A NEW GAME METATHEOREM FOR ASH-KNIGHT STYLE PRIORITY CONSTRUCTIONS

ANTONIO MONTALBÁN

ABSTRACT. We develop a new metatheorem for $0^{(\eta)}$-priority constructions for transfinite ordinals η. Our metatheorem simplifies previous metatheorems by Ash, Ash–Knight and the author. A drawback of this simplification is that its applications are more limited: It can only be used when the object being built is an ω-presentation of a structure. However, in practical terms, this is not a huge limitation and the majority of proofs in the literature that use the Ash–Knight metatheorem can be done using our metatheorem.

Often in computability theory we need to build a computable object using non-computable information — sometimes even non-computable information about the very same object we are building. The main tool for such constructions is the priority method, which has become increasingly more involved and sophisticated since the 1950s. Priority arguments are classified in terms of how much non-computable information is needed to verify the construction. The most common priority constructions are the finite-injury ones [Fri57, Muc56]. They are used when the information needed is $0'$-computable. Infinite-injury priority constructions [Sho61, Sac63] are used when $0''$-computable guesses are needed. There are various zero-triple priority constructions [Lac76] in the literature, but they are very complicated, and far less common. Beyond that point, humans cannot keep track of the combinatorics anymore. Well, that is unless the level-by-level combinatorics of the proof is uniform and one can describe the work done at all the levels by a single procedure. There have been various proposals for general $0^{(n)}$-injury constructions: Harrington's worker's method [Har76], Lempp and Lerman's trees of strategies [LL95, Ler10], Ash's [Ash86] and Ash–Knight's [AK00] η-systems, and Montalbán's systems of true stages [Mon14]. Among these methods, the Ash–Knight's η-system is the only one that contains a clear metatheorem, where if a certain combinatorial machinery can be put in place, one can then apply the metatheorem as a black box and produce the desired computable object out of Δ^0_η-information. The proof of the metatheorem is complicated. But the whole point of the metatheorem is that one does not need to know its proof to use it. A disadvantage for the metatheorem is that it is applicable in a limited number of situations. It has only been used in computable structure theory, where the combinatorial features needed for the metatheorem often occur naturally. Let me emphasize, though, that despite its limitations, it has been extremely useful in computable structure theory, and has been applied to a wide range of situations.

In this paper we propose a new metatheorem. The advantage of the new metatheorem is that it is much easier to use than the Ash–Knight's metatheorem. The disadvantage is that it is more limited. However, in most of the constructions from the computable-structure-theory literature where Ash–Knight's metatheorem is useful, our metatheorem is too. So the restriction in applicability does not seem to be too limiting in practice, while the simplification

Saved: December, 2019
Compiled: December 9, 2019
The author was partially supported by NSF grant DMS-1363310.

is considerable. At a meta-level, we believe that our metatheorem exhibits the interplay between the η-back-and-forth relations and Δ^0_η-information in the clearest possible way.

Montalbán's η-true stages method provides a different angle to view Ash's construction, giving it a more hands-on feeling, while adding some flexibility. For instance, Montalbán used his method in [Mon14] to prove the *tree of structures theorem*, a version of the pair-of-structures theorem which cannot be proved with Ash and Knight's system. It was also used by Greenberg and Turetsky [GT], and Day and Marks (in preparation).

The metatheorem we introduce here is not more but less flexible: it can only be used to build ω-presentations of structures. There are many proofs in computable structure theory that require exactly this: to build a ω-presentations with a particular property. After presenting the metatheorem in Section 2, we showcase its applicability by present proofs of various results from the literature using our metatheorem. We claim that these proofs are much simpler than the original ones, though we will let the reader be the judge on that. 'Simpler' might not be the right word, as what it actually does is to black box more the computational part of the construction, leaving only the key combinatorial steps that are pertinent to the situation. The proofs we will go through are: Ash and Knight's pair of structures theorem (Section 3); Watnick's and Ash-Jockusch-Knight's theorem that a linear ordering $\mathcal{L}$ has a $\Delta^0_{2\alpha+1}$ copy if and only if $\omega^\alpha \cdot \mathcal{L}$ has a computable copy (Section 4); and Ash's theorem that, under enough effectiveness conditions, Δ^0_α-categoricity implies relative Δ^0_α-categoricity (Section 5). We will also discuss how the copy-vs-diagonalizer constructions introduced by the author in [Mon13] to prove the low$_n$ property can be viewed as a particular case of the metatheorem (Section 6).

We leave the proof of the metatheorem to the end of the paper. We use Section 7 to reveiew η-true stage systems, and then use them to prove the metatheorem in Section 8.

1. Background and notation

Before stating the metatheorem in the next section, we need to lay down some basic background and notation.

1.1. **Diagrams.** Throughout this article we assume we have a computable vocabulary τ and all our structures are τ-structures. For simplicity, and without loss of generality, we may assume τ is relational. An ω-presentation is just a copy of a structure with domain ω. Given an ω-presentation $\mathcal{A}$, we use $D(\mathcal{A}) \in 2^{\mathbb{N}}$ to denote its atomic diagram, where $D(\mathcal{A})(i)$ gives the truth value of ith atomic formula. That is, $D(\mathcal{A})(i) = 1$ if and only if $\mathcal{A} \models \varphi^{at}_i[x_j \mapsto j]$, where $\{\varphi^{at}_i : i \in \mathbb{N}\}$ is some effective enumeration of the atomic formulas on the variables $x_0, x_1, \ldots$. Given a finite tuple $\bar{a} = \langle a_0, \ldots, a_{|\bar{a}|-1}\rangle \in \mathcal{A}^{<\mathbb{N}}$, we define the *atomic diagram of $\bar{a}$ in $\mathcal{A}$* to be the finite binary string $D_\mathcal{A}(\bar{a})$ of length $\ell_{|\bar{a}|}$, where, for $i < \ell_{|\bar{a}|}$,

$$D_\mathcal{A}(\bar{a})(i) = 1 \iff \mathcal{A} \models \varphi^{at}_i[x_j \mapsto a_j : j < |\bar{a}|],$$

and where ℓ_s is the number of atomic formulas using the variables $x_0, \ldots, x_{s-1}$ and the first s symbols from τ. We assume that in our enumeration $\{\varphi^{at}_i : i \in \mathbb{N}\}$ of the atomic formulas, those atomic formulas come first. It follows that $D(\mathcal{A}) = \bigcup_{s\in\mathbb{N}} D_\mathcal{A}(\langle 0, \ldots, s-1\rangle)$, where the union is a union of strings. Also, if $g \colon \omega \to \mathcal{A}$ is a bijection and $\mathcal{B}$ is the pull-back of $\mathcal{A}$ through g (i.e., g is an isomorphism $\mathcal{B} \to \mathcal{A}$), then $D(\mathcal{B}) = \bigcup_{s\in\mathbb{N}} D_\mathcal{A}(g \restriction s)$.

1.2. **Back-and-forth relations.** The back-and-forth relations measure how hard it is to distinguish two structures, or two tuples within a structure, or even two tuples from different structures. The idea is that two tuples are n-back-and-forth equivalent if we cannot differentiate them using only n Turing jumps. Basic model-theoretic information about these relations may be found in [Bar73] and Karp [Kar65], and computability-theoretic information in the work of Ash and Knight [AK00].

Definition 1.1. Given structures $\mathcal{A}$ and $\mathcal{B}$, tuples $\bar{a} \in A^{<\mathbb{N}}$, $\bar{b} \in B^{<\mathbb{N}}$, and an ordinal $\eta > 0$, we say that $(\mathcal{A}, \bar{a})$ is η-back-and-forth below $(\mathcal{B}, \bar{b})$, and write $(\mathcal{A}, \bar{a}) \leq_\eta (\mathcal{B}, \bar{b})$, if every $\Pi^{\mathtt{in}}_\eta$ formula true about $\bar{a}$ in $\mathcal{A}$ is also true about $\bar{b}$ in $\mathcal{B}$. I.e.,

$$(\mathcal{A}, \bar{a}) \leq_\eta (\mathcal{B}, \bar{b}) \iff \Pi^{\mathtt{in}}_\eta\text{-}tp_{\mathcal{A}}(\bar{a}) \subseteq \Pi^{\mathtt{in}}_\eta\text{-}tp_{\mathcal{B}}(\bar{b}),$$

Here, $\Pi^{\mathtt{in}}_\eta$ denotes de class of infinitary Π_η formulas, and $\Pi^{\mathtt{in}}_\eta\text{-}tp_{\mathcal{A}}(\bar{a})$ denotes the $\Pi^{\mathtt{in}}_\eta$-type of $\bar{a}$ in $\mathcal{A}$. See [AK00, Chapter 6] and [MonP2] for more background on infinitary formulas.

For $\eta = 0$ we only look at finite segment of the quantifier-free types and we let $(\mathcal{A}, \bar{a}) \leq_0 (\mathcal{B}, \bar{b})$ if $D_{\mathcal{A}}(\bar{a}) \subseteq D_{\mathcal{B}}(\bar{b})$.

An equivalent definition for the back-and-forth relations is given in the following theorem.

Theorem 1.2 (Karp [Kar65])**.** *Consider structures $\mathcal{A}$ and $\mathcal{B}$, tuples $\bar{a} \in A^{<\mathbb{N}}$, $\bar{b} \in B^{<\mathbb{N}}$, and an ordinal $\eta > 0$. Then $(\mathcal{A}, \bar{a}) \leq_\eta (\mathcal{B}, \bar{b})$ if and only if, for every $\beta < \eta$ and every $\bar{d} \in B^{<\mathbb{N}}$, there exists $\bar{c} \in \mathcal{A}^{<\mathbb{N}}$ such that*

$$(\mathcal{A}, \bar{a}\bar{c}) \geq_\beta (\mathcal{B}, \bar{b}\bar{d}).$$

See[AK00, 15.1, 18.6] for a proof.

Given a list of computable ω-presentations $\mathbb{A} = (\mathcal{A}_i : i \in \mathbb{N})$ and a computable ordinal η, we say that the *back-and-forth relations in $\mathbb{A}$ are computable up to η* if the relations $\leq_\beta$ for $\beta \leq \eta$ are uniformly computable, that is, if the set of five-tuples

$$\{\langle \beta, i, \bar{a}, j, \bar{b} \rangle : \beta \in \eta + 1, (\mathcal{A}_i, \bar{a}) \leq_\beta (\mathcal{A}_j, \bar{b})\}$$

is computable.

This is essentially Ash and Knight's notion of η-friendliness, except that they required the set above to be only c.e.

Note that this notion is meaningful even when $\mathbb{A}$ consists of a single ω-presentation. That ω-presentation has to be extremely nice to allow us to compute all the back-and-forth relations among its tuples. When we understand a structure well, we expect to be able to understand its back-and-forth relations too. For example, every computable ordinal has an ω-presentation that is η-friendly. Furthermore, for every computable ordinal α, if we represent ω^α in the standard way using Cantor normal forms, we get an α-friendly ω-presentation of ω^α. A calculation of the back-and-forth relations on ordinals was done by Ash in [Ash86], see [AK00, Lemma 15.10]. The same applies if we take linear orderings of the form $\mathbb{Z}^\alpha$, where α is a computable ordinal. Ash also calculated the back-and-forth relations on super-atomic Boolean algebras in [Ash87, Lemma 5].

In [Mon10, Proposition 2.10], the author shows that if a class of structures is Σ_α-small, meaning that there are only countably many $\equiv_\alpha$-equivalence classes among the tuples within its structures, and it is Σ_α-small in an effective way, meaning that we can effectively understand how the back-and-forth relations work among these $\equiv_\alpha$-equivalence classes, then we can produce a computable list of structures in the class containing representatives of all $\equiv_\alpha$-equivalence classes, and where the back-and-forth relations are computable up to α.

Another point to make is that for every countable family of structures there is an oracle, namely the 2ηth jump of the list of ω-presentations, relative to which the back-and-forth relations up to η are computable. For many of the results below, we still get a meaningful result if we relativize to this oracle to be able to assume the back-and-forth relations up to η are computable.

1.3. $\Delta^0_{\eta+1}(X)$ questions.

We will deal with finite and transfinite iterations of the Turing jumps. The standard notation for these iterates has been set, unfortunately, so that for finite n, $0^{(n)}$ is Σ^0_n-complete, while for infinite α, it is $0^{(\alpha+1)}$ who is Σ_α-complete. One can encapsulate both statements by saying that $0^{(\eta+1)}$ is $\Sigma^0_{1+\eta}$ complete for all η — though this is rather cumbersome. Instead, in this paper we avoid using the $0^{(\alpha)}$ notation, and we only talk about Σ^0_η complete sets or of Δ^0_γ Turing complete sets.

Definition 1.3. Given $X \in 2^{\mathbb{N}}$, $n, e \in \mathbb{N}$ and a computable ordinal η, we say that n is *the answer to the eth $\Delta^0_{\eta+1}(X)$ question* if

$$n = \Phi_e^{S^X_\eta}(0),$$

where S^X_η is a $\Sigma^0_\eta(X)$ complete set, namely $X^{(\eta)}$ when η is finite and $X^{(\eta+1)}$ when η is infinite, and where Φ_e is the eth Turing functional.

Let us remark that any finite number of questions of the form $\varphi_{e_0}^{S^X_\eta}(k_0), \varphi_{e_1}^{S^X_\eta}(k_1), ..., \varphi_{e_\ell}^{S^X_\eta}(e_k)$ can be encoded into a single question using an index e such that $\varphi_e^{S^X_\eta}(0)$ outputs a number encoding the tuple $\langle \varphi_{e_0}^{S^X_\eta}(k_0), \varphi_{e_1}^{S^X_\eta}(k_1), ..., \varphi_{e_\ell}^{S^X_\eta}(k_\ell)\rangle$.

2. Game constructions

Let η be a computable ω-presentation of an ordinal. The reader may assume $\eta = 1$ on a first read of this section — the case $\eta = 1$ is already interesting, and the metatheorem is still quite useful and meaningful. Suppose we have a list of structures

$$\mathbb{A} = \{\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2, ...\},$$

where the back-and-forth relations are computable up to η. Let $\mathfrak{P}$ be a property of ω-presentations of structures, or equivalently, let $\mathfrak{P}$ be a subset of the set Mod_τ of all τ-ω-presentations.

We will now describe a type of construction that we will call an *η-$\mathbb{A}$-game*. This game involves three characters, the *engineer*, the *extender*, and the *oracle*. Together, when the game ends, they build an ω-presentation $\mathcal{L}$, which we call the *limit structure*. The goal of the engineer is for the limit structure $\mathcal{L}$ to satisfy property $\mathfrak{P}$. The extender is in charge of making $\mathcal{L}$ computable — he will not, in any way, coordinate his work with the engineer. The job of the oracle is to answer $\Delta^0_{\eta+1}(\mathcal{L})$ questions posed by the engineer. The game has infinitely many stages. At each stage $j \in \mathbb{N}$, first, the engineer plays a triple $\langle i_j, \bar{a}_j, e_j\rangle$ where $i_j, e_j \in \mathbb{N}$ and $\bar{a}_j \in \mathcal{A}_{i_j}^{<\mathbb{N}}$, and second, the extender plays a tuple $\bar{b}_j \in A_{i_j}^{<\mathbb{N}}$ extending $\bar{a}_j$ and the oracle plays a number n_j which must be the answer to the e_jth $\Delta^0_{\eta+1}(D(\mathcal{L}))$ question.

engineer	$i_0, \bar{a}_0, e_0$		$i_1, \bar{a}_1, e_1$		$i_2, \bar{a}_2, e_2$		$\cdots$
extender		b_0		b_1		b_2	$\cdots$
oracle		n_0		n_1		n_2	$\cdots$

At each stage $j > 0$, the tuple $\bar{a}_j$ played by the engineer must satisfy:

$$(\mathcal{A}_{i_j}, \bar{a}_j) \geq_\eta (\mathcal{A}_{i_{j-1}}, \bar{b}_{j-1}).$$

The tuple $\bar{b}_j$ played by the extender must be in the same structure just played by the engineer and must satisfy:

$$\bar{b}_j \supseteq \bar{a}_j.$$

After ω many moves we get

$$D_{\mathcal{A}_0}(\bar{a}_0) \subseteq D_{\mathcal{A}_0}(\bar{b}_0) \subseteq D_{\mathcal{A}_1}(\bar{a}_1) \subseteq D_{\mathcal{A}_1}(\bar{b}_1) \subseteq D_{\mathcal{A}_2}(\bar{a}_2) \subseteq \cdots$$

and hence we get a limit ω-presentation $\mathcal{L}$ whose atomic diagram is the union of the diagrams of the tuples played:

$$D(\mathcal{L}) = \bigcup_{j\in\mathbb{N}} D_{\mathcal{A}_{i_j}}(\bar{a}_j).$$

The numbers e_j represent $\Delta^0_{\eta+1}(\mathcal{L})$ questions as in Definition 1.3.[*] The engineer must ensure that all these questions converge. The numbers n_j played by the oracle must be the answers to these questions.

A *strategy* for the engineer is a function that tells the engineer what to play next given the previous moves by the extender and the oracle. We say that a strategy is *valid* if, on all possible plays by the extender, all the $\Delta^0_{\eta+1}(\mathcal{L})$ questions e_j converge.

Theorem 2.1. *Given $\mathbb{A}$ and η as described above, for every computable valid strategy for the engineer in the η-$\mathbb{A}$-game, there is a $\Delta^0_{\alpha+1}$ sequence of moves by the extender so that, if the engineer follows her strategy, the limit ω-presentation $\mathcal{L}$ is computable.*

Furthermore, we will prove that there is a uniform effective procedure that, given the strategy for the engineer, produces the ω-presentation $\mathcal{L}$ which comes from some sequence of moves by the extender.

To apply this theorem, one needs to describe a computable valid strategy for the engineer that, with the help of the oracle who is answering our $\Delta^0_{\eta+1}$ questions, will build an ω-presentation with property $\mathfrak{P}$ independently of what the extender does. One can then cite the theorem to conclude that, even if the construction relied on the $\Delta^0_{\eta+1}$ information provided by the oracle, the resulting ω-presentation is computable.

3. Pairs of structures

The pair of structures theorem of Ash and Knight [AK90] is one of the most useful applications of Ash and Knight's metatheorem. It gives a complete answer to the question of when are two structures hard to distinguish in a Σ^0_α way — under enough effectiveness conditions. Two of its main applications are the construction of the αth jump inversion [GHK$^+$05] and of a structure whose degree spectrum is exactly the non-hyperarithmetic degrees [GMS13].

Theorem 3.1. *Let η be a computable ordinal and let $\mathcal{A}_0$ and $\mathcal{A}_1$ be a pair of ω-presentations with their back-and-forth relations computable up to η.*

If $\mathcal{A}_0 \geq_{\eta+1} \mathcal{A}_1$, then, for every $\Sigma^0_{\eta+1}$ set $S \subseteq \omega$, there is a computable list $\{\mathcal{C}_i : i \in \omega\}$ of computable structures such that

$$\mathcal{C}_i \cong \begin{cases} \mathcal{A}_0 & \text{if } i \notin S \\ A_1 & \text{if } i \in S. \end{cases}$$

Proof. Let S be a $\Sigma^0_{\eta+1}$ subset of $\mathbb{N}$. Uniformly computably in $i \in \mathbb{N}$, we need to define a strategy for the engineer to build a structure that is isomorphic to $\mathcal{A}_1$ if $i \in S$, and to $\mathcal{A}_0$ if $i \notin S$. Theorem 2.1 will then guarantee that, also uniformly computably in i, we can choose a $\Delta^0_{\eta+1}$ sequence of moves by the extender so that the limit structure $\mathcal{C}_i$ is uniformly computably in i. The strategy must be computable, but the engineer is allowed to ask $\Delta^0_{\eta+1}$ questions along the way, and it will be the extender who, through Theorem 2.1, will guarantee that $\mathcal{C}_i$ is computable.

The question of whether $i \in S$ or not is not a $\Delta^0_{\eta+1}$ question, so we cannot ask the oracle directly about it. Instead, we use a computable list of indices e_0, e_1,... of $\Delta^0_{\eta+1}$ questions

[*]From now on we write $\Delta^0_{\eta+1}(\mathcal{L})$ instead of $\Delta^0_{\eta+1}(D(\mathcal{L}))$.

whose answers n_0, n_1, ... are either all zeros if $i \notin S$ or start with zeros and then change to all ones if $i \in S$.[†]

Fix $i \in \mathbb{N}$. We define a strategy for the engineer for the η-$\mathbb{A}$-game where $\mathbb{A} = \{\mathcal{A}_0, \mathcal{A}_1\}$ as follows: In its first move, play the empty tuple in $\mathcal{A}_0$, and ask about e_0 — i.e., play the triple $\langle 0, \langle\rangle, e_0\rangle$. In the $j+1$st move, if $n_{j+1} = n_j$, then play any tuple $\bar{a}_{j+1}$ in $\mathcal{A}_{n_j}$ of length at least j extending $\bar{b}_j$, and ask about e_{j+1} — i.e., play the triple $\langle n_{j+1}, \bar{a}_{j+1}, e_{j+1}\rangle$. If $n_{j+1} \neq n_j$, we must have $n_j = 0$ and $n_{j+1} = 1$. In this case, play a tuple $\bar{a}_{j+1} \in \mathcal{A}_1^{<\mathbb{N}}$ such that $(\mathcal{A}_1, \bar{a}_{j+1}) \geq_\eta (\mathcal{A}_0, \bar{b}_j)$. The existence of such $\bar{a}_{j+1}$ follows from the hypothesis that $\mathcal{A}_0 \geq_{\eta+1} \mathcal{A}_1$.

At the end of the game we get that, if $i \notin S$, then $\{\bar{a}_j : j \in \mathbb{N}\}$ is an increasing sequence of tuples in $\mathcal{A}_0$, and hence the limit structure is isomorphic to $\mathcal{A}_0$. If $i \in S$ and s_0 is the first stage with $n_{s_0} = 1$, then $\{\bar{a}_j : j \in \mathbb{N}, j \geq s_0\}$ is an increasing sequence of tuples in $\mathcal{A}_1$, and hence the limit structure is isomorphic to $\mathcal{A}_1$. □

The theorem above is only the successor-ordinal version of the Ash–Knight pairs-of-structures theorem. The way we developed our metatheorem in this paper only allows us to prove this case. To prove the limit-ordinal case, we would need to develop $<\eta$-$\mathbb{A}$-game constructions.

4. Linear ordering presentations

Here is another classical application of Ash's metatheorem. The theorem below was proved by Watnick [Wat84] for the case $\eta = 1$ and then extended to all η by Ash, Jockusch, and Knight [AJK90] using workers,and by Ash alone [Ash91] using 2η-systems.

Theorem 4.1. *Let $\mathcal{A}$ be a linear ordering with a first element. Then $\mathcal{A}$ has a $\Delta^0_{2\eta+1}$ copy if and only if $\omega^\eta \cdot \mathcal{A}$ has a computable copy.*

Proof. Assume that the least element of $\mathcal{A}$ is the 0 of its ω-presentation.

The pool $\mathbb{A}$ of structures that we use for our game will consists of all the structures of the form $\omega^\eta \cdot \mathcal{F}$, where $\mathcal{F}$ is a finite linear ordering whose domain is an initial segment of $\mathbb{N}$. The back-and-forth relations between these structures are computable up to $2\eta + 1$. Precise calculations of the back-and-forth relations among ordinals are calculated in [AK00, Lemma 15.10]. These calculations are based on two main lemmas: One, that a tuple is β-back-and-forth below another if the ordering of the elements within the tuple are the same, and the intervals in between the elements of the first tuple are β-back-and-forth below the corresponding intervals for the other tuple [AK00, Lemma15.8]; and, two, that for linear orderings $\mathcal{B}$ and $\mathcal{C}$, $\omega^\beta \cdot \mathcal{B} \leq_{2\beta+1} \omega^\beta \cdot \mathcal{C}$ if and only if $|\mathcal{B}| \geq |\mathcal{A}|$.

Another observation we need is that if $\mathcal{F}_0 \subseteq \mathcal{F}_1$ are linear orderings with the same first element 0, then $\omega^\eta \cdot \mathcal{F}_0$ is a $\Sigma^{\mathtt{in}}_{2\eta+1}$-elementary substructure of $\omega^\eta \cdot \mathcal{F}_1$, meaning that, for every tuple $\bar{b} \in \omega^\eta \cdot \mathcal{F}_0$, we have that $(\omega^\eta \cdot \mathcal{F}_0, \bar{b}) \geq_{2\eta+1} (\omega^\eta \cdot \mathcal{F}_1, \bar{b})$. The reason is that if an interval (b_i, b_j) of $\omega^\eta \cdot \mathcal{F}_0$ changes in $\omega^\eta \cdot \mathcal{F}_1$, it is because we add a few intervals of the form ω^η in between, that is, it changes from $\omega^\eta \cdot k_0 + \beta$ to $\omega^\eta \cdot k_1 + \beta$ for $0 < k_0 < k_1 \in \mathbb{N}$ and $\beta < \omega^\eta$, and we know from the lemma mentioned above, that $\omega^\eta \cdot k_1 \leq_{2\eta+1} \omega^\eta \cdot k_0$.

We describe a computable strategy for the engineer in the 2η-$\mathbb{A}$-game. At stage $j-1$, the engineer asks the oracle for a full description of $\mathcal{A} \restriction j$, i.e., the ordering $\mathcal{A}$ on the first j natural numbers. At the following stage, stage j, she choses the structure $\omega^\eta \cdot \mathcal{F}_j$ in $\mathbb{A}$ where $\mathcal{F}_j = \mathcal{A} \restriction j$. Note that $\mathcal{F}_j$ naturally extends $\mathcal{F}_{j-1}$. By our observation above that $\omega^\eta \cdot \mathcal{F}_{j-1}$ is a $\Sigma^{\mathtt{in}}_{2\eta+1}$-elementary substructure of $\omega^\eta \cdot \mathcal{F}_j$, we know that $(\omega^\eta \cdot \mathcal{F}_{j-1}, \bar{b}_{j-1}) \leq_\eta (\omega^\eta \cdot \mathcal{F}_j, \bar{b}_{j-1})$, and hence that the engineer can play any tuple $\bar{a}_j$ extending $\bar{b}_{j-1}$. All she needs to do is make

[†]Let W be a c.e. operator such that $i \in S \iff i \in W^{S_\eta}$ where S_η is a Σ^0_η-complete set. Then let e_j be an index so that $\Phi^{S_\eta}_{e_j}(0) = 1$ if i is enumerated in W^{S_η} in less than j steps.

sure that she ends up including all members of $\omega^\eta \cdot \mathcal{A}$ eventually. The limit structure will then be isomorphic to $\omega^\eta \cdot \mathcal{A}$. □

The theorem is still true if $\mathcal{A}$ has no least element. In that case, instead of using the finite linear ordering $\mathcal{A}\restriction j$ to approximate $\mathcal{A}$, we need to use ω^* as follows. One needs to show that there is an $\mathcal{A}$-computable sequence of computable embeddings $f_j : \omega^* \to \omega^*$ whose direct limit is $\mathcal{A}$. For this, consider the isomorphism between ω^* and $\omega^* + \mathcal{A}\restriction j$, and embed $\omega^* + \mathcal{A}\restriction j$ into $\omega^* + \mathcal{A}\restriction j+1$ using the inclusion of $\mathcal{A}_j \subseteq \mathcal{A}_{j+1}$ and use the ω^* part for elements of $\mathcal{A}_{j+1}$ that are below the least element of $\mathcal{A}_j$.

A similar proof would show that $\mathcal{A}$ has a $\Delta^0_{2\eta+1}$ copy if and only if $\mathbb{Z}^\eta \cdot \mathcal{A}$ has a computable copy.

5. Δ^0_η-categoricity

A computable structure $\mathcal{A}$ is *Δ^0_α-categorical* if, for every computable copy $\mathcal{B}$ of $\mathcal{A}$, there is a Δ^0_α isomorphism between $\mathcal{A}$ and $\mathcal{B}$. Downey, Kach, Lempp, Lewis, Montalbán, and Turetsky [DKL+15] proved that this property cannot be characterized structurally. However, a variant of it, namely the on-a-cone version, can [AKMS89]. Given $X \in 2^{\mathbb{N}}$, $\mathcal{A}$ is *$\Delta^0_\alpha(X)$-categorical* if, for every X-computable copy $\mathcal{B}$ of $\mathcal{A}$, there is a $\Delta^0_\alpha(X)$ isomorphism between $\mathcal{A}$ and $\mathcal{B}$. $\mathcal{A}$ is *relatively Δ^0_α-categorical* if it is $\Delta^0_\alpha(X)$-categorical for all X. $\mathcal{A}$ is *Δ^0_α-categorical on a cone* if there is a $C \in 2^{\mathbb{N}}$ such that $\mathcal{A}$ is $\Delta^0_\alpha(X)$-categorical for all X for all $X \geq_T C$. There are several measures for different aspects of the complexity of a structure that turned out to be equivalent to $\boldsymbol{\Delta}^0_\alpha$-categoricity on a cone, as for example, the structure having a $\Sigma^{\mathtt{in}}_{\alpha+2}$ Scott sentence [Mon15]. Because of this robustness, we have proposed that the rank of $\mathcal{A}$ should be defined to be the least α such that $\mathcal{A}$ is $\boldsymbol{\Delta}^0_\alpha$-categorical on a cone.

Unfortunately, the three notions of plain, relative, and on-a-cone Δ^0_α-categoricity are not equivalent. Examples of this non-equivalence were build by Goncharov, Harizanov, Knight, McCoy, R. Miller, and Solomon [GHK+05]. However, they are equivalent for most natural structures one encounters. Ash and Knight proved that these notions are equivalent if we have enough structural information about $\mathcal{A}$. To understand that result, we need to consider the notion of α-freeness:

Definition 5.1 (Ash [Ash87], [AK00, Section 17.4]). We say that the $\Pi^{\mathtt{in}}_\alpha$-type of a tuple $\bar{a} \in \mathcal{A}^{<\mathbb{N}}$ is *$\Sigma^{\mathtt{in}}_\alpha$-supported* if there is a $\Sigma^{\mathtt{in}}_\alpha$-formula $\varphi(\bar{x})$ true of $\bar{a}$ which implies all $\Pi^{\mathtt{in}}_\alpha$ formulas $\psi(\bar{x})$ true about $\bar{a}$. I.e.,

$$\mathcal{A} \models \varphi(\bar{a}) \quad \& \; \forall \bar{x}\big(\varphi(\bar{x}) \to \bigwedge\!\!\!\bigwedge_{\psi \in \Pi^{\mathtt{in}}_\alpha\text{-}tp_{\mathcal{A}}(\bar{a})} \psi(\bar{x})\big).$$

If the $\Pi^{\mathtt{in}}_\alpha$-type of $\bar{a}$ is not supported by a $\Sigma^{\mathtt{in}}_\alpha$ formula, we say that $\bar{a}$ is *α-free*. We say that $\bar{a}$ is *α-free over $\bar{p}$* if it is α-free in the structure $(\mathcal{A}, \bar{p})$.

Using ideas from [AKMS89, AK00], one can show that a structure $\mathcal{A}$ is $\boldsymbol{\Delta}^0_\alpha$-categorical on a cone if and only if there is a tuple $\bar{p} \in \mathcal{A}^{<\mathbb{N}}$ such that every tuple $\bar{a} \in \mathcal{A}^{<\mathbb{N}}$ is α-free over $\bar{p}$ (see [Mon15]). It can be shown using [Mon10, Lemma 2.2] that a tuple $\bar{a}$ is $(\eta+1)$-free if and only if, for every $\bar{b} \supseteq \bar{a}$, there exist tuples $\bar{a}' \subseteq \bar{b}'$ such that

$$\bar{b} \leq_\eta \bar{b}' \quad \text{but} \quad \bar{a} \not\leq_{\eta+1} \bar{a}'.$$

In practice, when we understand the back-and-forth relations on a structure very well we can effectively decide which tuples are α-free over which tuples, and we can effectively find witness for the tuples that are not α-free. When that is the case we say that *α-freeness is computable* in $\mathcal{A}$. Under this assumptions, and the computability of the back-and-forth relations, is that Ash [Ash87] proved that plain, relative, and on-a-cone Δ^0_η categoricity coincide. That proof has a structural part and a computably-theoretic part. The computably-theoretic part can be captured in the next lemma, which we state only for the successor case.

Lemma 5.2. *Let $\mathcal{A}$ be a computable ω-presentation where the back-and-forth relations up to $\eta+1$ and $\eta+1$-freeness are computable. If over every tuple $\bar{p}$ there is a tuple that is $\eta+1$-free, then $\mathcal{A}$ is not $\Delta^0_{\eta+1}$-categorical.*

This is essentially the η-$\mathcal{A}$-game version of the proof that computable categoricity implies relative computable categoricity for 2-decidable structures (i.e., the case $\eta = 0$ due to Goncharov following ideas of Nurtazin; see for instance [MonP1, Section VIII.4]). We recommend the reader to read that proof before reading the following argument.

Proof. We build $\mathcal{A}$ by defining a computable strategy for the engineer in an η-$\mathcal{A}$-game construction (i.e. $\mathbb{A} = \{\mathcal{A}\}$). We want to end up defining a copy of $\mathcal{A}$, so we will make sure that the tuples $\bar{a}_j$ stabilize in the limit. i.e., that for each $n \in \mathbb{N}$, $\lim_{j\to\infty} \bar{a}_j(n)$ exists — call it $g(n)$. We will then end up with a function $g\colon \omega \to \mathcal{A}$ and we will get that the limit structure $\mathcal{L}$ is exactly the pull-back of $\mathcal{A}$ through g (see [MonP1, Section I.1.7]). The objective is to build $\mathcal{L}$ so that it is not $\Delta^0_{\eta+1}$ isomorphic to $\mathcal{A}$.

We use a finite-injury priority construction with infinitely many requirements R_e for $e \in \mathbb{N}$. Requirement R_e ensures $\Phi_e^{S^0_\eta}$ is not an isomorphism from $\mathcal{L}$ to $\mathcal{A}$, where S^0_η is $0^{(\eta)}$ if η is finite and is $0^{(\eta+1)}$ if η is infinite, or in other words, S^0_η is a Σ^0_η-complete set. In practical terms, R_e will try to guarantee that $\Phi_e^{S^0_\eta} \circ g^{-1}$ is not an automorphism of $\mathcal{A}$ by finding a tuple $\bar{c}'_e$ so that its image $\tilde{c}_e$ through $\Phi_e^{S^0_\eta} \circ g^{-1}$ is not $\eta+1$-back-and-forth equivalent to $\bar{c}'_e$, and in particular, not automorphic to $\bar{c}'_e$. Since we do not know when or where $\Phi_e^{S^0_\eta}$ converges, we cannot ask about its values directly to the $\Delta^0_{\eta+1}$ oracle. All we can ask is, given a tuple $\bar{n}$ and a number s, whether $\Phi_e^{S^0_\eta}$ converges on the numbers in $\bar{n}$ within s steps.

At each step j, an initial segment $R_0,...,R_{k_j}$ of the list of requirements are active. The engineer goes through them one at the time checking if they *require attention* (defined below). Each requirement R_{e-1} outputs a tuple $\bar{p}_e[j] \subseteq \bar{b}_{j-1}$, which lower priority requirements are not allowed to modify. The tuple $\bar{p}_e[j]$ is then given to R_e as input, and R_e's output must extend it. (We will usually omit the $[j]$ from the notation when the stage is understood from context.) When a stronger priority requirement acts, the weaker requirement is deactivated and its $\bar{p}_e$ becomes undefined to be re-defined later. We will see, however, that for each e, $\bar{p}_e[j]$ will stabilize as $j \to \infty$ and hence we will end up with a limit function $g\colon \omega \to \mathcal{A}$, where $g(n) = \lim_{j\to\infty} \bar{p}_e[j](n)$.

If none of the requirements R_e for $e \leq k_j$ requires attention, the engineer initializes the first inactive requirement, namely, R_{k_j+1}: That means, for $e = k_j + 1$, the engineer looks for a tuple $\bar{c}_e$ that is η-free over $\bar{p}_e$ and adds it to the tuple, say on position $\bar{n}_e \in \mathbb{N}^{<\mathbb{N}}$. That is, she plays the tuple $\bar{a}_j = \bar{b}_{j-1}{}^\frown\bar{c}_e{}^\frown d$ and asks the oracle whether $\Phi_e^{S^0_\eta}(\bar{n}_e)$ converges within j steps, where d is the least element in $\mathcal{A}$ not yet played, and $\bar{n}_e = \langle|\bar{b}_{j-1}|, |\bar{b}_{j-1}+1|, ..., |\bar{b}_{j-1}{}^\frown\bar{c}_e| - 1\rangle$. (As we will see, we will have $\bar{p}_e \subseteq \bar{b}_{j-1}$.) She will keep on asking about this convergence, but with larger time bounds, at every later stage $j' > j$ until she gets an answer. Actually, it is for all $e \leq k_j + 1$ simultaneously that she asks whether $\Phi_e^{S^0_\eta}(\bar{n}_e)$ converges within j steps, encapsulating all questions into one question as in the last paragraph of Section 1.3.

What do we mean by requiring attention, and what does the engineer do then? If the oracle answered that for some $e < k_j$, $\Phi_e^{S^0_\eta}(\bar{n}_e)$ converges within $j-1$ steps, we say that the least such e *requires attention.* Suppose that $\Phi_e^{S^0_\eta}(\bar{n}_e) = \tilde{c}_e$. So we have that $\bar{p}_{e+1}$ maps $\bar{n}_e$ to $\bar{c}_e$ while $\Phi_e^{S^0_\eta}$

maps $\bar{n}_e$ to $\tilde{c}_e$. Then, she checks if $\bar{c}_e \leq_\eta \tilde{c}_e$. If not, she does not need to do anything, as we then know that $\bar{c}_e$ and $\tilde{c}_e$ are not non-automorphic. She plays $\bar{a}_j = \bar{b}_{j-1}{}^\frown d$ where d is the least element in $\mathcal{A}$ not yet played, declares R_e satisfied (unless is later deactivated), deactivates lower priority requirements, and lets $k_{j+1} = e-1$. If yes, that is if $\bar{c}_e \leq_\eta \tilde{c}_e$, she will switch $\bar{c}_e$ by a tuple $\bar{c}'_e \not\geq_{\eta+1} \bar{c}_e$, getting that $\bar{c}'_e$ and $\tilde{c}_e$ are non-automorphic. To find such $\bar{c}'_e$, we use the $(\eta+1)$-freeness of $\bar{c}_e$ over $\bar{p}_e$, applied to the tuple $\bar{b}_{j-1} \supseteq \bar{p}_e\bar{c}_e$. By $(\eta+1)$-freeness, we then know there is a tuple $\bar{c}'_e$ and a tuple $\bar{a}'_j \supseteq \bar{p}_e\bar{c}'_e$ such that $\bar{b}_{j-1} \leq_\eta \bar{a}'_j$ but $\bar{p}_e\bar{c}_e \not\leq_{\eta+1} \bar{p}_e\bar{c}'_e$. The engineer now plays $\bar{a}_j = \bar{a}'_j{}^\frown d$ where d is the least element in $\mathcal{A}$ not yet played, declares R_e satisfied (unless it is later deactivated), deactivates lower priority requirements, and lets $k_{j+1} = e-1$.

If R_e is never deactivated, we will end up with $\bar{p}_{e+1}[j] \subseteq \bar{a}_{j'}$ for all $j' \geq j$, and hence with $\bar{p}_{e+1}[j] \subseteq g$. We would have then satisfied R_e as either $\Phi_e^{S^0_\eta}$ is not total or $\Phi_e^{S^0_\eta} \circ g^{-1}$ maps either $\bar{c}_e$ or $\bar{c}'_e$ to $\tilde{c}_e$, and in either case these tuples are not $\eta+1$-back-and-forth equivalent. □

6. The low$_n$ property

Downey and Jockusch [DJ94] proved that Boolean algebras have the *low property*, that is, that every low Boolean algebra has a computable copy. Jockusch and Soare [JS91] had already shown this is not the case for linear orderings. A year later, Thurber [Thu95] showed that Boolean algebras have the low$_2$ property, that is, that every low$_2$ Boolean algebra has a computable copy. A few years later, Knight and Stob showed that Boolean algebras have the low$_4$ property. We still don't know whether Boolean algebras have the low$_n$ property for all $n \in \mathbb{N}$, or even if they have the low$_5$ property. Harris and Montalbán [HM14] showed that the low$_5$ problem for Boolean algebras is qualitatively more difficult that the previous ones: While for $n = 1, 2, 3, 4$, every low$_n$ Boolean algebra is $0^{(n+2)}$-isomorphic to a computable one, they built a low$_5$ Boolean algebra not $0^{(7)}$-isomorphic to any computable one.

One of the most interesting examples of a class with the low property is differentially closed fields of characteristic zero, as recently proved by Marker and Miller [MM17]. This allowed them to give a full description of the degree spectra of differentially closed fields of characteristic zero.

In [Mon13], we provided a framework for low-property proofs that separates the combinatorial/algebraic part of the proof from the computational part of the proof, hiding the computational part inside the framework, and letting the prover concentrate only on the combinatorial/algebraic part. We called that framework the *copy-vs-diagonalizer game*. In this section we just want to observe that the copy-vs-diagonalizer game is, essentially, a particular case of our new metatheorem.

We proved in [MonP1] that if a class of structures has the low$_1$ property, it must be Σ-small, meaning that there are countably many 1-back-and-forth equivalence classes among all tuples among all structures in the class. This can be easily extended to show that if a class of structures has the low$_n$ property, it must be Σ_n-small. Being Σ_n-small is not a a sufficient condition, but a necessary one that helps setting up the constructions. All natural Σ_n-small classes are *effectively* Σ_n*-small* (see [Mon10, Definition 2.3][Mon13, Definition 4.1][MonP1, Definition X.24]) meaning that we have a complete effective understanding of the interactions between the m-back-and-forth equivalence classes for $m \leq n$. We showed in [Mon10] that we can then build a computable list of structure representing all n-back-and-forth equivalence classes where the back-and-forth relations are computable up to n. Call it $\mathbb{K}_n$. Actually, having of such a list of structures is equivalent to having a computable representation of the n-back-and-forth structure of the class. We thus have a setting to perform an n-$\mathbb{K}_n$-game construction. When the engineer is playing tuples in $\mathbb{K}_n$, all that matters is the n-back-and-forth type of that tuple. So, if we know the structure of the n-back-and-forth types and we are given a way to represent them nicely, we may think that instead of playing structures in $\mathbb{K}_n$, we are playing is n-back-and-forth types.

If we are given a low$_n$ structure $\mathcal{A}$ in the class, the Δ_{n+1} oracle has access to its n-jump, and hence it knows how its tuples are m-back-and-forth related to the tuples of the structures in $\mathbb{K}_n$ — well, kind of: Given a tuple $\bar{k}$ in a structure $\mathcal{K}_i$ in $\mathbb{K}_n$, and a tuple $\bar{a} \in \mathcal{A}^{<\mathbb{N}}$, it is Π_n in the structure to tell if $(\mathcal{K}_i, \bar{k}) \leq_n (\mathcal{A}, \bar{a})$ (see [Mon10, Lemma 2.2]), but it is harder to tell if $(\mathcal{K}_i, \bar{k}) \geq_n (\mathcal{A}, \bar{a})$.[‡] What we showed in [Mon13] is that if we are given the nth jump of a structure we can produce an n-approximation to it. In the setting of the current paper, an *n-approximation* to a structure is a sequence of pairs $\langle i_0, \bar{k}_0\rangle, \langle i_1, \bar{k}_1\rangle, \langle i_2, \bar{k}_2\rangle, \ldots$ where $\bar{k}_j \in \mathbb{K}_{i_j}^{<\mathbb{N}}$ and

$$(\mathcal{K}_{i_0}, \bar{k}_0) \leq_n (\mathcal{K}_{i_1}, \bar{k}_1) \leq_n (\mathcal{K}_{i_2}, \bar{k}_2) \leq_n \cdots,$$

and the given structure $\mathcal{A}$ is the limit of this sequence in the sense that $D(\mathcal{A}) = \bigcup_j D_{\mathcal{K}_{i_j}}(\bar{k}_j)$.

The engineer can thus, at stage j, ask the oracle for the pair $\langle i_j, \bar{k}_j\rangle$. The engineer does not need to play in $\mathcal{K}_{i_j}$ in the next move, tough that may be a good idea. The engineer would like to copy $\mathcal{A}$. Her obstacle is that the extender is extending her tuples, and the fact that $(\mathcal{K}_{i_{j-1}}, \bar{k}_{j-1}) \leq_n (\mathcal{K}_{i_j}, \bar{k}_j)$ does not mean that every tuple $\bar{b}_{j-1}$ extending $\bar{a}_{j-1}$ has a matching tuple in $\mathcal{K}_{i_j}$. If she can overcome that obstacle, she will be able to produce a proof of the low$_n$-property.

7. η-true-stage systems

In this section we review η-true-stage systems, introduced by the author in [Mon14], so we can use them in the proof of the metatheorem in the next section.

Let η be a computable ω-presentation of an ordinal where we can decide whether an ordinal is a successor or a limit, and we can calculate successors.

An *η-true-stage system* is a computable family $\{\leq_\xi : \xi \leq \eta\}$ of partial orderings on $\mathbb{N}$ that satisfies the following properties:

(TS0) $\leq_0$ is just the standard ordering on $\mathbb{N}$.
(TS1) The sequence of relations is *nested* (i.e., if $\gamma \leq \xi$ and $s \leq_\xi t$, then $s \leq_\gamma t$).
(TS2) For every ξ, there exist an infinite $\leq_\xi$-increasing sequence $t_0 <_\xi t_1 <_\xi \cdots$.
(TS3) The sequence of relations is *continuous* (i.e., if λ is a limit ordinal, then $\leq_\lambda = \bigcap_{\xi<\lambda} \leq_\xi$).
(♣) For every ξ, and every $r < s < t$, if $r \leq_{\xi+1} t$ and $s \leq_\xi t$, then $r \leq_{\xi+1} s$.

$$\begin{tikzcd} r \arrow[rr, bend left, dash, "<_{\xi+1}"] \arrow[r, bend right, dotted, dash, "<_{\xi+1}"'] & s \arrow[r, bend right, dash, "<_{\xi}"'] & t \end{tikzcd}$$

For each $\xi \leq \eta$, we say that t is a *ξ-true stage* if it belongs to an infinite $\leq_\xi$-increasing sequence. We define $\mathfrak{T}^\xi \in \mathbb{N}^\mathbb{N}$ to be the sequence of ξ-true stages listed in increasing order. One could thus define $\mathfrak{T}^\xi$ as the union of all infinite $\leq_\xi$-increasing sequences. One can the prove using (♣) that $\mathfrak{T}^\xi$ is itself a $\leq_\xi$-increasing sequence. $\mathfrak{T}^\xi$ is thus the maximal $\leq_\xi$-increasing sequence. For successor ordinals, one can show using (♣) that

$$t \text{ is a } \xi+1\text{-true stage} \quad \iff \quad t \text{ is } \xi\text{-true and } (\forall s \geq_\xi t \text{ with } s\ \xi\text{-true})\ s \geq_{\xi+1} t.$$

[‡]The asymmetry comes from the assumption that the back-and-forth relation in $\mathcal{K}_i$ are computable, while in $\mathcal{A}$ they probably aren't.

For limit ordinals λ, t is a λ-true stage if and only if it is a ξ-true stage for all $\xi < \lambda$. This implies that the the set of ξ-true stages is Π^0_ξ. $\mathfrak{T}^\xi$ does not have the same m-degree as the Π^0_ξ, but the theorem below states that it may have the same Turing degree.

Definition 7.1. We say that an η-true-stage system is *complete* if $\mathfrak{T}^\xi$ is $\Delta^0_{\xi+1}$-Turing-complete for all $\xi \leq \eta$, uniformly in ξ.

Theorem 7.2. *There exists a complete η-true-stage system.*

This theorem was proved in [Mon14, Lemma 7.8]. A different construction was later given by Greenberg and Turetsky in [GT]. In [MonP2] we present yet a different construction which incorporates ideas from [GT]. In any case, for applications of the η-true-stage method, all that matters is that such a system exists. Let us now describe the key idea behind these systems.

We say that s is an *apparent ξ-true stage at t* if $s \leq_\xi t$. Given ξ and t, we define the *stage-t approximation to $\mathfrak{T}^\xi$*, denoted $\mathfrak{T}^\xi_t$, as the tuple enumerating the apparent ξ-true stages at t:

$$\mathfrak{T}^\xi_t = \langle s : s \leq_\xi t \rangle$$

Note that

$$s \leq_\xi t \iff \mathfrak{T}^\xi_s \subseteq \mathfrak{T}^\xi_t,$$

where the inclusion is as strings, and that

$$t \text{ is } \xi\text{-true} \iff \mathfrak{T}^\xi_t \subseteq \mathfrak{T}^\xi.$$

An η-system thus provides a combinatorial framework to approximate the iterations of the jumps. The properties (TS1)-(TS3) and (♣) describe the behavior of these approximations.

8. The proof of the game metatheorem

We split the proof in two independent parts by first considering a simplification of the game. In this simplification, the engineer does not get to choose which $\Delta^0_{\eta+1}$ question to ask, and the oracle just plays $\mathfrak{T}^\xi \restriction j$ at stage j, namely the string consisting of the first j many ξ-true stages. Recall that $\mathfrak{T}^\xi$ is $\Delta^0_{\eta+1}$ Turing complete, so if the engineer had a $\Delta^0_{\eta+1}$ question in mind, she will eventually be able to figure out the answer. Let us call this version of the game, the *simplified game*. Let us start seeing how to transform the general version of the game into an instance of the simplified version.

Proof of Theorem 2.1 from the simplified version of Theorem 2.1. Let σ be a computable valid strategy for the engineer in the η-$\mathbb{A}$-game from Theorem 2.1. We will build a computable ω-presentation for a limit-structure obtained from a certain sequence by the extender where the engineer follows her strategy σ. This construction will use a number ℓ as a parameter — so we are actually building a different ω-presentation $\mathcal{L}_\ell$ for each $\ell \in \mathbb{N}$. We will then use the recursion theorem to find a computable index ℓ_0 for the diagram of the limit-structure computed using ℓ_0 as a parameter. I.e.,

$$D(\mathcal{L}_{\ell_0}) = \Phi_{\ell_0}.$$

Thus, we may assume the parameter ℓ is an index for the computable diagram we are building. For this to work, we must produce a computable ω-presentation $\mathcal{L}_\ell$ even if the ℓth computable function Φ_ℓ is not total.

We will build a computable valid strategy $\hat{\sigma}$ for engineer in the simplified η-$\mathbb{A}$-game, and we will do it uniformly in ℓ. We will do it in a way that, for every run of the simplified game following $\hat{\sigma}$, there is a run of the original game following σ that produces the same limit structure. The simplified version of Theorem 2.1 (which we prove below) will give a sequence of moves by the extender such that, when we follow $\hat{\sigma}$, produces a computable ω-presentation $\mathcal{L}_\ell$. We omit the subindex ℓ in what follows.

Here is how we define $\hat{\sigma}$: Let Γ be a computable operator such that $\Gamma^{\mathfrak{T}^\eta}(e)$ is the answer to the eth $\Delta^0_{\eta+1}(\Phi_\ell)$ question. Let $\hat{\sigma}$'s first move be the same as σ's; that is let $\hat{\sigma}(\langle\rangle) = \sigma(\langle\rangle)$. At each following stage, $\hat{\sigma}$ may either *pass* or *emulate* σ, depending on whether on not the oracle has given her enough information to answer the last $\Delta^0_{\eta+1}$-question she asked. At a stage $j+1$, whether she passes or emulates σ gets decided as follows: Suppose the extender has just played $\bar{b}_j$, and the oracle played the string $\mathfrak{T}^\eta \upharpoonright j$. Let j_0 be the last stage at which $\hat{\sigma}$ emulated σ — suppose it was the kth time where $\hat{\sigma}$ emulated σ. At that stage, σ asked a $\Delta^0_{\eta+1}(\mathcal{L})$-question, say e_k. If $\Gamma^{\mathfrak{T}^\eta\upharpoonright j}(e_k)\downarrow$ let $\hat{\sigma}$ emulate σ and play the string σ would play if the extender had played $\bar{b}_j$ and the oracle has played $n_k = \Gamma^{\mathfrak{T}^\eta\upharpoonright j}(e_k)$. If $\Gamma^{\mathfrak{T}^\eta\upharpoonright j}(e_k)\uparrow$, let $\hat{\sigma}$ pass, that is, let it play $i_{j+1} = i_j$ and $\bar{a}_{j+1}$ be any proper extension of $\bar{b}_j$ in $\mathcal{A}_{i_j}$.

If we apply the simplified version of Theorem 2.1 to this strategy $\hat{\sigma}$, we end up building a computable ω-presentation $\mathcal{L}$. The diagram of $\mathcal{L}$ is total independently of whether Φ_ℓ is total and of whether we ever get answers to the $\Delta^0_{\eta+1}(\mathcal{L})$-questions e_k. This is because if $\hat{\sigma}$ ends up passing from some point j_0 onwards, then the limit structure will end up isomorphic to $\mathcal{A}_{i_{j_0}}$. Thus, when ℓ_0 is given to us by the recursion theorem as above, we get that Φ_{ℓ_0} is total and is equal to the diagram of the limit structure $\mathcal{L}_{\ell_0}$ we just obtained. Since σ is a valid strategy, all the $\Delta^0_{\eta+1}(\mathcal{L})$-questions it asks converge, and hence, for all k, $\Gamma^{\mathfrak{T}^\eta\upharpoonright j}(e_k)$ eventually converges. This means that there are infinitely many stages at which $\hat{\sigma}$ emulates σ, and we thus get that for every sequence of moves by the extender in the simplified game, there is a sequence of moves in the original game which give us the same limit structure. We then get a run of the original game where the limit structure is computable. □

The following lemma is the key property about the back-and-forth relations that allows us to simplify the metatheorems of Ash and Knight [AK00] and Montalbán [Mon14] in the particular case where the object we are building is an ω-presentation of a structure. Those metatheorems are more general than the one of this paper, and when they are applied in the literature the lemma below is used in the middle of each of the constructions. One of the ways in which our metatheorem simplifies the previous ones is by absorbing the lemma inside the proof of the metatheorem, so that the user of the metatheorem does not have to deal with it.

Lemma 8.1. *(Ash [Ash87, AK00]) Suppose we have a finite sequence of τ-structures $\mathcal{A}_0, ..., \mathcal{A}_k$, ordinals $\xi_{k-1} > \cdots > \xi_1 > \xi_0$, and tuples $\bar{a}_i \in A_i^{<\mathbb{N}}$ for $i \leq k$, such that*

$$(\mathcal{A}_k, \bar{a}_k) \leq_{\xi_{k-1}+1} (\mathcal{A}_{k-1}, \bar{a}_{k-1}) \leq_{\xi_{k-2}+1} \cdots \leq_{\xi_1+1} (\mathcal{A}_1, \bar{a}_1) \leq_{\xi_0+1} (\mathcal{A}_0, \bar{a}_0).$$

There exist a tuple $\bar{b} \in A_k^{<\mathbb{N}}$ extending $\bar{a}_k$ such that $(\mathcal{A}_j, \bar{a}_j) \leq_{\xi_j} (\mathcal{A}_k, \bar{b})$ for all $j < k$.

Proof. We will define a sequence of tuples $\bar{b}_j \in A_j^{<\mathbb{N}}$ extending $\bar{a}_j$ by induction on $j \leq k$, ending with $\bar{b} = \bar{b}_k$. We will make sure by induction that $(\mathcal{A}_i, \bar{a}_i) \leq_{\xi_i} (\mathcal{A}_j, \bar{b}_j)$ for all $i < j$.

$$\begin{tikzcd}
(\mathcal{A}_k, \bar{a}_k) \arrow[d, dotted, no head, "\rotatebox{-90}{$\subseteq$}" description] \arrow[rrd, dotted, no head, "\leq_{\xi_{k-1}+1}" sloped] & \leq_{\xi_{k-1}+1} & (\mathcal{A}_{k-1}, \bar{a}_{k-1}) \arrow[d, dotted, no head, "\rotatebox{-90}{$\subseteq$}" description] \arrow[rrd, dotted, no head, "\leq_{\xi_{k-2}+1}" sloped] & \leq_{\xi_{k-2}+1} \cdots & \ddots \arrow[rrd, dotted, no head, "\leq_{\xi_1+1}" sloped] & \leq_{\xi_1+1} & (\mathcal{A}_1, \bar{a}_1) \arrow[d, dotted, no head, "\rotatebox{-90}{$\subseteq$}" description] \arrow[rrd, dotted, no head, "\leq_{\xi_0+1}" sloped] & \leq_{\xi_0+1} & (\mathcal{A}_0, \bar{a}_0) \arrow[d, dotted, no head, "\parallel" description] \\
(\mathcal{A}_k, \bar{b}_k) & \geq_{\xi_{k-1}} & (\mathcal{A}_{k-1}, \bar{b}_{k-1}) & \geq_{\xi_{k-2}} & \cdots & \geq_{\xi_1} & (\mathcal{A}_1, \bar{b}_1) & \geq_{\xi_0} & (\mathcal{A}_0, \bar{b}_0)
\end{tikzcd}$$

Let $\bar{b}_0 = \bar{a}_0$. Given $\bar{b}_j$, since $(\mathcal{A}_{j+1}, \bar{a}_{j+1}) \leq_{\xi_j+1} (\mathcal{A}_j, \bar{a}_j)$, and $\bar{a}_j \subseteq \bar{b}_j$, there exists $\bar{b}_{j+1} \supseteq \bar{a}_{j+1} \in A_{j+1}^{<\mathbb{N}}$ such that $(\mathcal{A}_{j+1}, \bar{b}_{j+1}) \geq_{\xi_j} (\mathcal{A}_j, \bar{b}_j)$. We end up with $(\mathcal{A}_k, \bar{b}_k) \geq_{\xi_j} (\mathcal{A}_j, \bar{b}_j) \supseteq (\mathcal{A}_j, \bar{a}_j)$. □

Proof of Theorem 2.1 for the simplified game. We build a computable sequence of pairs $\langle i_s, \bar{a}_s\rangle$ with $\bar{a}_s \in \mathcal{A}_{i_s}$ which satisfies that, for all $\xi \leq \eta$, and $r < s \in \mathbb{N}$,

$$r \leq_\xi s \quad \text{implies} \quad (\mathcal{A}_{i_r}, \bar{a}_r) \leq_\xi (\mathcal{A}_{i_s}, \bar{a}_s). \tag{1}$$

In particular we get that if $s \leq t$, then $(\mathcal{A}_{i_s}, \bar{a}_s) \leq_0 (\mathcal{A}_{i_t}, \bar{a}_t)$, which means that $D_{\mathcal{A}_{i_s}}(\bar{a}_s) \subseteq D_{\mathcal{A}_{i_t}}(\bar{a}_t)$, and hence that the limit sequence with diagram $\bigcup_s D_{\mathcal{A}_{i_s}}(\bar{a}_s)$ is computable. To show that this structure is the limit structure under some run of the game following the engineer's strategy σ, we will show that if we restrict ourselves to the sequence of η-true stages $t_0 \leq_\eta t_1 \leq_\eta t_2 \leq_\eta \cdots$, then the sequence $\langle i_{t_0}, \bar{a}_{t_0}\rangle, \langle i_{t_1}, \bar{a}_{t_1}\rangle, \langle i_{t_2}, \bar{a}_{t_2}\rangle, ...$ can be seen as the sequence of moves by the engineer following σ for some particular sequence of moves by the extender.

For each s, will also define a tuple $\bar{b}_s$ that belongs to A_{i_t} for the largest $t < s$ with $t \leq_\eta s+1$. (This is the tuple we will use as the move by the extender later.) The tuple $\bar{b}_s$ will satisfy that for every $\xi \leq \eta$ and every $r < s$,

$$r \leq_\xi s+1 \quad \text{implies} \quad (\mathcal{A}_{i_r}, \bar{a}_r) \leq_\xi (\mathcal{A}_{i_t}, \bar{b}_s) \tag{2}$$

Thus, if we then define i_{s+1} and $\bar{a}_{s+1}$ satisfying $(\mathcal{A}_{i_t}, \bar{b}_s) \leq_\eta (\mathcal{A}_{i_{s+1}}, \bar{a}_{s+1})$, we will immediately get property (1). We will define $\bar{b}_s$ using the previous lemma, so we first need to find the right setting to apply it. For each $\xi \leq \eta$, let $t_\xi < s+1$ be the largest t such that $t \leq_\xi s+1$. (It does not hurt to assume that $0 \leq_\xi s+1$ for all ξ and s, so such a t_ξ always exists.) Notice that if $r <_\xi s+1$, then $r \leq t_\xi$ and then by (♣) $r \leq_\xi t_\xi \leq_\xi s+1$. So, to satisfy property (2), it is enough to get $\bar{b}_s$ so that $(\mathcal{A}_{i_{t_\xi}}, \bar{a}_{t_\xi}) \leq_\xi (\mathcal{A}_{i_t}, \bar{b}_s)$ for all $\xi \leq \eta$.

There are infinitely many ξ's, but only finitely many possible values for $t_\xi < s+1$, so they must repeat a lot. Since the relations $(\leq_\xi)_{\xi \leq \eta}$ are nested, if $\xi \leq \zeta \leq \eta_0$ then $t_\zeta \leq t_\xi$. We now want to define stages $s_k < .... < s_0 < s+1$ so that $\{s_k, ..., s_0\} = \{t_\xi : \xi \leq \eta_0\}$ as sets, but we need to define them effectively. Let $s_0 = t_0 = s$. Given s_j, let $\xi_j \leq \eta_0$ be the greatest such that $s_j = t_{\xi_j}$, i.e., the greatest ξ such that $s_j \leq_\xi s+1$. We notice that such a greatest ordinal exists by the continuity of $(\leq_\xi)_{\xi \leq \eta}$. If $\xi_j = \eta$, then we let $k = j$ and that finishes the definition of $s_0, ..., s_k$. Otherwise, let $s_{j+1} = t_{\xi_j+1}$. Since we know $s_j \not\leq_{\xi_j+1} s$, we must have $s_{j+1} < s_j$. By (♣) we then have $s_{j+1} \leq_{\xi_j+1} s_j$, and hence $(\mathcal{A}_{i_{s_{j+1}}}, \bar{a}_{s_{j+1}}) \leq_{\xi_j+1} (\mathcal{A}_{i_{s_j}}, \bar{a}_{s_j})$. We now apply Lemma 8.1 to the sequence

$$(\mathcal{A}_{i_{s_k}}, \bar{a}_{s_k}) \leq_{\xi_{k-1}+1} (\mathcal{A}_{i_{s_{k-1}}}, \bar{a}_{s_{k-1}}) \leq_{\xi_{k-2}+1} \cdots \leq_{\xi_1+1} (\mathcal{A}_{i_{s_1}}, \bar{a}_{s_1}) \leq_{\xi_0+1} (\mathcal{A}_{i_{s_0}}, \bar{a}_{s_0}),$$

to get $\bar{b}_s$ satisfying (2).

The last step is to define $\bar{a}_{s+1}$ using the strategy σ for the engineer in the simplified game. Let $0 = t_0, ..., t_j$ are the apparent η-true stages below $s+1$. Note that $\mathfrak{T}^\eta_{t_i} = \mathfrak{T}^\eta_{s+1} \restriction i$. We then let

$$\langle i_{s+1}, \bar{a}_{s+1}\rangle = \sigma(\bar{b}_{t_1-1}, \mathfrak{T}^\xi_{t_1}, \bar{b}_{t_2-1}, \mathfrak{T}^\xi_{t_2}, ..., \bar{b}_{t_j-1}, \mathfrak{T}^\xi_{t_j}, \bar{b}_s, \mathfrak{T}^\xi_{s+1}).$$

That is, $\langle i_{s+1}, \bar{a}_{s+1}\rangle$ is what the engineer would play in her $j+1$st move if she was following σ and the previous moves by the extender where $\bar{b}_{t_1-1}, \bar{b}_{t_2-1}, ...\bar{b}_{t_j-1}, \bar{b}_s$ and the previous plays by the oracle $\mathfrak{T}^\xi_{s+1} \restriction 1$,$\mathfrak{T}^\xi_{s+1} \restriction 2$,..., $\mathfrak{T}^\xi_{s+1}$.

Now, consider the sequence $t_1 <_\eta t_2 <_\eta \cdots$ of η-true stages. We get that the following is a run of the simplified game following σ:

engineer	$i_0, \bar{a}_0$		$i_{t_1}, \bar{a}_{t_1}$		$i_{t_2}, \bar{a}_{t_2}$		$\cdots$
extender		b_{t_1-1}		b_{t_2-1}		b_{t_3-1}	$\cdots$
oracle		$\mathfrak{T}^\xi \restriction 1$		$\mathfrak{T}^\xi \restriction 2$		$\mathfrak{T}^\xi \restriction 3$	$\cdots$

It follows that the limit structure of this run of the game is the computable structure with diagram $\bigcup_s D_{\mathcal{A}_{i_s}}(\bar{a}_s)$ that we mentioned above. □

References

[AJK90] C. J. Ash, C. G. Jockusch, Jr., and J. F. Knight. Jumps of orderings. *Trans. Amer. Math. Soc.*, 319(2):573–599, 1990.

[AK90] C. J. Ash and J. F. Knight. Pairs of recursive structures. *Ann. Pure Appl. Logic*, 46(3):211–234, 1990.

[AK00] C.J. Ash and J. Knight. *Computable Structures and the Hyperarithmetical Hierarchy.* Elsevier Science, 2000.

[AKMS89] Chris Ash, Julia Knight, Mark Manasse, and Theodore Slaman. Generic copies of countable structures. *Ann. Pure Appl. Logic*, 42(3):195–205, 1989.

[Ash86] C. J. Ash. Stability of recursive structures in arithmetical degrees. *Ann. Pure Appl. Logic*, 32(2):113–135, 1986.

[Ash87] C. J. Ash. Categoricity in hyperarithmetical degrees. *Ann. Pure Appl. Logic*, 34(1):1–14, 1987.

[Ash91] C. J. Ash. A construction for recursive linear orderings. *J. Symbolic Logic*, 56(2):673–683, 1991.

[Bar73] J. Barwise. Back and forth through infinitary logic. In M. D. Morley, editor, *Studies in model theory*, pages 5–34. The Mathematical Association of America, Buffalo, N.Y., 1973.

[DJ94] Rod Downey and Carl G. Jockusch. Every low Boolean algebra is isomorphic to a recursive one. *Proc. Amer. Math. Soc.*, 122(3):871–880, 1994.

[DKL$^+$15] Rodney G. Downey, Asher M. Kach, Steffen Lempp, Andrew E. M. Lewis-Pye, Antonio Montalbán, and Daniel D. Turetsky. The complexity of computable categoricity. *Advances in Mathematics*, 268:423–466, 2015.

[Fri57] Richard M. Friedberg. Two recursively enumerable sets of incomparable degrees of unsolvability (solution of Post's problem, 1944). *Proc. Nat. Acad. Sci. U.S.A.*, 43:236–238, 1957.

[GHK$^+$05] Sergey Goncharov, Valentina Harizanov, Julia Knight, Charles McCoy, Russell Miller, and Reed Solomon. Enumerations in computable structure theory. *Ann. Pure Appl. Logic*, 136(3):219–246, 2005.

[GMS13] N. Greenberg, A. Montalbán, and T. A. Slaman. Relative to any non-hyperarithmetic set. *Journal of Mathematical Logic*, 13(1), 2013.

[GT] Noam Greenberg and Daniel Turetsky. Completeness of the hyperaritmetic isomorphism equivalence relation. Submitted for publication.

[Har76] L. Harrington. Mclaughlin's conjecture. Handrwitten notes, 11 pages, September 76.

[HM14] Kenneth Harris and A. Montalbán. Boolean algebra approximations. *Transactions of the AMS*, 366(10):5223–5256, 2014.

[JS91] Carl G. Jockusch, Jr. and Robert I. Soare. Degrees of orderings not isomorphic to recursive linear orderings. *Ann. Pure Appl. Logic*, 52(1-2):39–64, 1991. International Symposium on Mathematical Logic and its Applications (Nagoya, 1988).

[Kar65] Carol R. Karp. Finite-quantifier equivalence. In *Theory of Models (Proc. 1963 Internat. Sympos. Berkeley)*, pages 407–412. North-Holland, Amsterdam, 1965.

[Lac76] Alistair H. Lachlan. A recursively enumerable degree which will not split over all lesser ones. *Ann. Math. Logic*, 9(4):307–365, 1976.

[Ler10] Manuel Lerman. *A framework for priority arguments*, volume 34 of *Lecture Notes in Logic.* Association for Symbolic Logic, La Jolla, CA, 2010.

[LL95] Steffen Lempp and Manuel Lerman. A general framework for priority arguments. *Bull. Symbolic Logic*, 1(2):189–201, 1995.

[MM17] David Marker and Russell Miller. Turing degree spectra of differentially closed fields. *J. Symb. Log.*, 82(1):1–25, 2017.

[Mon10] Antonio Montalbán. Counting the back-and-forth types. *Journal of Logic and Computability*, page doi: 10.1093/logcom/exq048, 2010.

[Mon13] Antonio Montalbán. Copyable structures. *Journal of Symbolic Logic*, 78(4):1025–1346, 2013.

[Mon14] Antonio Montalbán. Priority arguments via true stages. *Journal of Symbolic Logic*, 79(4):1315–1335, 2014.

[Mon15] Antonio Montalbán. A robuster Scott rank. *Proc. Amer. Math. Soc.*, 143(12):5427–5436, 2015.

[MonP1] Antonio Montalbán. Computable structure theory: Beyond the arithmetic. In preparation, P1.

[MonP2] Antonio Montalbán. Computable structure theory: Beyond the arithmetic. In preparation, P2.

[Muc56] A. A. Muchnik. On the unsolvability of the problem of reducibility in the theory of algorithms. *Dokl. Akad. Nauk SSSR, N.S.*, 108:194–197, 1956.

[Sac63] Gerald E. Sacks. *Degrees of unsolvability.* Princeton University Press, Princeton, N.J., 1963.

[Sho61] J. R. Shoenfield. Undecidable and creative theories. *Fund. Math.*, 49:171–179, 1960/61.

[Thu95] John J. Thurber. Every low$_2$ Boolean algebra has a recursive copy. *Proc. Amer. Math. Soc.*, 123(12):3859–3866, 1995.

[Wat84] Richard Watnick. A generalization of Tennenbaum's theorem on effectively finite recursive linear orderings. *J. Symbolic Logic*, 49(2):563–569, 1984.

Department of Mathematics, University of California, Berkeley
E-mail address: antonio@math.berkeley.edu
URL: www.math.berkeley.edu/∼antonio