\mathcal{U}^{\mathcal{M}_s}_{E_s} = \langle \mathcal{U}^{\mathcal{M}_s}_{D_s} \rangle$ by Lemma 9.6.24 and by $\mathcal{M}_s$-cohesiveness of the class $\mathcal{U}^{\mathcal{M}_s}_{D_s}$.

$$\mathcal{U}^{\mathcal{M}_{s+1}}_{D_{s+1}} \subseteq \mathcal{U}^{\mathcal{M}_{s+1}}_{F_{s+1}} = \mathcal{U}^{\mathcal{M}_s}_{E_s} = \langle \mathcal{U}^{\mathcal{M}_s}_{D_s} \rangle$$

Notion of forcing. The notion of forcing is a variant of Mathias forcing whose conditions are triples (σ_0, σ_1, X), where[7]

7: This notion of forcing is very similar to the one of Theorem 9.7.1, with $\mathcal{M}_{n-1}$ playing the role of the ideal $\mathcal{N}$.

1. (σ_i, X) is a Mathias condition for each $i < 2$;
2. $\sigma_i \subseteq A_i$; $X \in \langle \mathcal{U}^{\mathcal{M}_{n-2}}_{D_{n-2}} \rangle$;
3. $X \in \mathcal{M}_{n-1}$.

The interpretation $[\sigma_0, \sigma_1, X]$ of a condition (σ_0, σ_1, X), the notion of extension, the definition of a valid part of a condition are exactly the same as in Theorem 9.7.1. The following lemma also holds, with the same proof as Lemma 9.7.3. Therefore, for every sufficiently generic filter $\mathcal{F}$ with valid part i, $G_{\mathcal{F},i}$ is infinite and belongs to $\langle \mathcal{U}^{\mathcal{M}_{n-2}}_{D_{n-2}} \rangle$.

Lemma 11.3.2. Let $p = (\sigma_0, \sigma_1, X)$ be a condition with valid part i and let $\mathcal{V} \supseteq \langle \mathcal{U}_{D_{n-2}}^{\mathcal{M}_{n-2}} \rangle$ be a large $\Sigma_1^0(\mathcal{M}_{n-2})$ class. There is an extension (τ_0, τ_1, Y) of p such that $[\tau_i] \subseteq \mathcal{V}$. ★

Forcing question at lower levels. In the proof of Theorem 9.7.1, we defined a non-disjunctive $\Pi_2^0(\mathcal{N})$ forcing question for Σ_1^0-formulas and a disjunctive $\Sigma_1^0(\mathcal{N})$ forcing question for Σ_2^0-formulas. The generalization to Theorem 11.3.1 goes as follows: the non-disjunctive forcing question will be extended to every Σ_s^0-formula, for $s \in \{1, \dots, n-1\}$, yielding a $\Pi_1^0(\mathcal{M}_s)$ forcing question for Σ_s^0-formulas, and one will keep the same disjunctive $\Sigma_1^0(\mathcal{M}_{n-1})$ forcing question for Σ_n^0-formulas.

Definition 11.3.3. 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 the following class is large:[8]

$$\mathcal{U}_{D_0}^{\mathcal{M}_0} \cap \{Z : \exists \rho \subseteq Z \ \varphi(\sigma \cup \rho)\}$$

Given a string $\sigma \in 2^{<\mathbb{N}}$ and a Σ_s^0-formula $\varphi(G) \equiv \exists x \psi(G, x)$ for $s \in \{2, \dots, n-1\}$, define $\sigma \mathrel{?\vdash} \varphi(G)$ to hold if the following class is large:[9]

$$\mathcal{U}_{D_{s-1}}^{\mathcal{M}_{s-1}} \cap \{Z : \exists \rho \subseteq Z \ \exists x \ \sigma \cup \rho \mathrel{?\vdash} \psi(G, x)\}$$

8: Note that for Σ_s^0-formulas, we consider largeness with respect to $\mathcal{U}_{D_{s-1}}^{\mathcal{M}_{s-1}}$. The advantage is that it yields a better definitional complexity than using $\mathcal{U}_{D_{n-1}}^{\mathcal{M}_{n-1}}$, but it requires to have some compatibility between $\mathcal{U}_{D_{s-1}}^{\mathcal{M}_{s-1}}$ and $\mathcal{U}_{D_{n-1}}^{\mathcal{M}_{n-1}}$. This was the purpose of the construction of $D_0, \dots, D_{n-2}$.

9: As usual, ψ is Π_{s-1}^0, so $\sigma \cup \rho \mathrel{?\vdash} \psi(G, x)$ is a shorthand for $\sigma \cup \rho \mathrel{?\nvdash} \neg\psi(G, x)$.

By induction over the complexity of the formulas and using Lemma 9.6.15, one can prove that for Σ_s^0-formulas, the relation $\sigma \mathrel{?\vdash} \varphi(G)$ is $\Pi_1^0(D_{s-1} \oplus M'_{s-1})$ uniformly in σ and φ. Since $M'_{s-1}, D_{s-1} \in \mathcal{M}_s$, the relation is $\Pi_1^0(\mathcal{M}_s)$. Before proving the validity of Definition 11.3.3, one first needs to focus on the forcing relation for Π_s^0-formulas, for $s \in \{2, \dots, n\}$. Recall that in the proof of Theorem 9.7.1, we defined a custom syntactic forcing relation for Π_2^0-formulas, implying the semantic forcing relation only on the valid parts. It becomes more convenient to define a syntactic relation at every level, both for Σ_s^0 and Π_s^0-formulas.

Definition 11.3.4. Let $p = (\sigma_0, \sigma_1, X)$ be a condition and $i < 2$ be a part. We define the relation $\Vdash$ for Σ_s^0 and Π_s^0-formulas for $s \in \{1, \dots, n\}$ inductively as follows. For a Δ_0^0-formula $\psi(G, x)$,

1. $p \Vdash \exists x \psi(G_i, x)$ if $\psi(\sigma_i, x)$ holds for some $i < 2$;
2. $p \Vdash \forall x \neg\psi(G_i, x)$ if $(\forall \rho \subseteq X)(\forall x)\neg\psi(\sigma_i \cup \rho, x)$.

For a Π_{s-1}^0-formula $\psi(G, x)$ with $s \in \{2, \dots, n\}$

1. $p \Vdash \exists x \psi(G_i, x)$ if $p \Vdash \psi(G_i, x)$ for some $x \in \mathbb{N}$;
2. $p \Vdash \forall x \neg\psi(G_i, x)$ if $(\forall \rho \subseteq X)(\forall x)\sigma_i \cup \rho \mathrel{?\vdash} \neg\psi(G_i, x)$. ◊

The first property that one expects of a forcing relation is that it is stable under condition extension. This is left as an exercise.

Exercise 11.3.5. Let p and q be two conditions, and $i < 2$. Show that for every $s \in \{1, \dots, n\}$ and every Σ_s^0 and Π_s^0-formula $\varphi(G)$, if $p \Vdash \varphi(G_i)$ and $q \le p$, then $q \Vdash \varphi(G_i)$.[10] ★

10: Note that the closure under extension of the syntactic question also holds if the side is not valid.

There is an interplay between the syntactic forcing relation and the forcing questions. Indeed, the proof that the syntactic forcing relation for Π_s^0-formulas implies the semantic ones uses the validity of the forcing question for lower levels, while the proof of validity of the forcing question involves the syntactic forcing relation at the same level. We therefore start with the proof of validity of Definition 11.3.3, which is a straightforward generalization of Lemma 9.7.5 and is left as an exercise.

Exercise 11.3.6. Let $p = (\sigma_0, \sigma_1, X)$ be a condition with valid part i and $\varphi(G)$ be a Σ^0_s-formula for $s \in \{1, \dots, n-1\}$. Prove that

1. if $\sigma_i \mathrel{?{\vdash}} \varphi(G)$, then there is an extension q of p such that $q \Vdash \varphi(G_i)$;
2. if $\sigma_i \mathrel{?{\nvdash}} \varphi(G)$, then there is an extension q of p such that $q \Vdash \neg\varphi(G_i)$. ⋆

The following trivial lemma shows that if a Π^0_s-formula is syntactically forced on a valid part, then progress can be made on forcing the Π^0_s-formula.

Lemma 11.3.7. Let $p = (\sigma_0, \sigma_1, X)$ be a condition with valid part i and $\varphi(G) \equiv \forall x \psi(G, x)$ be a Π^0_s-formula for some $s \in \{2, \dots, n\}$. If $p \Vdash \varphi(G_i)$, then for every $x \in \mathbb{N}$, there is an extension $q \leq p$ such that $q \Vdash \psi(G_i, x)$. ⋆

Proof. Fix $x \in \mathbb{N}$. Since $p \Vdash \varphi(G_i)$, then in particular, for $\rho = \emptyset$, $\sigma_i \mathrel{?{\vdash}} \psi(G, x)$. By Exercise 11.3.6, there is an extension q of p such that $q \Vdash \psi(G_i, x)$. ■

We are now ready to prove that the syntactic forcing relation implies the semantic one on valid sides.

Lemma 11.3.8. Let p be a condition, $i < 2$ be a side and $\varphi(G)$ be a Σ^0_s or Π^0_s-formula for some $s \in \{1, \dots, n\}$. If $p \Vdash \varphi(G_i)$, then $\varphi(G_{\mathcal{F},i})$ holds for every sufficiently generic filter $\mathcal{F}$ containing p and whose side i is valid.[11] ⋆

11: Recall that a side $i < 2$ is *valid* in a filter $\mathcal{F}$ if the side is valid for every $p \in \mathcal{F}$. Every filter has at least a valid side.

Proof. By induction over the complexity of the formula φ. The case $s = 1$ is easy and $\varphi(G_{\mathcal{F},i})$ even holds for every filter $\mathcal{F}$ containing p, with no regard to genericity or to validity of the side. Suppose $s \geq 2$. If $\varphi(G) \equiv \exists x \psi(G, x)$ for some Π^0_{s-1}-formula ψ, then by definition, there is some $x \in \mathbb{N}$ such that $p \Vdash \psi(G_i, x)$, so by induction hypothesis, $\psi(G_{\mathcal{F},i}, x)$ holds for every sufficiently generic filter $\mathcal{F}$ containing p and whose side i is valid. In particular, $\varphi(G_{\mathcal{F},i})$ holds for every such filter $\mathcal{F}$. If $\varphi(G) \equiv \forall x \neg\psi(G, x)$ for some Π^0_{s-1}-formula ψ, then we claim that for every $x \in \mathbb{N}$, the following class $\mathcal{D}_x$ is dense below p:

$$\mathcal{D}_x = \{q : \text{ side } i \text{ of } q \text{ is not valid } \vee q \Vdash \neg\psi(G_i, x)\}$$

Indeed, fix $x \in \mathbb{N}$ and let $r = (\tau_0, \tau_1, Y)$ be an extension of p. If side i of r is not valid, then $r \in \mathcal{D}_x$, in which case we are done. Otherwise, by Exercise 11.3.5, $r \Vdash \varphi(G_i)$, so, unfolding the definition, for $\rho = \emptyset$, $\tau_i \mathrel{?{\vdash}} \neg\psi(G_i, x)$, so by Exercise 11.3.6, there is an extension $q \leq r$ such that $q \Vdash \neg\psi(G_i, x)$, in which case $q \in \mathcal{D}_x$. Thus, $\mathcal{D}_x$ is dense below p.

Let $\mathcal{F}$ be a sufficiently generic filter containing p and whose side i is valid. Since $\mathcal{D}_x$ is dense below p for every $x \in \mathbb{N}$, $\mathcal{F} \cap \mathcal{D}_x \neq \emptyset$ for every $x \in \mathbb{N}$. Moreover, since side i is valid in $\mathcal{F}$, then for $q \in \mathcal{F} \cap \mathcal{D}_x$, we have $q \Vdash \neg\psi(G_i, x)$. By induction hypothesis, $\neg\psi(G_{\mathcal{F},i}, x)$ holds, and this for every $x \in \mathbb{N}$, so $\varphi(G_{\mathcal{F},i}, x)$ holds. ■

Forcing question on top level. The design of the forcing question for Σ^0_n formulas is exactly the one of Theorem 9.7.1. It consists of defining two forcing questions: a disjunctive one which works if both sides of the condition are valid, and in case one side is invalid, one designs a degenerate non-disjunctive forcing question exploiting the failure of validity. We define both forcing questions and leave their proofs as exercises.

> **Definition 11.3.9.** Given a condition $p = (\sigma_0, \sigma_1, X)$ and a pair of Σ^0_n formulas $\varphi_0(G)$ and $\varphi_1(G)$, with $\varphi_i(G) \equiv \exists x \psi_i(G, x)$, define $p \mathrel{?}\!\vdash \varphi_0(G_0) \vee \varphi_1(G_1)$ to hold if for every 2-partition $Z_0 \cup Z_1 = X$, there is some $i < 2$, some $x \in \mathbb{N}$ and some $\rho \subseteq Z_i$ such that $\sigma_i \cup \rho \mathrel{?}\!\vdash \psi_i(G, x)$. $\diamond$

Exercise 11.3.10. Let $p = (\sigma_0, \sigma_1, X)$ be a condition with both valid parts and $\varphi_0(G), \varphi_1(G)$ be two Σ^0_n-formulas. Prove that

1. if $p \mathrel{?}\!\vdash \varphi_0(G_0) \vee \varphi_1(G_1)$, then there is an extension q of p such that $q \Vdash \varphi(G_i)$ for some $i < 2$;
2. if $p \mathrel{?}\!\nvdash \varphi_0(G_0) \vee \varphi_1(G_1)$, then there is an extension q of p such that $q \Vdash \neg\varphi(G_i)$ for some $i < 2$. ★

A *witness of invalidity* of part i of a condition $p = (\sigma_0, \sigma_1, X)$ is a $\Sigma^0_1(\mathcal{M}_{n-2})$ large class $\mathcal{V} \supseteq \langle \mathcal{U}^{\mathcal{M}_{n-2}}_{D_{n-2}} \rangle$ such that $X \cap A_i \notin \mathcal{V}$.

> **Definition 11.3.11.** Let $p = (\sigma_0, \sigma_1, X)$ be a condition with witness of invalidity $\mathcal{V}$ on part $1 - i$, and let $\varphi(G) \equiv \exists x \psi(G, x)$ be a Σ^0_n formula. Define $p \mathrel{?}\!\vdash^{\mathcal{V}} \varphi(G_i)$ to hold if for every 2-partition $Z_0 \sqcup Z_1 = X$ such that $Z_{1-i} \notin \mathcal{V}$, there is some $x \in \mathbb{N}$ and some $\rho \subseteq Z_i$ such that $\sigma_i \cup \rho \mathrel{?}\!\vdash \psi_i(G, x)$. $\diamond$

Exercise 11.3.12. Let $p = (\sigma_0, \sigma_1, X)$ be a condition with witness of invalidity $\mathcal{V}$ on part $1 - i$, and let $\varphi(G)$ be a Σ^0_n formula. Prove that

1. if $p \mathrel{?}\!\vdash^{\mathcal{V}} \varphi(G_i)$, then there is an extension of p forcing $\varphi(G_i)$;
2. if $p \mathrel{?}\!\nvdash^{\mathcal{V}} \varphi(G_i)$, then there is an extension $q \leq p$ such that $q \Vdash \neg\varphi(G_i)$. ★

By compactness, both forcing questions for Σ^0_n-formulas are $\Sigma^0_1(\mathcal{M}_{n-1})$. We are now ready to prove Theorem 11.3.1.

Suppose first there is a condition p with some invalid part $1 - i$. Let $\mathcal{F}$ be a sufficiently generic filter containing p and let $G_i = G_{\mathcal{F},i}$. Then part i is valid in $\mathcal{F}$. By Lemma 11.3.7, the syntactic forcing relation implies the semantic forcing relation on part i. By Exercise 11.3.12 and by adapting Theorem 9.3.5, for every Turing functional Φ_e, there is some condition $q \in \mathcal{F}$ forcing $\Phi_e^{G_i^{(n-1)}} \neq C$, so C is not $\Delta^0_n(G_i)$.

Suppose now that for every condition, both parts are valid. Let $\mathcal{F}$ be a sufficiently generic filter, and let $G_i = G_{\mathcal{F},i}$ for $i < 2$. By Lemma 11.3.7, the syntactic forcing relation implies the semantic forcing relation on both parts. By Exercise 11.3.10 and by adapting Exercise 11.2.5, for every pair of Turing functionals Φ_{e_0}, Φ_{e_1}, there is some condition $q \in \mathcal{F}$ forcing $\Phi_{e_0}^{G_0^{(n-1)}} \neq C \vee \Phi_{e_1}^{G_1^{(n-1)}} \neq C$. By a pairing argument, there is some $i < 2$ such that C is not $\Delta^0_n(G_i)$. This completes the proof of Theorem 11.3.1. ∎

11.4 Computable ordinals

In order to extend iterated jump control to transfinite levels, one first needs to develop a theory of computable ordinals. There are often two approaches to define a mathematical structure : the axiomatic approach (top-down) and the constructive one (bottom-up). For instance, an ordinal can either be defined as the order type of a well-order, or using von Neumann definition, as the set of its smaller ordinals. We shall see that the effective counterparts of these definitions coincide, yielding a robust notion of computable ordinal.[12]

12: We assume the reader has some familiarity with the classical theory of ordinals.

Definition 11.4.1. An ordinal α is *computable* if it is finite or it is the order-type of a computable[13] well-order on $\mathbb{N}$. ◇

13: Actually, one could have replaced "computable" by "polynomial-time computable", "arithmetic", or even "hyperarithmetic", this would have yielded exactly the same class of ordinals, even-though the equivalence is highly non-trivial.

First, note from the above definition that every computable ordinal is witnessed by the program of a computable well-order. There are therefore only countably many ordinals. We first show that one can replace "computable" by "c.e." in the above definition of a computable ordinal.

Lemma 11.4.2. Let $<_R$ be a c.e. total order on $\mathbb{N}$. Then $<_R$ is computable.★

Proof. By totality of $<_R$, $(a, b) \notin <_R$ iff $a = b$ or $(b, a) \in <_R$. Thus, $<_R$ is both c.e. and co-c.e., hence is computable. ■

We shall now prove that the computable ordinals form an initial segment of the ordinals.

Lemma 11.4.3. Let $<_R$ be a c.e. total order on an infinite set $A \subseteq \mathbb{N}$. Then there is a c.e. total order $<_S$ on $\mathbb{N}$ with the same order type as $<_R$. ★

Proof. First, note that A is c.e., since $A = \{a \in \mathbb{N} : \exists b((a, b) \in <_R \vee (b, a) \in <_R)\}$ by totality of $<_R$. Thus, there is a computable bijection $f : \mathbb{N} \to A$. Then, $<_S = \{(f^{-1}(a), f^{-1}(b) : (a, b) \in <_R\}$. ■

Suppose now that α is a computable ordinal, as witnessed by a computable well-order $<_R$ on $\mathbb{N}$, and let $\beta < \alpha$. Then either β is finite, in which case it is computable by definition, or β is the order type of $<_R$ restricted to $\{b \in \mathbb{N} : b <_R a\}$ for some $a \in \mathbb{N}$ with infinitely many predecessors. Then by Lemma 11.4.3 and Lemma 11.4.2, β is the order type of a computable well-order on $\mathbb{N}$, thus is a computable ordinal. Since the computable ordinals form a countable initial segment of the ordinals, then there is a least non-computable ordinal.

Definition 11.4.4. Let ω_1^{ck} denote the least non-computable ordinal.[14] ◇

14: "ck" stands for "Church Kleene", who introduced the concept in [91].

The representation of a computable ordinal using well-orders is not the most effective, in that given a computable well-order $<_R$ on $\mathbb{N}$ and some $a \in \mathbb{N}$, one cannot computably decide wether a is a successor element or a limit. We now give an alternative and more constructive definition of the computable ordinals, which can be seen as an effective counterpart of von Neumann definition.

15: The choice of 2^b to code the successor of b and $3 \cdot 5^e$ to code for a limit ordinal with cofinal sequence Φ_e is arbitrary. The only requirement is to have a unique notation to be able to deconstruct the inductive definition and distinguish the successor and limit cases. For instance, one could have defined 3^{e+1} instead of $3 \cdot 5^e$.

Definition 11.4.5 (Kleene's O). Let $<_\mathcal{O}$ be the least partial order on $\mathbb{N}$ such that $1 <_\mathcal{O} 2$, satisfying the following closures:[15]

(1) If $a <_\mathcal{O} b$ then $a <_\mathcal{O} 2^b$
(2) For every total function $\Phi_e : \mathbb{N} \to \mathbb{N}$, if $\forall n(\Phi_e(n) <_\mathcal{O} \Phi_e(n+1))$,

then for every $n \in \mathbb{N}$, $\Phi_e(n) <_{\mathcal{O}} 3 \cdot 5^e$.

Let $\mathcal{O}$ be the domain of $<_{\mathcal{O}}$.[16] ◇

The above definition might seem quite cryptic, and deserves some explanation. Each element a of $\mathcal{O}$ can be evaluated into a computable ordinal $|a|$, by transfinite induction[17] as follows: First, $|1| = \mathbb{0}$. If $2^a \in \mathcal{O}$, then $|2^a| = |a| + \mathbb{1}$. Last, if $3 \cdot 5^e \in \mathcal{O}$, then $|3 \cdot 5^e| = \sup_n |\phi_e(n)|$. To avoid confusion, we write $\mathbb{0}, \mathbb{1}, \dots$ for the finite ordinals and keep the standard font $0, 1, \dots$ for their codes.[18]

Definition 11.4.6. An ordinal α is *constructible* if $\alpha = |a|$ for some $a \in \mathcal{O}$.◇

The main advantage of constructible ordinals is that one can directly know from a code a whether it codes for $\mathbb{0}$, for a successor ordinal, or is a limit ordinal. In the latter case, one can even effectively find a cofinal sequence of codes.

Exercise 11.4.7. Show that the constructible ordinals are downward-closed.★

Every finite ordinal n admits a unique code in $\mathcal{O}$, namely, the n-fold power of two. The ordinal ω, on the other hand, admits infinitely many codes in $\mathcal{O}$, since there exist countably many computable strictly increasing sequences of finite ordinals. More generally, the limit step introduces infinitely many codes, and one can thus see $\mathcal{O}$ as a tree, which is ω-branching at limit steps. A maximal path[19] through this tree is a linearly ordered subset of $\mathcal{O}$ which is downward-closed, and cofinal in ω_1^{ck}.

Exercise 11.4.8. Show that for every $a \in \mathcal{O}$, the set $\{b \in \mathcal{O} : b <_{\mathcal{O}} a\}$ is uniformly c.e. and linearly ordered.[20] ★

The same way Turing-invariant operators on sets induce operations on the Turing degrees, one can study the effectivity of operations on ordinals by defining functions over their codes. The following exercise shows that ordinal addition is computable.

Exercise 11.4.9. Let $+_{\mathcal{O}} : \mathbb{N}^2 \to \mathbb{N}$ be total computable function defined by $a +_{\mathcal{O}} 1 = a$, $a +_{\mathcal{O}} 2^b = 2^{a +_{\mathcal{O}} b}$, $a +_{\mathcal{O}} 3 \cdot 5^e = 3 \cdot 5^{f(e,a)}$, where $f(e, a)$ is the code of a function[21] such that $\Phi_{f(e,a)}(n) = a +_{\mathcal{O}} \Phi_e(n)$, and $a +_{\mathcal{O}} b = 1$ if b is not in any of those forms. Show that for every $a, b \in \mathcal{O}$, $|a| + |b| = |a +_{\mathcal{O}} b|$.★

Given a non-empty c.e. set of codes of constructible ordinals, its supremum is again constructible, but not uniformly computable. One can however uniformly compute an upper bound:

Lemma 11.4.10 (Sacks [93]). There is a total computable function $f : \mathbb{N} \to \mathbb{N}$ such that if $W_e \subseteq \mathcal{O}$, then $f(e) \in \mathcal{O}$ and $\sup_{a \in W_e} |a| \leq |f(e)|$.[22] ★

Proof. One can without loss of generality assume that W_e is infinite, by enumerating all the constructible codes of finite ordinals. For every $e \in \mathbb{N}$, let $f(e) = 3 \cdot 5^a$ where $\Phi_a(n)$ returns the finite ordinal sum (using Exercise 11.4.9) of the n first distinct elements enumerated in W_e, different from 1 (the code of $\mathbb{0}$). One therefore has $\Phi_a(n) <_{\mathcal{O}} \Phi_a(n+1)$ for every $n \in \mathbb{N}$, hence $3 \cdot 5^a \in \mathcal{O}$. Moreover, by construction, $\sup_{a \in W_e} |a| \leq \sup_n |\Phi_a(n)| = |3 \cdot 5^a| = |f(e)|$. ■

16: The sets $<_{\mathcal{O}}$ and $\mathcal{O}$ are both Π^1_1-complete.

17: In order to be allowed to use transfinite induction, one must actually first check that $<_{\mathcal{O}}$ is a well-founded partial ordering. One can define an natural enumeration of $<_{\mathcal{O}}$ by transfinite induction on the ordinals, such that if $a <_{\mathcal{O}} b$ and $b <_{\mathcal{O}} c$, then $a <_{\mathcal{O}} b$ is enumerated at an earlier stage than $b <_{\mathcal{O}} c$. It follows that any infinite decreasing $<_{\mathcal{O}}$-sequence would yield an infinite decreasing sequence of ordinals.

18: One must be careful in distinguishing the constructible code 1 from the ordinal $\mathbb{1}$. Indeed, the code 1 denotes the ordinal $\mathbb{0}$.

19: As noted Chong and Liu [92], not every path can be extended into a maximal path. Indeed, with poor choices at the ω-branching levels, one might obtain only ω^2 for instance.

20: Although $<_{\mathcal{O}}$ is Π^1_1, the restriction of the order to $\{b \in \mathcal{O} : b <_{\mathcal{O}} a\}$ is uniformly c.e. in a.

21: Note that this definition involves Kleene's fixpoint theorem, as the definition of f uses $+_{\mathcal{O}}$. Also note that $a \leq_{\mathcal{O}} a +_{\mathcal{O}} b$ but not necessarily $b \leq_{\mathcal{O}} a +_{\mathcal{O}} b$ because of the limit case.

22: Note that we do not require $<_{\mathcal{O}}$ to be total on W_e. In other words, the inequality holds for ordinals, one does not satisfy $a <_{\mathcal{O}} f(e)$ for every $a \in W_e$.

We shall now prove that the constructible ordinals coincide with the computable ones. Following the intuition, a code for a constructible ordinal carries more information than a computable well-order, in that one can computably transform a code $a \in \mathbb{O}$ into a program for a computable well-order of order type $|a|$, while the reverse translation is not computable.

Theorem 11.4.11 (Kleene, Markwald)
Computable and constructible ordinals coincide.

Proof. Let $a \in \mathbb{O}$ be a code for a constructible ordinal α. If $\alpha < \omega$, then it is computable by definition. If α is infinite, then the relation $<_{\mathbb{O}}$ restricted to $\{b \in \mathbb{O} : b <_{\mathbb{O}} a\}$ is c.e. By Lemma 11.4.3 and Lemma 11.4.2, there is a computable order over $\mathbb{N}$ with the same order type, thus α is computable.

Suppose now that α is a computable ordinal. If $\alpha < \omega$, then the α-fold power of 2 yields a constructible code for α, hence hence α is constructible. If α is infinite, then there is a computable well-order $<_R$ on $\mathbb{N}$ of order type α. Let $f : \mathbb{N} \to \mathbb{N}$ be the function of Lemma 11.4.10, and let $g : \mathbb{N} \to \mathbb{N}$ be the total computable function which on a computes the code e_a of the c.e. set $W_{e_a} = \{g(b) : b <_R a\}$, and outputs $f(e_a)$. One can prove by induction over a that $g(a) \in \mathbb{N}$ and $|g(a)|$ is at least the order type of $<_R$ restricted to the elements below a. Let $W_e = \{g(a) : a \in \mathbb{N}\}$, then $|f(e)| \geq \sup_a |g(a)|$, so $|f(e)|$ is at least the order type of $<_R$.[23] ∎

23: One could be tempted to rather consider $3 \cdot 5^i$ where $\Phi_i(a) = g(a)$. However, although $|g(a)| < |g(a+1)|$, one does not have in general $g(a) <_{\mathbb{O}} g(a+1)$, thus $3 \cdot 5^i$ is not a valid constructible code.

11.5 Hyperarithmetic hierarchy

The arithmetic hierarchy corresponds to the finite levels of the effective counterpart to the Borel hierarchy over $\mathbb{N}$, equipped with the discrete topology.[24] We now generalize the arithmetic hierarchy to transfinite levels, and prove the corresponding generalization of Post theorem, namely, every level of the hierarchy is effectively open relative to the appropriate iteration of the halting set.

24: It seems at first sight that this is just a complicated reformulation of a simple notion. However, the topological considerations are very useful to understand why Post theorem holds for the arithmetic hierarchy, but not for classes over $2^{\mathbb{N}}$. Indeed, since the Borel hierarchy collapses over the discrete topology, every Borel set is open, hence is effectively open relative to an appropriate oracle, while the Borel hierarchy is strict on the Cantor space, hence some Π^0_2 classes are not $\Pi^0_1(A)$ for any oracle A.

Although the arithmetic hierarchy is usually defined in terms of alternations of quantifiers, the generalization to transfinite levels which require to use infinitary effective conjunctions and disjunctions to handle the limit cases. One therefore rather defines the hyperarithmetic hierarchy in terms of codes.

Definition 11.5.1. The hyperarithmetic codes are defined by induction over the computable ordinals[25][26].

1. A Σ^0_1*-code* of a set A is a pair $\langle 0, e\rangle$ such that $W_e = A$.
2. A Π^0_α*-code* of a set A is a pair $\langle 1, e\rangle$, where e is a Σ^0_α-code of the set $\mathbb{N} \setminus A$.
3. A Σ^0_α*-code* of a set $A = \bigcup_n A_n$ is a pair $\langle 2, e\rangle$ where W_e is non-empty, and enumerates $\Pi^0_{\beta_n}$-codes of sets A_n such that $\sup_n(\beta_n + \mathbb{1}) = \alpha$.

◊

25: One could actually define the notion of Σ^0_α-code for arbitrary ordinals. However, an easy induction along the ordinals shows that every Σ^0_α-code is Σ^0_β for some $\beta < \omega_1^{ck}$, hence the hierarchy does not go beyond the computable ordinals.

26: Because Σ^0_α-codes do not distinguish the successor case from the limit case, one cannot uniformly compute a constructible code $a \in \mathbb{O}$ from a $\Sigma^0_{|a|}$-code.

A set A is Σ^0_α (resp. Π^0_α) if it admits a Σ^0_α-code (resp. a Π^0_α-code). A set A is Δ^0_α if it is both Σ^0_α and Π^0_α. An easy induction shows that the finite levels correspond to the arithmetic hierarchy.

Exercise 11.5.2. Show that the Σ^0_α sets are closed under effective countable unions and finite intersections. Moreover, those closure are uniform in Σ^0_α-codes. ★

Exercise 11.5.3. Show that if A is either Σ^0_α or Π^0_α, then A is $\Delta^0_{\alpha+\mathbb{1}}$ uniformly in a Σ^0_α or a Π^0_α-code of A. ★

The following lemma requires a bit more work, thus is fully proven.

Lemma 11.5.4. If A is Δ^0_α and B is $\Sigma^0_1(A)$, then B is Σ^0_α uniformly in a Δ^0_α-code of A and a c.e. index of B.[27] ★

27: A Δ^0_α-code is nothing but a pair of a Σ^0_α-code and a Π^0_α-code.

Proof. Say $B = W^A_e$. Then $B = \{n : \exists\sigma\ (n \in W^\sigma_e \wedge \forall i < |\sigma|\ ((\sigma(i) = 0 \wedge i \notin A) \vee (\sigma(i) = 1 \wedge i \in A))\}$. By induction on α, given $\sigma \in 2^{<\mathbb{N}}$ and $i < 2$, one can uniformly compute a Σ^0_α-code of a set $A_{\sigma,i}$ such that $A_{\sigma,i} = \mathbb{N}$ if $\sigma(i) = A(i)$ and $A_{\sigma,i} = \emptyset$ otherwise. Then $B = \bigcup_\sigma (W^\sigma_e \cap \bigcap_{i<|\sigma|} A_{\sigma,i})$. By Exercise 11.5.2, B is Σ^0_α. ■

The following exercise is proven by a simple induction over codes, and will be useful later.

Exercise 11.5.5. Let $f : \mathbb{N} \to \mathbb{N}$ be a total computable function and A be a Σ^0_α-set. Show that $f[A] = \{f(n) : n \in A\}$ is Σ^0_α uniformly in a Σ^0_α-code of A and a c.e. index of f. ★

We now define transfinite iterations of the Turing jump to state the generalized Post theorem. In the limit case, one naturally wants to join a cofinal sequence of previous iterations. This raises some canonicity issues, as there exist infinitely many cofinal sequences already at the level of ω, and they yield different sets[28]. We will therefore iterate the jump along constructible codes of ordinals.[29]

28: One could for instance define $\emptyset^{(\omega)}$ as $\bigoplus_n \emptyset^{(n)}$, but also as $\bigoplus_n \emptyset^{(2n)}$, among many possibilities.

29: Since constructible codes are integers, it would be confusing to write $\emptyset^{(a)}$ for an $|a|$-iteration of the Turing jump. One therefore traditionally uses the notation H_a, standing for "hyperarithmetic".

Definition 11.5.6. For every $a \in \mathcal{O}$, let H_a be defined inductively as follows.

1. $H_1 = \emptyset$
2. $H_{2^a} = H'_a$
3. $H_{3\cdot 5^e} = \bigoplus_n H_{\Phi_e(n)}$. ◇

By Spector [94], if a and b are two constructible codes for an ordinal α, then $H_a \equiv_T H_b$. Therefore, this hierarchy defines iterations of the Turing jump over the Turing degrees, and one can write $\mathbf{0}^{(\alpha)}$ for the α-iterate of the Turing jump. The following proposition might be surprising at first, as the transfinite iterations are shifted with respect to the finite levels.

Proposition 11.5.7. For every constructible code $a \in \mathcal{O}$ with $|a| \geq \omega$, H_a is $\Delta^0_{|a|}$ uniformly in a. ★

Proof. By induction along $\mathcal{O}$ starting with $|a| = \omega$.

Suppose first $a = 2^b$ codes of a successor ordinal. Then, by induction hypothesis, H_b is $\Delta^0_{|b|}$ uniformly in b. By Lemma 11.5.4, $H_a = H'_b$ is $\Sigma^0_{|b|}$ uniformly in b, so by Exercise 11.5.3, H_a is $\Delta^0_{|a|}$ uniformly in a.

Suppose now $a = 3 \cdot 5^e$ codes for a limit ordinal. Here, for every n, we have two cases: either $\Phi_e(n)$ is a constructible code of a finite ordinal, in which case Post's theorem yields that $H_{\Phi_e(n)}$ is $\Sigma^0_{|\Phi_e(n)|+1}$ uniformly in n and e, or $\Phi_e(n)$ is a constructible code of an infinite ordinal. In the latter case, by induction hypothesis, $H_{\Phi_e(n)}$ is $\Delta^0_{|\Phi_e(n)|}$ uniformly in n and e, in which case by Exercise 11.5.3 it is again $\Sigma^0_{|\Phi_e(n)|+1}$ uniformly in n and e. Note that one can computably decide in which case we are, since being a constructible code of a finite ordinal is decidable. Thus, we can assume in both cases that $H_{\Phi_e(n)}$ is $\Sigma^0_{|\Phi_e(n)|+1}$ uniformly in n and e.

By Exercise 11.5.5, for each n, the set $B_n = \{\langle m, n\rangle : m \in H_{\Phi_e(n)}\}$ is $\Sigma^0_{|\Phi_e(n)|+1}$ uniformly in n and e. Then $H_a = \bigcup_n B_n$ is $\Sigma^0_{|\alpha|}$ uniformly in a. By Exercise 11.5.3, $\overline{H}_{\Phi_e(n)}$ is $\Sigma^0_{|\Phi_e(n)|+2}$ uniformly in n and e. By Exercise 11.5.5, for each n, the set $C_n = \{\langle m, n\rangle : m \in \overline{H}_{\Phi_e(n)}\}$ is $\Sigma^0_{|\Phi_e(n)|+2}$ uniformly in n and e. Thus, $\overline{H}_a = \bigcup_n C_n$ is $\Sigma^0_{|\alpha|}$ uniformly in a. It follows that H_a is $\Delta^0_{|\alpha|}$ uniformly in a. ■

Corollary 11.5.8
For every constructible code $a \in \mathcal{O}$,

1. *if $|a| < \omega$, then H_a is $\Sigma^0_{|a|}$ uniformly in a;*
2. *if $|a| \geq \omega$, then H_{2^a} is $\Sigma^0_{|a|}$ uniformly in a.*

Proof. The first case holds by Post's theorem. The second case is immediate by Proposition 11.5.7 and Lemma 11.5.4. ■

The bound is actually tight, and one can prove with some extra work that H_{2^a} is $\Sigma^0_{|a|}$-complete when $|a| \geq \omega$. Together with Post's theorem, this yields the following generalized Post theorem:

Theorem 11.5.9 (Monin and Patey [4])
Fix some $a \in \mathcal{O}$.

1. *If $|a| < \omega$, then the set H_a is $\Sigma^0_{|a|}$-complete uniformly in a.*
2. *If $|a| \geq \omega$, then the set H_{2^a} is $\Sigma^0_{|a|}$-complete uniformly in a.*

11.6 Higher recursion theory

Beyond the definition of a robust notion of computable ordinal, and the extension of the arithmetic hierarchy to transfinite levels, there is a whole theory generalizing computability theory along computable ordinals, called *higher recursion theory*. Its development goes far beyond the scope of this book. We however state some of its main concepts and theorems, which will be useful for transfinite jump control. One might refer to Sacks [93], Chong and Yu [92] or to Monin and Patey [4] for an introduction to higher recursion theory.

11.6.1 Hyperarithmetic reduction

Many natural properties on sets induce operations or relations over sets by considering their relativized form. The most basic example is the notion of

Turing machine, whose relativization yields the Turing reduction. One can also relativize the arithmetic hierarchy, yielding the arithmetic reduction by letting X be *arithmetically reducible* to Y if X is $\Sigma^0_n(X)$ for some $n \in \mathbb{N}$. Similarly, one can naturally define the notion of Y-computable ordinal, with ω_1^Y denoting the least non-Y-computable ordinal. The $\Pi^1_1(Y)$ set $\mathcal{O}^Y$ of Y-constructible codes is defined accordingly, with all c.e. operators replaced by Y-c.e. operators.[30] One then defines $\Sigma^0_\alpha(Y)$ classes for $\alpha < \omega_1^Y$ and the sets H_a^Y for $a \in \mathcal{O}^Y$. All the theorems of the previous sections are uniform in Y. In particular, $H^Y_{2^a}$ is uniformly $\Sigma^0_{|a|_Y}$ if $|a|_Y \geq \omega$.

30: If $a \in \mathcal{O}^X \cap \mathcal{O}^Y$, the interpretation $|a|_Y$ of a Y-constructible code might differ from its interpretation $|a|_X$. For convenience, we might assume that for every $a \in \mathcal{O} \cap \mathcal{O}^Y$, $|a| = |a|_Y$.

We shall see that most sets Y satisfy $\omega_1^Y = \omega_1^{ck}$. In other words, it is an "anomaly" to compute non-computable ordinals. However, even if $\omega_1^Y = \omega_1^{ck}$, computable ordinals will have in general more codes in $\mathcal{O}^Y$ than in $\mathcal{O}$.

Definition 11.6.1. A set X is *hyperarithmetically reducible*[31] to a set Y (written $X \leq_h Y$) if it is $\Sigma^0_\alpha(Y)$ for some $\alpha < \omega_1^Y$, or equivalently if there is some $a \in \mathcal{O}^Y$ and $e \in \mathbb{N}$ such that $X = \Phi_e^{H_a^Y}$. ◊

31: It is very important to note that $a \in \mathcal{O}^Y$ and not simply $a \in \mathcal{O}$. Indeed, Y might compute some non-computable ordinals.

The hyperarithmetic reduction is a very robust notion, in that it admits various characterizations of very different nature. A set $X \subseteq \mathbb{N}$ is $\Sigma^1_1(Y)$ if it can be written of the form $\{n \in \mathbb{N} : \exists X \varphi(X, Y, n)\}$, where φ is an arithmetic formula.[32] A set X is $\Pi^1_1(Y)$ if its complement is $\Sigma^1_1(Y)$, and $\Delta^1_1(Y)$ if it is both $\Sigma^1_1(Y)$ and $\Pi^1_1(Y)$. A Y*-modulus* of a set X is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every $g : \mathbb{N} \to \mathbb{N}$ dominating[33] f, $g \oplus Y \geq_T X$. Last, a set X is X*-computably encodable* if for every infinite set $A \subseteq \mathbb{N}$, there is an infinite subset $B \subseteq A$ such that $B \oplus Y \geq_T X$. The following theorem shows that all these definitions coincide.

32: By Kleene's normal form theorem, φ can even be taken Π^0_1.

33: A function g *dominates* f if $g(x) \geq f(x)$ for every x. Some authors define it as $g(x) \geq f(x)$ *for all but finitely many* x. This difference does not matter in this context.

Theorem 11.6.2 (Groszek and Slaman [95], Solovay [19], Kleene [96])
Let X and Y be two sets. The following are equivalent:

1. $X \leq_h Y$;
2. X *is* $\Delta^1_1(Y)$;
3. X *admits a* Y*-modulus;*
4. X *is* Y*-computably encodable.*

There exists a whole correspondence[34] between classical computability theory and higher recursion theory. In this correspondence, the Π^1_1 sets play the role of higher c.e. sets, the hyperarithmetic sets are both the higher finite and higher computable sets, and Kleene's $\mathcal{O}$ is the higher halting set.

34: This correspondence is imperfect, in particular because the true higher counterpart of the integers is ω_1^{ck}. It follows that there is a better correspondence between classical computability theory and *metarecursion theory*, a theory which studies the subsets of ω_1^{ck} from a computational viewpoint. See Sacks [93] for an introduction to both theories.

The following theorem is known as the Σ^1_1 *majoration theorem*.

Theorem 11.6.3 (Spector [94])
Let $X \subseteq \mathcal{O}$ be a Σ^1_1 set. Then $\sup_{a \in X} |a| < \omega_1^{ck}$.[35]

35: This theorem is actually uniform in the following sense: one can computably find a constructible code $b \in \mathcal{O}$ such that $\sup_{a\in X} |a| \leq |b|$ from a Σ^1_1-code of X.

Corollary 11.6.4
Let $f : \mathbb{N} \to \mathcal{O}$ be a total Π^1_1-function.[36] *Then $\sup_n |f(n)| < \omega_1^{ck}$.*

36: A function is Π^1_1 if its graph is Π^1_1.

Proof. The graph G_f of f can be written of the form $\{(x, y) : \forall X \Phi_e^X(x, y)\downarrow\}$. Since f is total, $G_f = \{(x, y) : \forall z \exists X (z \neq y \to \Phi_e^X(x, z)\uparrow\}$, which is a Σ^1_1 set, so f is Δ^1_1. In particular, the range of f is a Σ^1_1 subset of $\mathcal{O}$, so by the Σ^1_1 majoration theorem, $\sup_n |f(n)| < \omega_1^{ck}$. ■

11.6.2 Hyperjump operator

As mentioned, Kleene's $\mathcal{O}$ is the higher counterpart of the halting set. The relativization of the halting set induces an operation on the Turing degrees called the Turing jump. Similarly, the map $X \mapsto \mathcal{O}^X$ is compatible with the hyperarithmetic reduction, and therefore induces an operation on the hyperarithmetic degrees, called the *hyperjump*.

Recall that given two sets X, Y, $X \leq_T Y$ iff $X' \leq_m Y'$. The following theorem states its higher counterpart.

Theorem 11.6.5 (Sacks [93])
Fix two sets X, Y. Then $X \leq_h Y$ iff $\mathcal{O}^X \leq_m \mathcal{O}^Y$.

Proof. Suppose first $X \leq_h Y$. Then X is $\Delta^1_1(Y)$ by Theorem 11.6.2, but since $\mathcal{O}^X$ is $\Pi^1_1(X)$, then $\mathcal{O}^X$ is $\Pi^1_1(Y)$.[37] Since $\mathcal{O}^Y$ is $\Pi^1_1(Y)$-complete for the many-one reduction[38], $\mathcal{O}^X \leq_m \mathcal{O}^Y$.

Suppose now $\mathcal{O}^X \leq_m \mathcal{O}^Y$. Since X and $\overline{X}$ are $\Pi^1_1(X)$, then $X \leq_m \mathcal{O}^X$ and $\overline{X} \leq_m \mathcal{O}^X$. It follows by transitivity of the many-one reduction that $X \leq_m \mathcal{O}^Y$ and $\overline{X} \leq_m \mathcal{O}^Y$. Since $\mathcal{O}^Y$ is $\Pi^1_1(Y)$, both X and $\overline{X}$ are $\Pi^1_1(Y)$, so X is $\Delta^1_1(Y)$, hence $X \leq_h Y$ by Theorem 11.6.2. ■

37: This is true in general: if X is $\Pi^1_1(Y)$ and Y is $\Delta^1_1(Z)$, then X is $\Pi^1_1(Z)$.

38: The proof that $\mathcal{O}$ is Π^1_1-complete for the many-one reduction relativizes in a strong way: for every set Y and every $\Pi^1_1(Y)$ set X, there is a *computable* function $f : \mathbb{N} \to \mathbb{N}$ such that $X = \{n : f(n) \in \mathcal{O}^Y\}$.

One deduces from the previous theorem that the hyperjump operator is a hyperdegree-theoretic operation. The following theorem states in a relativized form that the notion of computable ordinal is robust, in that any hyperarithmetic ordinal is computable.

Theorem 11.6.6 (Spector [94])
Fix two sets X, Y. If $X \leq_h Y$, then $\omega_1^X \leq \omega_1^Y$.

Proof. Let $f : \mathbb{N} \to \mathbb{N}$ be the partial Y-computable function witnessing the uniformity of the Σ^1_1 majoration theorem relativized to Y (Theorem 11.6.3), that is, if $A \subseteq \mathcal{O}^Y$ is a $\Sigma^1_1(Y)$ set with $\Sigma^1_1(Y)$-code c, then $f(c) \in \mathcal{O}^Y$ is such that $\sup_{a \in A} |a|_Y \leq |f(c)|_Y$.

We prove, by transfinite induction over the X-constructible codes, the existence of a partial Y-computable function $g : \mathbb{N} \to \mathbb{N}$ such that for every $a \in \mathcal{O}^X$, $g(a) \in \mathcal{O}^Y$ and $|a|_X \leq |g(a)|_Y$. Let $a \in \mathcal{O}^X$.

Suppose first $a = 1$ codes for $\mathbb{0}$. Letting $g(a) = 1$, we have $|a|_X = |g(a)|_Y$.

Suppose now $a = 2^b$ codes for a successor ordinal. Then by induction hypothesis, $g(b) \in \mathcal{O}^Y$ and $|b|_X \leq |g(b)|_Y$. Letting $g(a) = 2^{g(b)}$, we have $|a|_X = |b|_X + \mathbb{1} \leq |g(b)|_Y + \mathbb{1} = |g(a)|_Y$.

Suppose last $a = 3 \cdot 5^e$ codes for a limit ordinal. Then for every n, by induction hypothesis, $g(\Phi_e^X(n)) \in \mathcal{O}^Y$ and $|\Phi_e^X(n)|_X \leq |g(\Phi_e^X(n))|_Y$. Since X is $\Delta^1_1(Y)$, the set $A = \{g(\Phi_e^X(n)) : n \in \mathbb{N}\} \subseteq \mathcal{O}^Y$ is $\Sigma^1_1(Y)$. Furthermore, a $\Sigma^1_1(Y)$-code c of A can be found uniformly in e. Let $g(a) = f(c)$. ■

Last, the following theorem relates the hypercomputation of Kleene's $\mathcal{O}$ to the computation of a non-computable ordinal. It implies in particular that the hyperjump is strictly increasing in the hyperdegrees.

Theorem 11.6.7 (Spector [94])
Let X be a set. Then $X \geq_h \mathcal{O}$ iff $\omega_1^X > \omega_1^{ck}$.[39]

39: This statement relativizes as follows: let X, Y be sets such that $X \geq_h Y$. Then $X \geq_h \mathcal{O}^Y$ iff $\omega_1^X > \omega_1^Y$. In particular, the hypothesis $X \geq_h Y$ is necessary for the equivalence to hold.

11.6.3 Classes of reals

One can define an effective Borel hierarchy for the Cantor space as one did for the discrete topology on $\mathbb{N}$. This yields the notions of Σ^0_α and Π^0_α classes of reals for every $\alpha < \omega_1^{ck}$. The notions of Σ^0_α-code and Π^0_α-code for classes are defined accordingly.

Many previous theorems about the arithmetic hierarchy relativize uniformly in the oracle. They enable to give canonical representations of the effective Borel hierarchy using iterations of the halting set. Recall that every Σ^0_k class of reals is of the form $\{X : n \in X^{(k)}\}$ for some $n \in \mathbb{N}$. The generalization to the transfinite levels yields the following theorem.

Theorem 11.6.8 (Monin and Patey [4])
Fix some $a \in \mathcal{O}$ such that $|a| \geq \omega$. A class $\mathcal{A} \subseteq 2^{\mathbb{N}}$ is $\Sigma^0_{|a|}$ iff there is some $n \in \mathbb{N}$ such that $\mathcal{A} = \{X : n \in H^X_{2^a}\}$.[40]

40: Note again the shift in indices between the finite levels and the transfinite levels.

Given a set Y and $\beta < \omega_1^Y$, we let $\mathcal{O}^Y_{<\beta} = \{a \in \mathcal{O} : |a|_Y < \beta\}$. Among the classes of reals, we shall be particularly interested in the following family of classes:

Theorem 11.6.9 (Spector [94])
For every $n \in \mathbb{N}$ and $a \in \mathcal{O}$, the class $\{X : n \in \mathcal{O}^X_{<|a|}\}$ is $\Sigma^0_{|a|+\mathbb{1}}$ uniformly in n and a.

11.7 Transfinite jump control

Transfinite jump control involves different sets of techniques, depending on whether one wants to control a fixed level in the hyperarithmetic hierarchy, or the hyperjump itself. Indeed, α-jump control for a fixed level $\alpha < \omega_1^{ck}$ is achieved by designing a Σ^0_α-preserving forcing question for Σ^0_α-classes, while hyperjump control furthermore requires to consider G-computable ordinals $\alpha < \omega_1^G$, where G is the generic set being built. This section is therefore divided into two parts, each focusing on one problematic.

11.7.1 α-jump control

As usual, we illustrate the technique with the simplest notion of forcing, namely, Cohen forcing, and with α-jump cone avoidance.

Theorem 11.7.1 (Feferman [90])
Fix a non-zero $\alpha < \omega_1^{ck}$ and let C be a non-Δ^0_α set. For every sufficiently Cohen generic filter $\mathcal{F}$, C is not $\Delta^0_\alpha(G_{\mathcal{F}})$.

PROOF. This proof is a generalization of Theorem 11.2.1 to transfinite levels. Contrary to finite levels which can be represented by arithmetic formulas, defining a notion of Σ^0_α-formula for $\alpha \geq \omega$ would require to work with some effective infinitary logic, with effective countable disjunctions and intersections. It is therefore more convenient to define the forcing relation in terms of classes.

Definition 11.7.2. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition, and $\mathcal{B} \subseteq 2^{\mathbb{N}}$ be a Σ^0_α class for $\alpha < \omega_1^{ck}$.[41]

1. For $\alpha = \mathbb{1}$, let $\sigma \mathrel{?\vdash} \mathcal{B}$ hold if there is some $\tau \succeq \sigma$ such that $[\tau] \subseteq \mathcal{B}$.
2. For $\alpha > \mathbb{1}$, $\mathcal{B}$ is of the form $\bigcup_n \mathcal{B}_{\beta_n}$ where $\mathcal{B}_{\beta_n}$ is $\Pi^0_{\beta_n}$. Let $\sigma \mathrel{?\vdash} \mathcal{B}$ hold if there is some $\tau \succeq \sigma$ and some $n \in \mathbb{N}$ such that $\tau \mathrel{?\vdash} \mathcal{B}_{\beta_n}$.[42] $\diamond$

41: The notation $\sigma \mathrel{?\vdash} \mathcal{B}$ is a shorthand for $\sigma \mathrel{?\vdash} G \in \mathcal{B}$. At finite levels, $\mathcal{B}$ can be written as $\{X \in 2^{\mathbb{N}} : \varphi(X)\}$ for some Σ^0_n-formula φ and $\sigma \mathrel{?\vdash} \mathcal{B}$ iff $\sigma \mathrel{?\vdash} \varphi(G)$.

42: The class $\mathcal{B}_{\beta_n}$ is $\Pi^0_{\beta_n}$, and the forcing question for Π-formulas is induced from the one for Σ-formulas. Thus, $\tau \mathrel{?\vdash} \mathcal{B}_{\beta_n}$ is a shorthand for $\tau \mathrel{?\nvdash} (2^{\mathbb{N}} \setminus \mathcal{B}_{\beta_n})$

We start by proving that the forcing question for Σ^0_α-classes is Σ^0_α-preserving uniformly in its parameters, for $\alpha < \omega_1^{ck}$.

Lemma 11.7.3. For every non-zero $\alpha < \omega_1^{ck}$, every Σ^0_α class $\mathcal{B} \subseteq 2^{\mathbb{N}}$ and every Cohen condition $\sigma \in 2^{<\mathbb{N}}$. The relation $\sigma \mathrel{?\vdash} \mathcal{B}$ is Σ^0_α uniformly in σ and a Σ^0_α-code c of $\mathcal{B}$. ⋆

PROOF. By induction over α. For $\alpha = \mathbb{1}$, $c = \langle 0, e\rangle$ and $\mathcal{B} = \bigcup_{\tau \in W_e} [\tau]$. Thus, $\sigma \mathrel{?\vdash} \mathcal{B}$ iff there is some $\tau \in W_e$ such that $[\sigma] \cap [\tau] \neq \emptyset$, which is a Σ^0_1 relation uniformly in σ and $\langle 0, e\rangle$.

For $\alpha > \mathbb{1}$, $c = \langle 2, e\rangle$ and $\mathcal{B} = \bigcup_n \mathcal{B}_n$ where $\mathcal{B}_n$ is a $\Pi^0_{\beta_n}$ class of $\Pi^0_{\beta_n}$-code $c_n \in W_e$. Then $\sigma \mathrel{?\vdash} \mathcal{B}$ iff there is some $n \in \mathbb{N}$ and some $\tau \succeq \sigma$ such that $\tau \mathrel{?\nvdash} (2^{\mathbb{N}} \setminus \mathcal{B}_n)$. By induction hypothesis, the relation $\tau \mathrel{?\vdash} (2^{\mathbb{N}} \setminus \mathcal{B}_n)$ is $\Sigma^0_{\beta_n}$ uniformly in a $\Sigma^0_{\beta_n}$-code of $(2^{\mathbb{N}} \setminus \mathcal{B}_n)$, thus $\tau \mathrel{?\vdash} \mathcal{B}_n$ is $\Pi^0_{\beta_n}$ uniformly in a $\Pi^0_{\beta_n}$-code of $\mathcal{B}_n$. Thus, the overall relation is $\Sigma^0_{\sup_n(\beta_n+\mathbb{1})}$, hence is Σ^0_α. ■

The following lemma shows that the definition of the forcing question meets a strong version of its specifications.

Lemma 11.7.4. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition and $\mathcal{B} \subseteq 2^{\mathbb{N}}$ be a Σ^0_α class for $\alpha < \omega_1^{ck}$.

1. If $\sigma \mathrel{?\vdash} \mathcal{B}$, then there is an extension $\tau \succeq \sigma$ forcing $G \in \mathcal{B}$.
2. If $\sigma \mathrel{?\nvdash} \mathcal{B}$, then σ forces $G \notin \mathcal{B}$. ⋆

PROOF. We prove simultaneously both items inductively on α.

Base case: $\alpha = \mathbb{1}$. If $\sigma \mathrel{?\vdash} \mathcal{B}$, then, letting $\tau \succeq \sigma$ be such that $[\tau] \subseteq \mathcal{B}$, for every filter $\mathcal{F}$ containing τ, $G_{\mathcal{F}} \in \mathcal{B}$. It follows that τ is an extension of σ forcing $G \in \mathcal{B}$. Conversely, if σ does not force $G \notin \mathcal{B}$, then there is a filter $\mathcal{F}$ containing σ such that $G_{\mathcal{F}} \in \mathcal{B}$. Then, since $\mathcal{B}$ is open in Cantor space, there is a finite $\tau \prec G_{\mathcal{F}}$ such that $[\tau] \subseteq \mathcal{B}$. Since $\sigma \prec G_{\mathcal{F}}$, by taking τ long enough, one has $\sigma \prec \tau$, thus $\sigma \mathrel{?\vdash} \mathcal{B}$.

Inductive case: $\alpha > \mathbb{1}$. Say $\mathcal{B} = \bigcup_n \mathcal{B}_n$, where $\mathcal{B}_n$ is $\Pi^0_{\beta_n}$. If $\sigma \mathrel{?\vdash} \mathcal{B}$, then there is some $n \in \mathbb{N}$ and some $\tau \succeq \sigma$ such that $\tau \mathrel{?\vdash} \mathcal{B}_n$. By induction hypothesis, there is some $\rho \succeq \tau$ forcing $G \in \mathcal{B}_n$. In particular, ρ is an extension of σ forcing $G \in \mathcal{B}$. If $\sigma \mathrel{?\nvdash} \mathcal{B}$, then for every $n \in \mathbb{N}$ and every $\tau \succeq \sigma$, $\tau \mathrel{?\nvdash} \mathcal{B}_n$. By induction hypothesis, for every $n \in \mathbb{N}$ and every $\tau \succeq \sigma$, there is some $\rho \succeq \tau$ forcing $G \notin \mathcal{B}_n$. In other words, for every $n \in \mathbb{N}$, the set of all ρ forcing $G \notin \mathcal{B}_n$ is dense below σ. Thus, for every sufficiently generic filter $\mathcal{F}$ containing σ and for every $n \in \mathbb{N}$, there is some $\rho \in \mathcal{F}$ forcing $G \notin \mathcal{B}_n$, hence $G \notin \bigcup_n \mathcal{B}_n$. In other words, σ forces $G \notin \mathcal{B}$. ■

The following diagonalization lemma is a straightforward generalization of Lemma 3.2.2. Fix some $a \in \mathcal{O}$ such that $|a| = \alpha$. Recall that a set is H_a^Y-computable iff $\alpha < \omega$ and it is $\Delta^0_{\alpha+\mathbb{1}}(Y)$, or $\alpha \geq \omega$ and it is $\Delta^0_\alpha(Y)$. For simplicity, we shall handle only the case $\alpha \geq \omega$, since the finite case is Lemma 11.2.4.

Lemma 11.7.5. For every Cohen condition $\sigma \in 2^{<\mathbb{N}}$ and every Turing index e, there is an extension $\tau \geq \sigma$ forcing $\Phi_e^{H_a^G} \neq C$. ⋆

Proof. Consider the following set[43]

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

43: By Corollary 11.5.8, for $\alpha \geq \omega$, the following class is Σ^0_α uniformly in x and v:

$$\mathcal{B}_{x,v} = \{X : \Phi_e^{H_a^X}(x)\downarrow = v\}$$

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

- Case 1: $(x, 1 - C(x)) \in U$ for some $x \in \mathbb{N}$. By Lemma 11.7.4(1), there is an extension $\tau \geq \sigma$ forcing $\Phi_e^{H_a^G}(x)\downarrow = 1 - C(x)$.
- Case 2: $(x, C(x)) \notin U$ for some $x \in \mathbb{N}$. By Lemma 11.7.4(2), there is an extension $\tau \geq \sigma$ forcing $\Phi_e^{H_a^G}(x)\uparrow$ or $\Phi_e^{H_a^G}(x)\downarrow \neq C(x)$.
- Case 3: None of Case 1 and Case 2 holds. Then U is a Σ^0_α graph of the characteristic function of C, hence C is Δ^0_α. This contradicts our hypothesis. ■

We are now ready to prove Theorem 11.7.1. Let $\mathcal{F}$ be a sufficiently generic filter for Cohen forcing, and let $G_\mathcal{F} = \bigcup \mathcal{F}$. By genericity of $\mathcal{F}$, $G_\mathcal{F}$ is an infinite binary sequence. If $\alpha < \omega$, by Lemma 11.2.4 $C \not\leq G_\mathcal{F}^{(\alpha-1)}$. If $\alpha \geq \omega$, by Lemma 11.7.5, $C \not\leq_T H_a^{G_\mathcal{F}}$. In both cases, C is not $\Delta^0_\alpha(G_\mathcal{F})$. This completes the proof of Theorem 11.7.1. ■

Exercise 11.7.6. Let $(\mathbb{P}, \leq)$ be the primitive recursive Jockusch-Soare forcing, that is, $\mathbb{P}$ is the set of all infinite primitive recursive binary trees $T \subseteq 2^{<\mathbb{N}}$, partially ordered by inclusion. Fix a non-zero $\alpha < \omega_1^{ck}$.

1. Adapt the proof of Theorem 9.4.1 to design a Σ^0_α-preserving forcing question for Σ^0_α-formulas.
2. Deduce that for every non-Δ^0_α set C and every sufficiently generic $\mathbb{P}$-filter $\mathcal{F}$, C is not $\Delta^0_\alpha(G_\mathcal{F})$. ⋆

11.7.2 Hyperjump control

Hyperjump control can be seen as the higher counterpart of first-jump control. Recall that the hyperjump of a set X is the set $\mathcal{O}^X$, that is, Kleene's O relative to X. The goal of this section is to develop a set of tools to prove that, given a sufficiently generic filter $\mathcal{F}$, $\omega_1^{G_\mathcal{F}} = \omega_1^{ck}$. From this, it follows that the levels of the relativized hyperarithmetic hierarchy are left unchanged, reducing hyperjump control to α-jump control for every $\alpha < \omega_1^{ck}$.

For this, we first need to define sets and classes slightly more complex than the hyperarithmetic hierarchy, but still in the Borel realm. Recall that, although the notion of Σ^0_α-code can be defined for every ordinal α, by the Σ^1_1 majoration theorem, the corresponding hierarchy collapses at the level of ω_1^{ck}, that is, every Σ^0_α set is Σ^0_β for some $\beta < \omega_1^{ck}$. One can however extend the family of sets and classes by considering effective unions along Π^1_1 sets of ordinals. A *hyperarithmetic code* is a Σ^0_α-code for some $\alpha < \omega_1^{ck}$, and a Π^1_1-*code* of a set $A \subseteq \mathbb{N}$ is a code of a Π^1_1-formula defining A.

Definition 11.7.7.

1. A $\Sigma^0_{\omega_1^{ck}}$-*code* of a class $\mathcal{B} \subseteq 2^{\mathbb{N}}$ is a pair $\langle 3, e\rangle$, where e is Π^1_1-code of set $A \subseteq \mathbb{N}$ such that $\mathcal{B} = \bigcup_{e\in A} \mathcal{B}_e$, where $\mathcal{B}_e$ is the class of hyperarithmetic code e.[44]
2. A $\Pi^0_{\omega_1^{ck}}$-*code* of a class $\mathcal{B} \subseteq 2^{\mathbb{N}}$ is a pair $\langle 1, e\rangle$, where e is a $\Sigma^0_{\omega_1^{ck}}$-code of the class $2^{\mathbb{N}} \setminus \mathcal{B}$.
3. A $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$-*code* of a class $\mathcal{B} = \bigcup_n \mathcal{B}_n$ is a pair $\langle 2, e\rangle$ where W_e is non-empty and enumerates $\Pi^0_{\omega_1^{ck}}$-codes of the classes $\mathcal{B}_n$. ◇

44: As explained, this notion does not coincide with the naive definition of $\Sigma^0_{\omega_1^{ck}}$ in terms of effective countable union of hyperarithmetic sets. The set of hyperarithmetic codes of the union must be non-Σ^1_1 in order to properly extend the hyperarithmetic hierarchy.

A class $\mathcal{B} \subseteq 2^{\mathbb{N}}$ is $\Sigma^0_{\omega_1^{ck}}$ ($\Pi^0_{\omega_1^{ck}}$, $\Sigma^0_{\omega_1^{ck}+1}$) if it admits a corresponding code. One can define the notions of $\Sigma^0_{\omega_1^{ck}}$, $\Pi^0_{\omega_1^{ck}}$ and $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ for sets accordingly. In the case of sets, Π^1_1 and $\Sigma^0_{\omega_1^{ck}}$ sets coincide. For classes on the other hand, every $\Sigma^0_{\omega_1^{ck}}$ class is Π^1_1, but the converse is not true.[45]

45: From a topological viewpoint, every $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ class is Borel. The Borel hierarchy does not collapse on the Cantor space, and there exists effectively co-analytic (Π^1_1) classes which are not Borel. On the other hand, as mentioned before, every set of integers is open in the discrete topology on $\mathbb{N}$, so there is no contradiction to the equivalence between Π^1_1 and $\Sigma^0_{\omega_1^{ck}}$ sets.

It will be sometimes more convenient to represent a $\Sigma^0_{\omega_1^{ck}}$ class as a countable union along $\mathcal{O}$. The following lemma shows that the two definitions are equivalent.

Lemma 11.7.8. A class $\mathcal{B} \subseteq 2^{\mathbb{N}}$ is $\Sigma^0_{\omega_1^{ck}}$ iff $\mathcal{B} = \bigcup_{a\in\mathcal{O}} \mathcal{D}_a$, where $\mathcal{D}_a$ is hyperarithmetic uniformly in a.[46] ★

46: Note that one can computably switch from one representation to the other.

Proof. Suppose first $\mathcal{B} = \bigcup_{e\in A} \mathcal{B}_e$, where A is Π^1_1 and $\mathcal{B}_e$ is the class of hyperarithmetic code e. Since $\mathcal{O}$ is Π^1_1-complete for the many-one reduction, there is a total computable function $f : \mathbb{N} \to \mathbb{N}$ such that $e \in A$ iff $f(e) \in \mathcal{O}$. One can furthermore suppose that f is injective and increasing, since given a code $a \in \mathcal{O}$ and $n \in \mathbb{N}$, $2^a_n \in \mathcal{O}$ iff $a \in \mathcal{O}$.[47] In particular, the range of f is computable. For every $a \in \mathcal{O}$, $\mathcal{D}_a = \mathcal{B}_{f^{-1}(a)}$ if a is in the range of f, and $\mathcal{D}_a = \emptyset$ otherwise. Note that $\mathcal{D}_a$ is Σ^0_β for some $\beta < \omega_1^{ck}$, and a Σ^0_β-code of $\mathcal{D}_a$ can be found uniformly in a. By construction, $\mathcal{B} = \bigcup_{a\in\mathcal{O}} \mathcal{D}_a$.

47: The function $(a, n) \mapsto 2^a_n$ is defined inductively by $2^a_0 = a$ and $2^a_{n+1} = 2^{2^a_n}$.

Suppose now $\mathcal{B} = \bigcup_{a\in\mathcal{O}} \mathcal{D}_a$, where $\mathcal{D}_a$ is hyperarithmetic uniformly in a. Let $f : \mathbb{N} \to \mathbb{N}$ be a partial computable function such that $f(a)$ is a hyperarithmetic code of $\mathcal{D}_a$ for every $a \in \mathcal{O}$. Here again, one can suppose that f is injective and increasing, since one can computably transform a hyperarithmetic code into a larger hyperarithmetic code of the same class. Let $A = \{f(a) : a \in \mathcal{O}\}$. The set A is Π^1_1 as it is the image of a Π^1_1 set by a computable injective function. Thus $\mathcal{B} = \bigcup_{e\in A} \mathcal{B}_e$, where $\mathcal{B}_e$ is the class of hyperarithmetic code e. ■

As usual, Cohen forcing provides a simple example to illustrate the use of the forcing question. We therefore prove that Cohen genericity preserves ω_1^{ck}.

Theorem 11.7.9 (Feferman [90])
For every sufficiently Cohen generic filter $\mathcal{F}$, $\omega_1^{G_\mathcal{F}} = \omega_1^{ck}$.

Proof. Suppose $\omega_1^G > \omega_1^{ck}$, then there is an element $a \in \mathcal{O}^G$ which codes for ω_1^{ck}. Since ω_1^{ck} is a limit ordinal, $a = 3 \cdot 5^e$, where $\forall n \Phi^G_e(n)\downarrow \in \mathcal{O}^G_{<\omega_1^{ck}}$ and

48: The set $\mathcal{O}^G_{<\alpha}$ is the set of all codes $a \in \mathcal{O}^G$ such that $|a|_G < \alpha$. Note that $\mathcal{O}^G_{<\omega_1^{ck}} \neq \mathcal{O}$ in general. We can however assume for convenience that $\mathcal{O} \subseteq \mathcal{O}^G_{<\omega_1^{ck}}$.

with $\sup_n |\Phi_e^G(n)|_G = \omega_1^{ck}$.[48] We shall therefore naturally work with $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ classes. We first extend the forcing question to $\Sigma^0_{\omega_1^{ck}}$ and $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ classes, assuming the existence of a Σ^0_α-preserving forcing question for Σ^0_α-formulas (see the proof of Theorem 11.7.1).

Definition 11.7.10. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition, and $\mathcal{B} = \bigcup_{a \in \mathcal{O}} \mathcal{B}_a$ be a $\Sigma^0_{\omega_1^{ck}}$ class.[49] Let $\sigma \mathrel{?{\vdash}} \mathcal{B}$ hold if there is some $a \in \mathcal{O}$ and some $\tau \geq \sigma$ such that $\tau \mathrel{?{\vdash}} \mathcal{B}_a$. ◊

49: By Lemma 11.7.8, $\mathcal{B}$ can be written of this form.

The forcing question for a $\Sigma^0_{\omega_1^{ck}}$-class $\mathcal{B}$ is $\Sigma^0_{\omega_1^{ck}}$ uniformly in a $\Sigma^0_{\omega_1^{ck}}$-code of $\mathcal{B}$. One easily proves that the forcing question meets its specifications. The proof is left as an exercise.

Exercise 11.7.11. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition, and $\mathcal{B} = \bigcup_{a \in \mathcal{O}} \mathcal{B}_a$ be a $\Sigma^0_{\omega_1^{ck}}$ class. Prove that

1. if $\sigma \mathrel{?{\vdash}} \mathcal{B}$, then there is an extension of σ forcing $G \in \mathcal{B}$;
2. if $\sigma \mathrel{?{\nvdash}} \mathcal{B}$, then there is an extension of σ forcing $G \notin \mathcal{B}$. ★

We now extend the forcing question to $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ classes.

Definition 11.7.12. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition, and $\mathcal{B} = \bigcup_n \mathcal{B}_n$ be a $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ class. Let $\sigma \mathrel{?{\vdash}} \mathcal{B}$ hold if there is some $n \in \mathbb{N}$ and some $\tau \geq \sigma$ such that $\tau \mathrel{?{\vdash}} \mathcal{B}_n$.[50] ◊

50: The class $\mathcal{B}_n$ is $\Pi^0_{\omega_1^{ck}}$, so $\tau \mathrel{?{\vdash}} \mathcal{B}_n$ is a shorthand for $\tau \mathrel{?{\nvdash}} (2^{\mathbb{N}} \setminus \mathcal{B}_n)$. The forcing question for $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$-classes is $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$-preserving, but we are not going to use this fact in the proof.

The forcing question for $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ classes meets its specification, but one can actually prove a stronger version of it, in the negative case. Recall that, given a set Y and $\beta < \omega_1^Y$, we let $\mathcal{O}^Y_{<\beta} = \{a \in \mathcal{O} : |a|_Y < \beta\}$.

Lemma 11.7.13. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition, and $\mathcal{B} = \bigcup_n \bigcap_{a \in \mathcal{O}} \mathcal{B}_{n,a}$ be a $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ class, where $\mathcal{B}_{n,a}$ is hyperarithmetic uniformly in n and a.[51]

51: Every $\Sigma^0_{\omega_1^{ck}+\mathbb{1}}$ class can be written of this form thanks to Lemma 11.7.8.

1. If $\sigma \mathrel{?{\vdash}} \mathcal{B}$, then there is an extension of σ forcing $G \in \mathcal{B}$;
2. If $\sigma \mathrel{?{\nvdash}} \mathcal{B}$, then there is some $\beta < \omega_1^{ck}$ and an extension of σ forcing $G \notin \bigcup_n \bigcap_{a \in \mathcal{O}_{<\beta}} \mathcal{B}_{n,a}$.[52] ★

52: Note that $\mathcal{B} \subseteq \bigcup_n \bigcap_{a \in \mathcal{O}_{<\beta}} \mathcal{B}_{n,a}$.

Proof. Suppose $\sigma \mathrel{?{\vdash}} \mathcal{B}$. Then there is some $n \in \mathbb{N}$ and some $\tau \geq \sigma$ such that $\tau \mathrel{?{\vdash}} \bigcap_{a \in \mathcal{O}} \mathcal{B}_{n,a}$. By Exercise 11.7.11, there is an extension $\rho \geq \tau$ forcing $G \in \bigcap_{a \in \mathcal{O}} \mathcal{B}_{n,a}$, hence forcing $G \in \mathcal{B}$.

Suppose $\sigma \mathrel{?{\nvdash}} \mathcal{B}$. For every n and every $\tau \geq \sigma$, $\tau \mathrel{?{\nvdash}} \bigcap_{a \in \mathcal{O}} \mathcal{B}_{n,a}$, in other words, $\tau \mathrel{?{\vdash}} \bigcup_{a \in \mathcal{O}} (2^{\mathbb{N}} \setminus \mathcal{B}_{n,a})$. Unfolding the definition, for every n, and every $\tau \geq \sigma$, there is some $\rho \geq \tau$ and some $a \in \mathcal{O}$ such that $\rho \mathrel{?{\vdash}} (2^{\mathbb{N}} \setminus \mathcal{B}_{n,a})$. Given $n \in \mathbb{N}$ and $\tau \geq \sigma$, let $f(n, \tau) = a$ for some $a \in \mathcal{O}$ such that there some $\rho \geq \tau$ for which $\rho \mathrel{?{\vdash}} (2^{\mathbb{N}} \setminus \mathcal{B}_{n,a})$. The function f is Π^1_1 and total, so by Corollary 11.6.4, there is some $\beta < \omega_1^{ck}$ such that $\sup_{n,\tau \geq \sigma} |f(n, \tau)| < \beta$. Thus, for every $n \in \mathbb{N}$ and every $\tau \geq \sigma$, there is some $\rho \geq \tau$ and some $a \in \mathcal{O}_{<\beta}$ such that $\rho \mathrel{?{\vdash}} (2^{\mathbb{N}} \setminus \mathcal{B}_{n,a})$, and by definition of the forcing question, there is some $\mu \geq \rho$ forcing $G \notin \mathcal{B}_{n,a}$. For every n, let D_n be the set of μ such that for some $a \in \mathcal{O}_{<\beta}$, μ forces $G \notin \mathcal{B}_{n,a}$. The set D_n is dense below σ for every $n \in \mathbb{N}$, so for every sufficiently generic filter $\mathcal{F}$ containing σ, $\mathcal{F} \cap D_n \neq \emptyset$, and thus $G_{\mathcal{F}} \notin \bigcup_n \bigcap_{a \in \mathcal{O}_{<\beta}} \mathcal{B}_{n,a}$. ■

The following lemma is an immediate application of Lemma 11.7.13. The core argument actually lies in Lemma 11.7.13 rather than Lemma 11.7.14.

Lemma 11.7.14. Let $\sigma \in 2^{<\mathbb{N}}$ be a Cohen condition and Φ_e be a Turing functional. There is an extension $\tau \geq \sigma$ forcing one of the following:

1. $\exists n\ \forall \alpha < \omega_1^{ck}\ \Phi_e^G(n) \notin \mathcal{O}_{<\alpha}^G$;
2. $\exists \beta < \omega_1^{ck}\ \forall n\ \Phi_e^G(n) \in \mathcal{O}_{<\beta}^G$. ⋆

Proof. By Spector [94], the class $\mathcal{B}_{n,a} = \{X : \Phi_e^X(n) \notin \mathcal{O}_{<|a|}^X\}$ is hyperarithmetic uniformly in $n \in \mathbb{N}$ and $a \in \mathcal{O}$. It follows that the class $\mathcal{B} = \bigcup_n \bigcap_{a\in\mathcal{O}} \mathcal{B}_{n,a}$ is $\Sigma^0_{\omega_1^{ck}+1}$. If $\sigma\ ?\!\vdash \mathcal{B}$, then by Lemma 11.7.13(1), there is an extension forcing $G \in \mathcal{B}$, in other words forcing $\exists n\ \forall \alpha < \omega_1^{ck}\ \Phi_e^G(n) \notin \mathcal{O}_{<\alpha}^G$. If $\sigma\ ?\!\nvdash \mathcal{B}$, then by Lemma 11.7.13(2), there is some $\beta < \omega_1^{ck}$ and an extension of σ forcing $G \notin \bigcup_n \bigcap_{a\in\mathcal{O}_{<\beta}} \mathcal{B}_{n,a}$, in other words forcing $\forall n \Phi_e^G(n) \in \mathcal{O}_{<\beta}^G$. ■

We are now ready to prove Theorem 11.7.9. Let $\mathcal{F}$ be a sufficiently generic filter for Cohen forcing. Suppose for the contradiction that $\omega_1^{G_\mathcal{F}} > \omega_1^{ck}$. Then there is some $a \in \mathcal{O}^{G_\mathcal{F}}$ which codes for ω_1^{ck}. Since ω_1^{ck} is a limit ordinal, $a = 3 \cdot 5^e$, where $\forall n \Phi_e^{G_\mathcal{F}}(n) \downarrow \in \mathcal{O}^{G_\mathcal{F}}_{<\omega_1^{ck}}$ and with $\sup_n |\Phi_e^{G_\mathcal{F}}(n)|_G = \omega_1^{ck}$. By Lemma 11.7.14, either $\exists n\ \forall \alpha < \omega_1^{ck}\ \Phi_e^{G_\mathcal{F}}(n) \notin \mathcal{O}^{G_\mathcal{F}}_{<\alpha}$, or $\exists \beta < \omega_1^{ck}\ \forall n\ \Phi_e^{G_\mathcal{F}}(n) \in \mathcal{O}^{G_\mathcal{F}}_{<\beta}$, in which case $\sup_n |\Phi_e^G(n)|_G \leq \beta < \omega_1^{ck}$. In both cases, this yields a contradiction, so $\omega_1^{G_\mathcal{F}} = \omega_1^{ck}$. This completes the proof of Theorem 11.7.9. ■

Combining Theorem 11.7.9 and Theorem 11.7.1, we obtain cone avoidance for the hyperarithmetic reduction.

Corollary 11.7.15 (Feferman [90])
Let C be a non-hyperarithmetic set. For every sufficiently generic Cohen filter $\mathcal{F}$, $C \nleq_h G_\mathcal{F}$.

Proof. Let $\mathcal{F}$ be a sufficiently generic Cohen filter. By Theorem 11.7.1, C is not $\Delta^0_\alpha(G_\mathcal{F})$ for any $\alpha < \omega_1^{ck}$, and by Theorem 11.7.9, $\omega_1^{G_\mathcal{F}} = \omega_1^{ck}$. It follows that C is not $\Delta^0_\alpha(G_\mathcal{F})$ for any $\alpha < \omega_1^{G_\mathcal{F}}$, thus $C \nleq_h G_\mathcal{F}$. ■

The following contains the core property to prove that every sufficiently generic filter preserves ω_1^{ck}.

Definition 11.7.16. Given a notion of forcing $(\mathbb{P}, \leq)$, a forcing question is $\Sigma^0_{\omega_1^{ck}+1}$*-majoring* if for every $\Sigma^0_{\omega_1^{ck}+1}$ class $\mathcal{B} = \bigcup_n \bigcap_{a\in\mathcal{O}} \mathcal{B}_{n,a}$ where $\mathcal{B}_{n,a}$ is hyperarithmetic uniformly in n and a, for every condition $p \in \mathbb{P}$ such that $p\ ?\!\nvdash \mathcal{B}$, there is some $\beta < \omega_1^{ck}$ and an extension $q \leq p$ forcing $G \notin \bigcup_n \bigcap_{a\in\mathcal{O}_{<\beta}} \mathcal{B}_{n,a}$. ◊

We leave the abstract theorem as an exercise.

Exercise 11.7.17. Let $(\mathbb{P}, \leq)$ be a notion of forcing, with a $\Sigma^0_{\omega_1^{ck}+1}$-majoring forcing question. Prove that for every sufficiently generic filter $\mathcal{F}$, $\omega_1^{G_\mathcal{F}} = \omega_1^{ck}$.⋆

Exercise 11.7.18. Let $(\mathbb{P}, \leq)$ be the primitive recursive Jockusch-Soare forcing, that is, $\mathbb{P}$ is the set of all infinite primitive recursive binary trees $T \subseteq 2^{<\mathbb{N}}$, partially ordered by inclusion.

1. Show the existence of a $\Sigma^0_{\omega_1^{ck}+1}$-majoring forcing question.
2. Deduce that for every sufficiently generic filter $\mathcal{F}$, $\omega_1^{G_\mathcal{F}} = \omega_1^{ck}$. ★