A Failure of Π^1_{n+3}-Reduction in the Presence of Σ^1_{n+3}-Separation

Stefan Hoffelner*

December 5, 2023

Abstract

We show that one can force over L that Σ^1_3-separation holds, while Π^1_3-reduction fails, thus separating these two principles for the first time. The construction can be lifted to canonical inner models M_n with n-many Woodin cardinals, yielding that assuming the existence of M_n, Σ^1_{n+3}-separation can hold, yet Π^1_{n+3}-reduction fails.

1 Introduction

Descriptive Set Theory serves as a fundamental framework for investigating the structure and properties of sets of real numbers. Two central concepts within this theory, the Separation Property, introduced in the early 1920's and the Reduction Property, introduced by Kuratowski in the mid 1930's, have garnered significant attention due to their profound implications for properties of projective subsets of the real numbers.

Definition 1.1. *We say that a projective pointclass $\Gamma \in \{\Sigma^1_n \mid n \in \omega\} \cup \{\Pi^1_n \mid n \in \omega\}$ has the separation property (or just separation) iff every pair A_0 and A_1 of disjoint elements of Γ has a separating set C, i.e. a set C such that $A_0 \subset C$ and $A_1 \subset \omega^\omega \backslash C$ and such that $C \in \Gamma \cap \check{\Gamma}$, where $\check{\Gamma}$ denotes the dual pointclass of Γ.*

Definition 1.2. *We say that a projective pointclass $\Gamma \in \{\Sigma^1_n \mid n \in \omega\} \cup \{\Pi^1_n \mid n \in \omega\}$ satisfies the Γ-reduction property (or just reduction) if every pair B_0, B_1 of Γ-subsets of the reals can be reduced by a pair of Γ-sets R_0, R_1, which means that $R_0 \subset B_0$, $R_1 \subset B_1$, $R_0 \cap R_1 = \emptyset$ and $R_0 \cup R_1 = B_0 \cup B_1$.*

*Supported by the Deutsche Forschungsgemeinschaft (DFG German Research Foundation) under Germanys Excellence Strategy EXC 2044 390685587, Mathematics Münster: Dynamics-Geometry-Structure.

It follows immediately from the definitions that Γ-reduction implies $\check{\Gamma}$-separation. It is very natural to ask whether the reverse direction is also true.

Since their introduction, many results have been proved which shed light on how separation and reduction can behave among the projective pointclasses. These results can be obtained using two very different set theoretic assumptions, which draw very different scenarios of the properties of separation and reduction.

The first assumption is $V = L$, or rather the existence of a Σ^1_2-definable, good projective well-order of the reals. Recall that a good Σ^1_2-definable well-order is a Σ^1_2-well-order with the additional property that also the relation $\text{InSeg}(x, y) :\Leftrightarrow \{(x)_i \mid i \in \omega\} = \{z \mid z <_L y\}$ is Σ^1_2, where $(x)_i$ denotes some recursive decoding of x into ω-many reals, $(x)_i$ being the i-th real decoded out of r. By results of J. Addison ([1]) the existence of a good Σ^1_n-wellorder implies Π^1_m-uniformization for every $m \geqslant n$. As Π^1_m-uniformization implies Π^1_m reduction, the assumption of $V = L$ implies that for every $n \geqslant 1$, Σ^1_n-reduction and Π^1_n-separation is true (the case $n = 1$ follows from Kondo's theorem that Π^1_1-uniformization is true).

The second assumption which settles the behaviour of reduction and separation on the projective hierarchy is projective determinacy (PD). By the results of Y. Moschovakis ([13]), under PD, for every $n \in \omega$, Π^1_{2n+1} and Σ^1_{2n+2} sets have the scale property, which in particular implies that Π^1_{2n+1} and Σ^1_{2n+2} sets have the uniformization property and so Π^1_{2n+1} and Σ^1_{2n+2} reduction is true. By the famous theorems of D. Martin and J. Steel (see [11]) on the one hand, and H. Woodin (see [10]) on the other hand, determinacy assumptions on projective sets and large cardinal assumptions are two sides of the very same coin.

Note that under $V = L$ and also under PD, Γ separation holds because already the stronger $\check{\Gamma}$-reduction (in fact $\check{\Gamma}$-uniformization) holds. So these results do not shed light on the question stated above, whether Γ-separation and $\check{\Gamma}$-reduction are different properties at all. A partial answer to the question was first given by R. Sami in his PhD thesis from 1976 [14]). In it he showed (among other interesting results) that after adding a single Cohen real to L, the resulting universe will satisfy that Π^1_3-separation holds, yet Σ^1_3-reduction fails. And additionally Σ^1_n-reduction holds again for $n \geqslant 4$. His results inspired L. Harrington to produce a model in which $\boldsymbol{\Pi^1_3}$-separation holds but there is a (lightface) Σ^1_3 set, which can not be reduced by any pair of $\boldsymbol{\Sigma^1_3}$-sets, thus Σ^1_3-reduction fails (a write up of Harrington's proof can be found in [9]).

The question for the other side of the projective hierarchy, namely for Σ^1_3-separation and Π^1_3-reduction remained open though since then.

Goal of this article is to produce the counterpart to L. Harrington's result:

Theorem. *One can force over L a model of $\boldsymbol{\Sigma^1_3}$-separation over which there* *is a pair of Π^1_3-sets, which can not be reduced by any pair of $\boldsymbol{\Pi^1_3}$-sets.*

We add that our result uses completely different techniques and methods than Sami's and Harrington's theorems. It follows from a modification of the arguments from [7], that the proofs can be transferred from L to M_n, the canonical inner model with n-many Woodin cardinal.

Theorem. *Assuming that M_n exists, there is a model of $\boldsymbol{\Sigma^1_{n+3}}$-separation over which there is a pair of Π^1_{n+3}-sets, which can not be reduced by any pair of $\boldsymbol{\Pi^1_{n+3}}$-sets.*

The proof relies on the construction and ideas from [5], where a universe with the $\boldsymbol{\Sigma^1_3}$-separation property is produced via forcing over L. However, we will introduce some simplifications of the original argument, yielding a cleaner presentation. We will use the coding machinery form [6] which is basically the same as in [7]. In [6] a universe is forced over L, where Π^1_3-reduction holds but the stronger Π^1_3-uniformization fails. Our proof presented here will take a quite different direction though and uses a more direct diagonalization argument, where we actively work towards two Π^1_3-sets B_0, B_1 which can not be reduced by a pair of $\boldsymbol{\Pi^1_3}$-sets. As we simultaneously have to work towards a stronger failure of Π^1_3-reduction, we need to substantially alter the original definitions and arguments for forcing the Σ^1_3-separation.

1.1 Notation

The notation we use will be mostly standard, we hope. Diverging from the conventions we write $\mathbb{P} = (\mathbb{P}_\alpha : \alpha < \gamma)$ for a forcing iteration of length γ with initial segments $\mathbb{P}_\alpha$. The α-th factor of the iteration will be denoted with $\mathbb{P}(\alpha)$, this is nonstandard as typically one writes $\dot{\mathbb{Q}}_\alpha$. Note here that we drop the dot on $\mathbb{P}(\alpha)$, even though $\mathbb{P}(\alpha)$ is in fact a $\mathbb{P}_\alpha$-name of a partial order. If $\alpha' < \alpha < \gamma$, then we write $\mathbb{P}_{\alpha'\alpha}$ to denote the intermediate forcing of $\mathbb{P}$ which happens in the interval $[\alpha', \alpha)$, i.e. $\mathbb{P}_{\alpha'\alpha}$ is such that $\mathbb{P} \cong \mathbb{P}_{\alpha'} * \mathbb{P}_{\alpha'\alpha}$.

We write $\Sigma_n(X)$, for X an arbitrary set, to denote the set of formulas which are Σ_n and use elements from X as a parameter.

We write $\mathbb{P} \Vdash \varphi$ whenever every condition in $\mathbb{P}$ forces φ, and make deliberate use of restricting partial orders below conditions, that is, if $p \in \mathbb{P}$ is such that $p \Vdash \varphi$, we let $\mathbb{P}' := \mathbb{P}_{\leqslant p} := \{q \in \mathbb{P} : q \leqslant p\}$ and use $\mathbb{P}'$ instead of $\mathbb{P}$. This is supposed to reduce the notational load of some definitions and arguments. We also write $V[\mathbb{P}] \models \varphi$ to indicate that for every $\mathbb{P}$-generic filter G over V, $V[G] \models \varphi$, and use $V[\mathbb{P}]$ to denote the generic extension of V by $\mathbb{P}$ in case the particular choice of the generic filter does not matter in the current context.

2 Independent Suslin trees in L, almost disjoint coding

The coding method of our choice utilizes Suslin trees, which can be generically destroyed in an independent way of each other.

Definition 2.1. *Let $\vec{T} = (T_\alpha : \alpha < \kappa)$ be a sequence of Suslin trees. We say that the sequence is an independent family of Suslin trees if for every finite set of pairwise distinct indices $e = \{e_0, e_1, ..., e_n\} \subset \kappa$ the product $T_{e_0} \times T_{e_1} \times \cdots \times T_{e_n}$ is a Suslin tree again.*

Note that an independent sequence of Suslin trees $\vec{T} = (T_\alpha : \alpha < \kappa)$ has the property that whenever we decide to generically add branches to some of its members, then all the other members of $\vec{T}$ remain Suslin in the resulting generic extension. Indeed, if $A \subset \kappa$ and we form $\prod_{i \in A} T_i$ with finite support, then in the resulting generic extension $V[G]$, for every $\alpha \notin A$, $V[G] \models$ "T_α is a Suslin tree".

One can easily force the existence of independent sequences of Suslin trees with products of Jech's or Tennenbaum's forcing, or with just products of ordinary Cohen forcing. On the other hand independent sequences of length ω_1 already exist in L.

Theorem 2.2. *Assume $V = L$, then there is a $\Sigma_1(\{\omega_1\})$-definable, independent sequence $\vec{S} = (S_\alpha \mid \alpha < \omega_1)$ of Suslin trees.*

Proof. We fix a $\diamondsuit$-sequence $(a_\alpha \subset \alpha \mid \alpha < \omega_1)$. Next we alter the usual construction of a Suslin tree from $\diamondsuit$ to construct an ω_1-sequence of Suslin trees $\vec{T} = (T^\alpha \mid \alpha < \omega_1)$. We consider a partition of ω_1 into ω_1-many stationary sets $\{B_\alpha \mid \alpha < \omega_1\}$ using the canonically defined $\diamondsuit$-sequence. Hence we can assume that the partition is $\Sigma_1(\{\omega_1\})$-definable over L.

If α is a limit stage, and β is such that $\alpha \in B_\beta$, then we want to construct the $\alpha + 1$-th level of T^β, denoted by $T^\beta_{\alpha+1}$ under the assumption that T^β_α is already defined. First we assume that α is a not a limit point of B_β, then we define $T^\beta_{\alpha+1}$ to be T^β_α and put infinitely many successors on each of the top nodes of T^β_α. Second we assume that α is a limit point of B_β. Then we define $T^\beta_{\alpha+1}$ as follows. We let e be an element of $[\omega_1]^{<\omega}$ and we assume that for each $\delta \in e$, we have a tree T^δ_α defined already. We consider $a_\alpha \subset \alpha$. If a_α happens to be a maximal antichain A in $\prod_{\gamma \in e} T^\gamma_\alpha$, then we seal that antichain off at level $\alpha + 1$ for $\prod_{\gamma \in e} T^\gamma_{\alpha+1}$, that is we chose $\prod_{\gamma \in e} T^\gamma_{\alpha+1}$ in such a way that A remains a maximal antichain in all further extensions of $\prod_{\gamma \in e} T^\gamma_{\alpha+1}$. Otherwise we just extend T^β_α via adding top nodes on countably many branches through T^β_α.

We let $T^\beta := \bigcup_{\alpha < \omega_1} T^\beta_\alpha$ and claim that $(T^\beta \mid \beta < \omega_1)$ is an independent sequence of Suslin trees.

Indeed, if A is an antichain in some $\prod_{\gamma \in e} T^\gamma$, then there is a club α such that $A \cap \alpha$ is an antichain in $\prod_{\gamma \in e} T^\gamma_\alpha$. But then A got sealed off in the next step of $\prod_{\gamma \in e} T^\gamma_{\alpha+1}$

The definability of $\vec{S}$ comes from the fact that the canonical $\diamondsuit$-sequence in L is $\Sigma_1(\{\omega_1\})$-definable. We can use L_{ω_1} (which is $\Sigma_1(\{\omega_1\})$ to correctly define $\diamondsuit$ over it and consequentially $\vec{S}$ becomes definable over L_{ω_1} as well. $\square$

Whenever we force with a Suslin tree $(T, <_T)$, i.e. we force with its nodes to add an uncountable branche, we denote the forcing with T again.

We briefly introduce the almost disjoint coding forcing due to R. Jensen and R. Solovay. We will identify subsets of ω with their characteristic function and will use the word reals for elements of 2^ω and subsets of ω respectively. Let $D = \{d_\alpha \;\; \alpha < \aleph_1\}$ be a family of almost disjoint subsets of ω, i.e. a family such that if $r, s \in D$ then $r \cap s$ is finite. Let $X \subset \omega$ be a set of ordinals. Then there is a ccc forcing, the almost disjoint coding $\mathbb{A}_D(X)$ which adds a new real x which codes X relative to the family D in the following way

$$\alpha \in X \text{ if and only if } x \cap d_\alpha \text{ is finite.}$$

Definition 2.3. *The almost disjoint coding $\mathbb{A}_D(X)$ relative to an almost disjoint family D consists of conditions $(r, R) \in [\omega]^{<\omega} \times D^{<\omega}$ and $(s, S) < (r, R)$ holds if and only if*

1. *$r \subset s$ and $R \subset S$.*

2. *If $\alpha \in X$ and $d_\alpha \in R$ then $r \cap d_\alpha = s \cap d_\alpha$.*

We shall briefly discuss the L-definable, $\aleph_1^L$-sized almost disjoint family of reals D we will use throughout this article. The family D is the canonical almost disjoint family one obtains when recursively adding the $<_L$-least real x_β not yet chosen and replace it with $d_\beta \subset \omega$ where that d_β is the real which codes the initial segments of x_β using some recursive bijections between ω and $\omega^{<\omega}$.

3 Coding machinery

We continue with the construction of the appropriate notions of forcing which we want to use in our proof. The goal is to first define a coding forcings $\text{Code}(x)$ for reals x, which will force for x that a certain Σ^1_3-formula $\Phi(x)$ becomes true in the resulting generic extension. The coding method is almost as in [7] and [6].

In a first step we destroy all members of $\vec{S}$ via generically adding an ω_1-branch, that is we first form $\prod_{\alpha \in \omega_1} S_\alpha$ with finite support and force with it over L. Note that this is an $\aleph_1$-sized, ccc forcing over L, so in the generic extension $\aleph_1$ is preserved and CH remains to be true. We use W to denote this generic extension of L.

We let W be our ground model now. Let $x \in W$ be a real, and let $m, k \in \omega$. We simply write (x, m, k) for a real w which codes the triple (x, m, k) in a recursive way. The forcing $\text{Code}(x, m, k, 1)$, which codes the triple (x, m, k) into $\vec{S^1}$ is defined as a two step iteration

$$\text{Code}(x, m, k, 1) := (\mathbb{C}(\omega_1))^L * \dot{\mathbb{A}}(\dot{Y}),$$

where $(\mathbb{C}(\omega_1))^L$ is the usual ω_1-Cohen forcing (i.e. adding an ω_1-set with countable conditions), as defined in L, and $\dot{\mathbb{A}}(\dot{Y})$ is the (name of) an almost disjoint coding forcing, coding a particular set $\dot{Y}$ (to be defined as we proceed in the discussion) into as real. Note that as $\mathbb{C}(\omega_1)$ is defined in L instead of W, we can write the two step iteration $(\mathbb{C}(\omega_1))^L * \dot{\mathbb{A}}(\dot{Y})$ as defined over W as a three step iteration $((\prod_{\alpha \in \omega_1} S_\alpha) \times \mathbb{C}(\omega_1)) * \dot{\mathbb{A}}(\dot{Y})$ over L. As $\mathbb{C}(\omega_1)$ is σ-closed, $\vec{S}$ is still Suslin in $L[\mathbb{C}(\omega_1)]$, hence the forcing can be rewritten as $(\mathbb{C}(\omega_1) \times (\prod_{\alpha \in \omega_1} S_\alpha)) * \dot{\mathbb{A}}(\dot{Y})$. Consequentially the coding forcing does preserve $\aleph_1^L$.

Next we shall describe the factor $\dot{\mathbb{A}}(\dot{Y})$ in detail. We let $g \subset \omega_1$ be a $\mathbb{C}(\omega_1)^L$-generic filter over L, and let $\rho : [\omega_1]^\omega \to \omega_1$ be some canonically definable, constructible bijection between these two sets. We use ρ and g to define the set $h \subset \omega_1$, which eventually shall be the set of indices of ω-blocks of $\vec{S}$, where we code up the characteristic function of the real $((x, y, m)$. Let

$$h := \{\rho(g \cap \alpha) \,:\, \alpha < \omega_1\}$$

and let

$$A := \{\omega\gamma + 2n \mid \gamma \in h, n \notin (x, m, k)\} \cup \{\omega\gamma + 2n + 1 \mid \gamma \in h, n \in (x, m, k).\}$$

$X \subset \omega_1$ be the $<$-least set (in some previously fixed well-order of $H(\omega_2)^W[g]$ which codes the following objects:

> The $<$-least set of ω_1-branches in W through elements of $\vec{S^1}$ which code (x, y, m) at ω-blocks which start at values in h, that is we collect $\{b_\beta \subset S^1_\beta \,:\, \beta = \omega\gamma + 2n, \gamma \in h \wedge n \in \omega \wedge n \notin (x, y, m)\}$ and $\{b_\beta \subset S^1_\beta \,:\, \beta = \omega\gamma + 2n + 1, \gamma \in h \wedge n \in \omega \wedge n \in (x, y, m)\}$.

Note that, when working in $L[X]$ and if $\gamma \in h$ then we can read off (x, m, k) via looking at the ω-block of $\vec{S^1}$-trees starting at γ and determine which tree has an ω_1-branch in $L[X]$:

($*$) $n \in (x, m, k)$ if and only if $S^1_{\omega \cdot \gamma + 2n + 1}$ has an ω_1-branch, and $n \notin (x, m, k)$ if and only if $S^1_{\omega \cdot \gamma + 2n}$ has an ω_1-branch.

Indeed if $n \notin (x, m, k)$ then we added a branch through $S^1_{\omega \cdot \gamma + 2n}$. If on the other hand $S^1_{\omega \cdot \gamma + 2n}$ is Suslin in $L[X]$ then we must have added an ω_1-branch through $S^1_{\omega \cdot \gamma + 2n+1}$ as we always add an ω_1-branch through either $S^1_{\omega \cdot \gamma + 2n+1}$ or $S^1_{\omega \cdot \gamma + 2n}$ and adding branches through some S^1_α's will not affect that some S^1_β is Suslin in $L[X]$, as $\vec{S}^1$ is independent.

We note that we can apply an argument resembling David's trick [1] in this situation. We rewrite the information of $X \subset \omega_1$ as a subset $Y \subset \omega_1$ using the following line of reasoning. It is clear that any transitive, $\aleph_1$-sized model M of ZF^- which contains X will be able to correctly decode out of X all the information. Consequentially, if we code the model $(M, \in)$ which contains X as a set $X_M \subset \omega_1$, then for any uncountable β such that $L_\beta[X_M] \models \mathsf{ZF}^-$ and $X_M \in L_\beta[X_M]$:

$$L_\beta[X_M] \models \text{"The model decoded out of } X_M \text{ satisfies } (*) \text{ for every } \gamma \in h\text{"}.$$

In particular there will be an $\aleph_1$-sized ordinal β as above and we can fix a club $C \subset \omega_1$ and a sequence $(M_\alpha : \alpha \in C)$ of countable elementary submodels of $L_\beta[X_M]$ such that

$$\forall \alpha \in C (M_\alpha \prec L_\beta[X_M] \wedge M_\alpha \cap \omega_1 = \alpha)$$

Now let the set $Y \subset \omega_1$ code the pair (C, X_M) such that the odd entries of Y should code X_M and if $Y_0 := E(Y)$ where the latter is the set of even entries of Y and $\{c_\alpha : \alpha < \omega_1\}$ is the enumeration of C then

1. $E(Y) \cap \omega$ codes a well-ordering of type c_0.
2. $E(Y) \cap [\omega, c_0) = \emptyset$.
3. For all β, $E(Y) \cap [c_\beta, c_\beta + \omega)$ codes a well-ordering of type $c_{\beta+1}$.
4. For all β, $E(Y) \cap [c_\beta + \omega, c_{\beta+1}) = \emptyset$.

We obtain

(**) For any countable transitive model M of ZF^- such that $\omega_1^M = (\omega_1^L)^M$ and $Y \cap \omega_1^M \in M$, M can construct its version of the universe $L[Y \cap \omega_1^N]$, and the latter will see that there is an $\aleph_1^M$-sized transitive model $N \in L[Y \cap \omega_1^N]$ which models $(*)$ for (x, m, k) and every $\gamma \in h \cap M$.

Thus we have a local version of the property $(*)$.

In the next step $\dot{\mathbb{A}}(\dot{Y})$, working in $W[g]$, for $g \subset \mathbb{C}(\omega_1)$ generic over W, we use almost disjoint forcing $\mathbb{A}_D(Y)$ relative to our previously defined, almost disjoint family of reals $D \in L$ (see the paragraph after Definition 2.5)

[1] see [2] for the original argument, where the strings in Jensen's coding machinery are altered such that certain unwanted universes are destroyed. This destruction is emulated in our context as seen below.

to code the set Y into one real r. This forcing only depends on the subset of ω_1 we code, thus $\mathbb{A}_D(Y)$ will be independent of the surrounding universe in which we define it, as long as it has the right ω_1 and contains the set Y.

We finally obtained a real r such that

(∗∗∗) For any countable, transitive model M of ZF^- such that $\omega_1^M = (\omega_1^L)^M$ and $r \in M$, M can construct its version of $L[r]$ which in turn thinks that there is a transitive ZF^--model N of size $\aleph_1^M$ such that N believes (∗) for (x, m, k) and every $\gamma \in h \cap M$.

Note that (∗∗∗) is a Π^1_2-formula in the parameters r and (x, m, k), as the set $h \cap M \subset \omega_1^M$ is coded into r. We will often suppress the reals $r, (x, m, k)$ when referring to (∗∗∗) as they will be clear from the context. We say in the above situation that the real (x, m, k) *is written into* $\vec{S}^1$, or that (x, m, k) *is coded into* $\vec{S}^1$ and r witnesses that (x, m, k) is coded. Likewise a forcing $\mathbb{P}_{(x,m,k),0}$ is defined for coding the real (x, m, k) into $\vec{S}^0$.

The projective and local statement (∗∗∗), if true, will determine how certain inner models of the surrounding universe will look like with respect to branches through $\vec{S}$. That is to say, if we assume that (∗∗∗) holds for a real (x, m, k) and is the truth of it is witnessed by a real r. Then r also witnesses the truth of (∗∗∗) for any transitive ZF^--model M which contains r (i.e. we can drop the assumption on the countability of M). Indeed if we assume that there would be an uncountable, transitive M, $r \in M$, which witnesses that (∗∗∗) is false. Then by Löwenheim-Skolem, there would be a countable $N \prec M$, $r \in N$ which we can transitively collapse to obtain the transitive $\bar{N}$. But $\bar{N}$ would witness that (∗∗∗) is not true for every countable, transitive model, which is a contradiction.

Consequentially, the real r carries enough information that the universe $L[r]$ will see that certain trees from $\vec{S}^1$ have branches in that

$$n \in w = (x, y, m) \Rightarrow L[r] \models \text{“}S^1_{\omega\gamma+2n+1} \text{ has an } \omega_1\text{-branch”}.$$

and

$$n \notin w = (x, y, m) \Rightarrow L[r] \models \text{“}S^1_{\omega\gamma+2n} \text{ has an } \omega_1\text{-branch”}.$$

Indeed, the universe $L[r]$ will see that there is a transitive ZF^--model N which believes (∗) for every $\gamma \in h \subset \omega_1$, the latter being coded into r. But by upwards Σ_1-absoluteness, and the fact that N can compute $\vec{S}^1$ correctly, if N thinks that some tree in $\vec{S}^1$ has a branch, then $L[r]$ must think so as well.

Next we define the set of forcings which we will use in our proof. We aim to iterate the coding forcings we defined in the last section. As the first factor is always $(\mathbb{C}(\omega_1))^L$, the iteration we aim for is actually a hybrid of an iteration and a product. We shall use a mixed support, that is we use countable support on the product-like coordinates which use $(\mathbb{C}(\omega_1))^L$, and finite support on the iteration-like coordinates which use almost disjoint coding forcing.

Definition 3.1. *A mixed support iteration $\mathbb{P} = (\mathbb{P}_\beta : \beta < \alpha)$ is called allowable (or 0-allowable, to anticipate later developments) if $\alpha < \omega_1$ and there exists a bookkeeping function $F : \alpha \to H(\omega_2)^2$ such that $\mathbb{P}$ is defined inductively using F as follows:*

- *If $F(0) = (x, i)$, where x is a real, $i \in \{0, 1\}$, then $\mathbb{P}_0 = \mathrm{Code}(x, i)$. Otherwise $\mathbb{P}_0$ is the trivial forcing.*
- *Assume that $\beta > 0$ and $\mathbb{P}_\beta$ is defined, $G_\beta \subset \mathbb{P}_\beta$ is a generic filter over W. Moreover assume that $F(\beta) = (\dot{x}, i)$, where $\dot{x}$ is a $\mathbb{P}_\beta$-name of a real, and i is a $\mathbb{P}_\beta$-name of an element of $\{0, 1\}$ and $\dot{x}^{G_\beta} = x$. Then let $\mathbb{P}(\beta) = \mathrm{Code}(x, i^{G_\beta}) = \mathbb{C}(\omega_1))^L * \mathbb{A}(Y)$, for the reshaped $Y \subset \omega_1$ as being defined in the last section, and let $\mathbb{P}_{\beta+1} = \mathbb{P}_\beta \times (\mathbb{C}(\omega_1))^L * \dot{\mathbb{A}}(\dot{Y})$. Otherwise we force with just $(\mathbb{C}(\omega_1))^L$.*

We use finite support on the iteration-like parts where almost disjoint coding is used and countable support on the product-like parts where ω_1-Cohen forcing, as computed in L is used.

Informally speaking, a (0-) allowable forcing just decides to code the reals which the bookkeeping F provides into either $\vec{S}^0$ or $\vec{S}^1$. Note that the notion of allowable can be defined in exactly the same way over any $W[G]$, where G is a $\mathbb{P}$-generic filter over W for an allowable forcing.

We also add that we could have defined allowable in an equivalent way if we first added, over L, ω_1-many Cohen subsets of ω_1, $\vec{C} = (C_\alpha : \alpha < \omega_1)$ with a countably supported product, then, in a second step destroy all the Suslin trees from $\vec{S}$ (note $\vec{S}$ remains independent after adding the ω_1-many ω_1-Cohen subsets) and dub the resulting universe W'. Then we can define allowable over the new ground model W' as just a finitely supported iteration of almost disjoint coding forcings which select at each step injectively one element C from $\vec{C}$ and the real given by the bookkepping F and the $i \in \{0, 1\}$ and then code up all the branches of the trees from $\vec{S}^0$ or $\vec{S}^1$ according to the real x we code for every ω-block with starting value in $h \subset \omega_1$ derived from C as in the last section. That is to say, we could have moved the product factors in an iteration of allowable forcings right at the beginning of our iteration, which we are allowed to do anyway, as it is a product. Our current and equivalent approach is a bit easier in terms of notation for later parts of the proof, so we defined allowable the way we did.

We obtain the following first properties of allowable forcings:

Lemma 3.2. *1. If $\mathbb{P} = (\mathbb{P}(\beta) : \beta < \delta) \in W$ is allowable then for every $\beta < \delta$, $\mathbb{P}_\beta \Vdash |\mathbb{P}(\beta)| = \aleph_1$, thus every factor of $\mathbb{P}$ is forced to have size $\aleph_1$.*

2. Every allowable forcing over W preserves ω_1.

3. The product of two allowable forcings is allowable again.

Proof. The first assertion follows immediately from the definition.

To see the second item we exploit some symmetry. Indeed, every allowable $\mathbb{P} = *_{\beta<\delta}P(\beta) = *_{\beta<\delta}(((\mathbb{C}(\omega_1))^L * \dot{\mathbb{A}}(\dot{Y}_\beta)) \in W$ can be rewritten as $(\prod_{\beta<\delta}(\mathbb{C}(\omega_1))^L) * (*_{\beta<\delta}\dot{\mathbb{A}}_D(\dot{Y}_\beta))$ (again with countable support on the $(\mathbb{C}(\omega_1))^L$ part and finite support on the almost disjoint coding forcings). Using that $W = L[\prod_{\alpha\in\omega_1} S_\alpha]$ we can write $\mathbb{P}$ as a forcing over L as follows: $(\prod_{\alpha\in\omega_1} S_\alpha) \times \prod_{\beta<\delta}(\mathbb{C}(\omega_1))^L) * (*_{\beta<\delta}\dot{\mathbb{A}}_D(\dot{Y}_\beta))$. This is the same as $(\prod_{\beta<\delta}(\mathbb{C}(\omega_1))^L \times \prod_{\alpha\in\omega_1} S_\alpha)\ ) * (*_{\beta<\delta}\dot{\mathbb{A}}_D(\dot{Y}_\beta))$

The latter representation is easily seen to be of the form $\mathbb{P}_1 \times \mathbb{P}_2 * (*_{\beta<\delta}\dot{\mathbb{A}}_D(\dot{Y}_\beta))$, where $\mathbb{P}_1$ is σ-closed, $\mathbb{P}_2$ has the ccc, and the third part is a finite support iteration of ccc forcings, hence ω_1 is preserved.

To see that the third item is true, we note that the definition of almost disjoint coding forcing only depends on the subset of ω_1 we want to code and is independent of the surrounding universe $V \supset W$ over which it is defined as long as $Y \in V$. In particular, if $(Y_\alpha \subset \omega_1 : \alpha < \beta)$ is a sequence of subsets of ω_1 in some ground model, then the finitely supported iteration $*_{\alpha<\beta}\dot{\mathbb{A}}(\check{Y}_\alpha)$ is isomorphic to the finitely supported product $\prod_{\alpha<\beta}\mathbb{A}(Y_\alpha)$. So we immediately see that if

$$\mathbb{P}^1 = *_{\beta<\delta^1}P(\beta) = \prod_{\beta<\delta^1}(((\mathbb{C}(\omega_1))^L) *_{\beta<\delta^1} \dot{\mathbb{A}}(\dot{Y}_\beta))$$

and

$$\mathbb{P}^2 = *_{\beta<\delta^2}P(\beta) = \prod_{\beta<\delta^2}(((\mathbb{C}(\omega_1))^L *_{\beta<\delta^2} \dot{\mathbb{A}}(\dot{Y}_\beta))$$

then

$$\mathbb{P}^1 \times \mathbb{P}^2 = \prod_{\beta<\delta^1+\delta^2}((\mathbb{C}(\omega_1))^L *_{\beta<\delta^1} \dot{\mathbb{A}}(\dot{Y}_\beta)) *_{\beta<\delta^2} \dot{\mathbb{A}}(\dot{Y}_\beta)$$

which is allowable. □

The proof of the second assertion of the last lemma immediately gives us the following:

Corollary 3.3. *Let $\mathbb{P} = (\mathbb{P}(\beta) : \beta < \delta) \in W$ be an allowable forcing over W. Then $W[\mathbb{P}] \models \mathsf{CH}$. Further, if $\mathbb{P} = (\mathbb{P}(\alpha) : \alpha < \omega_1) \in W$ is an ω_1-length iteration such that each initial segment of the iteration is allowable over W, then $W[\mathbb{P}] \models \mathsf{CH}$.*

Let $\mathbb{P} = (\mathbb{P}(\beta) : \beta < \delta)$ be an allowable forcing with respect to some $F \in W$. The set of (names of) reals which are enumerated by F we call the set of reals coded by $\mathbb{P}$. That is, for every β, if we let $\dot{x}_\beta$ be the (name) of a real listed by $F(\beta)$ and if we let $G \subset \mathbb{P}$ be a generic filter over W and finally if we let $\dot{x}_\beta^G =: x_\beta$, then we say that $\{x_\beta : \beta < \alpha\}$ is the set of reals coded by $\mathbb{P}$ and G (though we will suppress the G). Next we show, that iterations of 0-allowable forcings will not add unwanted witnesses to the Σ^1_3-formula $\psi(w,i)$ which corresponds to the formula $(***)$:

$$\psi(w,i) \equiv \exists r \forall M (M \text{ is countable and transitive and } M \models \mathsf{ZF}^- \\ \text{and } \omega_1^M = (\omega_1^L)^M \text{ and } r, w \in M \rightarrow M \models \varphi(w,i))$$

where $\varphi(w,i)$ asserts that in M's version of $L[r]$, there is a transitive, $\aleph_1^M$-sized ZF^--model which witnesses that w is coded into $\vec{S}^i$.

Lemma 3.4. *If $\mathbb{P} \in W$ is allowable, $\mathbb{P} = (\mathbb{P}_\beta : \beta < \delta)$, $G \subset \mathbb{P}$ is generic over W and $\{x_\beta : \beta < \delta\}$ is the set of reals which is coded by $\mathbb{P}$. Let $\psi(v_0)$ be the distinguished formula from above. Then in $W[G]$, the set of reals which satisfy $\psi(v_0)$ is exactly $\{x_\beta : \beta < \delta\}$, that is, we do not code any unwanted information accidentally.*

Proof. Let G be $\mathbb{P}$ generic over W. Let $g = (g_\beta : \beta < \delta)$ be the set of the δ many ω_1 subsets added by the $(\mathbb{C}(\omega_1))^L$-part of the factors of $\mathbb{P}$. We let $\rho : ([\omega_1]^\omega)^L \rightarrow \omega_1$ be our fixed, constructible bijection and let $h_\beta = \{\rho(g_\beta \cap \alpha) : \alpha < \omega_1\}$. Note that the family $\{h_\beta : \beta < \delta\}$ forms an almost disjoint family of subsets of ω_1. Thus there is $\alpha < \omega_1$ such that for arbitrary distinct β_1, $\beta_2 < \delta$, $\alpha > h_{\beta_1} \cap h_{\beta_2}$ and additionally, assume that α is an index which does not show up in the set of indices of the trees we code with $\mathbb{P}$.

We let $S^1_\alpha \in \vec{S}^1$. We claim that there is no real in $W[G]$ such that $W[G] \models L[r] \models$ "S^1_α has an ω_1-branch". We show this by pulling out the forcing S^1_α out of $\mathbb{P}$. Indeed if we consider $W[\mathbb{P}] = L[\mathbb{Q}^0][\mathbb{Q}^1][\mathbb{Q}^2][\mathbb{P}]$, and if S^1_α is as described already, we can rearrange this to $W[\mathbb{P}] = L[\mathbb{Q}^0][\mathbb{Q}'^1 \times S^1_\alpha][\mathbb{Q}^2][\mathbb{P}] = W[\mathbb{P}'][S^1_\alpha]$, where $\mathbb{Q}'^1$ is $\prod_{\beta \neq \alpha} S^1_\beta$ and $\mathbb{P}'$ is $\mathbb{Q}^0 * \mathbb{Q}'^1 * \mathbb{Q}^2 * \mathbb{P}$.

Note now that, as S^1_α is ω-distributive, $2^\omega \cap W[\mathbb{P}] = 2^\omega \cap W[\mathbb{P}']$, as S_α is still a Suslin tree in $W[\mathbb{P}']$ by the fact that $\vec{S}^0$ and $\vec{S}^1$ are independent, and no factor of $\mathbb{P}'$ besides the trees from $\vec{S}^0$ and $\vec{S}^1$ used in $\mathbb{P}'$ destroys Suslin trees. But this implies that

$$W[\mathbb{P}'] \models \neg \exists r L[r] \models \text{"}S^1_\alpha \text{ has an } \omega_1\text{-branch"}$$

as the existence of an ω_1-branch through S^1_α in the inner model $L[r]$ would imply the existence of such a branch in $W[\mathbb{P}']$. Further and as no new reals appear when passing to $W[\mathbb{P}]$ we also get

$$W[\mathbb{P}] \models \neg \exists r L[r] \models \text{"}S^1_\alpha \text{ has an } \omega_1\text{-branch"}.$$

On the other hand any unwanted information, i.e. any $(x,m) \notin \{(x_\beta, m_\beta) : \beta < \delta\}$ such that $W[G] \models \psi((x,i,m))$ will satisfy that there is a real r such that

$$n \in (x,i,m) \rightarrow L[r] \models \text{“}S^1_{\omega\gamma+2n+1} \text{ has an } \omega_1\text{-branch”}$$

and

$$n \notin (x,i,m) \rightarrow L[r] \models \text{“}S^1_{\omega\gamma+2n} \text{ has an } \omega_1\text{-branch”}.$$

by the discussion of the last subsection for ω_1-many γ's.

But by the argument above, only trees which we used in one of the factors of $\mathbb{P}$ have this property, so there can not be unwanted codes on the $\vec{S}^1$-side. But the very same argument shows the assertion also for the $\vec{S}^0$-side. So for our fixed α, there is no real r which codes an ω_1 branch over $L[r]$. But any unwanted information would need not only one but even $\aleph_1$-many such α's chosen as above. This shows that there can not be unwanted information in $W[G]$, as claimed.

$\square$

3.1 α-allowable forcings

The notion of 0-allowable will form the base case of an inductive definition. Let $\alpha \geqslant 0$ be an ordinal and assume we defined already the notion of α-allowable. Then we can inductively define the notion of $\alpha+1$-allowable as follows.

Suppose that $\gamma < \omega_1$, F is a bookkeeping function,

$$F : \gamma \rightarrow H(\omega_2)^5$$

and

$$\mathbb{P} = (\mathbb{P}_\beta \,:\, \beta < \gamma)$$

is a allowable forcing relative to F (in fact relative to some bookkeeping F' determined by F in a unique way - the difference here is not relevant).

Suppose that

$$E = E_0 \cup E_1 = \{(\dot{y}_\delta, m_\delta, k_\delta) \,:\, \delta \leqslant \alpha\} \cup \{(\dot{x}_\delta, i_\delta) \,:\, \delta \leqslant \alpha, i_\delta \in \{0,1\}\}$$

where $m_\delta, k_\delta \in \omega$ and every $\dot{x}_\delta, \dot{y}_\delta$ is a $\mathbb{P}$-name of a real and for every two ordinals $\beta, \gamma < \alpha$, if $\dot{y}_\beta$ and $\dot{y}_\gamma$ are not the empty set, then $\mathbb{P} \Vdash (\dot{y}_\beta, m_\beta, k_\beta) \neq (\dot{y}_\gamma, m_\gamma, k_\gamma)$. Intuitively, E_0 will serve as the set of pairs of boldface Σ^1_3-sets, for which we already obtained rules which allow us to separate them; whereas E_1 is the set of (names of) reals which we decided to never code along our α-allowable iteration using the coding forcing $\text{Code}(x,a,b,i)$ for two fixed natural numbers a and b. The latter plays a role in establishing the eventual failure of Π^1_3-reduction.

Suppose that for every $\delta \leqslant \alpha$, $(\mathbb{P}_\beta \,:\, \beta < \gamma)$ is δ-allowable with respect to $E \upharpoonright \delta = (E_0 \upharpoonright \delta) \cup (E_1 \upharpoonright \delta) = \{(\dot{y}_\eta, m_\eta, k_\eta) \,:\, \eta < \delta\} \cup \{\dot{x}_\eta \,:\, \eta < \delta\}$ and F.

We assume first that $\dot{x}_{\alpha+1}$ is the empty set and $\dot{y}_{\alpha+1}$ is a $\mathbb{P}$-name for a real and $m_{\alpha+1}, k_{\alpha+1} \in \omega$ such that $\mathbb{P} \Vdash \forall \delta \leqslant \alpha((\dot{y}_\delta, m_\delta, k_\delta) \neq (\dot{y}_{\alpha+1}, m_{\alpha+1}, k_{\alpha+1}))$. Then we say that $(\mathbb{P}_\beta : \beta < \gamma)$ is $\alpha + 1$-allowable with respect to $E \cup \{\dot{y}_{\alpha+1}, m_{\alpha+1}, k_{\alpha+1})\}$ and F if it obeys the following rules.

1. Whenever $\beta < \gamma$ is odd and such that there is a $\mathbb{P}_\beta$-name $\dot{x}$ of a real and a $\mathbb{P}_\beta$-name for an integer i such that

$$F(\beta) = (\dot{x}, \dot{y}_{\alpha+1}, m_{\alpha+1}, k_{\alpha+1}, i)$$

and $\dot{y}_{\alpha+1}$ is in fact a $\mathbb{P}_\beta$-name, and for G_β a $\mathbb{P}_\beta$-generic over W, $W[G_\beta]$ thinks that

$$\exists \mathbb{Q}(\mathbb{Q} \text{ is } \alpha\text{-allowable with respect to } E \wedge \\ \mathbb{Q} \Vdash x \in A_m(y_{\alpha+1})),$$

where $x = \dot{x}^G$, and $y_\alpha = \dot{y}^G_{\alpha+1}$. Then continuing to argue in $W[G_\beta]$, we let

$$\mathbb{P}(\beta) = \text{Code}((x, y, m, k), 0).$$

Note that we confuse here the quadruple (x, y, m, k) with one real which codes this quadruple.

2. Whenever $\beta < \gamma$ is such that there is a $\mathbb{P}_\beta$-name $\dot{x}$ of a real and a $\mathbb{P}_\beta$-name i of an integer in $\{0, 1\}$ such that

$$F(\beta) = (\dot{x}, \dot{y}_{\alpha+1}, m_{\alpha+1}, k_{\alpha+1}, i)$$

and for G_β which is $\mathbb{P}_\beta$-generic over W, $W[G_\beta]$ thinks that

$$\forall \mathbb{Q}_1(\mathbb{Q}_1 \text{ is } \alpha\text{-allowable with respect to } E \\ \rightarrow \neg(\mathbb{Q}_1 \Vdash x \in A_m(\dot{y}_{\alpha+1})))$$

but there is a forcing $\mathbb{Q}_2$ such that $W[G_\beta]$ thinks that

$$\mathbb{Q}_2 \text{ is } \alpha\text{-allowable with respect to } E \text{ and} \\ \mathbb{Q}_2 \Vdash x \in A_k(\dot{y}_{\alpha+1})$$

Then continuing to argue in $W[G_\beta]$, we force with

$$\mathbb{P}(\beta) := \text{Code}((x, y, m, k), 1).$$

Note that we confuse here again the quadruple (x, y, m, k) with one real w which codes this quadruple.

3. If neither 1 nor 2 is true, then either
$$\mathbb{P}(\beta) = \text{Code}((x, y, m, k), 1)$$
or
$$\mathbb{P}(\beta) = \text{Code}((x, y, m, k), 0)$$
depending on whether $i^{G_\beta} \in \{0, 1\}$ was 0 or 1.

4. If $F(\beta) = (\dot{x}, \dot{y}, m, k, i)$ and for our $\mathbb{P}_\beta$-generic filter G, $W[G] \models \forall \delta \leqslant \alpha + 1((\dot{y}^G, m, k) \notin E^G)$, then, working over $W[G_\beta]$ let
$$\mathbb{P}(\beta) = \text{Code}((x, y, m, k), i^{G_\beta})$$
depending on whether $i^{G_\beta} \in \{0, 1\}$ was 0 or 1.

If, on the other hand $\dot{y}_{\alpha+1}$ is the empty set as is $m_{\alpha+1}, k_{\alpha+1}$ and $\dot{x}_{\alpha+1}$ is the $\mathbb{P}$-name of a real, then we define $\alpha + 1$-allowable with respect to F and $E \cup \{(\dot{x}_{\alpha+1}, i)\}$, where $i \in \{0, 1\}$ to be α-allowable relative to E and F plus the additional rule, that we will not use factors in our iteration which contain $\text{Code}(\dot{x}_{\alpha+1}, a, b, i)$.

This ends the definition for the successor step $\alpha \to \alpha + 1$. For limit ordinals α, we say that a allowable forcing $\mathbb{P}$ is α allowable with respect to E and F if for every $\eta < \alpha$, $(\mathbb{P}_\beta : \beta < \gamma)$ is η-allowable with respect to $E \upharpoonright \eta$ and some F'.

We add a couple of remarks concerning the last definition.

- By definition, if $\delta_2 < \delta_1$ and $\mathbb{P}_1$ is δ_1-allowable with respect to $E = \{(\dot{y}_\beta, m_\beta, k_\beta) : \beta \leqslant \delta_1\} \cup \{(\dot{x}_\beta, i_\beta) : \beta \leqslant \delta_1$ and some F_1, then $\mathbb{P}_1$ is also δ_2-allowable with respect to $E \upharpoonright \delta_2 = \{(\dot{y}_\beta, m_\beta, k_\beta) : \beta \leqslant \delta_2\} \cup \{(\dot{x}_\beta, i_\beta) : \beta \leqslant \delta_2, i_\beta \in \{0, 1\}\}$ and an altered bookkeeping function F'.
- The notion of α-allowable can be defined in a uniform way over any allowable extension W' of W.
- We will often just say that some iteration $\mathbb{P}$ is α-allowable, by which we mean that there is a set E and a bookkeeping F such that $\mathbb{P}$ is α-allowable with respect to E and F.

4 Closure under products

Lemma 4.1. *Let α be an ordinal, assume that W' is some α-allowable generic extension of W, and that $\mathbb{P}^1 = (\mathbb{P}^1_\beta : \beta < \delta)$ and $\mathbb{P}^2 = (\mathbb{P}^2_\beta : \beta < \delta)$ are two α-allowable forcings over W' with respect to a common set $E = E_0 \cup E_1 = \{(\dot{y}_\delta, m_\delta, k_\delta) : \delta < \alpha\} \cup \{(\dot{x}_\delta, i_\delta) : \delta < \alpha\}$ and bookkeeping functions F_1 and F_2 respectively. Then there is a bookkeeping function F such that $\mathbb{P}_1 \times \mathbb{P}_2$ is α-allowable over W' with respect to E and F.*

Proof. We define $F \upharpoonright \delta_1$ to be F_1. For values $\delta_1 + \beta > \delta_1$ we let $F(\delta_1 + \beta)$ be such that its value on the first four coordinates equal the first four coordinates of $F_2(\beta)$, i.e. $F(\delta_1 + \beta) = (\dot{x}, \dot{y}, m, k, i)$ for some $i \in \{1, 2\}$ where $F_2(\beta) = (\dot{x}, \dot{y}, m, k, i')$. We claim now that we can define the remaining value of $F(\beta)$, in such a way that the lemma is true. This is shown by induction on $\beta < \delta_2$.

First we note that for $E = E_0 \cup E_1$ we can fully concentrate on the set E_0 in our argument, that is the odd stages β of our iteration, as E_1 defines a set of coding forcings we must not use in an α-allowable forcing with respect to E, and this is clearly closed under products.

Let $(\mathbb{P}_2)_\beta$ be the iteration of $\mathbb{P}_2$ up to the odd stage $\beta < \delta_2$. Assume, that $\mathbb{P}_1 \times (\mathbb{P}_2)_\beta$ is in fact an α-allowable forcing relative to E and F. Then we have that $F(\delta_1 + \beta) \upharpoonright 5 = F_2(\beta) \upharpoonright 5 = (\dot{x}, \dot{y}, m, k)$, and we claim that at that odd stage,

Claim 1. *If we should apply case 1,2, 3, or 4 when considering the forcing $\mathbb{P}_1 \times \mathbb{P}_2$ as an α-allowable forcing relative to $E = E_0 \cup E_1$ over the model W', we must apply the same case when considering $\mathbb{P}_2$ as an α-allowable forcing over the model W' relative to E.*

Once the claim is shown, the lemma can be proven as follows by induction on $\beta < \delta_2$: we work in the model $W'[\mathbb{P}_1][(\mathbb{P}_2)_\beta]$, consider $F(\delta_1 + \beta) \upharpoonright 5 = F_2(\beta) \upharpoonright 5$, and ask which of the four cases has to be applied. By the claim, it will be the same case, as when considering $\mathbb{P}_2$ over W' as an α-allowable forcing relative to E and F_2. In particular the forcing $\mathbb{P}_2(\beta)$ we define at stage β will be a choice obeying the rules of α-allowable even when working over the model $W'[\mathbb{P}_1][(\mathbb{P}_2)_\beta]$. This shows that $\mathbb{P}_1 \times \mathbb{P}_2$ is an α-allowable forcing relative to E and some F over W'.

The proof of the claim is via induction on α. So we assume that $\alpha = 1$ and both $\mathbb{P}_1$ and $\mathbb{P}_2$ are 1-allowable with respect to $E = E_0 \cup E_1$. As the case $E = E_1$ is clear, we can assume that $E = E_0 = \{\dot{y}, m, k\}$. We shall show that there is a bookkeeping F such that $(\mathbb{P}_2)_\beta : \beta < \delta_2\})$ is still 1-allowable with respect to E, even when considered in the universe $W'[\mathbb{P}_1]$. We assume first that at stage $\delta_1 + \beta$ of $\mathbb{P}_1 \times \mathbb{P}_2$ case 1 in the definition of 1-allowable applies, when working in the model $W'[\mathbb{P}_1][(\mathbb{P}_2)_\beta]$ relative to E and F. Thus

$$F(\beta) \upharpoonright 5 = (\dot{x}, \dot{y}, m, k)$$

and $(\dot{y}, m, k) \in E$ and for any $G^1 \times G_\beta$ which is $\mathbb{P}_1 \times (\mathbb{P}_2)_\beta$-generic over W', if $\dot{x}^{G_\beta} = x$ and $\dot{y}^{G_\beta} = y$, the universe $W'[G_1 \times G_\beta]$ thinks that

$$\exists \mathbb{Q} (\mathbb{Q} \text{ is 0-allowable with respect to } E \text{ and some F } \wedge \\ \mathbb{Q} \Vdash x \in A_m(y)).$$

Thus, if we work over $W'[G_\beta]$ instead it will think

$$\exists (\mathbb{P}_1 \times \mathbb{Q})(\mathbb{P}_1 \times \mathbb{Q} \text{ is 0-allowable } \wedge \\ \mathbb{P}_1 \times \mathbb{Q} \Vdash x \in A_m(y)).$$

Thus, at stage β, we are in case 1 as well, when considering $\mathbb{P}_2$ as an 1-allowable forcing over W' relative to E.

If, at stage β, case 2 applies, when considering $\mathbb{P}_1 \times \mathbb{P}_2$ as a 1-allowable forcing with respect to E over W', then we argue first that case 1 is impossible when considering $\mathbb{P}_2$ as a 1-allowable forcing over W'. Indeed, assume for a contradiction that case 1 must be applied, then, by assumption, $\mathbb{P}_2(\beta)$ will force that $x \in A_m(y)$. Yet, by Shoenfield absoluteness, $\mathbb{P}_2(\beta)$ would witness that we are in case 1 at stage β when considering $\mathbb{P}_1 \times \mathbb{P}_2$ as 1-allowable with respect to E over W', which is a contradiction.

Thus we can not be in case 1 and we shall show that we are indeed in case 2, i.e. there is a 0-allowable forcing $\mathbb{Q}$, such that $\mathbb{Q} \Vdash x \in A_k(y)$, but such a $\mathbb{Q}$ exists, namely $\mathbb{P}_2(\beta)$,

Finally, if at stage β, case 3 applies when considering $\mathbb{P}_2$ as a 1-allowable forcing with respect to E over $W'[\mathbb{P}_1]$, we claim that we must be in case 3 as well, when considering $\mathbb{P}_2$ over just W'. If not, then we would be in case 1 or 2 at β. Assume without loss of generality that we were in case 1, then, as by assumption $\mathbb{P}_2$ is 1-allowable over W', $\mathbb{P}_2(\beta)$ will force $\Vdash x \in A_m(y)$. But this is a contradiction, so we must be in case 3 as well. This finishes the proof of the claim for $\alpha = 1$.

We shall argue now that the Claim is true for $\alpha + 1$-allowable forcings provided we know that it is true for α-allowable forcings. Again we can focus on the case when $\alpha + 1$ allowable forcings are obtained via enlarging E_0, as enlarging E_1 just means to avoid certain coding forcings, which is trivial to be closed under products. We shall show the claim via induction on β. So assume that $\mathbb{P}_1 \times (\mathbb{P}_2)_\beta$ is $\alpha+1$-allowable with respect to $E = E \upharpoonright \alpha \cup \{(\dot{y}, m_\alpha, k_\alpha)\}$ and an F whose domain is $\delta_1 + \beta$. We look at

$$F(\delta_1 + \beta) \upharpoonright 5 = F_2(\beta) \upharpoonright 5 = (\dot{x}, \dot{y}, m_\alpha, k_\alpha)$$

We concentrate on the case where β is such that case 2 applies when considering $\mathbb{P}_1 \times (\mathbb{P}_2)_\beta$ over W'. The rest follows similarly. Our goal is to show that case 2 must apply when considering the β-th stage of the forcing using F_2 and E over $W'[(\mathbb{P}_2)_\beta]$ as well.

Assume first for a contradiction, that, when working over $W'[(\mathbb{P}_2)_\beta]$, at stage β, case 1 applies. Then, for any $(\mathbb{P}_2)_\beta$-generic filter G_β over W',

$$\begin{aligned} W'[G_\beta] \models \exists \mathbb{Q}(&\mathbb{Q} \text{ is } \alpha\text{-allowable with respect to } E \upharpoonright \alpha \text{ and some } F' \text{ and} \\ &\mathbb{Q} \Vdash x \in A_m(y)) \end{aligned}$$

Now, as $\mathbb{P}_2$ is α-allowable, we know that $\mathbb{P}_2(\beta)$ is such that $\mathbb{P}_2(\beta) \Vdash x \in A_m(y)$.

Thus, using the upwards-absoluteness of Σ^1_3-formulas, at stage β of the $\alpha + 1$-allowable forcing determined by F and E, there is an α-allowable forcing $\mathbb{Q}$ with respect to $E \upharpoonright \alpha$ which forces $x \in A_m(y)$, namely $\mathbb{P}_2(\beta)$.

But this is a contradiction, as we assumed that when considering $\mathbb{P}_1 \times (\mathbb{P}_2)_\beta$ over W' at stage β, case 1 does not apply, hence such an α-allowable forcing should not exist.

So we know that case 1 is not true. We shall show now that case 2 must apply at stage β when considering $\mathbb{P}_2$ over the universe W'. By assumption we know that

$$W'[\mathbb{P}_1][(\mathbb{P}_2)_\beta] \models \exists \mathbb{Q}_2 (\mathbb{Q}_2 \text{ is } \alpha\text{-allowable with respect to } E \upharpoonright \alpha \text{ and } \\ \mathbb{Q}_2 \Vdash x \in A_k(y)$$

As $\mathbb{P}_1$ is $\alpha + 1$-allowable with respect to E and F_1, it is also α-allowable with respect to $E \upharpoonright \alpha$ and some altered F_1', thus, as a consequence from the induction hypothesis, we obtain that

$$W'[(\mathbb{P}_2)_\beta] \models \mathbb{P}_1 \times \mathbb{Q}_2 \text{ is } \alpha\text{-allowable and } \mathbb{P}_1 \times \mathbb{Q}_2 \Vdash x \in A_k(y).$$

But then, $\mathbb{P}_1 \times \mathbb{Q}_2$ witnesses that we are in case 2 as well when at stage β of $\mathbb{P}_2$ over W'. This ends the proof of the claim and so we have shown the lemma. □

5 Ideas for the proof

This section will be used to briefly explain the set up and the structure of the proof of the main theorem. There are two goals we aim to accomplish. First, we want to force $\boldsymbol{\Sigma^1_3}$-separation. For this, we will use an ω_1-length iteration. We list all Σ^1_3-formulas in two free variables $(\varphi_n(v_0, v_1) \mid n \in \omega)$. We shall use a bookkeeping device which enumerates simultaneously in an ω_1-length list all pairs of natural numbers $(m, k) \in \omega^2$ and (names of) reals $\dot{y}$. These objects $m, k, \dot{y}$ correspond to pairs of $\boldsymbol{\Sigma^1_3}$-sets $A_m(\dot{y})$ and $A_k(\dot{y})$ (the (name of a) real $\dot{y}$ serves as a parameter in the k-th and m-th Σ^1_3-formula φ_m and φ_k) we want to separate.

At the same time we want to create a universe over which there are two (lightface) Π^1_3-sets B_0 and B_1 which we will design in such a way, that no (boldface) pair of Π^1_3-sets exists, which reduces B_0 and B_1.

We settle to work towards Σ^1_3-separation on the odd stages of our iteration, whereas we work towards a failure of Π^1_3-reduction on the even stages of the iteration. The iteration itself will consist of the coding forcings $\text{Code}(x, y, m, k)$ applied over L to make certain reals of the form (x, y, m, k) to satisfy our two Σ^1_3-formulas $\Phi_0(x, y, m, k)$ or $\Phi_1(x, y, m, k)$. The final goal is that for any fixed pair of natural numbers m, k, and any parameter $y \in \omega^\omega$, there is a real parameter $R_{y,m,k}$ and a fixed Σ^1_3 formula σ such that the sets

$$D^0_{y,m,k}(R_{y,m,k}) := \{x \mid \Phi_0(x, y, m, k) \wedge \sigma(x, R_{y,m,k})\}$$

and

$$D^1_{y,m,k}(R_{y,m,k}) := \{x \mid \Phi_1(x, y, m, k) \wedge \sigma(x, R_{y,m,k})\}$$

will become the separating sets for the pair of $\Sigma^1_3(y)$-definable sets $A_m(y)$ and $A_k(y)$, i.e. $A_m(y) \subset D^0_{y,m,k}, A_k(y) \subset D^1_{y,m,k}$ and $D^0_{y,m,k}(R_{y,m,k}) \cup D^1_{y,m,k}(R_{y,m,k}) = \omega^\omega$ and $D^0_{y,m,k}(R_{y,m,k}) \cap D^1_{y,m,k}(R_{y,m,k}) = \emptyset$.

At the same time we need to work towards a failure of Π^1_3-reduction which will be done on the even stages of the iteration. We aim to accomplish the failure of Π^1_3-reduction via exhibiting two Π^1_3-sets (lightface) B_0 and B_1 which are chosen in such a way that the question of whether some real x is in B_0 or B_1 can be changed using the coding forcings Code without interfering with the coding forcings we have to use in order to work for the $\boldsymbol{\Sigma^1_3}$-separation. This freedom will be used to define our ω_1-length iteration of coding forcings so that eventually B_0 and B_1 can not be reduced by any pair of (boldface) $\boldsymbol{\Pi^1_3}$-sets, thus yielding a slightly stronger failure than just a failure of $\boldsymbol{\Pi^1_3}$-reduction.

6 The first step of the iteration

We let
$$\vec{\varphi} := (\varphi_n(v_0, v_1) \mid n \in \omega)$$
be a fixed recursive list of the Σ^1_3-formulas in two free variables. We allow that v_0 or v_1 actually do not appear in some of the φ_n's, so our list also contains all Σ^1_3-formulas in one free variable. We use ψ_n to denote $\neg\varphi_n$, i.e. ψ_n is the n-th Π^1_3-formula in the recursive list of Π^1_3-formulas induced by $\vec{\varphi}$.

We fix two Σ^1_3-formulas φ_a, φ_b which provably have non-empty intersection, e.g. $\varphi_a(v_0) = \exists v_0(v_0 = v_0)$ and $\varphi_b = \exists v_0(v_0 = v_0 \wedge v_0 = 1)$. As a consequence, we need not to separate A_a and A_b, thus coding forcings of the form $\text{Code}(\dot{x}, a, b, i)$ for any (name of a) real $\dot{x}$, for a, b our fixed natural numbers and for $i \in \{0, 1\}$ can be used freely in our definition of the iteration to come.

Next we assume for notational simplicity that $\neg\varphi_0$ and $\neg\varphi_1$, i.e. the negation of the first and the negation of the second formula in our list look like this:

$$\neg\varphi_0(v_0) = \psi_0(v_0) = \text{“}(v_0, a, b) \text{ is not coded into the } \vec{S}^0\text{-sequence”}$$
$$\neg\varphi_1(v_0) = \psi_1(v_0) = \text{“}(v_0, a, b) \text{ is not coded into the } \vec{S}^1\text{-sequence”}$$

The resulting sets will be defined like this:

$$B_0 = \{x \in \omega^\omega \mid \psi_0(x)\}$$
$$B_1 = \{x \in \omega^\omega \mid \psi_1(x)\}$$

Note that for any given real $x \in \omega^\omega$, we can manipulate the truth value of $\psi_0(x)$ and $\psi_1(x)$ via using the coding forcings $\text{Code}(x, a, b, 0)$ and $\text{Code}(x, a, b, 1)$ respectively from true to false (and once false it will remain false because of upwards absoluteness of Σ^1_3-formulas). This in particular will not interfere with the yet to be defined procedure of forming the α_β-allowable forcings, we will need in order to force $\mathbf{\Sigma^1_3}$-separation. Thus we gain some amount of flexibility in how we can make the sets B_0 and B_1 behave. We will use this to diagonalise against all possible $\mathbf{\Pi^1_3}$-sets B_m, b_k in such a way that none of those can reduce B_0 and B_1. This ensures the failure of Π^1_3-reduction.

7 Towards $\mathbf{\Sigma^1_3}$-separation

We are finally in the position to define the iteration which will yield a universe of $\mathbf{\Sigma^1_3}$-separation and a failure of Π^1_3-reduction.

The iteration we are about to define inductively will be an allowable iteration, whose tails are α-allowable and α-increases along the iteration. We start with fixing a bookkeeping function

$$F : \omega_1 \to H(\omega_1)^4$$

which visits every element cofinally often. The role of F is to list all the quadruples of the form $(\dot{x}, \dot{y}, m, k)$, where $\dot{x}, \dot{y}$ are names of reals in the forcing we already defined, and m and k are natural numbers which represent Σ^1_3-formulas or Π^1_3-formulas with two free variables, cofinally often. Assume that we are at stage $\beta < \omega_1$ of our iteration. By induction we will have constructed already the following list of objects.

- An ordinal $\alpha_\beta \leqslant \beta$ and a set $E_{\alpha_\beta} = E^0_{\alpha_\beta} \cup E^1_{\alpha_\beta}$ which is of the form $\{\dot{y}_\eta, m_\eta, k_\eta \; : \; \eta < \alpha_\beta\} \cup \{(\dot{x}_\eta, i_\eta) \; : \; \eta < \alpha_\beta\}$, where $\dot{y}_\eta, \dot{x}_\eta$ are $\mathbb{P}_\beta$-names of a reals, m_η, k_η are natural numbers and $i_\eta \in \{0,1\}$. As a consequence, for every bookkeeping function F', we do have a notion of η-allowable relative to E and F' over $W[G_\beta]$.

- We assume by induction that for every $\eta < \alpha_\beta$, if $\beta_\eta < \beta$ is the η-th stage in $\mathbb{P}_\beta$, where we add a new member to E_{α_β}, then $W[G_{\beta_\eta}]$ thinks that the $\mathbb{P}_{\beta_\eta \beta}$ is η-allowable with respect to $E_{\alpha_\beta} \upharpoonright \eta$.

- If $(\dot{y}_\eta, m_\eta, k_\eta) \in E_{\alpha_\beta}$, then we set again β_η to be the η-th stage in $\mathbb{P}_\beta$ such that a new member to E_{α_β} is added. In the model $W[G_{\beta_\eta}]$, we can form the set of reals R_η which were added so far by the use of a coding forcing in the iteration up to stage β_η, and which witness $(***)$ holds for some (x, y, m, k);

  Note that R_η is a countable set of reals and can therefore be identified with a real itself, which we will do. The real R_η is an error term and indicates the set of coding areas we must avoid when expecting correct codes, at least for the codes which contain $\dot{y}_\eta, m_\eta$ and k_η.

Assume that β is odd, $F(\beta) = (\dot{x}, \dot{y}, m, k)$, assume that $\dot{x}$, $\dot{y}$ are $\mathbb{P}_\beta$-names for reals, and $m, k \in \omega$ correspond to the Σ^1_3-formulas $\varphi_m(v_0, v_1)$ and $\varphi_k(v_0, v_1)$. Assume that G_β is a $\mathbb{P}_\beta$-generic filter over W. Let $\dot{x}^{G_\beta} = x$ and $\dot{y}_1^{G_\beta} = y_1, \dot{y}_2^{G_\alpha} = y_2$. We turn to the forcing $\mathbb{P}(\beta)$ we want to define at stage β in our iteration. Again we distinguish several cases.

(A) Assume that $W[G_\beta]$ thinks that there is an α_β-allowable forcing $\mathbb{Q}$ relative to E_{α_β} and some F' such that

$$\mathbb{Q} \Vdash \exists z (z \in A_m(y) \cap A_k(y)).$$

Then we pick the $<$-least such forcing, where $<$ is some previously fixed wellorder. We denote this forcing with $\mathbb{Q}_1$ and use

$$\mathbb{P}(\beta) := \mathbb{Q}_1.$$

We do not change R_β at such a stage.

(B) Assume that (A) is not true.

(i) Assume however that there is an α_β-allowable forcing $\mathbb{Q}$ in $W[G_\beta]$ with respect to E_{α_β} such that

$$\mathbb{Q} \Vdash x \in A_m(y).$$

Then we set

$$\dot{\mathbb{Q}}_\beta = \mathbb{P}(\beta) := \mathrm{Code}(x, y, m, k, 0).$$

In that situation, we enlarge the E-set as follows. We let $(\dot{y}, m, k) =: (\dot{y}_{\alpha_\beta}, m_{\alpha_\beta}, k_{\alpha_\beta})$ and

$$E_{\alpha_\beta + 1} := E_{\alpha_\beta} \cup \{(\dot{y}, m, k)\}.$$

Further, if we let r_η be the real which is added by $\mathrm{Code}((x, y, m, k), 0)$ at stage η of the iteration which witnesses $(***)$ of some quadruple $(x_\eta, y_\eta, m_\eta, k_\eta)$. Then we collect all the countably many such reals we have generically added so far in our iteration up to stage β and put them into one set R and let

$$R_{\alpha_\beta + 1} := R.$$

(ii) Assume that (i) is wrong, but there is an α_β-allowable forcing $\mathbb{Q}$ with respect to E_{α_β} in $W[G_\beta]$ such that

$$\mathbb{Q} \Vdash x \in A_k(y).$$

Then we set

$$\mathbb{P}(\beta) := \text{Code}(x, y, m, k, 1).$$

In that situation, we enlarge the E-set as follows. We let the new E value $(\dot{y}_{\alpha_\beta}, m_{\alpha_\beta}, k_{\alpha_\beta})$ be $(\dot{y}, m, k)$ and

$$E_{\alpha_\beta+1} := E_{\alpha_\beta} \cup \{(\dot{y}, m, k)\}.$$

Further, if we let r_η be the real which is added by Code(x, y, m, k, i), $i \in \{0, 1\}$ at stage η of the iteration which witnesses $(***)$ of some quadruple $(x_\eta, y_\eta, m_\eta, k_\eta)$. Then we collect all the countably many such reals we have added so far in our iteration up to stage β and put them into one set R and let

$$R_{\alpha_\beta+1} := R.$$

(iii) If neither (i) nor (ii) is true, then there is no α_β-allowable forcing $\mathbb{Q}$ with respect to E_{α_β} which forces $x \in A_m(y)$ or $x \in A_k(y)$, and we set

$$\mathbb{P}(\beta) := \text{Code}((x, y, m, k), 1).$$

Further, if we let r_η be the real which is added by Code$((x, y, m, k), 1)$ at stage η of the iteration which witnesses $(***)$ of some quadruple $(x_\eta, y_\eta, m_\eta, k_\eta)$. Then we collect all the countably many such reals we have added so far in our iteration up to stage β and put them into one set R and let

$$R_{\alpha_\beta+1} := R.$$

Otherwise we force with the trivial forcing.

8 Towards a failure of Π^1_3-reduction

Assume that $\beta < \omega_1$ is an even stage of our iteration. Our induction hypothesis includes that we have created already the iteration $\mathbb{P}_\beta$ up to stage β, and that we defined the notion of α_β-allowable forcings for an ordinal $\alpha_\beta < \omega_1$. Assume that $F(\beta) = (m, k, \dot{y})$, where $m, k \in \omega$ and $\dot{y}$ is the $\mathbb{P}_\beta$-name of a real number. We consider the following cases:

8.1 Case 1

We first assume that, working over $L[G_\beta]$, there is a further α_β-allowable forcing $\mathbb{P} \in L[G_\beta]$ such that $\mathbb{P}$ adds two reals x_0 and x_1 and such that for a $\mathbb{P}$-generic filter G over $L[G_\beta]$, $L[G_\beta][G]$ satisfies:

1. $x_0 \in B_m \backslash B_k$ and (x_0, y, a, b) is neither coded into $\vec{S}^0$ nor $\vec{S}^1$.

2. $x_1 \in B_k \backslash B_m$ and (x_1, y, a, b) is neither coded into $\vec{S}^0$ nor $\vec{S}^1$.

In that situation we first use the $<_L$-least $\mathbb{P}_\beta$-name of such a forcing $\mathbb{P}$ to add the reals x_0 and x_1. After forcing with $\mathbb{P}$ we code x_0 and x_1 both into $\vec{S}^1$ via forcing with $\mathrm{Code}(x_0, a, b, 1) \times \mathrm{Code}(x_1, a, b, 1)$. The forcing

$$\mathbb{P} * (\mathrm{Code}(x_0, a, b, 1) \times \mathrm{Code}(x_1, a, b, 1)) =: \mathbb{P}(\beta) = \dot{\mathbb{Q}}_\beta$$

is the forcing we use at stage β in the situation of case 1. We also change the notion of α_β-allowable to $\alpha_\beta + 1$-allowable which is defined to be α_β-allowable together with the additional demand to neither use the forcing $\mathrm{Code}(x_0, a, b, 0)$ nor $\mathrm{Code}(x_1, a, b, 0)$. In other words we let $E_{\alpha_\beta+1} := E_{\alpha_\beta} \cup \{(x_0, 0), (x_1, 0)\}$. Note that this choice ensures that x_0 and x_1 will both be elements of B_0 in all outer $\alpha_\beta + 1$-allowable extensions.

Note that the choice of $\mathbb{P}(\beta) = \dot{\mathbb{Q}}_\beta$ and $\alpha_\beta + 1$-allowability already ensures that $B_m(y), B_k(y)$ can not reduce B_0 and B_1 in all possible $\alpha_\beta + 1$-generic extensions of $L[G_{\beta+1}]$.

Lemma 8.1. *The sets $B_m(y)$ and $B_k(y)$ can not reduce B_0 and B_1 in all outer models of $L[G_{\beta+1}]$ which are obtained by a further $\alpha_\beta + 1$-allowable forcing.*

Proof. Indeed, we shall consider three subcases to see this. We work in $M \supset L[G_{\beta+1}]$ where M is an outer model obtained by a further $\alpha_\beta + 1$-allowable forcing.

Case 1a: First we assume that $x_0 \in B_m \backslash B_k$ and $x_1 \in B_k \backslash B_m$ still holds in M. In this situation neither $B_m \subset B_1$ nor $B_k \subset B_1$ can hold as witnessed by x_0 and x_1. In particular, $B_m(y)$ and $B_k(y)$ can not reduce B_0 and B_1. Note that this will remain true in all further outer models obtained by an additional $\alpha_\beta + 1$-allowable forcing as long as $x_0 \in B_m \backslash B_k$ and $x_1 \in B_k \backslash B_m$. If x_i will drop out of B_m or B_k in some $\alpha_\beta + 1$-allowable extension, then case 1b and case 1c will apply.

Case 1b: We assume that $x_0 \in B_m \backslash B_k$ but $x_1 \notin B_m \cup B_k$ holds in M. In this situation we can not have $B_0 \cup B_1 = B_m \cup B_k$ as $x_1 \in B_0$ and $x_1 \notin B_m \cup B_k$. Note that $x_1 \in B_0$ will remain true in outer $\alpha_\beta + 1$-allowable models, as we settled to never use $\mathrm{Code}(x_1, y, a, b, 0)$ and $x_1 \notin B_m \cup B_k$ by the upwards absoluteness of the Σ^1_3-formulas $\neg\psi_m$ and $\neg\psi_k$. As a consequence, $B_m(y)$ and $B_k(y)$ can not reduce B_0 and B_1 in all outer $\alpha_\beta + 1$-allowable models M.

Case 1c: In the dual case we assume that $x_1 \in B_m(y) \backslash B_k(y)$ but $x_0 \notin B_m(y) \cup B_k(y)$ holds in M to derive, as above, that $B_0 \cup B_1 \neq B_m(y) \cup B_k(y)$.

Case 1d: If x_0 and x_1 are both not in $B_m(y)$ and $B_k(y)$, then again $B_m(y) \cup B_k(y) \neq B_0 \cup B_1$.

$\square$

To summarize: In the described case 1, we defined an α_β-allowable forcing $\dot{\mathbb{Q}}_\beta$ and the notion of $\alpha_\beta + 1$-allowable forcings, such that α_β-allowable forcings are also $\alpha_\beta + 1$-allowable (but not vice versa). Additionally in all further $\alpha_\beta + 1$-allowable generic extensions of $L[G_{\beta+1}]$, the sets $B_m(y)$ and $B_k(y)$ can not reduce B_0 and B_1.

8.2 Case 2

In the second case, we assume that case 1 does not apply. As a consequence, whenever we work over $L[G_\beta]$ and apply a further α_β-allowable forcing $\mathbb{P}$ which adds two reals $x_0 \neq x_1$ and does neither have $\text{Code}(x_0, a, b, i)$ nor $\text{Code}(x_1, a, b, i)$ as a factor for $i \in \{0, 1\}$, then x_0, x_1 will not satisfy that $x_0 \in B_m(y) \backslash B_k(y)$ and $x_1 \in B_k(y) \backslash B_m(y)$.

Again, we shall split into subcases:

Case 2a: There is an α_β-allowable $\mathbb{P}$ and a $G \subset \mathbb{P}$ such that in $L[G_\beta][G]$ there are $x_0 \neq x_1$ such that $x_0, x_1 \in B_m \backslash B_k$. In this situation, we force over $L[G_\beta][G]$ with $\text{Code}(x_0, a, b, 1) \times \text{Code}(x_1, a, b, 0)$. Let $H_0 \times H_1 \subset \text{Code}(x_0, a, b, 1) \times \text{Code}(x_1, a, b, 0)$ be a $L[G_\beta][G]$-generic filter; and define $G_{\beta+1} := G_\beta * G * (H_0 \times H_1)$. We also define $\alpha_\beta + 1$-allowable as being α_β-allowable with the additional rule that $\text{Code}(x_0, a, b, 0)$ and $\text{Code}(x_1, a, b, 1)$ will not be used anymore as factors. (Note that as a consequence of this $x_0 \in B_0$ and $x_1 \in B_1$ will remain true in all $\alpha_\beta + 1$-allowable generic extensions of $L[G_{\beta+1}]$). That is, we define $E_{\alpha_\beta+1} := E_{\alpha_\beta} \cup \{(x_0, 0), (x_1, 1)\}$. Then we argue as follows:

Lemma 8.2. *Let M be an outer model of $L[G_{\beta+1}]$ obtained via an $\alpha_\beta + 1$-allowable forcing. Then $B_m(y)$ and $B_k(y)$ can not reduce B_0 and B_1 over M.*

Proof. We split into several cases. Assume first that in $M \supset L[G_{\beta+1}]$, $x_0, x_1 \in B_m(y)$ still holds true. Then, as $x_0 \in B_0 \backslash B_1$ and $x_1 \in B_1 \backslash B_0$, the set $B_m(y)$ can neither reduce B_0 nor B_1. Indeed x_0 witnesses that $B_m(y)$ is not a subset of B_1 and x_1 witnesses that $B_m(y)$ is not a subset of B_0. If, in a further $\alpha_\beta + 1$-allowable extension of $M \supset L[G_{\beta+1}]$, $x_0 \notin B_m(y)$ or $x_1 \notin B_m(y)$ then this will be dealt with in the next subcases.

If in $M \supset L[G_{\beta+1}]$, $x_0 \notin B_m(y)$, yet $x_1 \in B_m(y)$ then $x_0 \notin B_m(y) \cup B_k(y)$ and $x_0 \in B_0$ in all $\alpha_\beta + 1$-allowable generic extensions of M. So $B_m(y) \cup B_k(y) \neq B_0 \cup B_1$ and $B_m(y), B_k(y)$ can not reduce B_0, B_1. The same argument works if $x_1 \notin B_m(y)$ and $x_0 \in B_m(y)$.

If in M, $x_0, x_1 \notin B_m(y)$ anymore, then again $B_m(y), B_k(y)$ can not reduce B_0, B_1.

□

Case 2b: The dual situation in which we can force $x_0 \neq x_1$ such that $x_0, x_1 \in B_k(y) \backslash B_m(y)$ is dealt with in the analogous way.

Case 2c: If neither 2a nor 2b are true, then one can not force two distinct reals x_0, x_1 into $B_m(y) \backslash B_k(y)$ with α_β-allowable forcings which do not contain $\mathrm{Code}(x_0, a, b, i)$ or $\mathrm{Code}(x_1, a, b, i)$ for $i \in \{0, 1\}$. And the same holds true for $B_k(y) \backslash B_m(y)$. But it is straightforward to use an α_β-allowable forcing $\mathbb{P}$ over $L[G_\beta]$ which will add $\aleph_1$-many new reals $(z_i \mid i < \omega_1)$ and neither $\mathrm{Code}(z_i, a, b, 0)$ nor $\mathrm{Code}(z_i, a, b, 1)$ is a factor of $\mathbb{P}$. We force with such a $\mathbb{P}$, let $G \subset \mathbb{P}$ be generic over $L[G_\beta]$ and let $L[G_{\beta+1}] = L[G_\beta][G]$. We set $\alpha_\beta + 1$-allowable forcing to be α_β-allowable with the additional constraint that neither $\mathrm{Code}(z_i, a, b, 0)$ nor $\mathrm{Code}(z_i, a, b, 1)$ must be used.

Lemma 8.3. *If M is an outer model of $L[G_{\beta+1}]$, obtained with a further $\alpha_\beta + 1$-allowable forcing. Then $B_m(y)$ and $B_k(y)$ can not reduce B_0, B_1.*

Proof. For every $\alpha_\beta + 1$-allowable extension M of $L[G_{\beta+1}]$, $\forall i < \omega_1 (z_i \in B_0 \cup B_1)$ must hold in M, yet for any pair $z_i \neq z_j$, $M \models z_i, z_j \in B_m(y) \cap B_k(y)$ or $M \models z_i, z_j \notin (B_m(y) \cup B_k(y))$. In both cases, z_i, z_j witness that $B_m(y)$ and $B_k(y)$ can not reduce B_0, B_1 over M.

□

To summarize: in both cases we defined an extension $L[G_{\beta+1}]$ of $L[G_\beta]$, and the notion of $\alpha_\beta + 1$-allowable forcings. Additionally we found reals which witness that $B_m(y)$ and $B_k(y)$ can not reduce B_0 and B_1 in all further outer models M of $L[G_{\beta+1}]$ which are obtained with a further $\alpha_\beta + 1$-allowable forcing.

At limit stages β, we use the mixed support to define the limit partial order $\mathbb{P}_\beta$ and set $E_{\alpha_\beta} = \bigcup_{\eta < \beta} E_{\alpha_\eta}$. This ends the definition of $\mathbb{P}_{\omega_1}$.

9 Discussion of the resulting universe

We let G_{ω_1} be a $\mathbb{P}_{\omega_1}$-generic filter over W. As $W[G_{\omega_1}]$ is a proper extension of W, ω_1 is preserved. Moreover CH remains true.

A second observation is that for every stage β of our iteration and every $\eta > \beta$, the intermediate forcing $\mathbb{P}_{[\beta, \eta)}$, defined as the factor forcing of $\mathbb{P}_\beta$ and $\mathbb{P}_\eta$, is always an α_β-allowable forcing relative to E_{α_β} and some bookkeeping. This is clear as by the definition of the iteration, we force at every stage

β with a α_β-allowable forcing relative to E_{α_β} and α_β-allowable becomes a stronger notion as we increase α_β.

A third observation is, that B_0 and B_1 can not be reduced by any pair of boldface Π^1_3-sets $B_m(y), B_k(y)$. This follows immediately from the definition and the discussion of the iteration on even stages. Indeed if $B_m(y)$ and $B_k(y)$ are arbitrary $\boldsymbol{\Pi^1_3}$-sets, then, if $\dot{y}$ is some nice name of y, the triple $(m, k, \dot{y})$ will be guessed by the bookkeeping function cofinally often on the even stages. But on the first stage β where it is guessed, we ensured that B_0 and B_1 can not be reduced by $B_m(y)$ and $B_k(y)$. And this remains true for all further $\alpha_\beta + 1$-allowable extensions of $L[G_{\beta+1}]$. As $L[G_{\omega_1}]$ is an $\alpha_\beta + 1$-allowable extension of $L[G_{\beta+1}]$, the assertion follows.

We still need to show that in $L[G_{\omega_1}]$ the $\boldsymbol{\Sigma^1_3}$-separation property is true. For a pair of disjoint $\Sigma^1_3(y)$-sets, $A_m(y)$ and $A_k(y)$, we consider the least stage β such that there is a $\mathbb{P}_\beta$-name $\dot{z}$ such that $\dot{z}^{G_\beta} = z$ and (z, y, m, k) are considered by F at stage β. We consider the real R_{α_β} which codes up all the reals which witness an instance of $(***)$ for some quadruple (x', y', m', k'). This real R_{α_β} contains the information for all coding areas and quadruples (x', y', m', k') we have coded up so far into $\vec{S}$ in our iteration. As mentioned already, the role of R_{α_β} is that of an error term. We might have added false patterns in our iteration so far, but these false patterns will appear in a coded form in R_{α_β}. Our definitions will ensure that for the pair $A_m(y)$ and $A_k(y)$, modulo R_{α_β}, the set of reals x for which the quadruple (x, y, m, k) is coded into $\vec{S}^0$, and the set of reals x, for which the quadruple (x, y, m, k) is coded into $\vec{S}^1$ will separate $A_m(y)$ and $A_k(y)$:

$$\begin{aligned} x \in D^0_{y,m,k}(R_{\alpha_\beta}) \Leftrightarrow \exists r(L[r, R_{\alpha_\beta}] \models\ & (x, y, m, k) \text{ can be read off from a code} \\ & \text{written on an } \omega_1\text{-many } \omega\text{-blocks of elements of} \\ & \vec{S}^0 \text{ and the coding area of } (x, y, m, k) \\ & \text{is almost disjoint from each coding area in } R_{\alpha_\beta}). \end{aligned}$$

and

$$\begin{aligned} x \in D^1_{y,m,k}(R_{\alpha_\beta}) \Leftrightarrow \exists r(L[r, R_{\alpha_\beta}] \models\ & (x, y, m, k) \text{ can be read off from a code} \\ & \text{written on an } \omega_1\text{-many } \omega\text{-blocks of elements of} \\ & \vec{S}^1 \text{ and the coding area of } (x, y, m, k) \\ & \text{is almost disjoint from each coding area in } R_{\alpha_\beta}). \end{aligned}$$

It follows from the definition of the iteration that for any real parameter y and any $m, k \in \omega$, $D^0_{y,m,k}(R_{\alpha_\beta}) \cup D^1_{y,m,k}(R_{\alpha_\beta}) = \omega^\omega$. Indeed, as our bookkeeping function visits each name of a real in our iteration $\aleph_1$-many times, it will list each real $x \in W[G_{\omega_1}]$ unboundedly often below ω_1. Thus (x, y, m, k) will be coded into $\vec{S}^0$ or $\vec{S}^1$ at stages $\beta' > \beta$, which gives that

$D^0_{y,m,k}(R_{\alpha_\beta}) \cup D^1_{y,m,k}(R_{\alpha_\beta}) = \omega^\omega$ The next lemma establishes that the sets are indeed separating.

Lemma 9.1. *In $W[G_{\omega_1}]$, let y be a real and let $m, k \in \omega$ be such that $A_m(y) \cap A_k(y) = \emptyset$. Then there is an real R such that the sets $D^0_{y,m,k}(R)$ and $D^1_{y,m,k}(R)$ partition the reals.*

Proof. Let β be the least stage such that there is a real x such that $F(\beta) = (\dot{x}, \dot{y}, m, k)$ with $\dot{x}^G_\beta = x$, $\dot{y}^G_\beta = y$. Let R be R_{α_β} for R_{α_β} being defined as above. Then, as $A_m(y)$ and $A_k(y)$ are disjoint in $W[G_{\omega_1}]$, by the rules of the iteration, case B must apply at β.

Assume now for a contradiction, that $D^0_{y,m,k}(R)$ and $D^1_{y,m,k}(R)$ do have non-empty intersection in $W[G_{\omega_1}]$. Let $z \in D^0_{y,m,k}(R) \cap D^1_{y,m,k}(R)$ and let $\gamma > \beta$ be the first stage of the iteration which sees that z is in the intersection. Then, by the rules of the iteration and without loss of generality, we must have used case B(i) at the first stage $\delta \geqslant \beta$ of the iteration where (z, y, m, k) is listed by the bookkeeping F, and case B(ii) at stage γ. But this would imply, that at stage δ, there is an α_δ-allowable forcing $\mathbb{Q}_\delta$ with respect to E_{α_δ}, which forces $z \in A_m(y)$, yet at stage $\gamma > \delta$, there is an α_γ-allowable forcing which forces $z \in A_k(y)$. As $\gamma > \delta$, the α_γ-allowable forcing $\mathbb{Q}_\gamma$ which witnesses that we are in case B (ii) at stage γ in our iteration, is also α_δ-allowable. But this means that, over $W[G_\delta]$, the intermediate forcing $\mathbb{P}_{\delta,\gamma}$, which is also α_δ-allowable can be extended to the α_δ-allowable forcing which first uses $\mathbb{P}_{\delta,\gamma}$ and then $\mathbb{Q}_\gamma$ (denotes by $\mathbb{P}_{\delta,\gamma} \frown \mathbb{Q}_\gamma$), as it is, very strictly speaking not an iteration but a hybrid of an iteration and a product)) yielding an α_δ-allowable forcing which forces $z \in A_k(y)$.

On the other hand, in $W[G_\delta]$ we do have $\mathbb{Q}_\delta$ which forces $z \in A_m(y)$, as we assumed that at stage δ we are in case B (i). Now the product $\mathbb{Q}_\delta \times (\mathbb{P}_{\delta,\gamma} \frown \mathbb{Q}_\gamma)$ is α_δ-allowable, and by upwards absoluteness of Σ^1_3-formulas we get that

$$\mathbb{Q}_\delta \times (\mathbb{P}_{\delta,\gamma} \frown \mathbb{Q}_\gamma) \Vdash z \in A_m(y) \cap A_k(y).$$

But this would mean that at stage δ, we are in case A in the definition of our iteration, which is a contradiction.

□

Lemma 9.2. *In $W[G_{\omega_1}]$, for every pair m, k and every parameter $y \in \omega^\omega$ such that $A_m(y) \cap A_k(y) = \emptyset$ there is a real R such that*

$$A_m(y) \subset D^0_{y,m,k}(R) \wedge A_k(y) \subset D^1_{y,m,k}(R)$$

Proof. The proof is by contradiction. Assume that m, k and y are such that for every real R there is z such that $z \in A_m(y) \cap D^1_{y,m,k}(R)$ or $z \in A_k(y) \cap D^1_{y,m,k}(R)$. We consider the smallest ordinal $\beta < \omega_1$ such that $F(\beta)$ lists a quadruple of the form (x, y, m, k) for which $W[G_{\omega_1}] \models x \in A_m(y) \cap D^1_{y,m,k}$

and let $R = R_{\alpha_\beta}$. As $A_m(y)$ and $A_k(y)$ are disjoint we know that at stage β we were in case B. As x is coded into $\vec{S}^1$ after stage β and by the last Lemma, Case B(i) is impossible at β. Hence, without loss of generality we may assume that case B(ii) applies at β. As a consequence, there is a forcing $\mathbb{Q}_2 \in W[G_\beta]$ which is α_β-allowable with respect to E_{α_β} which forces $\mathbb{Q}_2 \Vdash x \in A_k(y)$. Note that in that case we collect all the reals which witness $(***)$ for some quadruple to form the set R_{α_β}.

As $x \in A_m(y) \cap D^1_{y,m,k}(R)$, we let $\beta' > \beta$ be the first stage such that $W[G_{\beta'}] \models x \in A_m(y)$. By Lemma 4.1, $W[G_\beta]$ thinks that $\mathbb{Q}_2 \times \mathbb{P}_{\beta\beta'}$ is α_β-allowable with respect to E_{α_β}, yet $\mathbb{Q}_2 \times \mathbb{P}_{\beta\beta'} \Vdash x \in A_m(y) \cap A_k(y)$. Thus, at stage β, we must have been in case A. This is a contradiction. $\square$

The next lemma will finish the proof of our theorem:

Lemma 9.3. *In $W[G_{\omega_1}]$, if $y \in \omega^\omega$ is an arbitrary parameter, R a real and m, k natural numbers, then the sets $D^0_{y,m,k}(R)$ and $D^1_{y,m,k}(R)$ are $\Sigma^1_3(R)$-definable.*

Proof. The proof is a standard calculation using the obvious modification of $(***)$ employing R as the set which codes the coding areas a real must avoid to be in $D^0_{y,m,k}(R)$ or $D^1_{y,m,k}(R)$. $\square$

10 Lifting to M_n

The ideas presented can be used to construct a universe in which $\boldsymbol{\Sigma^1_{n+3}}$-separation can be separated from Π^1_{n+3}-reduction. Its proof is a direct application of the ideas from [7] which can be used to translate the argument for the third level of the projective hierarchy to M_n. As there are no new ideas needed, and the translation works in a very similar manner to [7], we will not go into any details.

11 Open questions

We end with several questions which are related to this article.

Question 1. *Does there exist a universe in which $\boldsymbol{\Sigma^1_3}$-separation holds, but Π^1_n-reduction fails for any $n \geqslant 3$?*

Note for this question that in our construction of the failure of Π^1_3-reduction, we used Shoenfield absoluteness, or rather the upwards absoluteness of Σ^1_3-formulas, several times. Thus a failure of Π^1_n-reduction would need new ideas and arguments.

Question 2. *Can one force a universe of $\boldsymbol{\Sigma^1_4}$-separation in which Π^1_4-reduction fails over just L?*

By a classical result of Novikov, for any projective pointclass Γ, it is impossible to have Γ and $\check{\Gamma}$-reduction simultaneously. The case for separation is still unknown.

Question 3. *Can one force a universe where Σ^1_3- and Π^1_3-separation hold simultaneously?*

References

[1] J. Addison *Some consequences of the axiom of constructibility*, Fundamenta Mathematica, vol. 46 (1959), pp. 337–357.

[2] R. David *A very absolute Π^1_2-real singleton.* Annals of Mathematical Logic 23, pp. 101-120, 1982.

[3] S. D. Friedman and D. Schrittesser *Projective Measure without Projective Baire.* Memoirs of the American Mathematical Society vol. 267, 1298. 2020.

[4] S. Hoffelner *NS_{ω_1} Δ_1-definable and saturated.* Journal of Symbolic Logic 86 (1), pp. 25 - 59, 2021.

[5] S. Hoffelner *Forcing the Σ^1_3-separation property.* Journal of Mathematical Logic 22, No. 2, 2022.

[6] S. Hoffelner *Forcing the Π^1_3-reduction property and a failure of Π^1_3-uniformization property*, Annals of Pure and Applied Logic, Volume 174, Issue 8, 2023.

[7] S. Hoffelner *Forcing the Π^1_n-uniformization property*, submitted. Prerprint available Arxiv.

[8] R. Jensen and R. Solovay *Some Applications of Almost Disjoint Sets.* Studies in Logic and the Foundations of Mathematics Volume 59, pp. 84-104, 1970.

[9] V. Kanovei and V. Lyubetsky *On Harrington's model in which Separation holds but Reduction fails at the 3rd projective level, and on some related models of Sami*, Arxiv.

[10] S. Müller, R. D. Schindler and W. H. Woodin *Mice with finitely many Woodin cardinals from optimal determinacy hypotheses*, Journal of Mathematical Logic, vol. 20, 2020.

[11] D. Martin and J. Steel *A Proof of Projective Determinacy.* Journal of the American Mathematical Society (2), pp.71-125, 1989.

[12] Y. Moschovakis *Descriptive Set Theory.* Mathematical Surveys and Monographs 155, AMS.

[13] Y. Moschovakis *Uniformization in a playful Universe.* Bulletin of the American Mathematical Society 77, no. 5, 731-736, 1971.

[14] R. Sami *Questions in descriptive set theory and the determinacy of infinite games*, Ph.D. Dissertation, Univ. of California, Berkeley, 1976.