

FORCING AS A COMPUTATIONAL PROCESS

JOEL DAVID HAMKINS, RUSSELL MILLER, AND KAMERYN J. WILLIAMS

ABSTRACT. We investigate how set-theoretic forcing can be seen as a computational process on the models of set theory. Given an oracle for information about a model of set theory $\langle M, \in^M \rangle$, we explain senses in which one may compute M -generic filters $G \subseteq \mathbb{P} \in M$ and the corresponding forcing extensions $M[G]$. Specifically, from the atomic diagram one may compute G , from the Δ_0 -diagram one may compute $M[G]$ and its Δ_0 -diagram, and from the elementary diagram one may compute the elementary diagram of $M[G]$. We also examine the information necessary to make the process functorial, and conclude that in the general case, no such computational process will be functorial. For any such process, it will always be possible to have different isomorphic presentations of a model of set theory M that lead to different non-isomorphic forcing extensions $M[G]$. Indeed, there is no Borel function providing generic filters that is functorial in this sense.

1. INTRODUCTION

The method of forcing, introduced by Paul Cohen to show the consistency of the failure of the continuum hypothesis, has become ubiquitous within set theory.¹ In this paper we analyze this method from the perspective of computable structure theory. To what extent is forcing an effective process?

Main Question. *Given an oracle for a countable model of set theory M , to what extent can we compute its various forcing extensions $M[G]$?*

We answer this, considering multiple levels of information which might be given by the oracle.

Main Theorem 1 (Forcing as a computational process).

- (1) *For each countable model M of set theory and for any forcing notion $\mathbb{P} \in M$ there is an M -generic filter $G \subseteq \mathbb{P}$ computable from the atomic diagram of M .*
- (2) *Given the Δ_0 -diagram² of M and a forcing notion $\mathbb{P} \in M$, we can uniformly compute such a G , the atomic diagram of $M[G]$, and moreover the Δ_0 -diagram of $M[G]$.*

2010 *Mathematics Subject Classification.* 03C57, 03E40.

Key words and phrases. Forcing, computable structure theory.

We thank the anonymous referee for their helpful comments.

The second author was supported by NSF grant # DMS-1362206, Simons Foundation grant # 581896, and several PSC-CUNY research awards.

Commentary can be made about this article on the first author's blog at <http://jdh.hamkins.org/forcing-as-a-computational-process>.

¹We will give an overview of the construction during Section 4. For full details we refer the reader to [Sho71] or [Kun80]. See also [Cho09] for a conceptual overview.

² Δ_0 in the sense of the Lévy hierarchy, as described below.

- (3) From \mathbb{P} and the elementary diagram of M , we can uniformly compute the elementary diagram of $M[G]$, and this holds level-by-level for the Σ_n -diagrams.

These statements are proved in Section 4 as Theorems 8, 10, and 11 below. In Section 5 we will extend these results to look at the generic multiverse of a countable model of set theory, i.e. those models obtained from the original model by taking both forcing extensions and grounds, where taking grounds is the process inverse to building forcing extensions. In Section 6 we also consider versions of these results for class forcing instead of set forcing.

We remark here that in this context the Δ_0 -*diagram* of M refers to the set of formulae true in M that are Δ_0 in the Lévy hierarchy. In this hierarchy, the standard one used within set theory, Δ_0 -formulae are those whose only quantifiers are bounded by sets, that is, of the form $\exists x \in y$ or $\forall x \in y$. This differs from Δ_0 in the arithmetical hierarchy, whose formulae's quantifiers are only those bounded by the order relation on ω , that is of the form $\exists x < y$ or $\forall x < y$. In Section 2 we will show that very little can be computed from the atomic diagram for a model of set theory. From the diagram we are not even able to compute relations as simple as $x \subseteq y$. We view this as evidence that for the computable structure theory of set theory the Lévy Δ_0 -diagram is an appropriate choice of the basic information to be used. The usual signature of set theory—just equality and the membership relation—is too spartan to say anything of use. In Section 3 we introduce an expansion of the signature for set theory which captures the strength of the Lévy Δ_0 -diagram. We show that the Σ_n -formulae in the Lévy hierarchy are precisely those that are Σ_n in the arithmetical hierarchy with the expanded signature.

We end the paper, in Sections 7 and 8, by investigating how much information is required to make the process of computing $M[G]$ from M functorial. Without significant detail about the dense subsets of \mathbb{P} , it will not be so. Recall that a *presentation* of a countable structure M is simply a structure isomorphic to M whose domain is ω . The following theorem emphasizes the importance of not conflating the isomorphism type of M with a specific presentation of M .

Main Theorem 2 (Nonfunctoriality of forcing). *There is no computable procedure and indeed no Borel procedure which performs the tasks of Main Theorem 1 in a uniform way so that distinct presentations of the model M will result in isomorphic presentations of the extension $M[G]$.*

We also show that forcing can be made a functorial process. One way to do this is to add extra information to the signature of the model. Another way is to restrict to a special class of models, namely the pointwise definable models.

2. THE ATOMIC DIAGRAM OF A MODEL OF SET THEORY KNOWS VERY LITTLE

In this section, we show that very little about a model of set theory can be computed from its atomic diagram. In particular, many basic set-theoretic relations are not decidable from the atomic diagram. As a warmup, let us first see that the atomic diagram does not suffice to identify even a single fixed element.

Proposition 3. *For any countable model of set theory $\langle M, \in^M \rangle$ and any element $b \in M$, no algorithm will pick out the number representing b uniformly given an oracle for the atomic diagram of a copy of M .*

For example, given the atomic diagram of a copy of M , one cannot reliably find the empty set, nor the ordinal ω , nor the set \mathbb{R} of reals.

Proof. Fix $b \in M$ and fix an oracle for the atomic diagram of a copy of M . Suppose we are faced with an algorithm purported to identify b . Run the algorithm until it has produced a number that it claims is representing b . This algorithm inspected only finitely much of the atomic diagram of M . The number b it produced has at most finitely many elements in that portion of the atomic diagram. But M has many sets that extend that pattern of membership, and so we may find an alternative copy M' of M whose atomic diagram agrees with the original one on the part that was used by the computation, but disagrees afterwards in such a way that the number for b now represents a different set. So the algorithm will get the wrong answer on M' . \square

This idea can be extended to characterize which relations on M are computably from the atomic diagram. In particular, any such relation must contain both finite and infinite sets.

Theorem 4. *Let $\langle M, \in^M \rangle \models \text{ZF}$ be a countable model of set theory and let $X \subseteq M^n$ for some $1 \leq n < \omega$. The following are equivalent. (In the following, Σ_1 in all cases refers to the arithmetical hierarchy, not the Lévy hierarchy.³)*

- (1) *X is uniformly relatively intrinsically computably enumerable in the atomic diagram of M . That is, there is a single computably enumerable operator which given the atomic diagram of a presentation of M will output the copy of X for that presentation.*
- (2) *Membership of each single \bar{a} in X is witnessed by a finite pattern of \in in the transitive closures of $\{a_0\}, \dots, \{a_{n-1}\}$, and the list of finite patterns witnessing membership is enumeration-reducible to the Σ_1 -diagram of M .*
- (3) *There is a $\mathcal{L}_{\omega_1, \omega}$ -formula $\varphi(x)$ so that $\varphi(x)$ is Σ_1 and defines X , and the set of (finite) disjuncts in this formula is enumeration-reducible to the Σ_1 -diagram of M .*

Proof. (1 \Leftrightarrow 3) is a standard fact in computable structure theory, first established in [AKMS89], relativized here since M does not have a computable presentation.

(2 \Rightarrow 1) For notational simplicity we will present the argument for the $n = 1$ case. Suppose membership of a in $X \subseteq M$ is witnessed by a finite pattern of \in in the transitive closure of $\{a\}$. That is, a is in X if and only if one of a certain list of finite graphs can be found in the pointed graph $(\text{TC}(\{a\}), a, \in^M)$. By hypothesis this list is e -reducible to the Σ_1 -diagram of M , which we can enumerate, since we have the atomic diagram of M as an oracle. Therefore, we can list out these finite graphs, one by one. Meanwhile, we enumerate \in^M and M , and continually check whether the most recent pair in \in^M has completed a copy of a graph from this list. If such happens, then we search through the $a \in M$ we have already enumerated and check whether a has one of these graphs appearing below its transitive closure. We output all the a for which this happens. We also check whether the most recent $a \in M$ has a copy of one of the graphs in the portion of its transitive closure so far enumerated. If that happens, we output a . This process will find every $a \in M$ which has a copy of this graph below its transitive closure, so it will exactly enumerate X .

³See Section 3 for further discussion of the distinction.

(1 \Rightarrow 2) Again, we will present the proof for the $n = 1$ case for notational simplicity. Assume that X is uniformly relatively intrinsically computable enumerable from the atomic diagram of M . That is, the decision of whether $a \in X$ must be made by a finite portion of the graph \in^M where we are only allowed to know the index of a itself. We can separate this information into two pieces, that inside the transitive closure of $\{a\}$ and that outside. We claim that only the first information can be relevant, which then implies (2). To see this, observe that every possible finite pattern of finite graphs (with a as a constant) occurs outside of the transitive closure of $\{a\}$ in every model of set theory. So the information outside is true for every $a \in M$ and thus cannot have any effect on the decision of whether to output a in the enumeration. \square

As a consequence of this theorem, the atomic diagram of a model of set theory does not know any of the predicates and relations named in Lemma 7 below, which are each easily checked to fail item (2) of Theorem 4. In short, the atomic diagram of a model of set theory knows little about the model. Before addressing forcing, therefore, we now present a natural expansion of the signature, under which the atomic diagram will present the information that set theorists normally consider to be basic.

3. THE LÉVY DIAGRAM

The signature ordinarily used in set theory is simple: it consists of the binary relation \in along with equality. This signature suffices all by itself to express all of mathematics (possibly using assumptions beyond ZFC). But from the perspective of computable structure theory we cannot say much using just the atomic diagram in this signature. As we see it, the lesson here is that this atomic diagram is too weak to take as the basic information for the computable structure theory of models of set theory. Instead, we will work with the Δ_0 -diagram, in the sense of the *Lévy hierarchy*, in which set-bounded quantification is not counted when determining quantifier complexity of a formula.

This conflicts with the convention in computability theory, where “bounded quantification” refers to bounded in the sense of arithmetic—i.e. of the form $\exists x < y$ or $\forall x < y$ —rather than bounded in the sense of a relation of the structure. In short, Δ_0 in the Lévy hierarchy is not the same as Δ_0 in the arithmetical hierarchy, as $\forall x \in a \varphi(x)$ may express infinitely many independent atomic statements.

In this section we describe how we can make the Lévy hierarchy line up with the arithmetical hierarchy, by expanding the signature to capture the content of the (Lévy) Δ_0 -diagram. Let us call this expanded signature the *Lévy signature*. This provides an alternate way to think of the Δ_0 -diagram. One may instead work with the atomic diagram in this expanded signature, with the correspondence between the two being effective. And this correspondence continues upward, with Σ_n in the Lévy hierarchy corresponding to Σ_n in the arithmetical hierarchy with this expanded signature. In the sequel, we will speak of the Δ_0 -diagram or Σ_n -diagram, referring to the Lévy hierarchy, but the reader who prefers the other approach may freely translate over the statements.⁴

⁴As further evidence for the naturalness of this expansion of the language, we note that it has arisen in other contexts. Venturi and Viale [VV19] studied model companions for set theory. For their work they also found it helpful to expand the signature from the spartan $\{\in\}$ to include symbols for each Δ_0 relation.

Definition 5. *The Lévy signature contains the binary relation symbol \in and also, for each Δ_0 -formula $\varphi(x_1, \dots, x_n)$, an n -ary predicate R_φ . Structures in the Lévy signature are assumed to satisfy the axiom schema (over all Δ_0 -formulae):*

$$\forall \vec{x} [R_\varphi(\vec{x}) \iff \varphi(\vec{x})].$$

For example, since inclusion is defined by a Δ_0 -formula using only \in , we see that

$$M \models b \subseteq c \iff M \models \forall y \in b \ y \in c \iff M \models R_{(\forall y \in x_1) y \in x_2}(b, c).$$

Since we can enumerate the Δ_0 -formulae effectively, the language of the Lévy signature is a computable language, and it includes equality (defined by $x \subseteq y \wedge y \subseteq x$).

Our choice of signature allows us to imitate the usual set-theoretic convention under which Δ_0 is the lowest possible complexity of a formula. Additionally, sets defined by Σ_n -formulae in the Lévy hierarchy will all be arithmetically Σ_n in models of ZF in the Lévy signature, by the following standard lemma.

Lemma 6. *For every formula $\varphi(t_1, \dots, t_n, x, y, z)$ in the Lévy hierarchy,*

$$\text{ZF} \vdash \forall \vec{t} [\forall x \in z \ \exists y \ \varphi(\vec{t}, x, y, z) \iff \exists Y \ \forall x \in z \ \exists y \in Y \ \varphi(\vec{t}, x, y, z)].$$

Proof. If $M \models \forall x \in z \ \exists y \ \varphi(\vec{a}, x, y, z)$ for a model $\langle M, \in^M \rangle$ of ZF with parameters \vec{a} , then by the Replacement axiom schema, there is some set Y in M such that $M \models \forall x \in z \ \exists y \in Y \ \varphi(\vec{a}, x, y, z)$. The reverse direction is trivial. \square

The lemma, applied repeatedly, allows us to turn every formula from the signature with just membership and equality into an equivalent formula which is in a prenex form, with all unbounded quantifiers preceding a Δ_0 matrix, and which has the same complexity (in the Lévy hierarchy) as the original formula. Therefore, every formula of Lévy complexity Σ_n may be expressed in the Lévy signature by a formula of arithmetic complexity Σ_n , and there is an effective procedure for producing the latter from the former.

To conclude this section, let us remark that many basic predicates of set theory are Δ_0 , and that the basic predicates for forcing are Δ_1 and hence computable from the Δ_0 -diagram.

Lemma 7. *For models M of ZFC, the following predicates are all Δ_0 .*

$x = \emptyset$.	x is a set of ordered pairs.
$x \subseteq y$.	x is a function.
$x = \{y, z\}$.	x is transitive.
$x = \bigcup y$.	x is an ordinal.
$x = \{z \in y : \varphi(z)\}$, for Δ_0 -formulae $\varphi(z)$.	x is inductive.
x is a (Kuratowski) ordered pair.	$x = \omega$.

We omit any proof of these standard facts. More pertinently for this article, let us briefly remark that the relations “ x is a \mathbb{P} -name”, $p \Vdash \sigma \in \tau$, and $p \Vdash \sigma = \tau$ are all Δ_1 and hence computable from the Δ_0 -diagram. This will be important in Section 4 to see that the Δ_0 -diagram suffices to compute a forcing extension.

4. COMPUTING FORCING EXTENSIONS

For the sake of the reader who may not be an expert in set theory, we interleave our proof of the first main theorem with an exposition of forcing.⁵ This exposition will follow the three parts of the main theorem. First we discuss generic filters and we show that a generic filter may be effectively built from just the atomic diagram. We next discuss how to build the forcing extension $M[G]$ from the ground model M and the generic G . Then we show that there is an effective procedure to construct (the atomic diagram of) the forcing extension from the Δ_0 -diagram of the ground model. Finally, we discuss how truth in the forcing extension is determined by the forcing relations in the ground model. Then we can show that elementary diagram of the forcing extension is effectively computable from the elementary diagram of the ground model.

In the ground universe M , a partially ordered set \mathbb{P} gives partial information about an object from outside the universe. A condition p is stronger than another condition q , written $p \leq q$, if it gives more information about this outside object. It is convenient to assume that \mathbb{P} has a maximum element 1 which gives no information. Two conditions are compatible, written $p \parallel q$, if there is a condition stronger than both of them. Otherwise, they are incompatible, written $p \perp q$. Formally, the outside object is an M -generic filter $G \subseteq \mathbb{P}$. That is, it is downward directed and upward closed, and it meets every dense $D \subseteq \mathbb{P}$ in M . Here, $D \subseteq \mathbb{P}$ is dense if every condition in \mathbb{P} can be strengthened to a condition in D . It is a straightforward exercise that if \mathbb{P} is splitting—any condition p extends to two incompatible conditions—then M itself contains no M -generic filter.

However, if M is countable, we can always externally construct an M -generic filter $G \subseteq \mathbb{P}$, and the construction works uniformly for all \mathbb{P} (and all $\Delta_0(M)$). We simply fix the first-encountered $p_0 \in \mathbb{P}$ as our starting point (using the external order $<$ on the domain ω of M to find it) and then ask of each element $s \in \omega$ in turn whether the current p_s can be extended to an element of $\mathbb{P} \cap s$. This is a Δ_0 question, and we either find the least $p_{s+1} \leq p_s$ lying in s if the $\Delta_0(M)$ indicates that one exists, or else set $p_{s+1} = p_s$. Finally, set $G = \{q \in \mathbb{P} : q \parallel p_n \text{ for some } n\}$. Of course, if s was a dense subset of \mathbb{P} , then we ensured that G does meet s , so this G is M -generic.

In fact, there exists a generic filter that is computable merely from the atomic diagram, as we now show.

Theorem 8. *Given an oracle for the atomic diagram of a model of set theory $\langle M, \in^M \rangle \models \text{ZF}$ and a notion of forcing $\mathbb{P} \in M$, there is a computable procedure to compute an M -generic filter $G \subseteq \mathbb{P}$.*

Proof. Fix an oracle for $\langle M, \in^M \rangle$. That is, we are presented on the oracle tape with an isomorphic copy of M as a binary relation on the natural numbers. Indeed, we might as well assume that $M = \mathbb{N}$ and \in^M is that relation on \mathbb{N} . Let $\mathbb{P} \in M$ be any notion of forcing in M . More specifically, we have in M an element $\mathbb{P} \in M$ for the underlying set of the partial order, and we also have a set $\leq_{\mathbb{P}}$ in M for what M thinks is the set of ordered pairs for that relation. We assume that the Kuratowski pairing function $(p, q) = \{\{p\}, \{p, q\}\}$ is used when coding ordered pairs.

Notice that from the atomic diagram we can decide whether a given number p represents an element of \mathbb{P} or not, since we need only ask the oracle whether $p \in^M \mathbb{P}$.

⁵See Footnote 1 for citations to references with more detail on the forcing construction.

Further, we can computably enumerate the pairs (p, q) of forcing conditions with $p \leq_{\mathbb{P}} q$. For this we must unwrap the Kuratowski pairing function, but this is possible as follows. We search for conditions p, q in \mathbb{P} and for elements $x \in \leq_{\mathbb{P}}$ that represent the pair (p, q) . This will happen when $x = \{y, z\}$, where $y = \{p\}$ and $z = \{p, q\}$. So we can search for the elements y and z which have $y \in^M x$ and $z \in^M x$ and $p \in^M y$ and $p, q \in^M z$. When this happens, we can be confident that $p \leq_{\mathbb{P}} q$. (This kind of computational inspection of the Kuratowski ordered pair also arose in [GH17].)

But actually, we can fully decide the relation $\leq_{\mathbb{P}}$, not just enumerate it. The reason is that M , being a model of set theory, has what it thinks is the set of pairs $(p, q) \in \mathbb{P}^2$ with $p \not\leq_{\mathbb{P}} q$. And so for any pair p, q , we can search for it to be enumerated by $\leq_{\mathbb{P}}$ or by the analogous procedure applied with $\not\leq_{\mathbb{P}}$, and thereby decide whether $p \leq_{\mathbb{P}} q$ or not.

Next, let $\mathcal{D} \in M$ be the set that M thinks is the set of all dense subsets of \mathbb{P} . From this data, we can enumerate the elements D_0, D_1, D_2, \dots of \mathcal{D} , listing all the elements of M that M thinks are dense subsets of \mathbb{P} . We simply run through all the natural numbers d , and ask the oracle whether $d \in^M \mathcal{D}$, and if so, we put it on the list.

Let us use this enumeration to compute a descending sequence of forcing conditions

$$p_0 \geq_{\mathbb{P}} p_1 \geq_{\mathbb{P}} p_2 \geq_{\mathbb{P}} \dots$$

with $p_n \in D_n$. To begin, we let p_0 be the first-encountered element of D_0 , which exists because D_0 is dense. Next, given p_n , we search through the natural numbers for the first-encountered condition $p_{n+1} \leq_{\mathbb{P}} p_n$ such that $p_{n+1} \in D_{n+1}$. There is always such a condition because D_{n+1} is dense.

Finally, having constructed the descending sequence, we define G as the set of conditions q for which $p_n \leq q$ for some n . This is an M -generic filter, because it is a filter and it meets every dense subset of \mathbb{P} in M . The elements of G can be enumerated by the processes above, because whenever we find a condition q for which $p_n \leq_{\mathbb{P}} q$ for some n , then we can enumerate q into the filter.

But actually, we claim that G is fully decidable from the oracle, not just enumerable. To see this, let $\perp_{\mathbb{P}}$ be the set in M that M thinks is the set of pairs (p, q) of incompatible conditions $p \perp q$. By genericity, it follows that if $q \notin G$, it must be that $q \perp_{\mathbb{P}} p_n$ for some n , since it is dense to either get below q or become incompatible with it. The algorithms above allow us to enumerate the pairs of incompatible conditions, and so for any condition q , we search for a p_n for which either $p_n \leq_{\mathbb{P}} q$ or $p_n \perp_{\mathbb{P}} q$, and in this way we can tell whether $q \in G$ or $q \notin G$, as desired. \square

The algorithm above is non-uniform in several senses. First, the algorithm makes use not only of the indices of the forcing notion (namely, the set \mathbb{P} , the set $\leq_{\mathbb{P}}$ and the complement of $\leq_{\mathbb{P}}$ in $\mathbb{P} \times \mathbb{P}$), but also the index of the set of dense subsets of \mathbb{P} , and the index $\perp_{\mathbb{P}}$ for the set of incomparable elements. The methods of Theorem 4 show that it is not possible in general to compute these just from the atomic diagram of $\langle M, \in^M \rangle$ and \mathbb{P} . Second, a more serious kind of non-uniformity arises from the fact that, even if we are given this additional data and even if we are given the full elementary diagram of the model M , the particular generic filter that we end up with will depend on the order in which M is represented in this presentation. The filter G is determined in part by the order in which the dense sets D_n appear in the presentation of M . If one rearranges the dense sets, then

at a certain stage, one might be led to place a different, incompatible condition on the sequence, and this will give rise to a different filter. In Sections 7 and 8, we shall examine uniformity in more detail and prove there is no uniform computable procedure nor even Borel function that always produces the same generic filter G from all isomorphic presentations of the same model M .

We also want to remark that it mattered which pairing function we used. Given sets x, y, p where we know p is a Kuratowski ordered pair it requires looking at only finitely many objects to check whether $p = (x, y)$. On the other hand, there are Δ_0 definitions for ordered pairs which we cannot computably unravel from just the atomic diagram. Consider the Morse pairing function

$$(x, y)^* = (\{0\} \times x) \cup (\{1\} \times y),$$

where the product here is defined via the Kuratowski ordered pair.⁶ This definition is Δ_0 in the Lévy hierarchy. But to recognize whether $p = (x, y)^*$ requires looking at infinitely many elements of the model when at least one of x and y is infinite, making it not effective (from just the atomic diagram). It follows from Theorem 4 that the relation $p = (x, y)^*$ is not decidable from the atomic diagram, even if we know p is a Morse pair.

We next turn to the construction of the full forcing extension. Given a generic filter G , which we have already seen how to construct, we need to determine the rest of the sets that will be the forcing extension of M . New sets are given by names in M , which are then interpreted by the generic. These \mathbb{P} -names are recursively defined as sets whose elements are pairs (σ, p) with σ a \mathbb{P} -name and $p \in \mathbb{P}$. This amounts to a recursive definition on rank, with each \mathbb{P} -name having an ordinal rank.

Lemma 9. *The property of being a \mathbb{P} -name is Δ_1 , hence decidable uniformly in the Δ_0 -diagram of M .*

Naively, one might attempt to prove that the class of \mathbb{P} -names is Σ_1 by, given an element x , querying the Δ_0 -diagram to check that all elements of x are pairs whose first coordinate is a \mathbb{P} -name, which is checked by querying the Δ_0 -diagram, and recursively continue this process until it halts. (And similarly to prove the class of \mathbb{P} -names is Π_1 .) The trouble with this approach is that this recursive procedure is transfinite; even in the case it is internally finite, M might be ω -nonstandard and thus have internally finite sets which are externally seen to be infinite. Nevertheless, the key of this idea is correct, and it can be made into a proper proof. The point is that instead of externally carrying out the recursive procedure, we instead look inside M for a certificate that the recursive procedure was carried out.

Proof of Lemma 9. To see that being a \mathbb{P} -name is Σ_1 , given a set x we check whether x is a \mathbb{P} -name by searching for a tree witnessing that x satisfies the recursive definition of a \mathbb{P} -name. Such a tree has x as its root, and the immediate children of the root are the first coordinates of elements of x . These nodes then have their own children by the same process, and this continues downward through the whole tree. This tree is necessarily well-founded (in the sense of M), because M thinks its membership relation is well-founded. And this tree witnesses that x is a \mathbb{P} -name

⁶This pairing function has the nice property that if $x, y \subseteq V_\alpha$ where α is a limit ordinal, then $(x, y)^* \subseteq V_\alpha$.

if the second coordinate of each node is in \mathbb{P} and no node has an element not represented among its children. It is clear that given a tree T it is Δ_0 to check whether it satisfies this property of witnessing that x is a \mathbb{P} -name, since this requires only quantifying over nodes in the tree and over \mathbb{P} . And thus the class of \mathbb{P} -names is Σ_1 .

To see that it is also Π_1 we carry out a similar procedure, except now we look for a tree witnessing that the recursive definition of being a \mathbb{P} -name fails for x . Being such a tree is again Δ_0 , because failure is witnessed by looking at and below some node. This then gives that being a \mathbb{P} -name is a Δ_1 property. \square

We would like to remark that this argument generalizes; any property defined by a set-theoretic recursion of Δ_0 properties will be Δ_1 .

A standard approach to forcing is to define the interpretation of a name σ by G , denoted σ_G , to be the set of τ_G for $(\tau, p) \in \sigma$ for some $p \in G$. But this recursive interpretation can only be carried out if M is well-founded, limiting which models it can be applied to. Fortunately, there is an alternative construction we can use, which applies to all models and which matches the recursive interpretation construction in case M is well-founded.

Every set theorist knows that there are a pair of definable relations,

$$\begin{aligned} p \Vdash_{\mathbb{P}} \sigma = \tau \\ p \Vdash_{\mathbb{P}} \sigma \in \tau, \end{aligned}$$

which determine membership and equality in the forcing extension. Namely, $\sigma_G \in \tau_G$ if and only if there is $p \in G$ so that $p \Vdash_{\mathbb{P}} \sigma \in \tau$, while $\sigma_G = \tau_G$ if and only if there is $p \in G$ so that $p \Vdash_{\mathbb{P}} \sigma = \tau$. We can use this property as the definition of the forcing extension. That is, define the following relations on the class of \mathbb{P} -names:

$$\begin{aligned} \sigma =_G \tau &\iff \exists p \in G \ p \Vdash_{\mathbb{P}} \sigma = \tau \\ \sigma \in_G \tau &\iff \exists p \in G \ p \Vdash_{\mathbb{P}} \sigma \in \tau. \end{aligned}$$

It is readily checked that the relation $=_G$ is an equivalence relation and indeed a congruence relation with respect to the relation \in_G . The forcing extension $M[G]$ can then be presented as the equivalence classes of names $[\sigma]_G$ by the relation $=_G$ with the membership relation induced by \in_G . This is essentially the Boolean ultrapower manner of constructing the forcing extension, rather than the value recursion method (see [HS06] for an account of how these constructions can differ).

Let us see that the above described process is indeed effective in the Δ_0 -diagram.

Theorem 10. *There is a uniform computable procedure that, given an oracle for the Δ_0 -elementary diagram of a model of set theory $\langle M, \in^M \rangle \models \text{ZF}$ and a notion of forcing $\mathbb{P} \in M$, computes an M -generic filter $G \subseteq \mathbb{P}$ and then computes the atomic diagram of a presentation of the extension $M[G]$. Indeed, with this oracle it can decide the Δ_0 -elementary diagram of this presentation.*

Proof. The diagram of the forcing extension $M[G]$ will be in the full forcing language

$$\langle M[G], \in^{M[G]}, \check{M}, \sigma \rangle_{\sigma \in M^{\mathbb{P}}},$$

with a predicate \check{M} for the ground model and constants for all the \mathbb{P} -names σ ; we denote the class of \mathbb{P} -names in M by $M^{\mathbb{P}}$. The class $M^{\mathbb{P}}$ of \mathbb{P} -names was seen in

Lemma 9 to be computable from $\Delta_0(M)$. The atomic forcing relations

$$\begin{aligned} p \Vdash_{\mathbb{P}} \sigma = \tau \\ p \Vdash_{\mathbb{P}} \sigma \in \tau, \end{aligned}$$

have complexity Δ_1 in the Lévy hierarchy, with \mathbb{P} as a parameter, because again these relations are the result of the solution of a recursion of a Δ_0 property (see [GHHSW] for explicit discussion of the forcing-relation-as-solution-to-a-recursion perspective). Similarly the negated atomic forcing relations are also Δ_1 with \mathbb{P} as a parameter. So given the Δ_0 -diagram and the generic G we can compute $=_G$, and thereby pick out representatives from the equivalence classes and compute a bijection onto $\omega = \text{dom } M[G]$ from the collection of these classes: 0 will denote the class of the $<$ -least \mathbb{P} -name, 1 the class of the least \mathbb{P} -name not in the class of 0, and so on. Similarly, we can also compute \in_G , giving us the atomic diagram of $M[G]$ (in the signature with just \in).

We delay the argument that we can decide the Δ_0 -diagram of $M[G]$ until after a discussion of truth in the forcing extension; see the proof of Theorem 11. \square

At last we come to truth in the forcing extension. As a first-order structure, truth in $M[G]$ is given by the usual Tarskian recursive construction. A key fact in forcing, however, is that truth in the forcing extension is closely tied to the ground model. We have already seen the start of this with the atomic forcing relations $p \Vdash \sigma \in \tau$ and $p \Vdash \sigma = \tau$. In general, for any formula $\varphi(x_0, \dots, x_n)$ there is a definable relation $p \Vdash \varphi(\sigma_0, \dots, \sigma_n)$ so that $M[G] \models \varphi((\sigma_0)_G, \dots, (\sigma_n)_G)$ if and only if there is $p \in G$ so that $p \Vdash \varphi(\sigma_0, \dots, \sigma_n)$. The map which sends φ to the formula $p \Vdash \varphi$ is effective. And as we explain in the proof, this map does not increase complexity; if φ is Δ_0 then $p \Vdash \varphi$ is Δ_1 and if φ is Σ_n (respectively Π_n) for $n \geq 1$ then $p \Vdash \varphi$ is Σ_n (respectively Π_n).

Using these forcing relations, we can compute the elementary diagram of the extension if we are given the elementary diagram of the ground model. Indeed, this goes level by level.

Theorem 11. *There is a uniform computable procedure that, given an oracle for the full elementary diagram of a model of set theory $\langle M, \in^M \rangle \models \text{ZF}$ and a notion of forcing $\mathbb{P} \in M$, decides the full elementary diagram of the presentation of $M[G]$ built in Theorem 10. Moreover, this goes level-by-level: given \mathbb{P} and an oracle for the Σ_n -elementary diagram of M , for $n \geq 1$, it can decide the Σ_n -elementary diagram of this presentation.*

Proof. By the process of Theorem 10 we can compute a generic G and the atomic diagram of $M[G]$. Consider next the forcing relations $p \Vdash_{\mathbb{P}} \varphi(\sigma_0, \dots, \sigma_n)$, where $p \in \mathbb{P}$ and φ is an assertion in the language of set theory with \mathbb{P} -name parameters $\sigma_i \in M^{\mathbb{P}}$. For each formula φ , the corresponding forcing relation (as a relation on p and the names σ_i) is definable in the ground model M . Furthermore, the proof that the forcing relations are definable is uniform, in the sense that from any formula φ , we can write down the formula defining the corresponding forcing relation.⁷ So it suffices to prove now that from the Σ_n -diagram of the ground model we can

⁷We are not claiming that the forcing relations are uniformly definable in M , since indeed as φ increases in complexity, the complexity of the definition of $p \Vdash \varphi(\sigma)$ similarly rises. Rather, we only claim here that there is a computational procedure that maps any formula φ to the formula $\text{force}_{\varphi}(p, \sigma)$ defining the forcing relation “ $p \Vdash \varphi(\sigma)$ ” in M .

compute the Σ_n -diagram of the extension. Given the full diagram of the ground model, the same process will compute the full diagram of the extension.

First, we claim that the forcing relation $p \Vdash \varphi(\sigma_0, \dots, \sigma_n)$ for any Δ_0 -formula φ is complexity Δ_1 in the Lévy hierarchy. This can be proved by induction on formulae. The only nontrivial case is the bounded-quantifier case $p \Vdash \exists x \in \tau \varphi(x, \sigma)$, which by the forcing recursion is equivalent to saying that there is a dense collection of conditions $q \leq p$ with some $\langle \rho, r \rangle \in \tau$ such that $q \leq r$ and $q \Vdash \varphi(\rho, \sigma)$. The point is that all these quantifiers remain bounded, and Δ_1 is closed under bounded quantification in set theory. This establishes that given the Δ_0 -diagram of M we can compute the Δ_0 -diagram of $M[G]$, completing the proof of Theorem 10.

We can now prove inductively that the forcing relation $p \Vdash \varphi(\sigma)$ for Σ_n -formulae φ has complexity Σ_n , for $n \geq 1$, and similarly the forcing relation on Π_n -formulae is Π_n .⁸ We can tell if $M[G] \models \varphi(\sigma_0, \dots, \sigma_n)$, for φ a Σ_n -formula, by looking for a condition $p \in G$ such that M satisfies the assertion $p \Vdash_{\mathbb{P}} \varphi(\sigma_0, \dots, \sigma_n)$.⁹ Thus, the Σ_n -diagram of the forcing extension $M[G]$ can be computed from the Σ_n -diagram of M , for $n \geq 1$, as desired. \square

We can also prove a version of this theorem for computable infinitary formulae. By definition the Σ_0^c -formulae are the Boolean combinations of atomic formulae—which, since we work in the Lévy signature, means precisely the Δ_0 -formulae—and these are also the Π_0^c -formulae. For computable ordinals α , the $\Sigma_{\alpha+1}^c$ -formulae are those $\Sigma_{\alpha+1}$ -formulae in $\mathcal{L}_{\omega_1, \omega}$ with finitely many free variables \vec{x} that are computable (countable) disjunctions of formulae $\exists \vec{y}_m \gamma_m(\vec{x}, \vec{y}_m)$, with each γ_m in Π_α^c . (The length k_m of the tuple $\vec{y}_m = (y_{m,1}, \dots, y_{m,k_m})$ of variables may vary over m , but must be computable from m .) The $\Pi_{\alpha+1}^c$ -formulae are their negations. For computable limit ordinals α , the Σ_α^c -formulae are computable disjunctions in which each individual disjunct is Π_β^c for some $\beta < \alpha$; any computable presentation of α may be used in this definition to give the β 's.

Theorem 12. *Fix a computable ordinal α . Then there is a single computable function $f : \omega^2 \rightarrow \omega$ such that, for every model of set theory $\langle M, \in^M \rangle \models \text{ZF}$ and every notion of forcing $p = \mathbb{P} \in M$, the function $f(p, \cdot) : \omega \rightarrow \omega$ is an m -reduction from the Σ_α^c -diagram of the structure $M[G]$ produced by the procedure in Theorem 10 to the Σ_α^c -diagram of M itself. (Here $p \in \omega = \text{dom } M$ is the domain element \mathbb{P} .)*

Proof. Knowing that the characteristic function of $\Delta_0(M[G])$ is given as $\Psi^{\Delta_0(M)}$ for some Turing functional Ψ , we explain the m -reduction between the Σ_1^c -diagrams claimed in the theorem. A Σ_1^c -formula

$$\bigvee_{m \in \omega} \exists \vec{y}_m \gamma_m([\sigma_1]_G, \dots, [\sigma_k]_G, \vec{y}_m),$$

using a computable sequence $\langle \gamma_m \rangle_{m \in \omega}$ of Δ_0 -formulae about a finite tuple from $M[G]$, holds in $M[G]$ just if there exist $m, s \in \omega$, elements $[\tau_1]_G, \dots, [\tau_{k_m}]_G$ in

⁸Note that this argument uses the Replacement axiom schema. This is as $p \Vdash \exists x \varphi(x)$ if and only if there are densely many $q \leq p$ so that $q \Vdash \varphi(\tau)$ for some name τ . It is the Replacement schema that allows us to pull the bounded quantifier over q inside the unbounded quantifiers in the definition for $q \Vdash \varphi(\tau)$ to obtain a Σ_n -formula.

⁹Recall that in this context we are working in the full forcing language with constants for every \mathbb{P} -name, so it is sensible to ask whether $M[G]$ satisfies $\varphi(\sigma_0, \dots, \sigma_n)$ where the σ_i here are constant symbols referring to $(\sigma_i)_G$.

$M[G]$, and a finite initial segment $\rho \subseteq \Delta_0(M)$ such that Ψ^ρ converges within s steps on the Gödel number of the formula $\gamma_m([\sigma_1]_G, \dots, [\sigma_k]_G, [\tau_1]_G, \dots, [\tau_{k_m}]_G)$ and outputs 1, meaning that this Δ_0 -formula holds in $M[G]$. This constitutes a Σ_1^c statement about M itself, quantifying over the $\rho \subseteq \Delta_0(M)$ which cause the program Ψ to halt with value 1 (as well as over m, s , and the \mathbb{P} -names). Since we can compute an index for this Σ_1^c statement about M from the original formula about $M[G]$, we have an m -reduction from $\Sigma_1^c(M[G])$ to $\Sigma_1^c(M)$, as claimed. This same function is an m -reduction from $\Pi_1^c(M[G])$ to $\Pi_1^c(M)$, and analogous arguments hold with any larger computable ordinal $\alpha + 1$ in place of 1, and also for limit ordinals. \square

5. THE GENERIC MULTIVERSE

In the previous section we investigated the computable structure theory of how a model M of set theory relates to a single forcing extension $M[G]$. We turn now to a broader perspective. Given a model $\langle M, \in^M \rangle$ of set theory, the *generic multiverse* of M is the smallest collection of models of set theory which is closed under extension by forcing and by grounds, where W is a *ground* of M if M is a forcing extension of W via a partial order in W . In this section we would like to investigate the extent to which the generic multiverse can be computed from a countable M , extending the analysis in Section 4. Let us begin by looking at grounds.

Lemma 13. *There is a uniform computable procedure which given an oracle for the Π_2 -elementary diagram of a model of set theory $\langle M, \in^M \rangle \models \text{ZFC}$ will compute a list of the Δ_0 -diagrams of the grounds of M .*

Proof. Because the membership relation of a ground of M is the restriction of the membership relation of M and thus M and its grounds agree on Δ_0 truth, all we need to compute is the domains of the grounds. The key fact is the ground model enumeration theorem [FHR15, Theorem 12], which asserts that the grounds of a model of ZFC are uniformly definable by a Π_2 -formula. (See also [BHTU16, Section 2].) That is, there is a Π_2 -formula $\varphi(x, r)$ so that for each r either $\{x : \varphi(x, r)\}$ is empty or else it is a ground [FHR15]. So given the Π_2 -elementary diagram of M we can compute whether $\{x : M \models \varphi(x, r)\}$ is nonempty, say by checking whether $\varphi(\emptyset, r)$ holds. We can thus compute the set $\{(n, x) : M \models \varphi(x, r_n)\}$ where r_n is the n th element r of M , according to the order on ω , so that $\{x : \varphi(x, r)\}$ is nonempty. From this we can get a list of the Δ_0 -diagrams of the grounds of M . \square

Let us highlight the assumption in the statement of this theorem that M satisfies the axiom of choice, an assumption that was missing in the results in Section 4. In [GJ14], Gitman and Johnstone showed for an ordinal δ that DC_δ , a version of the principle of dependent choice which is weaker than the full axiom of choice, suffices to establish that ground models are definable for a certain class of forcings, namely those with a gap at δ . (See their paper for definitions and details.) They conjectured that the ground model definability theorem fails for ZF. This remains an open problem, but Usuba has recently achieved some partial results [Usu19]. We assumed M satisfies the axiom of choice because in this case we do know that the grounds are uniformly definable. If Gitman and Johnstone's conjecture were to be refuted, then we could improve this theorem to assume the model satisfies only ZF instead of ZFC.

Corollary 14. *Given an oracle for the full elementary diagram of a model of set theory $\langle M, \in^M \rangle \models \text{ZFC}$, there is a computable procedure to compute a list of the full elementary diagrams of the grounds of M .*

Proof. This follows using the fact that the translation map on formulae $\varphi \mapsto \varphi^W$ is computable, if W is a definable class, since the translation is merely replacing unbounded quantifiers with quantifiers bounded by W . So from the full elementary diagram of M we can compute a listing of the elementary diagrams of the grounds W of M . \square

We turn now from the grounds to the full generic multiverse. It is not possible to compute a listing of all the models in the generic multiverse for the simple reason that the generic multiverse is uncountable. Even if we restrict to extensions from just a single simple forcing, for instance the forcing to add a Cohen real, there will still be uncountably many extensions. Nevertheless, the *computable generic multiverse* of M , that portion of the multiverse computable from the diagram of M , is close to the full multiverse in a sense we now describe.

Usuba's result that the grounds are strongly downward directed [Usu17] implies that every model in the generic multiverse of M is at most two steps away from M , namely it is a forcing extension of a ground of M . While we cannot in general hope that every model is computable from the elementary diagram of M , we can always compute a model which is a forcing extension of the same ground by the same poset.

Corollary 15. *Let $\langle \bar{N}, \in^{\bar{N}} \rangle$ be a model in the generic multiverse of M , where $\bar{N} = W[\bar{G}]$ for a distinguished ground W of M where \bar{G} is generic over W for a distinguished poset \mathbb{P} . Given an oracle for the full elementary diagram of M there is a computable procedure to compute a W -generic filter $G \subseteq \mathbb{P}$ and decide the full elementary diagram of $N = W[G]$.*

Proof. This follows immediately from Corollary 14 and Theorem 11. \square

On the other hand, there is a different sense in which the computable generic multiverse of M is far from the full generic multiverse of M . Namely, the computable generic multiverse is not dense in the generic multiverse. There are models in the generic multiverse so that no extension of them can be computed from the full diagram for M .

Theorem 16. *Let $\langle M, \in^M \rangle$ be a countable model of ZF. Then there is $M[G]$ a forcing extension of M by the forcing to add a Cohen-generic $G \subseteq \omega^M$ so that no outer model of $M[G]$ has a Δ_0 -diagram computable from the full elementary diagram of M .*

Proof. Let us first describe the generic G . Fix any real z , thought of as an ω -length binary sequence, which is not computable from the full elementary diagram of M . From the diagram of M we can compute a list of the dense subsets of $\text{Add}(\omega, 1)^M$. We use this list to build a generic, in the following manner. Start with $p_0 = \langle z(0) \rangle$. Having built p_n , first extend to meet the n th dense set, minimizing the length of the extension (if there is more than one minimal length extension to the n th dense set then pick arbitrarily). Then put $z(n+1)$ on the end to get p_{n+1} . Because we met every dense set, $G = \bigcup_n p_n$ is generic.¹⁰

¹⁰Note that this process works even if M is ω -nonstandard. In this case, we still have a list, whose ordertype is the real ω , of the dense subsets of $\text{Add}(\omega, 1)^M$. And since for each $a \in \omega^M$ it

Assume now that $\langle N, \in^N \rangle$ is an outer model of $M[G]$, where we think of the universe of N as being \mathbb{N} with \in^N being some binary relation on \mathbb{N} . Let us see how to compute z from the Δ_0 -diagram of N and the full diagram for M . Without loss of generality we may assume that the ordinals $\leq \omega$ in M and N are represented by the same natural numbers, as we may compute an isomorphic copy of N with this property from what we are given. Fix the index of G in N . From the Δ_0 -diagram of N we can compute $z(0)$, simply by asking what the first bit of G is. From the full diagram of M we know the shortest distance we have to extend past $p_0 = \langle z(0) \rangle$ to meet the 0th dense set. So we can compute the next coding point and thereby recover $z(1)$ and p_1 . Continuing this process upward, we can compute $z(n)$ for each n . Therefore, if we could compute the Δ_0 -diagram of N from the full diagram of M then we could compute z from the full diagram of M , which would be a contradiction. \square

Next we wish to discuss the extent to which the computable generic multiverse has the same structural properties as the full generic multiverse. Let us start with the following property, due essentially to Mostowski [Mos76] and recently extended in [HHKVW19]. Given a collection \mathcal{E} of models in the generic multiverse, say that \mathcal{E} is *amalgamable* when there is a model in the generic multiverse which contains every model in \mathcal{E} . Note that if each model in \mathcal{E} is a forcing extension of M it is equivalent to ask whether there is a forcing extension of M which contains every model in \mathcal{E} .

Theorem 17 (Mostowski). *Let I be a finite set and let \mathcal{A} be a family of subsets of I which contains all singletons and is closed under subsets. Let $\langle M, \in^M \rangle \models \text{ZF}$ be a countable model of set theory. Then there are reals $c_i \subseteq \omega^M$ for $i \in I$ so that each c_i is Cohen-generic over M and for $A \subseteq I$ the family $\{M[c_i] : i \in A\}$ is amalgamable if and only if $A \in \mathcal{A}$.*

The same phenomenon happens within the computable generic multiverse.

Theorem 18. *Let I be a finite set and let \mathcal{A} be a family of subsets of I which contains all singletons and is closed under subsets. Let $\langle M, \in^M \rangle \models \text{ZF}$ be a countable model of set theory. Then from an oracle for the elementary diagram of M there is a procedure to compute Cohen reals $c_i \subseteq \omega^M$ generic over M and the elementary diagrams of $M[c_i]$ for $i \in I$ so that for $A \subseteq I$ the family $\{M[c_i] : i \in A\}$ is amalgamable in the generic multiverse if and only if $A \in \mathcal{A}$.*

Proof. In the language of [HHKVW19], let z be a *catastrophic real* for M ; that is, z is a real so that no outer model of M can contain z . We claim that there is such z computable from the elementary diagram of M . Namely, we can take z to be an isomorphic copy of \in^M on ω , along with with an isomorphism onto M . Then, by the Mostowski collapse lemma, any model of ZF which contains z would have to contain \in^M itself as a set, which is impossible for an outer model of M .

Without loss of generality we may assume that $I, \mathcal{A} \in M$. For each $A \subseteq I$ let \mathbb{P}_A be the forcing $\prod_{i \in A} \text{Add}(\omega, 1) \in M$. From the elementary diagram of M we can compute a listing, in ordertype ω , of all pairs $\langle A, D \rangle$ with $A \in \mathcal{A}$ and $D \in M$ a dense subset of \mathbb{P}_A . We build the Cohen reals c_i by means of a descending sequence of conditions, which we think of as filling in an $\omega^M \times I$ matrix with 0s and 1s, with

is dense in $\text{Add}(\omega, 1)^M$ to have a condition with length $\geq a$, the G we produce is unbounded in ω^M .

the i th column growing into c_i . We will ensure that at each step of the construction we have built all columns up to the same height.

We start with a completely empty matrix, i.e. with $c_i^0 = \emptyset$ for all $i \in I$. Now suppose we have built up c_i^n . We are presented with $A_n \in \mathcal{A}$ and $D_n \subseteq \mathbb{P}_A$ dense. Extend the columns with index in A_n to collectively meet D_n , then pad with 1s to ensure the columns all have the same height. Next, pad the remaining columns with 0s to build them up to the same height, then extend each column by appending a row of 1s followed by a row of $z(n)$'s. These rows of 1s are the coding points which will be used to recover $z(n)$ if we are dealing with $A \notin \mathcal{A}$. Note that this process is computable given the Δ_0 -diagram of M , since from that we can compute the minimal length we need to extend to meet D_n and then pick one of the finitely many extensions of that length. So if we set c_i to be the generic determined by $\langle c_i^n : n \in \omega \rangle$ then c_i is computable from the diagram of M . And so, once we know the c_i 's are generic we know that we can, as before, compute the full diagrams of the $M[c_i]$'s.

It remains only to see that the $M[c_i]$'s have the desired amalgamability property. First, suppose that $A \in \mathcal{A}$. Then we built up $\{c_i : i \in A\}$ so that they met every dense subset of \mathbb{P}_A in M . So the c_i for $i \in A$ are mutually generic and so the family $\{M[c_i] : i \in A\}$ is amalgamable, witnessed by $M[c_i : i \in A]$. In particular, this shows that each c_i is generic over M . Now suppose that $A \in \mathcal{P}(I) \setminus \mathcal{A}$. Then, by the construction, the only rows in which each c_i for $i \in A$ has value 1 are the coding points identifying where the bits of z are coded. So no outer model of M which satisfies ZFC can contain each c_i for $i \in A$. \square

In [HHKVW19], the first and third author along with Habič, Klausner, and Verner extended Mostowski's theorem. We will not reproduce that article here, but we wish to note that the constructions therein are all effective. Given an oracle for the elementary diagram of M there are computable procedures to compute the desired generics for the results from that article. So the properties of the generic multiverse explored in that article are also enjoyed by the computable generic multiverse.

To close out this section, we remark that the existence of many (non-isomorphic) grounds for the same countable model of ZFC implies that in general it is impossible to recover M effectively—or even non-effectively—from an arbitrary copy of $M[G]$. M has a canonical embedding into $M[G]$ by the map sending each $x \in M$ to the class $[\check{x}]_G$, where the \mathbb{P} -name \check{x} is defined by recursion as $\{\langle \check{y}, 1 \rangle : y \in x\}$. With a $\Delta_0(M)$ -oracle, one can compute \check{x} from x , by the methods seen earlier for recursive definitions, and thereby compute the canonical embedding of M into $M[G]$. Its image will thus be $\Delta_0(M)$ -computably enumerable in $M[G]$, but it is not defined uniformly across copies of $M[G]$.

The question of how to recover a copy of M from a copy of $M[G]$ is closely tied to the question of whether $\Delta_0(M[G])$ can compute $\Delta_0(M)$, and if so, whether there is a uniform procedure for doing so from all copies of $M[G]$. The answer is not obvious. Indeed, one can imagine the possibility that the isomorphism type of the structure $M[G]$ may be simpler in some sense than that of M , and that therefore there may exist a copy of $M[G]$ that cannot compute any copy of M . For example, perhaps $M[G]$ satisfies GCH, whereas the map $\kappa \mapsto 2^\kappa$ on cardinals in M may have been far more chaotic and may have encoded some information not intrinsically recoverable from $M[G]$.

There is an analogy here to computable fields. Rabin’s Theorem states that for every computable field F , the algebraic closure \overline{F} also has a computable presentation, and that both a copy of \overline{F} and an embedding of F into that copy may be computed uniformly from the atomic diagram of F , in the signature with $+$ and \cdot . This much is analogous to our results above for a given (M, \mathbb{P}) . However, the uniformity carries over to countable fields F that are not computably presentable, and in this case the algebraic closure may be far simpler than any presentation of F , as every countable algebraically closed field has a computable presentation. Taking the algebraic closure smoothes out a field and eliminates complexity, and we ask whether the same might happen with a forcing extension of a model of set theory.

Question 19. *Let M be a countable model of ZFC, and $\mathbb{P} \in M$ a forcing notion, for which the procedure in Theorem 10 computes a filter G and the atomic diagram of a presentation of $M[G]$. Can there exist a presentation $\mathcal{A} \cong M[G]$ such that for every presentation $M^* \cong M$, we have $\Delta_0(M^*) \not\leq_T \Delta_0(\mathcal{A})$? And if so, can this presentation \mathcal{A} be the one computed by our procedure?*

6. CLASS FORCING

Elsewhere, we have restricted our attention to forcing notions which are set-sized. We detour in this section to consider proper class-sized forcing notions. There are two major approaches to formulate class forcing, and we consider both of them. The first approach, let us call it the first-order approach, is to work over ZF (possibly assuming more) and deal with a definable class. A generic then has to meet every definable dense subclass of the forcing notion.

Theorem 20. *Given an oracle for the full elementary diagram of a countable model $\langle M, \in^M \rangle \models \text{ZF}$ and given a definable pretame class forcing $\mathbb{P} \subseteq M$, there is a computable procedure to compute an M -generic filter $G \subseteq \mathbb{P}$ and decide the full elementary diagram of the forcing extension $M[G]$.*

See [Fri00, Section 2.2] for a definition of pretameness, which is equivalent to the preservation of ZF^- (in the language with a predicate for the generic filter). The reason to ask \mathbb{P} to be pretame is that the pretame forcings are precisely those which have a definable atomic forcing relation [HKS18].

Proof. From the full diagram of M we can compute a list of the definable dense subclasses of \mathbb{P} . So we can compute G as in Theorem 8. Now given the atomic forcing relation for \mathbb{P} there is a computable procedure to associate a formula φ with the formula defining the corresponding forcing relation $p \Vdash \varphi$. So we can compute $M[G]$ and its full elementary diagram as in Theorem 11. \square

This answers the question for the first-order approach. We turn now to the other approach, call it the second-order approach. For this approach classes are actual objects in our models, where we work (in first-order logic) with two-sorted structures. We work over a second-order set theory, such as Gödel–Bernays set theory GB or Kelley–Morse set theory KM.¹¹ We will use italic letters such M to

¹¹GB is the weaker of the two, stating the existence of classes defined predicatively—quantification is allowed only over sets. On the other hand, KM allows for impredicative comprehension, defining classes by quantifying over the classes. See [Wil19, Section 2] for precise axiomatizations of these two theories, as well as a discussion for their place in the hierarchies of second-order set theories.

refer to the sets and calligraphic letters such as \mathcal{M} to refer to the classes of a model of second-order set theory. Abusing notation slightly, we will also use \mathcal{M} to refer to the whole model; this is unambiguous, as the sets are definable from the classes. A class forcing notion \mathbb{P} is then a class in the model and a generic meets every dense subclass of \mathbb{P} in the model. We will write $\mathcal{M}[G]$ for the extension by a generic $G \subseteq \mathbb{P}$.¹²

Let Δ_0^1 refer to the class of formulae with only set quantifiers. Up to equivalence, this is the same as the class of formulae where all quantifiers are bounded, possibly by classes, because $\exists x \varphi(x)$ is equivalent to $\exists x \in V \varphi(x)$. So this is the second-order analogue of Δ_0 in the first-order Lévy hierarchy.

Theorem 21. *Let $\langle \mathcal{M}, \in^{\mathcal{M}} \rangle$ be a countable model of GB and suppose $\mathbb{P} \in \mathcal{M}$ is a class forcing notion with its atomic forcing relation a class in \mathcal{M} . Then, from an oracle for the Δ_0^1 -elementary diagram of \mathcal{M} there is a computable procedure to compute an \mathcal{M} -generic filter $G \subseteq \mathbb{P}$ and the Δ_0^1 -diagram of $\mathcal{M}[G]$.*

Proof. Begin by observing that it is Δ_0^1 to say that a class is a dense subclass of \mathbb{P} . So from the Δ_0^1 -diagram for \mathcal{M} we can compute a list of all the dense subclasses of \mathbb{P} . We can then compute an \mathcal{M} -generic filter $G \subseteq \mathbb{P}$ as in Theorem 8. Next, note that being a class \mathbb{P} -name is a Δ_0^1 property, because a class \mathbb{P} -name is a class whose elements are all pairs of elements of \mathbb{P} and a set \mathbb{P} -name, and being a set \mathbb{P} -name is a Δ_0^1 property. So we can decide which elements of \mathcal{M} are \mathbb{P} -names. Further observe that the relations

$$p \Vdash \sigma \in \tau, \quad p \Vdash \sigma = \tau, \quad p \Vdash \sigma \in T, \quad p \Vdash \Sigma = T$$

are all Δ_0^1 . As in the proof of Theorem 11 given the generic G we can define relations $=_G$ and \in_G on the \mathbb{P} -names, which are decidable from the Δ_0^1 -diagram. And so we can pick out representatives of the $=_G$ -equivalence classes, thereby computing the atomic diagram of $\mathcal{M}[G]$. Finally, observe that for a Δ_0^1 -formula φ that $p \Vdash \varphi(\Sigma_0, \dots, \Sigma_n)$ is Δ_0^1 , as it is defined from the atomic forcing relation by quantifying over sets. So we can decide the Δ_0^1 -diagram of $\mathcal{M}[G]$. \square

As in the set forcing case, from the full elementary diagram of \mathcal{M} we can compute the full elementary diagram of the class forcing extension.

Theorem 22. *Let $\langle \mathcal{M}, \in^{\mathcal{M}} \rangle$ be a countable model of GB and suppose $\mathbb{P} \in \mathcal{M}$ is a class forcing notion with its atomic forcing relation in \mathcal{M} . Then, from an oracle for the second-order elementary diagram of \mathcal{M} there is a computable procedure to compute an \mathcal{M} -generic filter $G \subseteq \mathbb{P}$ and the second-order elementary diagram of $\mathcal{M}[G]$.*

To clarify, by the *second-order elementary diagram* of a model of second-order set theory we mean second-order in the sense of allowing quantifying over the classes of the model—that is, in the Henkin semantics—not second-order in some external sense.

¹²Our approach is in first-order logic with two-sorted structures, but as is well-known one can equivalently work in second-order logic with Henkin semantics. In this semantics, one explicitly lists out the classes considered for the second-order part of semantics. Note, however, that to make this approach amenable to the context of computable structure theory we would have to attach to each model a list of the classes to be used for its second-order semantics. That is, we would need natural numbers to identify each class and include a relation for the set-class membership relation. This amounts to the same as the approach in first-order logic.

Proof. By Theorem 21 we can compute G and the Δ_0^1 -diagram of $\mathcal{M}[G]$. The result for the full diagram then follows from the fact that $p \Vdash \varphi(T_0, \dots, T_n)$ is second-order definable when φ is second-order. \square

It is natural to ask whether this goes level-by-level. The Σ_n^1 and Π_n^1 -formulae are inductively defined from the Δ_0^1 -formulae similar to how in first-order set theory the Σ_n and Π_n -formulae are defined from the Δ_0 -formulae. For instance, a formula is Σ_2^1 if it is of the form $\exists X \forall Y \varphi(X, Y)$, where both quantifiers are over the classes and φ is Δ_0^1 . To argue this goes level-by-level as in the set forcing case we would need that if φ is Σ_n^1 then $p \Vdash \varphi$ is Σ_n^1 . To prove the analogous fact for the set forcing case we used the Replacement schema.¹³ The same argument works in the class forcing case if our model satisfies the Class Collection schema, a second-order version of the Replacement schema.

Definition 23. *The Class Collection axiom schema asserts that if for every set there is a class satisfying some property, then there is a single class coding the “meta-class” consisting of a witnessing class for every set. Formally, instances of this schema take the form*

$$\forall \bar{P} [(\forall x \exists Y \varphi(x, Y, \bar{P})) \Rightarrow (\exists C \forall x \exists i \varphi(x, (C)_i, \bar{P}))],$$

where $(C)_i = \{y : (i, y) \in C\}$.¹⁴ For $n \in \omega$, the Σ_n^1 -Class Collection schema is the restriction of Class Collection to Σ_n^1 -formulae.

It is simple to check that Σ_n^1 -Class Collection implies Σ_n^1 -Comprehension. The converse does not hold. Even KM with the full impredicative Comprehension schema cannot prove Σ_0^1 -Class Collection [GHK]. However, adding Class Collection does not increase consistency strength—see [MM75, Theorem 2.5] for the KM/full Class Collection case and [Rat79] for the level-by-level case.

Corollary 24. *Let $\langle \mathcal{M}, \in^{\mathcal{M}} \rangle$ be a countable model of GB + Σ_n^1 -Class Collection and suppose $\mathbb{P} \in \mathcal{M}$ is a class forcing notion with its atomic forcing relation in \mathcal{M} . Then, from an oracle for the Σ_n^1 -elementary diagram of \mathcal{M} there is a computable procedure to produce an \mathcal{M} -generic filter $G \subseteq \mathbb{P}$ and the Σ_n^1 -elementary diagram of $\mathcal{M}[G]$. \square*

7. FUNCTORIALITY AND INTERPRETABILITY

In this article we are considering an effective procedure mapping one class of models on the domain ω to another such class (in fact, to the same class). In this section we recall some known theorems about this scenario and what it says about effective interpretability, relating these general facts to our specific case of a model M of set theory inside a forcing extension $M[G]$.

Theorems from [HTMMM17] and [HTMM18] relate such procedures to the interpretability of each output model (in the second class) in the corresponding input model (in the first class). Here we repeat the simplest versions of those theorems. The gist is that interpretations of one structure in another by $\mathcal{L}_{\omega_1, \omega}$ -formulae (and with no fixed arity on the domain of the interpretation) correspond bijectively to functors from the category of isomorphic copies of the second structure (with

¹³Cf. Footnote 8.

¹⁴Observe that in the presence of Global Choice—the assertion that every class can be well-ordered—we may equivalently ask that this index i for x be x itself.

isomorphisms as the morphisms in the category) into the category of isomorphic copies of the first. Rather than attempt to define all the terms here, we refer the reader to [HTMMM17, Definition 1.2], [HTMM18, Definition 2.1], and [Mon14, Definition 5.1] for the notions of interpretability, and to [HTMMM17, Definition 1.2], [HTMM18, Definition 2.7], and [MPSS18, Definition 3.1] for the notions about functors.

Theorem 25 (Theorem 1.5 of [HTMMM17]). *Let \mathcal{A} and \mathcal{B} be countable structures. Then \mathcal{A} is effectively interpretable in \mathcal{B} if and only if there exists a computable functor from the category $\text{Iso}(\mathcal{B})$ of isomorphic copies of \mathcal{B} (under isomorphism) to the corresponding category $\text{Iso}(\mathcal{A})$.*

Theorem 26 (Theorem 2.9 of [HTMM18]). *Let \mathcal{B} and \mathcal{A} be countable structures, possibly in different countable languages. For each Baire-measurable functor $F : \text{Iso}(\mathcal{B}) \rightarrow \text{Iso}(\mathcal{A})$ there is an infinitary interpretation \mathcal{I} of \mathcal{A} within \mathcal{B} , such that F is naturally isomorphic to the functor $F_{\mathcal{I}}$ associated to \mathcal{I} . Furthermore, if F is Δ_{α}^0 in the lightface Borel hierarchy, then the interpretation can be taken to use computable $\Delta_{\alpha}^0 \mathcal{L}_{\omega_1, \omega}$ -formulae and the isomorphism between F and $F_{\mathcal{I}}$ can be taken to be Δ_{α}^0 .*

Those sources also extend these theorems to situations of bi-interpretability between countable structures, and (of greater relevance here) to interpretations that hold uniformly from one class of structures to another, rather than interpretations merely of \mathcal{A} in \mathcal{B} . For us, the point is simply that results about functors and their complexity, such as the simple ones we derive in this section, correspond to results about interpretability. For example, a computable functor mapping copies of M to copies of $M[G]$ would correspond to an effective interpretation of $M[G]$ in M .

A computable functor is a particularly strong kind of operator on countable structures. Characteristically, attempting to create such an operator requires one to determine exactly which aspects of the structure are relevant to the operation. The following analysis is not difficult, but it serves as a good example of this principle.

We here extend the Lévy signature defined above to a larger signature. Along with the symbols from the former, this new signature has countably many constants p , c , and d_0, d_1, \dots . The intention (which can be expressed as an $\mathcal{L}_{\omega_1, \omega}$ -formula, but not by any finitary axiom) is that p denotes the partial order \mathbb{P} used for forcing in this structure, that the constants d_j name precisely the dense subsets of \mathbb{P} , and that c names a choice function on the nonempty subsets of \mathbb{P} in M . We consider models of ZFC here, not just ZF, to ensure that M will contain such a choice function, although in many cases \mathbb{P} will have a readily definable choice function. It is important that c be internal to M , and so formally we treat c as a constant symbol, rather than a function symbol. The domain of our functor is the category of models of ZFC on the domain ω , in our larger signature, satisfying these conditions. The morphisms between two such structures are exactly the isomorphisms of structures in this signature. The range of the functor is the category of models of ZFC on the domain ω in the Lévy signature, since in the forcing extension, the forcing notion is no longer relevant. (If desired, one can use the larger signature, with the check-name of p^M as $p^{M[G]}$ and similarly for c and the d_j .)

The construction of the generic G is now done effectively by a prescribed method, which is more precise than the method of Theorem 10. The added precision is

necessary to allow the functor to be computed with no more constants than these. In the new method, the functional Φ is given as an oracle the atomic diagram of M in the larger signature, namely $\Delta_0(M) \oplus \langle p^M, c^M, d_0^M, d_1^M, \dots \rangle$, and searches first for the maximum element p_0 of the forcing notion \mathbb{P} named by the constant p . The Δ_0 -diagram identifies it, as it is defined by $p_0 \in \mathbb{P} \wedge (\forall q \in \mathbb{P}) q \leq_{\mathbb{P}} p_0$. Next, for each s in turn, we find the element d_s^M and use c to select an extension $p_{s+1} \leq p_s$ in $d_s \cap \mathbb{P}$. We can do so because the set $x = \{q \in {}^M d_s : q \leq p_s\} \in M$ is Δ_0 -definable from parameters we have access to. So we can search for the element of M which satisfies this defining property and know that the search must terminate. Then we can search for $c(x) \in M$ and choose it to be p_{s+1} . While we use the $<$ relation on the domain ω of M to search for x and then $c(x)$, note that this procedure is nevertheless uniform as the search is looking for a uniquely determined object.

If $f : M \rightarrow M^*$ is an isomorphism (in the larger signature), then the same procedure Φ on $\Delta_0(M^*, p^{M^*}, \vec{d}^{M^*}, c^{M^*})$ will clearly produce a forcing extension $M^*[G^*]$ isomorphic to $M[G]$. The point of this method is that there is a Turing functional Φ_* which from the oracle

$$\Delta_0(M, p^M, \vec{d}^M, c^M) \oplus f \oplus \Delta_0(M^*, p^{M^*}, \vec{d}^{M^*}, c^{M^*})$$

computes the corresponding isomorphism from $M[G]$ onto $M^*[G^*]$. This map on isomorphisms will respect composition and, if given the identity isomorphism on M , will produce the identity isomorphism on $M[G]$. We do not consider it necessary to write out the proof of this result. This Φ_* , together with the functional Φ that computes $\Delta_0(M[G]) = \Phi^{\Delta_0(M)}$, constitutes a program for the forcing functor, showing that \mathcal{F} is in fact a computable functor, according to the definition in [MPSS18].

We remark additionally that the forcing functor works exactly the same way in the category with the same objects as above, but in which the morphisms now include all injective homomorphisms from any one forcing structure to any other. Of course, the image of a morphism g now need only be an injective homomorphism itself, not an isomorphism, but functoriality still holds, effectively.

None of the foregoing is at all difficult. With the new constant symbols enumerating the dense sets we get functoriality but, as will be established in the next section, if we omit them we do not get functoriality. Meanwhile, though, we can apply Theorem 25 to our computable functor and extract from it an effective interpretation, uniformly across all forcing extensions built by the functor.

Corollary 27. *There exist fixed computable infinitary Σ_1 -formulae that, for every ground model $(M, \Delta_0(M), p, c, d_0, d_1, \dots)$ of ZFC in the larger signature, give an effective interpretation of the forcing extension $M[G]$ in $(M, \Delta_0(M), p, c, d_0, d_1, \dots)$, provided G is built from this presentation as described in this section. (The interpretation allows tuples from M of arbitrary finite arity in its domain.)*

8. NON-FUNCTORIALITY

In the main theorem, we proved that there is a computable procedure

$$(M, \in^M, \mathbb{P}) \mapsto G$$

that takes as input the atomic diagram of a model of set theory $\langle M, \in^M, \mathbb{P} \rangle$ with a distinguished notion of forcing $\mathbb{P} \in M$, and produces an M -generic filter $G \subseteq \mathbb{P}$. We had observed, however, that the particular filter G arising from this process

can depend on how exactly the atomic diagram of M is presented to us. If we rearrange the copy of M as it is represented on the natural numbers (to be used as an oracle for the computation), then this can affect the order in which the dense sets d_s appear and therefore affect which conditions p_s arise in the construction. In short, the computational procedure we provided does not respect isomorphisms of presentations, since different isomorphic presentations of the same model can lead to non-isomorphic generic filters. Thus the process of computing the generic filter is not functorial in the category of presentations of models of set theory under isomorphism.

We claimed in Section 7 that in this signature this phenomenon is unavoidable. To justify that claim, we now show that even if one allows the full elementary diagram of the model M as input, there is no effective procedure that will pick out the same generic filter in all presentations of the same model.

Theorem 28. *If ZF is consistent, then there is no computable procedure that takes as input the elementary diagram of a model of set theory $\langle M, \in^M, \mathbb{P} \rangle$ with a partial order \mathbb{P} and produces an M -generic filter $G \subseteq \mathbb{P}$, such that isomorphic copies of the input model result always in the same corresponding isomorphic copy of G .*

In other words, there is no computable procedure to produce generic filters that is functorial in the category of presentations of models of set theory under isomorphism.

Proof. Assume toward contradiction that we have a computable procedure

$$\Phi : \Delta(M, \in^M, \mathbb{P}) \mapsto G,$$

where we assume $M = \mathbb{N}$ and $\langle M, \in^M, \mathbb{P} \rangle \models \text{ZF}$ and $\mathbb{P} \in M$ is a partial order in M , such that $G \subseteq \mathbb{P}$ is M -generic and this process is functorial, in the sense that isomorphic presentations of $\langle M, \in^M \rangle$ lead always to the same isomorphic copy of G .

We assumed that ZF is consistent. It follows by the Lévy–Montague reflection principle and a simple compactness argument that there is a countable model $M \models \text{ZF}$ such that $M_\kappa \prec M$ for some cardinal κ in M , where M_κ means the rank-initial segment $(V_\kappa)^M$.¹⁵ Fix any nontrivial forcing notion $\mathbb{P} \in M_\kappa$, such as the forcing to add a Cohen real.

The main idea of the proof is to try to run the computational process inside the model M . This doesn't make literal sense, of course, since in M we do not have the elementary diagram of M as a countable set. Nevertheless, in M we do have the elementary diagram of M_κ , which although uncountable, is a set in M and therefore has its diagram in M . It may be that M is ω -nonstandard and therefore may also have nonstandard-length formulae in its version of the diagram, which of course cannot be part of the real elementary diagram. But on the standard-length formulae, M will agree with us on the elementary diagram of M_κ .

Inside M , consider all the ways that we might place finitely much information about the elementary diagram of M_κ . Specifically, we enumerate finitely many

¹⁵In general, M may be ω -nonstandard. But note that the existence of well-founded M with $M_\kappa \prec M$ is consistent relative an inaccessible cardinal (indeed, less is necessary). However, this assumption is higher in consistency strength than just asking for a well-founded model of set theory.

elements of M_κ and then list off finitely many truth judgments about those elements, and use this as a partial oracle in the computational process of Φ .

If outside M we were to consider a full presentation of the elementary diagram of M_κ , then the process Φ will result in complete judgments about membership in an M_κ -generic filter $G \subseteq \mathbb{P}$, and furthermore these judgments will be independent of the presentation of the diagram we consider. It follows that for any particular condition $p \in \mathbb{P}$, there is a finite piece of the (full, actual) presentation that leads to a judgment by Φ either that $p \in G$ or that $p \notin G$.

The key observation is that this finite piece of the full actual diagram of M_κ will be an element of M and therefore M will be able to see the judgment that Φ makes on whether $p \in G$ or not. Furthermore, all minimal-length such pieces of the diagram of M_κ in M that decide whether $p \in G$ or not will agree on the outcome, since any such minimal-length piece will involve only standard-finite formulae and therefore in principle can be continued outside M to a presentation of the full actual diagram of M_κ , which always gives the same result about membership in G . Consequently, by searching inside M for these minimal-length supporting computations about M_κ , we conclude that $G \in M$.

But this is impossible, since all subsets of \mathbb{P} in M are in M_κ , since M_κ is a rank-initial segment of M , and so G will actually be M -generic, while also an element of M , which is impossible for nontrivial forcings. \square

It follows that, given suitable consistency assumptions, there also can be no functorial computable procedure for producing just the atomic diagram of the forcing extension $M[G]$, since there are some forcing notions \mathbb{P} , such as the self-encoding forcing defined in [FHR15], that produce unique generic filters in their extensions; for such forcing notions over a well-founded ground model, the extensions are isomorphic if and only if the generic filters are identical.

In the argument for the proof of Theorem 28 we produced a specific model which witnessed the failure of a would-be Turing functor Φ . It is natural to wonder whether given any Φ and given any countable model M of set theory we can witness the failure of Φ with an isomorphic copy of M .

The following theorem answers this question in the negative. If we restrict to the pointwise definable models then we can produce generics in a functorial manner. A model of set theory is *pointwise definable* if every element of the model is definable without parameters. For example, the Shepherdson–Cohen minimum transitive model of ZF is pointwise definable.

Theorem 29. *There is a computable functor Φ , in which Φ takes as input the elementary diagram of any pointwise definable model $\langle M, \in^M \rangle \models \text{ZFC}$ and a forcing notion $\mathbb{P} \in M$ and returns an M -generic $G \subseteq \mathbb{P}$ and the elementary diagram of $M[G]$. That is, if $\langle M^*, \in^{M^*} \rangle$ and $\langle M^\dagger, \in^{M^\dagger} \rangle$ are two isomorphic presentations of M then $\Phi(M^*, \in^{M^*}, \mathbb{P}^*) \cong \Phi(M^\dagger, \in^{M^\dagger}, \mathbb{P}^\dagger)$.*

Proof. The key step in getting functoriality is to ensure that isomorphic presentations give rise to corresponding isomorphic generic filters. That is, if $\pi : M^* \rightarrow M^\dagger$ is an isomorphism we want that $\pi \circ G^* = G^\dagger$, where G^* and G^\dagger are the generics produced by Φ . Given this it is then straightforward that the isomorphism between M^* and M^\dagger extends to an isomorphism between $M^*[G^*]$ and $M^\dagger[G^\dagger]$.

Let us see how to produce the generic. There is a canonical computable listing of the possible definitions in the language of set theory in ordertype ω . From

the elementary diagram of M we can decide which element is defined by which definition. So using the fact that M is pointwise definable we can produce a listing m_0, m_1, \dots of the elements of M , starting with the element defined by the zeroth definition, then the element defined by the first definition, and so on. This listing is canonical, in the sense that isomorphic copies of M will give rise to the same listing. More formally: if $\pi : M^* \rightarrow M^\dagger$ is an isomorphism then $\pi(m_i^*) = m_i^\dagger$ for all $i \in \omega$. We then use this listing to list out the dense subsets of \mathbb{P} and to define the generic, as in the argument for Theorem 8. Because isomorphic copies of M will use the same listing of their elements, this process will produce the same generic. \square

We claim that the non-functoriality result extends beyond computability to the case of Borel processes. Note that in the Borel context, from the atomic diagram of a model we can recover its full elementary diagram by a Borel function, and so the distinction between those cases evaporates.

Theorem 30. *Suppose ZF is consistent. Then there is no Borel function*

$$(M, \in^M, \mathbb{P}) \mapsto G$$

mapping codes for countable $\langle M, \in^M, \mathbb{P} \rangle \models \text{ZF}$ with a forcing notion $\mathbb{P} \in M$ to an M -generic filter $G \subseteq \mathbb{P}$, such that isomorphic models lead always to the same (isomorphic) filter. Indeed, we cannot even get such a Borel function so that if $\langle M^, \in^{M^*}, \mathbb{P}^* \rangle$ and $\langle M^\dagger, \in^{M^\dagger}, \mathbb{P}^\dagger \rangle$ are elementarily equivalent then so are $\langle M^*[G^*], \in^{M^*[G^*]} \rangle$ and $\langle M^\dagger[G^\dagger], \in^{M^\dagger[G^\dagger]} \rangle$.*

We first proved this result with a mild extra consistency assumption. Philipp Schlicht privately communicated to us an argument which needs only the minimum assumption that ZF is consistent. With his permission, it is his argument we present here.

Proof (Schlicht). Suppose that there is a Borel function $\Phi(M, \in^M, \mathbb{P}) = G$, defined from some real parameter y , such that if $\langle M^*, \in^{M^*}, \mathbb{P}^* \rangle$ and $\langle M^\dagger, \in^{M^\dagger}, \mathbb{P}^\dagger \rangle$ are codes for elementarily equivalent countable models of ZF (coded, say, as a binary relation \in^M on the natural numbers $M = \mathbb{N}$) equipped with forcing notions $\mathbb{P}^* \in M^*$ and $\mathbb{P}^\dagger \in M^\dagger$, then if $G^* = \Phi(M^*, \in^{M^*}, \mathbb{P}^*)$ and $G^\dagger = \Phi(M^\dagger, \in^{M^\dagger}, \mathbb{P}^\dagger)$ then $\langle M^*[G^*], \in^{M^*[G^*]} \rangle$ and $\langle M^\dagger[G^\dagger], \in^{M^\dagger[G^\dagger]} \rangle$ are elementarily equivalent. In particular, if $\langle M^*, \in^{M^*} \rangle$ and $\langle M^\dagger, \in^{M^\dagger} \rangle$ are isomorphic, then Φ produces elementarily equivalent forcing extensions.

The desired counterexample will be a pair of isomorphic presentations of a Cohen-generic extension of a pointwise definable model of $\text{ZF} + V = L$. Observe that the existence of such follows from the assumption that ZF is consistent: A famous result due to Gödel gives us a model of $\text{ZF} + V = L$ and using that model's definable global well-order we get Skolem functions so that the Skolem hull of the empty set is pointwise definable. Call this pointwise definable model $\langle N, \in^N \rangle$.

Work in N . Let $\mathbb{Q} \in N$ denote the Cohen forcing poset to add a single real (in the sense of N) and let $\rho \in N$ be a \mathbb{Q} -name for the set of finite variants of the generic real. Consider $\dot{\mathbb{P}}$ a \mathbb{Q} -name so that $\mathbb{1}_{\mathbb{Q}}$ forces that $\dot{\mathbb{P}}$ is the poset which chooses an element of ρ by lottery and codes that real into the continuum pattern

below \aleph_ω , via adding Cohen-generic subsets to the \aleph_n .¹⁶ Observe that if the generic real is modified on a finite domain, then this forcing does not change.

In V , let H be the transitive collapse of a countable elementary submodel of some large enough H_θ which contains y and a code for N . Observe that we can think of \mathbb{Q} as a forcing poset in H . Of course, N may be ω -nonstandard and so Cohen forcing in the sense of N need not be the real Cohen forcing. Nevertheless, H sees that \mathbb{Q} is a poset, as that is absolute even for nonstandard models of set theory. Now let x be H -generic for \mathbb{Q} . Then x must also be N -generic for \mathbb{Q} . Set $M = N[x]$ and $\mathbb{P} = \dot{\mathbb{P}}^x \in M$. Then $\langle M, \in^M, \mathbb{P} \rangle$ will yield the desired counterexample.

Because M is countable in $H[x]$, it has a real code. Let $\mu, \varepsilon, \pi \in H$ be \mathbb{Q} -names so that $\langle M^*, \in^{M^*}, \mathbb{P}^* \rangle = \langle \mu^x, \varepsilon^x, \pi^x \rangle$ is an isomorphic copy of $\langle M, \in^M, \mathbb{P} \rangle$ on the natural numbers. Moreover, fix a condition $q \in x$ which forces this. Then $\langle M^*, \in^{M^*}, \mathbb{P}^* \rangle$ is appropriate as input to Φ . Because Φ is Borel as defined using a real parameter $y \in H$, we have that $H[x]$ can compute $\Phi(M^*, \in^{M^*}, \mathbb{P}^*)$. Moreover, in the forcing extension of $\langle M^*, \in^{M^*}, \mathbb{P}^* \rangle$ via the generic output by Φ , call it $M^*[G^*]$, we can check the continuum pattern below \aleph_ω to determine which finite variant of x was used in the lottery sum.

Take p in this generic G^* which forces that the finite variant chosen to code into the continuum pattern is $x \triangle a$. Now define x' so that $x'(i) \neq x(i)$ for $i = \max(a \cup \{|p|, |q|\}) + 1$ and x' agrees with x on all other coordinates. Then x' extends q and is H -generic for \mathbb{Q} , and thus also N -generic. In particular, in $N[x']$ we have that p forces the continuum pattern in the extension to be $x' \triangle a$. And since x' and x differ on a single coordinate, $M = N[x] = N[x']$ and $\mathbb{P} = \dot{\mathbb{P}}^x = \dot{\mathbb{P}}^{x'}$. Thus, $\langle M^\dagger, \in^{M^\dagger}, \mathbb{P}^\dagger \rangle = \langle \mu^{x'}, \varepsilon^{x'}, \pi^{x'} \rangle$ is isomorphic to $\langle M^*, \in^{M^*}, \mathbb{P}^* \rangle$. Note now that $p \in G^\dagger = \Phi(M^\dagger, \in^{M^\dagger}, \mathbb{P}^\dagger)$ because $H[x] = H[x']$ and x' was defined to agree with x up to the amount of information needed about μ, ε , and π to have Φ put p in the generic. But the models $M^*[G^*]$ and $M^\dagger[G^\dagger]$ have different continuum patterns, disagreeing at \aleph_i .

The statements ‘‘GCH holds at \aleph_n ’’ or ‘‘GCH fails at \aleph_n ’’ show up in the theory of a model for standard natural numbers n . But if $M^*[G^*]$ and $M^\dagger[G^\dagger]$ are ω -nonstandard and i is nonstandard this may not be enough. This is where we use the assumption that the models extend a pointwise definable model of $V = L$; it follows that any nonstandard natural number in these models is definable. Thus, the statements show up in their theory, even if n is nonstandard. Thus they cannot be elementarily equivalent, giving us the desired counterexample. \square

Meanwhile, if we go to the projective level, then there will (consistently) be a functorial process. For example, if $V = L$, then in L given any countable oracle code for a structure, we can find the L -least isomorphic copy in a projective way, specifically at the level Δ^1_2 . If we now build $M[G]$ using this copy, it will be constant on the isomorphism class of M , which is a very strong way of respecting isomorphism. Whether we can push this lower down in the projective hierarchy remains open.

¹⁶Given a collection of forcing notions \mathbb{P}_i , their *lottery sum* is the poset which contains the \mathbb{P}_i as suborders, conditions from different \mathbb{P}_i being incomparable, and with a new maximum element placed above each \mathbb{P}_i . A generic for the lottery sum then chooses one of the \mathbb{P}_i by lottery and produces a generic for it. See [Ham00, Section 3] for a precise definition.

Question 31. *Is there an analytic (or co-analytic) functorial method to produce generics for models of set theory?*

REFERENCES

- [AKMS89] Chris Ash, Julia Knight, Mark Manasse, and Theodore Slaman. “Generic copies of countable structures”. *Annals of Pure and Applied Logic* 42.3 (1989), pp. 195–205. ISSN: 0168-0072. DOI: [https://doi.org/10.1016/0168-0072\(89\)90015-8](https://doi.org/10.1016/0168-0072(89)90015-8). <http://www.sciencedirect.com/science/article/pii/S0168007289900158>.
- [BHTU16] Joan Bagaria, Joel David Hamkins, Konstantinos Tsaprounis, and Toshimichi Usuba. “Superstrong and other large cardinals are never Laver indestructible”. *Arch. Math. Logic* 55.1-2 (2016). special volume in memory of R. Laver, pp. 19–35. ISSN: 0933-5846. DOI: 10.1007/s00153-015-0458-3. arXiv:1307.3486[math.LO]. <http://jdh.hamkins.org/superstrong-never-indestructible/>.
- [Cho09] Timothy Y. Chow. “A beginner’s guide to forcing”. In: *Communicating Mathematics*. Vol. 479. Contemporary Mathematics. 2009, pp. 25–40.
- [FHR15] Gunter Fuchs, Joel David Hamkins, and Jonas Reitz. “Set-theoretic geology”. *Annals of Pure and Applied Logic* 166.4 (2015), pp. 464–501. ISSN: 0168-0072. DOI: 10.1016/j.apal.2014.11.004. arXiv:1107.4776[math.LO]. <http://jdh.hamkins.org/set-theoreticgeology>.
- [Fri00] Sy D. Friedman. *Fine Structure and Class Forcing*. Reprint 2011 ed. De Gruyter Series in Logic and Its Applications, 3. De Gruyter, 2000. ISBN: 3110167778,9783110167771.
- [GH17] Michał Tomasz Godziszewski and Joel David Hamkins. “Computable Quotient Presentations of Models of Arithmetic and Set Theory”. In: *Logic, Language, Information, and Computation: 24th International Workshop, WoLLIC 2017, London, UK, July 18–21, 2017, Proceedings*. Ed. by Juliette Kennedy and Ruy J.G.B. de Queiroz. Springer, 2017, pp. 140–152. ISBN: 978-3-662-55386-2. DOI: 10.1007/978-3-662-55386-2.10. arXiv:1702.08350[math.LO]. <http://wp.me/p5M0LV-1tW>.
- [GHHSW] Victoria Gitman, Joel David Hamkins, Peter Holy, Philipp Schlicht, and Kameryn J. Williams. “The exact strength of the class forcing theorem”. *The Journal of Symbolic Logic* (). To appear. arXiv:1707.03700[math.LO].
- [GHK] Victoria Gitman, Joel David Hamkins, and Asaf Karagila. “Kelley-Morse set theory does not prove the class Fodor theorem”. *submitted for publication* (). eprint: 1904.04190.
- [GJ14] Victoria Gitman and Thomas A. Johnstone. “On ground model definability”. In: *Infinity, Computability, and Metamathematics: Festschrift in honour of the 60th birthdays of Peter Koepke and Philip Welch*. Series: Tributes. London, GB: College publications, 2014. eprint: 1311.6789.
- [Ham00] Joel David Hamkins. “The lottery preparation”. *Ann. Pure Appl. Logic* 101.2-3 (2000), pp. 103–146. ISSN: 0168-0072. DOI: 10.1016/S0168-0072(99)00010-X. arXiv:math/9808012[math.LO]. <http://jdh.hamkins.org/lotterypreparation/>.

- [HHKVW19] Miha Habič, Joel David Hamkins, Lukas Daniel Klausner, Jonathan Verner, and Kameryn Williams. “Set-theoretic blockchains”. *Archive for Mathematical Logic* 58.7–8 (2019), pp. 965–997.
- [HKS18] Peter Holy, Regula Krapf, and Philipp Schlicht. “Characterizations of pretameness and the Ord-cc”. *Annals of Pure and Applied Logic* (2018).
- [HS06] Joel David Hamkins and Daniel Seabold. “Well-founded Boolean ultrapowers as large cardinal embeddings” (2006), pp. 1–40. arXiv:1206.6075[math.LO]. <http://jdh.hamkins.org/boolean-ultrapowers/>.
- [HTMM18] Matthew Harrison-Trainor, Russell Miller, and Antonio Montalbán. “Borel functors and infinitary interpretations”. *The Journal of Symbolic Logic* 83.4 (2018), 14341456. DOI: 10.1017/jsl.2017.81.
- [HTMMM17] Matthew Harrison-Trainor, Alexander Melnikov, Russell Miller, and Antonio Montalbán. “Computable functors and effective interpretability”. *The Journal of Symbolic Logic* 82.1 (2017), 7797. DOI: 10.1017/jsl.2016.12.
- [Kun80] Kenneth Kunen. *Set theory. An introduction to independence proofs*. Vol. 102. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, 1980.
- [MM75] W. Marek and A. Mostowski. “On extendability of models of ZF set theory to the models of Kelley-Morse theory of classes”. In: *ISILC Logic Conference: Proceedings of the International Summer Institute and Logic Colloquium, Kiel 1974*. Ed. by Gert H. Müller, Arnold Oberschelp, and Klaus Potthoff. Berlin, Heidelberg: Springer Berlin Heidelberg, 1975, pp. 460–542. DOI: 10.1007/BFb0079429.
- [Mon14] Antonio Montalbán. “Computability theoretic classifications for classes of structures”. *Proceedings of ICM 2014* 2 (2014), pp. 79–101.
- [Mos76] Andrzej Mostowski. “A Remark on Models of the Gdel-Bernays Axioms for Set Theory”. In: *Sets and Classes on The Work by Paul Bernays*. Ed. by Gert H. Müller. Vol. 84. Studies in Logic and the Foundations of Mathematics. Elsevier, 1976, pp. 325–340.
- [MPSS18] Russell Miller, Bjorn Poonen, Hans Schoutens, and Alexandra Shlapentokh. “A computable functor from graphs to fields”. *The Journal of Symbolic Logic* 83.1 (2018), 326348. DOI: 10.1017/jsl.2017.50.
- [Rat79] Zygmunt Ratajczyk. “On sentences provable in impredicative extensions of theories”. *Dissertationes Mathematicae* 178 (1979). 40 pp.
- [Sho71] J. R. Shoenfield. “Unramified Forcing”. *Proceedings of Symposia in Pure Mathematics*. Axiomatic set theory 1.13 (1971). Ed. by D. S. Scott, pp. 383–395.
- [Usu17] Toshimichi Usuba. “The downward directed grounds hypothesis and very large cardinals”. *Journal of Mathematical Logic* 17.02 (2017), p. 1750009.
- [Usu19] Toshimichi Usuba. “Choiceless Löwenheim–Skolem property and uniform definability of grounds” (2019). manuscript. arXiv:1904.00895[math.LO].
- [VV19] Giorgio Venturi and Matteo Viale. “The model companions of set theory” (2019). submitted for publication. arXiv:1909.13372[math.LO].

- [Wil19] Kameryn J. Williams. “Minimum models of second-order set theories”. *Journal of Symbolic Logic* 8.2 (2019), pp. 589–620. DOI: 10.1017/jsl.2019.27.

(Joel David Hamkins) PROFESSOR OF LOGIC, FACULTY OF PHILOSOPHY, UNIVERSITY OF OXFORD & MATHEMATICS INSTITUTE, UNIVERSITY OF OXFORD & SIR PETER STRAWSON FELLOW, UNIVERSITY COLLEGE, HIGH STREET, OXFORD OX1 4BH, UNITED KINGDOM

E-mail address: joeldavid.hamkins@philosophy.ox.ac.uk

URL: <http://jdh.hamkins.org>

(Russell Miller) THE GRADUATE CENTER OF CUNY, PH.D. PROGRAMS IN MATHEMATICS & COMPUTER SCIENCE, 365 FIFTH AVENUE, NEW YORK, NY 10016, USA & QUEENS COLLEGE OF CUNY, MATHEMATICS DEPT., 65-30 KISSENA BLVD., FLUSHING, NY 11367, USA

E-mail address: Russell.Miller@qc.cuny.edu

URL: <http://qcpages.qc.cuny.edu/~rmiller>

(Kameryn J. Williams) UNIVERSITY OF HAWAII AT MĀNOA, DEPARTMENT OF MATHEMATICS, 2565 MCCARTHY MALL, KELLER 401A, HONOLULU, HI 96822, USA

E-mail address: kamerynw@hawaii.edu

URL: <http://kamerynjw.net>