Intrinsic Complexity of Recursive Functions on Natural Numbers with Standard Order

Nikolay Bazhenov
Sobolev Institute of Mathematics, Novosibirsk, Russia

Dariusz Kalociński[1]
Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Michał Wrocławski
Faculty of Philosophy, University of Warsaw, Poland

Abstract

The intrinsic complexity of a relation on a given computable structure is captured by the notion of its degree spectrum – the set of Turing degrees of images of the relation in all computable isomorphic copies of that structure. We investigate the intrinsic complexity of unary total recursive functions on nonnegative integers with standard order. According to existing results, the possible spectra of such functions include three sets consisting of precisely: the computable degree, all c.e. degrees and all Δ_2 degrees. These results, however, fall far short of the full classification. In this paper, we obtain a more complete picture by giving a few criteria for a function to have intrinsic complexity equal to one of the three candidate sets of degrees. Our investigations are based on the notion of block functions and a broader class of quasi-block functions beyond which all functions of interest have intrinsic complexity equal to the c.e. degrees. We also answer the questions raised by Wright [21] and Harrison-Trainor [10] by showing that the division between computable, c.e. and Δ_2 degrees is insufficient in this context as there is a unary total recursive function whose spectrum contains all c.e. degrees but is strictly contained in the Δ_2 degrees.

2012 ACM Subject Classification Theory of computation → Computability

Keywords and phrases Computable Structure Theory, Degree Spectra, ω-Type Order, c.e. Degrees, d.c.e. Degrees, Δ_2 Degrees, Learnability

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.8

Related Version *Full Version*: `https://arxiv.org/abs/2110.01268`

Funding *Nikolay Bazhenov*: The work of Bazhenov is supported by the Mathematical Center in Akademgorodok under agreement No. 075-15-2019-1613 with the Ministry of Science and Higher Education of the Russian Federation.
Dariusz Kalociński: The work of Kalociński is supported by the National Science Centre Poland, grant No. 2018/31/B/HS1/04018.

Acknowledgements We would like to thank Matthew Harrison-Trainor, Aleksander Iwanow, Mars Yamaleev for helpful discussions and anonymous reviewers for comments.

1 Introduction

In mathematics we study structures of various sorts like rings, fields or linear orders. In computability theory we investigate the complexity of countable objects. A combination of the two – computable structure theory – examines the relationship between complexity and structure in the above sense [1, 14]. One of the main research programs in computable structure theory consists in the study of how complexity of a relation on a given structure

[1] `dariusz.kalocinski@gmail.com`

behaves under isomorphisms (see, e.g., [17, 9, 11, 7]). Recall that a structure is computable if its domain and basic relations are uniformly computable. The complexity of a relation can be captured by a measure such as Turing degrees. This leads to the notion of the degree spectrum (of a computable relation on a computable structure) – the set of Turing degrees assumed by the images of that relation in all computable isomorphic copies of that structure. This notion captures what might otherwise be called the intrinsic complexity of a relation.

A natural motivation for investigating intrinsic complexity comes from treating computable copies of a structure as notations: we regard the elements of the copy as names for the members of the structure, with the underlying isomorphism acting as a naming function. A computable copy of a structure is thus a notation in which all the basic relations are computable (meaning that their images within the copy are computable). This is essentially Shapiro's idea, as studied, though in a very restricted sense, in [18]. But this analogy goes further. Shapiro insisted, not without reason, that computations are not performed directly on numbers but rather on their names (using the terminology of computable structure theory: computations are not performed on the underlying structure but on isomorphic copies). This intuition transfers to all computation-dependent notions, including complexity. In the end, the intricate notion of intrinsic complexity boils down to the study of how difficult it is to compute the relation in notations in which all the basic relations are computable.

Following Downey et al. [5] and Wright [21], we investigate degree spectra on the most common ordering: non-negative integers with the standard *less than* relation, denoted by $(\omega, <)$. We study this question in the restricted setting of specific binary relations of general interest – graphs of unary total computable functions. As an example of how isomorphism might influence the complexity of a such a function, consider the successor. By a well-known result (see, e.g., Example 1.3 in [2]), there is an isomorphic copy of $(\omega, <)$ in which the image of the successor computes the halting problem. In general, however, as one can easily observe, the range of intrinsic complexity of a computable relation on $(\omega, <)$ is restricted to Δ_2 degrees and, therefore, each isomorphic image of such a relation is learnable (i.e., it possesses a recursive approximation) or, equivalently, Turing reducible to the halting problem [8, 16, 19].

Several results from the literature partially characterize degree spectra of such functions. Moses [15] provided a syntactical characterization of intrinsically computable (i.e. having only the computable degree in their spectrum) n-ary relations on $(\omega, <)$. These results imply that a total unary recursive function is intrinsically computable if and only if it is almost constant or almost identity (see Proposition 6). In [5], Downey, Khoussainov, Miller and Yu examined degree spectra of unary relations on $(\omega, <)$. Their results show, among others, that the spectrum of any infinite coinfinite computable unary relation on $(\omega, <)$ contains all c.e. degrees (Theorem 1.1 in [5]). Wright extended their results by showing the following.

▶ **Theorem 1** (Wright [21]). *The spectrum of a computable n-ary relation which is not intrinsically computable contains all c.e. degrees.*

He was also able to show that a computable *unary* relation which is not intrinsically computable has Δ_2 degrees as a spectrum (see, also, [12]).

Wright asked in [21] whether the computable, the c.e. and the Δ_2 degrees exhaust possible degree spectra for computable n-ary relations on $(\omega, <)$. Roughly at about the same time, Harrison-Trainor posed a related question in [10] where he showed that there exists a computable relation R on $(\omega, <)$ such that its degree spectrum either

(1) contains the c.e. degrees but does not contain all of the Δ_2 degrees, or

(2) consists of exactly all Δ_2 degrees but R does not have this degree spectrum uniformly.

Harrison-Trainor conjectured that (1) holds for the relation he constructed. We construct a unary total computable function (hence, a computable binary relation) witnessing (1). This also answers Wright's question.

Results of this paper are heavily based on certain structural characteristics of functions, which we refer to as the block and (a weaker) quasi-block property. Intuitively, each block function on $(\omega, <)$ is defined by multiple sub-functions where each sub-function applies to a different finite $<$-interval of ω (Definition 10). A quasi-block function is one for which there are increasingly long initial $<$-segments such that no number from within the segment is sent outside. The usefulness of these properties is clear in view of the observation that any computable non-quasi-block function has exactly all c.e. degrees as a spectrum (Theorem 18). One of the main contributions of the paper consists in the complete characterization of degree spectra of block functions which have at most finitely many isomorphism types of their elementary sub-functions (Theorem 14). The second main contribution is Theorem 23 which answers Wright's and Harrison-Trainor's questions.

2 Definitions

▶ **Definition 2.** *$(\omega, \prec)$ is a computable copy of $(\omega, <)$ if $\prec$ is a computable ordering on ω and structures $(\omega, <)$ and $(\omega, \prec)$ are isomorphic.*

▶ **Definition 3.** *Let R be a relation on $(\omega, <)$, i.e. $R \subseteq \omega^k$, for some $k \in \omega$, and let $\mathcal{A}$ be a computable copy of $(\omega, <)$. If φ is an isomorphism from $(\omega, <)$ to $\mathcal{A}$, we write $R_\mathcal{A}$ for the image of R under φ.*

▶ **Definition 4.** *Let R be a relation on $(\omega, <)$. The degree spectrum or spectrum of R on $(\omega, <)$, in symbols $DgSp_{(\omega,<)}(R)$, is the set of Turing degrees of $R_\mathcal{A}$ over all computable copies $\mathcal{A}$ of $(\omega, <)$.*

Throughout the article, we use abbreviated forms: *spectrum of R* and $DgSp(R)$.

▶ **Definition 5.** *Let R be a relation on $(\omega, <)$. The relation R is intrinsically computable if $DgSp(R)$ contains only the computable degree.*

Let $\mathcal{A} = (A, <_\mathcal{A})$ be a linear order. If $a \leq_\mathcal{A} b$, then $[a; b]_\mathcal{A}$ and $[a; b)_\mathcal{A}$ denote the intervals $\{x : a \leq_\mathcal{A} x \leq_\mathcal{A} b\}$ and $\{x : a \leq_\mathcal{A} x <_\mathcal{A} b\}$, respectively. If the order $\mathcal{A}$ is clear from the context, then we omit the subscript $\mathcal{A}$. *Succ* is the successor function on $(\omega, <)$. $\langle \cdot, \cdot \rangle$ is the pairing function. Computability-related notation is standard and follows [20]. For example, $\leq_T$ denotes the Turing reduction.

If $X \subseteq \omega$ is a Δ_2 set, then one can choose its *computable approximation* $\xi(k, s)$, i.e. a $\{0, 1\}$-valued computable function such that $\lim_s \xi(k, s) = X(k)$, for all k. We often use notation $X_s(k)$ for $\xi(k, s)$.

3 Results

The following two statements will be useful (the proof of the first one is in the full version).

▶ **Proposition 6.** *Let f be a unary total computable function. Then f is intrinsically computable if and only if either f is almost constant, or f is almost identity.*

▶ **Proposition 7** (see, e.g., Example 1.3 in [2]). *The spectrum of successor is equal to the c.e. degrees.*

▶ **Theorem 8.** *Let f be a unary computable function with finite range. If f is not intrinsically computable then its spectrum is equal to the Δ_2 degrees.*

Proof. The proof is based on the ideas from Theorem 1.2 of [21]. We provide a detailed exposition, so that a reader could familiarize themselves with the proof techniques.

We fix $c_0 \neq c_1$ such that $f^{-1}(c_i)$ is infinite. Without loss of generality, one may assume that $c_0 = 0$ and $c_1 = 1$.

Let $X \subseteq \omega$ be an arbitrary Δ_2 set. We build a computable isomorphic copy $\mathcal{A} = (\omega, <_{\mathcal{A}})$ of the order $(\omega, <)$ such that $f_{\mathcal{A}}$ is Turing equivalent to the set X. Our construction will ensure that the following two conditions hold:

(i) $k \in X$ if and only if $f_{\mathcal{A}}(2k) = 1$, for all k;

(ii) the restriction of $f_{\mathcal{A}}$ to the set of odd numbers (i.e., $f_{\mathcal{A}} \upharpoonright \{2k+1 : k \in \omega\}$) is computable.

It is clear that these conditions imply $f_{\mathcal{A}} \equiv_T X$.

Let M be a large enough natural number such that

$$(\forall x > M)[\text{the } f\text{-preimage of } f(x) \text{ is infinite, and } x \notin \text{range}(f)].$$

Beforehand, we use odd numbers to copy the initial segment $[0; M]$ of $(\omega, <)$. More formally, we put $2k + 1 <_{\mathcal{A}} 2l + 1$ for all $k < l \leq M$. In addition, any newly added (to the copy $\mathcal{A}$) number will be strictly $\mathcal{A}$-greater than $2M + 1$.

Our construction satisfies the following requirements:

$$\begin{aligned} e \in X &\Leftrightarrow f_{\mathcal{A}}(2e) = 1, \\ e \notin X &\Leftrightarrow f_{\mathcal{A}}(2e) = 0. \end{aligned} \qquad (\mathcal{R}_e)$$

As usual, this will be achieved by working with a computable approximation $X_s(e)$.

By $\mathcal{A}_s$ we denote the finite structure built at a stage s. At each stage s, there is a natural isomorphic embedding h_s from $\mathcal{A}_s$ into $(\omega, <)$. If $\mathcal{A}_s$ consists of $a_0 <_{\mathcal{A}} a_1 <_{\mathcal{A}} a_2 <_{\mathcal{A}} \ldots <_{\mathcal{A}} a_n$, then we assume that $h_s(a_i) = i$, for all $i \leq n$.

This convention allows one to talk about values $f_{\mathcal{A}_s}(x)$ for elements $x \in \mathcal{A}_s$. We simply assume that

$$f_{\mathcal{A}_s}(a_i) = h_s^{-1} \circ f \circ h_s(a_i).$$

Our construction will ensure that $f_{\mathcal{A}}(x) = \lim_s f_{\mathcal{A}_s}(x)$, for all x. Sometimes (when the usage context is unambiguous), we write just $f_{\mathcal{A}}(x)$ in place of $f_{\mathcal{A}_s}(x)$.

Strategy $\mathcal{R}_e$ in isolation. Suppose that $(s_0 + 1)$ is the first stage of work for this strategy. Then we add $2e$ to the right end of $\mathcal{A}$. Since we want to ensure that $f_{\mathcal{A}_{s_0+1}}(2e) = X_{s_0+1}(e)$, we also add (if needed) finitely many fresh odd numbers in-between $\mathcal{A}_s$ and $2e$, i.e., we set

$$a <_{\mathcal{A}} 2k + 1 <_{\mathcal{A}} 2e,$$

for $a \in \mathcal{A}_{s_0}$ and newly added numbers $2k + 1$.

We say that $\mathcal{R}_e$ *requires attention* at a stage s if the current value $f_{\mathcal{A}_s}(2e)$ is not equal to $X_s(e)$. In order to deal with $\mathcal{R}_e$, we introduce the following important ingredient of our proof techniques. For the sake of future convenience, we give a *general* description of the module.

Pushing-to-the-right module (PtR-module). We split the (current finite) structure $\mathcal{A}_s$ into three intervals: $B <_{\mathcal{A}} C <_{\mathcal{A}} D$, where, say, we have $B = [a; b]_{\mathcal{A}}$, $C = \{c^0 <_{\mathcal{A}} c^1 <_{\mathcal{A}} \ldots <_{\mathcal{A}} c^m\}$, and $D = \{d^0 <_{\mathcal{A}} d^1 <_{\mathcal{A}} \ldots <_{\mathcal{A}} d^n\}$. Informally speaking, the module aims to achieve the following goal: while preserving all values $f_{\mathcal{A}}(x)$ for $x \in B \cup D$, we want to change the function $f_{\mathcal{A}} \upharpoonright C$ in such a way that $f_{\mathcal{A}}$ satisfies a particular requirement. In addition, we require that C remains an interval inside $\mathcal{F}$.

More formally, we extend the structure $\mathcal{A}_s$ to a finite structure $\mathcal{F}$ (which is intended to be an initial segment of $\mathcal{A}_{s+1}$) with the following properties:

- every element $x \in \mathcal{F} \setminus \mathcal{A}_s$ is a fresh odd number, and each such x satisfies either $B <_{\mathcal{A}} x <_{\mathcal{A}} C$ or $x >_{\mathcal{A}} C$;
- $f_{\mathcal{F}}(d^i) = f_{\mathcal{A}_s}(d^i)$ for all $i \leq n$;
- the new values $f_{\mathcal{F}}(c^j)$ satisfy some *target condition*.

In the future, when we talk about a particular instance of the module, we will always explicitly specify the desired target condition.

Roughly speaking, our module keeps the interval B fixed, while all elements from $C \cup D$ are pushed to the right (with the help of newly added odd numbers). In addition, the elements of C stick together.

Going back to $\mathcal{R}_e$: if $\mathcal{R}_e$ requires attention at a stage s, then we implement the following actions.

The PtR-module for the strategy $\mathcal{R}_e$. In our $\mathcal{R}_e$-setting, we choose the middle interval C as the singleton $\{2e\}$. The desired target condition is a natural one: we aim to satisfy $f_{\mathcal{A}}(2e) = X_s(e)$.

We build a finite structure $\mathcal{F}$ extending $\mathcal{A}_s$ as dictated by the PtR-module. Then we declare that $\mathcal{F}$ is the output of our module, and proceed further. This concludes the description of the $\mathcal{R}_e$-strategy.

Construction. At a stage $s+1$, we work with strategies $\mathcal{R}_e$, for $e \leq s$. So, a strategy $\mathcal{R}_e$ starts working at the stage $e+1$. For each $\mathcal{R}_e$ (in turn), our actions follow the description given above. After $\mathcal{R}_i$ finished its work, the PtR-module of the next strategy $\mathcal{R}_{i+1}$ works with the finite structure produced by $\mathcal{R}_i$. Since the described PtR-module preserves $f_{\mathcal{A}} \upharpoonright (B \cup D)$, our strategies do not injure each other. We define $\mathcal{A} = \bigcup_{s \in \omega} \mathcal{A}_s$, where $\mathcal{A}_{s+1}$ is the final content of our structure produced by the PtR-module of $\mathcal{R}_s$ at the end of stage $s+1$.

Verification. First, we show that in the construction, every application of a PtR-module is successful (i.e., one can always build a desired structure $\mathcal{F}$).

In order to prove this, we consider our structures from a different angle: the structure $(\omega, <, f)$ can be treated as an infinite string β over a finite alphabet $\Sigma = \mathrm{range}(f)$, where the i-th symbol $\beta(i)$ of the string is equal to $f(i)$, $i \in \omega$.

Then the construction of $\mathcal{F}$ in the PtR-module can be re-interpreted as follows. We are given three finite strings, namely σ, τ (of length one), and ρ (of length $n+1$), for the intervals B, $C = \{2e\}$, and D correspondingly. Our task is to find finite strings $\tau', \rho'_0, \rho'_1, \ldots, \rho'_n$ with the following property:

$$\sigma\, \tau'\, a\, \rho'_0\, \rho(0)\, \rho'_1\, \rho(1)\, \ldots\, \rho'_n\, \rho(n),$$

where $a = X_s(e)$, is an initial segment of β.

This task can be always implemented successfully – this is a consequence of the following simple combinatorial fact.

▶ Remark 9. Let Σ be a finite alphabet, and let $\alpha \in \Sigma^\omega$ be an infinite string over Σ. Suppose that every symbol from Σ occurs infinitely often in α. Then for every finite string $\sigma \in \Sigma^{<\omega}$ of length $m > 0$, one can find finite strings $\tau_0, \tau_1, \dots, \tau_{m-1}$ such that

$$\tau_0\, \sigma(0)\, \tau_1\, \sigma(1) \, \dots \, \tau_{m-1}\, \sigma(m-1) \text{ is an initial segment of } \alpha.$$ ⌟

So, we deduce that all applications of a PtR-module are successful. Hence, if $e \leq s$, then by the end of the stage $s+1$ we have $f_{\mathcal{A}_{s+1}}(2e) = X_s(e)$. This implies that every requirement $\mathcal{R}_e$ is satisfied.

Each element $a \in \mathcal{A}$ moves (to the right) only finitely often. Indeed, there are only finitely many even numbers $2e$ such that $2e \leq_{\mathcal{A}} a$. Consider a stage s^* such that the values $X_s(e)$ (for these $2e$) never change after s^*. Clearly, the element a never moves after the stage s^*.

We deduce that the structure $\mathcal{A}$ is a computable copy of $(\omega, <)$. For every k, after the value $f_{\mathcal{A}_s}(2k+1)$ is defined for the first time, this value never changes (since the PtR-module always preserves the restriction $f_{\mathcal{A}} \upharpoonright (B \cup D)$). Therefore, our structure $\mathcal{A}$ satisfies Conditions (i) and (ii) defined above. Theorem 8 is proved. ◀

3.1 Block and Quasi-Block Functions

From now on, we study some natural subclasses of unary total recursive functions with infinite range.

▶ **Definition 10.** *Let $f\colon \omega \to \omega$ be a total function. An interval I of $(\omega, <)$ is* f-closed *if for all $x \in I$, $f(x) \in I$ and $f^{-1}(x) \subseteq I$. For a finite non-empty interval $I \subset \omega$, the structure $(I, <, f \upharpoonright I)$ is an* f-block *if it has the following properties:*

- *I is an f-closed interval and it cannot be written as a disjoint union of several f-closed intervals;*
- *$\{x \in \omega : x < I\}$ is f-closed.*

The function f is a block function *if for every $a \in \omega$, there is an f-block containing a.*
If $(I, <, f \upharpoonright I)$ is an f-block, we refer to its isomorphism type as an f-type *(or a type).*

The second condition of the definition above ensures that for a block function f, every element is contained in a unique f-block. Observe that in Fig. 1, without this condition, the element 2 would be an f-block itself, which we would like to avoid.

▶ Remark 11. For any computable block function f there is a 1-1 computable enumeration of its types. f can be represented by the unique infinite string $\alpha_f : \omega \to [0; N)$, where $[0, N)$ is the domain of the enumeration, for some $N \in \omega \cup \{+\infty\}$. For example, if $I_0, I_1, \dots, I_N$ are all (isomorphism types of) f-blocks, then $(\omega, <, f)$ can be treated as an infinite string $\alpha_f : \omega \to \{n : 0 \leq n \leq N\}$, e.g. a string $012012012\dots$ corresponds to a disjoint sum of the following form: $I_0 + I_1 + I_2 + I_0 + I_1 + I_2 + I_0 + I_1 + I_2 + \dots$

▶ **Example 12.** $f(n) = 2 \cdot \lfloor \frac{n}{2} \rfloor$ is a block function. Its spectrum consists of all Δ_2 degrees by Theorem 14 below.

▶ **Example 13.** Consider finite structures $\mathcal{J}_n$ from Figure 1. Let g be the involution such that $(\omega, <, g) \cong \mathcal{J}_0 + \mathcal{J}_1 + \mathcal{J}_2 + \dots$ Clearly, g is a block function. In the full version, we show that its degree spectrum is all of the c.e. degrees.

▶ **Theorem 14.** *Let f be a computable block function such that it has only finitely many f-types and f is not almost identity. Then the spectrum $DgSp(f)$ consists of all Δ_2 degrees.*

Due to space constraints, the proof has been moved to the appendix.

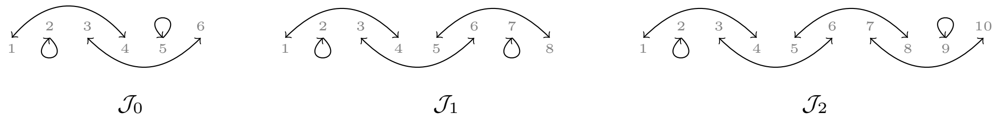

Figure 1 Structures $\mathcal{J}_n = ([1; 6+2n], <, f)$, for $n = 0, 1, 2$, where f is the involution such that $f(k) = k$ iff $k = 2$ or $k = 6 + 2n - 1$, and $f(k) = k + 3$ for odd numbers $\leq 6 + 2n - 3$.

The notion of a quasi-block function is a generalization of the notion of a block function. Unlike blocks which are disjoint and follow each other, quasi-blocks are increasingly larger and they are initial segments of ω.

▶ **Definition 15.** *We say that $f : \omega \to \omega$ is a* quasi-block function *if there are arbitrarily long finite initial segments of ω closed under f. For any such segment $I = [0; n]$, the structure $(I, <, f \restriction I)$ is an f-*quasi-block*.*

If f is a quasi-block function but not a block function, we call f a proper quasi-block function.

▶ **Example 16.** Euler's function is a function φ such that if $n > 0$, then $\varphi(n)$ is the number of such $m \leq n$ that m and n are relatively prime. φ is a proper quasi-block function. Since φ has a computable non-decreasing lower bound $\lfloor \sqrt{\frac{n}{2}} \rfloor$ diverging to ∞ (see, e.g., [13, p. 9]), the spectrum of φ is equal to the c.e. degrees by Theorem 19.

▶ **Example 17.** The function $nd : \omega \to \omega$ assigning to each $n > 0$ the number of its divisors is a proper quasi-block function.

Below we describe a method used to show that the degree spectrum of a certain unary recursive function f consists exactly of c.e. degrees.

Retrieving the Successor module (RS) on $(\omega, <, f)$, for f recursive, is a scheme of algorithms which, for any computable copy $\mathcal{A}$ of $(\omega, <)$ and an initial segment $\mathcal{I}_t$ of $\mathcal{A}$ satisfying some condition $\mathcal{R}$ (to be specified in a concrete implementation) computes, uniformly in t and relative to $f_{\mathcal{A}}$, a longer initial segment $\mathcal{I}_{t+1}$ of $\mathcal{A}$ satisfying $\mathcal{R}$, which enables us to construct an increasing sequence of initial segments $\mathcal{I}_0 \subset \mathcal{I}_1 \subset \ldots$.

Suppose that there exists a concrete implementation of the RS-module for $(\omega, <, f)$. We wish to show that the degree spectrum of f on $(\omega, <)$ consists of exactly c.e. degrees. To this aim, we want to show that $Succ_{\mathcal{A}}$ is Turing-reducible to $f_{\mathcal{A}}$. We also observe that the reduction in the other direction works. We conclude that $Succ_{\mathcal{A}} \equiv_T f_{\mathcal{A}}$, hence $DgSp(Succ) = DgSp(f)$, i.e. they consist of all c.e. degrees. This conclusion is based on Proposition 7.

Suppose that an initial segment of ω up to n (according to $<$) has already been determined, along with its isomorphic image $\mathcal{I}_t$ in $(\omega, \prec)$. Let us adopt a convention that the isomorphic image of each number i is k_i. Observe that for each number i such that $k_i \prec k_n$ we know how to determine its successor in $(\omega, \prec)$. In an application of the RS-module, given k_n – the rightmost element of $\mathcal{I}_t$ – we get some k_m and m such that $k_n \prec k_m$ and $[k_0; k_m]_{\mathcal{A}}$ satisfies $\mathcal{R}$. We know that in $\mathcal{A}$ there are exactly $m - n - 1$ elements between k_n and k_m. Since the ordering $\prec$ is recursive, we can check elements one by one until we determine what elements (and in what order) are between k_n and k_m. This way we extend the initial segment $\mathcal{I}_t$ of $\mathcal{A}$ to a larger initial segment $\mathcal{I}_{t+1}$ satisfying $\mathcal{R}$ and we are able to retrieve more values of the successor in this structure.

▶ **Theorem 18.** *The spectrum of any unary total computable non-quasi-block function is equal to the c.e. degrees.*

Proof. We show that the RS module can be used for $(\omega, <, f)$. Given a computable copy $\mathcal{A}$ of $(\omega, <)$, we set $\mathcal{I}_0$ as the image of some initial segment of $(\omega, <)$ such that for every position n outside of $\mathcal{I}_0$ there is $m \leq n$ such that $f(m) > n$. The condition $\mathcal{R}$ states that there is a position j within $\mathcal{I}_t$ such that $f(j) > n$. Then if we already know $\mathcal{I}_t$ and want to determine $\mathcal{I}_{t+1}$, we calculate both $f(j)$ and $f_{\mathcal{A}}(k_j)$ from the condition $\mathcal{R}$, obtaining some values of these functions m and k_m, each of them somewhere behind n and k_n in their sequences. ◀

▶ **Theorem 19.** *If f is a recursive proper quasi-block function with a computable non-decreasing lower bound diverging to $+\infty$, then its spectrum consists of exactly c.e. degrees.*

Proof. We claim that there exist only finitely many quasi-blocks closed under both f and f^{-1}. Observe that if there were infinitely many such quasi-blocks, then f would be a block function. Observe also that if f is as above, then we are able to calculate how many times each of its values is assumed.

We utilise the RS module. The segment $\mathcal{I}_0$ is any initial segment such that none of its super-quasi-blocks is closed under both f and f^{-1}. Assume we already have a segment $\mathcal{I}_t$ of $\mathcal{A}$ retrieved. $\mathcal{I}_t$ satisfies the condition $\mathcal{R}$ stating that it is an initial segment which is not closed under both f and f^{-1}.

We wish to algorithmically construct $\mathcal{I}_{t+1}$, a segment of $\mathcal{A}$, satisfying the same condition $\mathcal{R}$. If there is $n \in \mathcal{I}_t$ such that $f_{\mathcal{A}}(n) >_{\mathcal{A}} \mathcal{I}_t$, we set $\mathcal{I}_{t+1}$ as the segment consisting of all elements up to $f_{\mathcal{A}}(n)$. If not, then there must be $m \in \mathcal{I}_t$ such that for some $n >_{\mathcal{A}} \mathcal{I}_t$, $f_{\mathcal{A}}(n) = m$. What is more, for every such m there are only finitely many arguments satisfying this identity and we are able to determine what they are (by looking at their isomorphic images in the standard copy). If M is the largest of these elements, then we set $\mathcal{I}_{t+1}$ as the segment until M. ◀

▶ **Theorem 20.** *There exists a recursive proper quasi-block function f with a non-decreasing lower bound diverging to $+\infty$ but with no such computable bound, with all c.e. degrees as a spectrum.*

Proof. Consider a set $A \subseteq \omega$ which is Δ_2 but not computable. Observe that for each such set there is a recursive sequence g of natural numbers such that each natural number appears in g at most finitely many times and for any $n \in \omega$, $n \in A$ iff the number of occurrences of n in g is odd.

f is going to be g modified in such a way that we put some fixed points between elements of g, pushing these elements to the right, to ensure that f is a quasi-block function. We will be able to easily distinguish (within f) old elements of g from the new filler elements, because only the new elements are going to be fixed points of f.

We construct f by finite extension, starting from the empty function. Initially, all elements of sequence g are unused. At any given stage, suppose that $g(m)$ is the least unused element of sequence g and that n is the least argument such that $f(n)$ is not defined yet. If $g(m) \geq n$, then for each $i = n, \ldots, g(m)$ assign $f(i) = i$. Regardless of whether you performed the previous instruction, assign value $g(m)$ to the least i such that $f(i)$ has no value set yet. We declare that $g(m)$ is used and go to the next stage.

This is a quasi-block function because each argument n is either a fixed point or is a number from sequence g which has been pushed so far to the right that $f(n) < n$. Hence every finite initial segment of ω is closed under f. However, this is not a block function. If it were, then every m such that $f(m) = n$ would need to be in the same block as n. Then we would be able to count how many times n is assumed as the value of f and hence A would be decidable.

The lower bound of f diverges to ∞ because every value can be assumed only finitely often. However, no such bound is computable because otherwise we would be able to determine the last occurrence of every number in g and A would be computable. Observe we can assume that this bound is non-decreasing. We just need to set $f(n) =$ the largest m such that $f(i) \geq m$ whenever $i \geq n$.

If A is a c.e. set, then we utilise the RS module to show that the degree spectrum of f consists of exactly the c.e. degrees. We can assume without loss of generality that g assumes each of its values only once, then so does f if we ignore fixed points.

We take $\mathcal{I}_0$ such that behind it there are no quasi-blocks closed under f^{-1}. The condition $\mathcal{R}$ states that there is an element $n > \mathcal{I}_t$ such that $f(n) \in \mathcal{I}_t$. Observe that such element is determined uniquely. We want to retrieve $\mathcal{I}_{t+1} \supseteq \mathcal{I}_t$ satisfying $\mathcal{R}$. We need to look for n described above and then to fill in all the missing numbers between $\mathcal{I}_t$ and n. Since the segment thus obtained is not a block, it needs to satisfy $\mathcal{R}$. We call this segment $\mathcal{I}_{t+1}$. ◀

3.2 Unusual Degree Spectrum

In this section we answer Wright's question (Question 6.2 in [21]). The result we prove here is also relevant for Harrison-Trainor's question (p. 5 in [10]). Recall a representation of a block function f as an infinite sequence α_f of (the indices of) types (see, Remark 11).

▶ **Definition 21.** *Let f be a computable block function with infinitely many types. The counting function for f is defined by $c_f(n) = \#\{i : \alpha_f(i) = n\}$.*

▶ **Proposition 22.** *Let f be a computable block function with all types pairwise non-embeddable, each occurring finitely often. Then* $\deg(c_f)$ *is c.e. and $f_{\mathcal{A}} \geq_T c_f$ implies that deg($f_{\mathcal{A}}$) is c.e.*

Proof. $C_f^{\leq} := \{(k, n) : k \leq c_f(n)\}$ is c.e., $C_f^{\geq} := \{(k, n) : k \geq c_f(n)\}$ is co-c.e., so $\deg(C_f^{\leq} \oplus C_f^{\geq})$ is c.e. Since $C_f^{\leq} \oplus C_f^{\geq} \equiv_T c_f$, c_f is of c.e. degree.

Assume that $f_{\mathcal{A}} \geq_T c_f$. Then this implies $Succ_{\mathcal{A}} \leq_T f_{\mathcal{A}}$. Indeed, this fact can be illustrated by an example: with the oracle $f_{\mathcal{A}}$, one could recover that the structure $\mathcal{A}$ has, say, precisely two cycles of size 7. Since such a cycle is not embeddable into any other f-block, we could compute the precise positions of the two $f_{\mathcal{A}}$-cycles of size 7 (by looking at the standard copy of $\mathcal{A}$). Suppose that b is the rightmost element of the rightmost $f_{\mathcal{A}}$-cycle of size 7. Using this information, we could recover the values $Succ_{\mathcal{A}}(x)$ for all $x <_{\mathcal{A}} b$. Since all f-types are pairwise non-embeddable, we can "iterate" this process and compute $Succ_{\mathcal{A}}(x)$ for all x.

Note that $Succ_{\mathcal{A}} \geq_T f_{\mathcal{A}}$ always (for a computable f). Hence $f_{\mathcal{A}} \equiv_T Succ_{\mathcal{A}}$, and thus, by Proposition 7, $f_{\mathcal{A}}$ is of c.e. degree. ◀

▶ **Theorem 23.** *There exists a total computable function whose degree spectrum strictly contains all c.e. degrees and is strictly contained in the Δ_2 degrees.*

We construct a computable block function f with infinitely many types and each $c_f(n)$ finite. We want $c_f <_T \mathbf{0}'$ and a computable copy $\mathcal{A}$ of $(\omega, <)$ with $f_{\mathcal{A}}$ of non-c.e. degree. Combining this with Proposition 22 and a result by Cooper, Lempp and Watson from [4] (see Theorem 29) finishes the proof.

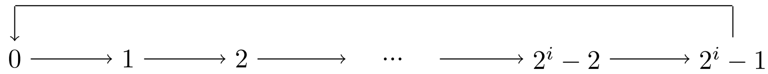

Figure 2 $\mathcal{C}_i = ([0; 2^i - 1], <, f_i)$, where the order $<$ is standard and f_i corresponds to the arrows.

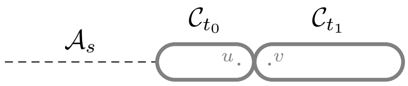

Figure 3 $\mathcal{A}_{s+1}$ after reserving $\langle u, v\rangle$ and tickets t_0, t_1, t_2 for $\mathcal{R}_{\langle e_1, e_2, n\rangle}$.

For each $e, e_1, e_2, n \in \omega$, we have the following requirements:

$$\mathcal{I}_e : I \not\simeq \Phi_e^J, \qquad \mathcal{J}_e : J \not\simeq \Phi_e^I, \qquad \text{and} \qquad R_{\langle e_1,e_2,n\rangle} : \Phi_{e_1}^{\Gamma_{f_\mathcal{A}}} \not\simeq W_n \vee \Phi_{e_2}^{W_n} \not\simeq \Gamma_{f_\mathcal{A}},$$

where $\Gamma_{f_\mathcal{A}}$ is the graph of $f_\mathcal{A}$. J is to make I incomplete while I is going to compute c_f. The non-c.e. degree requirements are based on [6, p. 195] (see, also, [3]).

At stage s we have finite sets I_s, J_s, and a finite structure $\mathcal{A}_s = (A_s, <_{\mathcal{A}_s}, f_{\mathcal{A}_s})$ with $f_{\mathcal{A}_s} : A_s \to A_s$ total. Eventually, we set $\mathcal{A} = \bigcup_{s\in\omega} \mathcal{A}_s$. We assume some recursive ω-type ordering of $\mathcal{I}_e$, $\mathcal{J}_e$, $\mathcal{R}_{\langle e_1,e_2,n\rangle}$, for all $e, e_1, e_2, n \in \omega$. During construction, requirements reserve numbers and, in order to be satisfied, they wait until those numbers meet certain conditions, in which case we say that they need attention.

- $\mathcal{I}_e$ (or $\mathcal{J}_e$) needs attention at stage $s+1$, if some x reserved for it at stage s and $I_s(x) = \Phi_{e,s}^{J_s}$ (or $J_s(x) = \Phi_{e,s}^{I_s}$).
- $R_{\langle e_1,e_2,n\rangle}$ needs attention at stage $s+1$ if, at stage s, some $\langle u, v\rangle$ is reserved for it, along with certain t_0, t_1, t_2 (called tickets), and, for some z, $\langle u, v\rangle < z < s$:

$$(\alpha)\ \Phi_{e_1,s}^{\Gamma_{f_{\mathcal{A}_s}}}[z] = W_{n,s}[z] \qquad \text{and} \qquad (\beta)\ \Phi_{e_2,s}^{W_{n,s}[z]}(\langle u, v\rangle) = \Gamma_{f_{\mathcal{A}_s}}(\langle u, v\rangle).$$

We use a variant of PtR (the proof of Theorem 8). In each application of PtR we distinguish E – the set of fresh numbers – for which we formulate an additional E-condition.

3.2.1 Construction

Let $(\mathcal{C}_i)_{i\in\omega}$ be a computable sequence of cycles, where $\mathcal{C}_i$ is of length 2^i (Figure 2). Put $I_0 = J_0 = \emptyset, \mathcal{A}_0 = (\emptyset, \emptyset, \emptyset)$. Requirements have no reserved numbers, no numbers are frozen. Below we describe stage $s+1$, for $s \in \omega$.

1. If no requirement needs attention at stage $s+1$, we choose the highest priority requirement with no reservation. If this is some $\mathcal{I}_e$ (or $\mathcal{J}_e$), we reserve for it the least fresh number x. If the highest priority requirement with no reservation is some $\mathcal{R}_{\langle e_1,e_2,n\rangle}$, we reserve for it the least number $\langle u, v\rangle$, fresh for $\mathcal{A}_s$ (i.e. u, v do not occur in $\mathcal{A}_s$), and three *consecutive* fresh numbers t_0, t_1, t_2, called tickets. We apply PtR by setting $B = \mathcal{A}_s$, $C = D = \emptyset$ and $E \supseteq \{u, v\}$ such that $|E| = 2^{t_0} + 2^{t_1}$ with every $x \in E$ being fresh for $\mathcal{A}_s$. We build a structure $\mathcal{E} = (E, <_\mathcal{E}, g)$ where $<_\mathcal{E}$ is a linear order satisfying the E-condition, depicted in Figure 3, which is:
   - $C_{t_0} + \mathcal{C}_{t_1} \cong \mathcal{E}$,
   - u is the $<_\mathcal{E}$-last element in the block corresponding to $\mathcal{C}_{t_0}$, and
   - v the $<_\mathcal{E}$-first element in the block corresponding to $\mathcal{C}_{t_1}$.

   We set $\mathcal{A}_{s+1} = \mathcal{A}_s + \mathcal{E}$. We have $\langle u, v\rangle \notin \Gamma_{f_{\mathcal{A}_{s+1}}}$. We enumerate ticket t_0 into I.

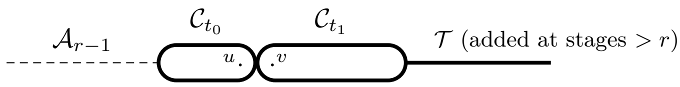

Figure 4 $\mathcal{A}_s$ when $R_{\langle e_1,e_2,n\rangle}$ receives attention for the first time with $\langle u, v\rangle$ and tickets t_0, t_1, t_2, assuming that the reservation has been made at stage r.

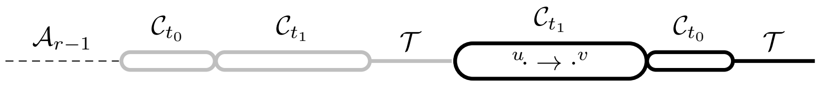

Figure 5 The result of reaction to *first attention* for $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ with reservation $\langle u, v\rangle$ and tickets t_0, t_1, t_2. Gray part is occupied by fresh numbers, thick part represents pushed numbers.

2. If a requirement needs attention, pick the highest one. We say it receives attention. If this is $\mathcal{I}_e$, some x is reserved for $\mathcal{I}_e$ at stage s and $I_s(x) = \Phi^{J_s}_{e,s}(x)$. Put x into I, freeze the computation $\Phi^{J_s}_{e,s}(x)$ and cancel all freezings and reservations for lower priority requirements. Deal with with $\mathcal{J}_e$ accordingly.
Suppose the highest priority requirement needing attention is some $\mathcal{R}_{\langle e_1,e_2,n\rangle}$. Some $\langle u, v\rangle$ is reserved for $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ at stage s with some tickets t_0, t_1, t_2. Below we describe reactions to first and second attention received by $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ with reservation $\langle u, v\rangle, t_0, t_1, t_2$.

  (i) Suppose the reservation for $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ has been made at stage r. After r and before $s + 1$ the structure $\mathcal{A}$ might have been extended by some $\mathcal{T}$ (thick line in Figure 4). The idea is that we push to the right all numbers that occupy the highlighted positions in Figure 4 and obtain the structure as in Figure 5.
  More formally, divide $\mathcal{A}_s$ into $\mathcal{A}_s = \mathcal{B} + \mathcal{C} + \mathcal{D}$, where $\mathcal{B} \cong \mathcal{A}_{r-1}$, $\mathcal{C} \cong \mathcal{C}_{t_0} + \mathcal{C}_{t_1}$ and $\mathcal{D} \cong \mathcal{T}$, and apply PtR. Take $|C \cup D|$ numbers, fresh for $\mathcal{A}_s$, and make F out of them. Build a structure $\mathcal{F} = (F, <_{\mathcal{F}}; g)$, where $<_{\mathcal{F}}$ is a linear order, satisfying the F-condition $\mathcal{F} \cong \mathcal{C} + \mathcal{D}$. We rebuild $\mathcal{C}$ to get $\mathcal{C}' = (C, <_{\mathcal{C}}; h)$ where $\mathcal{C}'$ satisfies the C-condition $\mathcal{C}' \cong \mathcal{C}_{t_1} + \mathcal{C}_{t_0}$. We set $\mathcal{A}_{s+1} = \mathcal{B} + \mathcal{F} + \mathcal{C}' + \mathcal{D}$ (Figure 5).
  Observe that pushed numbers from $\mathcal{C} + \mathcal{D}$ assume in $\mathcal{A}_{s+1}$ the same structure as in $\mathcal{A}_s$ except that the behavior of $f_{\mathcal{A}_{s+1}}$ (on numbers from $\mathcal{C}$) mimics $\mathcal{C}_{t_1} + \mathcal{C}_{t_0}$. This makes $\Gamma_{f_{\mathcal{A}_{s+1}}}(\langle u, v\rangle) = 1$ and thus $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ is satisfied at stage $s + 1$. We enumerate t_1 into I and invalidate all reservations and freezings for lower priority requirements.

  (ii) Suppose $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ has made the reservation at stage r and received the first attention at stage $p + 1$. By the time we got to stage $s + 1$, the structure $\mathcal{A}$ might have been extended by some $\mathcal{U}$ (Figure 6).
  The idea is that we push all numbers occupying the highlighted positions in Figure 6 and obtain the structure as in Figure 7.
  More formally, we divide $\mathcal{A}_s = \mathcal{B} + \mathcal{C} + \mathcal{D}$ in a way that $\mathcal{B} \cong \mathcal{A}_{r-1} + \mathcal{C}_{t_0} + \mathcal{C}_{t_1} + \mathcal{T}$, $\mathcal{C} \cong \mathcal{C}_{t_1} + \mathcal{C}_{t_0}$ and $\mathcal{D} \cong \mathcal{T} + \mathcal{U}$ with u, v residing in a copy of $\mathcal{C}_{t_1}$ within C. We apply PtR with $\mathcal{B}, \mathcal{C}, \mathcal{D}$ defined above. Let F be the set of $|C \cup D|$ numbers,

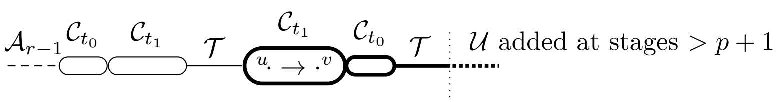

Figure 6 $\mathcal{A}_s$ when $R_{\langle e_1,e_2,n\rangle}$ receives attention for the second time with $\langle u, v\rangle$ and tickets t_0, t_1, t_2, assuming that the reservation has been made at stage r.

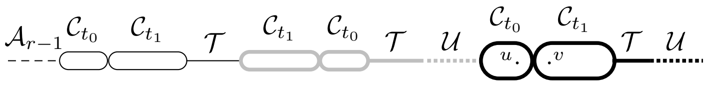

Figure 7 The result of reaction to *second attention* of $\mathcal{R}_{\langle e_1,e_2,n\rangle}$. Gray part is occupied by fresh numbers, thick part represents pushed numbers.

fresh for $\mathcal{A}_s$. We build a finite structure $\mathcal{F} = (F, <_{\mathcal{F}}, g)$, where $<_{\mathcal{F}}$ is a linear order, satisfying the F-condition $\mathcal{F} \cong \mathcal{C} + \mathcal{D}$. We rebuild $\mathcal{C}$ to get $\mathcal{C}' = (C, <_{\mathcal{C}}, h)$ satisfying the C-condition $\mathcal{C}' \cong C_{t_0} + C_{t_1}$. We set $\mathcal{A}_{s+1} = \mathcal{B} + \mathcal{F} + \mathcal{C}' + \mathcal{D}$. We have $\Gamma_{f_{\mathcal{A}_{s+1}}}(\langle u, v\rangle) = 0$. $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ is satisfied at stage $s+1$. We enumerate t_2 into I and invalidate all reservations and freezings for lower priority requirements.

3.2.2 Verification

Due to space constraints, the proof of the following lemma can be found in the full version.

▶ **Lemma 24.** *$\mathcal{A}$ is computable.*

▶ **Lemma 25.** *Every requirement is eventually satisfied. Hence, I, J are intermediate and $f_{\mathcal{A}}$ is of non-c.e. degree.*

Proof. This follows from finite injury. It remains to observe that each requirement can receive attention only finitely many times with the same numbers reserved for it. This is clear for $\mathcal{I}_e, \mathcal{J}_e$ (see, e.g. [20, Chap. VII.2]). We show that no $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ needs attention more than twice with the same $\langle u, v\rangle$ and tickets t_0, t_1, t_2 reserved for it. The PtR modules that we use to satisfy each $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ are carefully arranged to make the standard pattern of verification work (cf. [6, p. 196]). Suppose the reservation was made at stage r, the first attention was at stage $s+1$ and the second at stage $t+1$. Since $\langle u, v\rangle, t_0, t_1, t_2$ are reserved for $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ at stage $t \geq s+1$, no requirement with lower priority than $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ has received attention at any stage u, $t \geq u \geq s+1$. Actions performed at stage $t+1$ lead to $\mathcal{A}_{t+1} \upharpoonright A_s = \mathcal{A}_s \upharpoonright A_s$ (where A_s is the domain of $\mathcal{A}_s$). Therefore, $\Phi_{e_1}^{\Gamma_{f_{\mathcal{A}_{t+1}\upharpoonright A_s}}}[z] = \Phi_{e_1}^{\Gamma_{f_{\mathcal{A}_s\upharpoonright A_s}}}[z] = \Phi_{e_1}^{\Gamma_{f_{\mathcal{A}_s}}}[z] = W_{n,s}[z]$. At stage $s+1$ we had $\Phi_{e_2}^{W_{n,s}[z]}(\langle u, v\rangle) = \Gamma_{f_{\mathcal{A}_s}}(\langle u, v\rangle) \neq \Gamma_{f_{\mathcal{A}_t}}(\langle u, v\rangle)$. Since at stage $t+1$ we had $\Phi_{e_2}^{W_{n,t}[z]}(\langle u, v\rangle) = \Gamma_{f_{\mathcal{A}_t}}(\langle u, v\rangle)$ we must have $W_{n,t}[z] \neq W_{n,s}[z]$. Hence, for some x, $\Phi_{e_1}^{\Gamma_{f_{\mathcal{A}_{t+1}\upharpoonright A_s}}}(x) = W_{n,s}(x) \neq W_{n,t}(x)$. Now, observe that $\mathcal{A}_{t+1} \upharpoonright A_s$ does not change at any later stage at which $\langle u, v\rangle$ is reserved for $\mathcal{R}_{\langle e_1,e_2,n\rangle}$. Hence, for all such stages $w \geq t+1$, $\Phi_{e_1,w}^{\Gamma_{f_{\mathcal{A}_w\upharpoonright A_s}}}(w) \neq W_{n,w}(x)$ and $\mathcal{R}_{\langle e_1,e_2,n\rangle}$ does not need attention at stage $w+1$. ◀

▶ **Lemma 26.** *For every $n \in \omega$, $c_f(n)$ is finite and is never increased due to numbers $> n+2$ entering I.*

Proof. Suppose the contrary. Then there exists n such that $c_f(n)$ is increased because of some $k > n+2$ entering I. Let $s+1$ be the stage at which this happens. Since $c_f(n)$ is increased at stage $s+1$, $\mathcal{C}_n$ is present in $\mathcal{A}_{s+1}$. Since $c_f(n)$ is increased due to k entering I, k must be associated at stage $s+1$ with some R_i. Hence, k is one of the tickets t_0, t_1, t_2 paired with R_i at this point. There are three cases.

$(k = t_0)$ This is when R_i is initialized and receives tickets t_0, t_1, t_2 (see Figure 3). For $c_f(n)$ to increase, we must have $n = t_0$ or $n = t_1$. $n = t_0$ is not possible because then we would have $k = t_0 = n$ which contradicts $k > n+2$. $n = t_1$ is also not possible because we would have $k = t_0 = t_1 - 1 = n - 1$ which contradicts $k > n+2$.

($k = t_1$) This is when R_i receives first attention with tickets t_0, t_1, t_2 (see Figure 4). $\mathcal{C}_n$ must occur somewhere at the highlighted positions in Figure 4 because this fragment of the structure is copied leading to an increase of c_f. Hence, $n = t_0$ or $n = t_1$, or $\mathcal{C}_n$ occurs in $\mathcal{T}$. $n \neq t_0$ because otherwise $n = t_0$, $k = t_1 = t_0 + 1 = n + 1$ which contradicts $k > n + 2$. n cannot be t_1 because otherwise $n = t_1 = k$ which contradicts $k > n + 2$. Hence, $\mathcal{C}_n$ occurs in $\mathcal{T}$. However, this is also not possible for the following reason. We know that $k = t_1$ enters I so this is due to R_i acting when receiving the first attention with tickets t_0, t_1, t_2. This means that no requirement R_j of higher priority than R_i (i.e., with $j < i$) has received attention after R_i got associated with tickets t_0, t_1, t_2 (up to the current stage) – otherwise R_i's tickets would have been reassigned to numbers different than t_0, t_1, t_2. Therefore, $\mathcal{C}_n$ entered the construction *after* R_i was assigned to t_0, t_1, t_2. Hence, by the construction (i.e. the way we choose and assign tickets to requirements (re)entering the construction), n is a ticket for some lower priority requirement R_l ($l > i$). But when n enters the construction as a ticket of such R_l, n is chosen as a fresh number so, in particular, $n > t_1 = k$ which contradicts $k > n + 2$.

($k = t_2$) This is when $\mathcal{R}_i$ receives attention for the second time with tickets t_0, t_1, t_2 (see Figure 6). $\mathcal{C}_n$ occurs somewhere at the highlighted positions in Figure 6, i.e. $n = t_0$ or $n = t_1$, or $\mathcal{C}_n$ occurs in $\mathcal{T} + \mathcal{U}$. $n \neq t_0$ because otherwise $n = t_0$, $k = t_2 = t_0 + 2 = n + 2$ which contradicts $k > n + 2$. $n \neq t_1$ because otherwise $n = t_1$, $k = t_2 = t_1 + 1 = n + 1$ which contradicts $k > n + 2$. Therefore, n occurs in $\mathcal{T} + \mathcal{U}$. The rest of the argument is similar to the analogical place of the previous case ($k = t_1$). ◀

▶ **Lemma 27.** $c_f \leq_T I$.

Proof. To compute $c_f(n)$, find s such that $I_s[n+2] = I[n+2]$. By Lemma 26 and the fact that $c_f(n)$ is increased *only* due to numbers entering I, $c_f(n)$ is not increased at stages $> s$ (no additional copies of $\mathcal{C}_n$ are added to $f_{\mathcal{A}}$). Return the number of copies of $\mathcal{C}_n$ in $f_{\mathcal{A}_s}$. ◀

Due to space constraints, the proof of the following lemma can be found in the full version.

▶ **Lemma 28.** $f_{\mathcal{A}} \leq_T c_f$.

By Lemmas 25, 26, 27 and 28: $\mathbf{0} <_T f_{\mathcal{A}} \leq c_f \leq_T I <_T \mathbf{0}'$. The spectrum of f is not trivial by Proposition 6. By Theorem 1, $DgSp(f)$ contains all c.e. degrees. Since $f_{\mathcal{A}}$ is of non-c.e. degree, $DgSp(f) \neq$ the c.e. degrees. To show that $DgSp(f) \neq$ the Δ_2 degrees, we need the following theorem.

▶ **Theorem 29** (Cooper, Lempp and Watson, [4])**.** *Given c.e. sets $U <_T V$ there is a proper d.c.e. set C of properly d.c.e. degree such that $U <_T C <_T V$.*

Assume, for a contradiction, that $DgSp(f)$ consists of the Δ_2 degrees. By Theorem 29, $DgSp(f) \cap \{\deg(A) : c_f \leq_T A \leq_T \mathbf{0}'\}$ contains a proper d.c.e. degree. However, by Proposition 22, $DgSp(f) \cap \{\deg(A) : c_f \leq_T A \leq_T \mathbf{0}'\}$ contains only c.e. degrees. This is a contradiction, so the degree spectrum of f is different then the Δ_2 degrees. This completes the proof.

4 Conclusions and Open Questions

We have investigated the problem of intrinsic complexity of computable relations on $(\omega, <)$, as measured by their degree spectra, in the restricted setting of graphs of unary total computable functions. It has been known that possible candidates for intrinsic complexities of such functions include three sets consisting of precisely: the computable degree, all c.e.

degrees, and all Δ_2 degrees. Imposing certain structural constraints on such functions has led us to the notions of block functions (Definition 10) and a broader class of quasi-block functions (Definition 15). Non-quasi-block functions have intrinsic complexity equal to the c.e. degrees (Theorem 18) which redirects all focus to quasi-block functions. We have obtained several results on this class, most prominently the one on block-functions with finitely many types (Theorem 14) showing that their intrinsic complexity is either trivial or equal to the Δ_2 degrees. However, the most surprising result is that on an unusual degree spectrum (Theorem 23) which proves the existence of a block function having intrinsic complexity different from the already known three candidates. To the best of our knowledge, this theorem answers Question 6.2 from [21] formulated by Wright who asked whether there are relations on $(\omega, <)$ with other degree spectra (than the three known candidates). Harrison-Trainor obtained a related result though for a different relation. However, for his relation it is not known whether its spectrum is intermediate (see Section 1 for details, as well as [10]).

A few questions arise immediately. Note, for example, that our solution to Wright's question invites the hypothesis, possibly to be proven using some kind of permitting, that there exist infinitely many spectra of computable block functions on $(\omega, <)$. A parallel question is what degrees such nonstandard spectra contain. Observe that even for the function constructed in Theorem 23 the exact contents of its spectrum are unknown. We finish the paper with the general open question: what are the possible kinds of nonstandard spectra of computable block functions on $(\omega, <)$?

References

1. Chris J. Ash and Julia Knight. *Computable structures and the hyperarithmetical hierarchy*, volume 144 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, Amsterdam, 2000.
2. Jennifer Chubb, Andrey Frolov, and Valentina Harizanov. Degree spectra of the successor relation of computable linear orderings. *Archive for Mathematical Logic*, 48(1):7–13, 2009. `doi:10.1007/s00153-008-0110-6`.
3. S. Barry Cooper. *Degrees of unsolvability*. PhD thesis, University of Leicester, 1971.
4. S. Barry Cooper, Steffen Lempp, and Philip Watson. Weak density and cupping in the d-r.e. degrees. *Israel Journal of Mathematics*, 67(2):137–152, 1989. `doi:10.1007/BF02937291`.
5. Rod Downey, Bakhadyr Khoussainov, Joseph S. Miller, and Liang Yu. Degree spectra of unary relations on $(\omega, \leq)$. In *Logic, Methodology and Philosophy of Science: Proceedings of the Thirteenth International Congress*, pages 35–55. College Publications, 2009. Publisher: Citeseer. URL: `http://homepages.mcs.vuw.ac.nz/~downey/publications/LOJan24.pdf`.
6. Richard L. Epstein. *Degrees of Unsolvability: Structure and Theory*, volume 759 of *Lecture Notes in Mathematics*. Springer, Berlin Heidelberg, 1979.
7. Ekaterina B. Fokina, Valentina Harizanov, and Alexander Melnikov. Computable model theory. In R. Downey, editor, *Turing's legacy: Developments from Turing's ideas in logic*, volume 42 of *Lecture Notes in Logic*, pages 124–194. Cambridge University Press, Cambridge, 2014. `doi:10.1017/CBO9781107338579.006`.
8. E. Mark Gold. Limiting Recursion. *Journal of Symbolic Logic*, 30(1):28–48, 1965. `doi:10.2307/2270580`.
9. Valentina S. Harizanov. *Degree spectrum of a recursive relation on a recursive structure*. PhD thesis, University of Wisconsin-Madinson, 1987.
10. Matthew Harrison-Trainor. Degree spectra of relations on a cone. *Memoirs of the American Mathematical Society*, 253(1208):1–120, 2018. `doi:10.1090/memo/1208`.
11. Denis R. Hirschfeldt. Degree spectra of relations on computable structures. *Bulletin of Symbolic Logic*, 6(2):197–212, 2000. `doi:10.2307/421207`.

12 Carolyn Alexis Knoll. Degree spectra of unary relations on ω and ζ. Master's thesis, University of Waterloo, 2009.

13 Dragoslav S. Mitrinović, József Sándor, and Borislav Crstici. *Handbook of Number Theory*, volume 351 of *Mathematics and Its Applications*. Kluwer, 1995.

14 Antonio Montalbán. *Computable structure theory: Within the arithmetic*. Cambridge University Press, 2021.

15 Michael Moses. Relations Intrinsically Recursive in Linear Orders. *Mathematical Logic Quarterly*, 32(25-30):467–472, 1986. doi:10.1002/malq.19860322514.

16 Hilary Putnam. Trial and Error Predicates and the Solution to a Problem of Mostowski. *Journal of Symbolic Logic*, 30(1):49–57, 1965. doi:10.2307/2270581.

17 Linda Jean Richter. Degrees of structures. *Journal of Symbolic Logic*, 46(4):723–731, 1981. Publisher: Cambridge University Press. doi:10.2307/2273222.

18 Stewart Shapiro. Acceptable notation. *Notre Dame Journal of Formal Logic*, 23(1):14–20, January 1982. doi:10.1305/ndjfl/1093883561.

19 Joseph R. Shoenfield. On degrees of unsolvability. *Annals of Mathematics*, 69:644–653, 1959. doi:10.2307/1970028.

20 Robert I. Soare. *Recursively Enumerable Sets and Degrees*. Springer-Verlag, New York, NY, USA, 1987.

21 Matthew Wright. Degrees of relations on ordinals. *Computability*, 7(4):349–365, 2018. doi:10.3233/COM-180086.

A Proof of Theorem 14

▶ **Theorem 14.** *Let f be a computable block function such that it has only finitely many f-types and f is not almost identity. Then the spectrum $DgSp(f)$ consists of all Δ_2 degrees.*

Proof. Let $I_0, I_1, \dots, I_N$ be all (isomorphism types of) f-blocks. We represent the structure $\mathcal{B} = (\omega, <, f)$ by α_f according to Remark 11. As in Theorem 8, we fix a Δ_2 set X. Our goal is to construct a computable copy $\mathcal{A} = (\omega, <_{\mathcal{A}})$ of $(\omega, <)$ such that $f_{\mathcal{A}} \equiv_T X$. In general, we follow the notations of Theorem 8 (e.g., $f_{\mathcal{A}_s}(x)$ is defined in the same way as in the previous proof).

Beforehand, we choose a large enough number M such that:

- M lies at the right end of its f-block (inside $\mathcal{B}$),
- for every $x > M$, the isomorphism type of its f-block occurs infinitely often in $\mathcal{B}$.

As in the proof of Theorem 8, we copy the interval $[0; M]$ into our structure $\mathcal{A}$, and all new elements will be added to the right of this interval.

The proof is split into three cases which depend on the properties of the string α_f (each of the cases requires a separate construction):

(a) There are two different finite strings σ and τ such that:
  - the lengths of σ and τ are the same;
  - τ can be obtained via a permutation of σ, i.e., there is a permutation h of the set $\{0, 1, \dots, |\sigma| - 1\}$ such that $\tau(i) = \sigma(h(i))$, for all $i < |\sigma|$;
  - both σ and τ occur infinitely often in α_f.

(b) There is only one block I_k such that k occurs infinitely often in α_f.

(c) Neither of the previous two cases holds.

Case (a). For the sake of simplicity, we give a detailed proof for the case when $\sigma = 01$ and $\tau = 10$. After that, we explain how to deal with the general case.

Our construction satisfies the following requirements:

$$\begin{aligned} e \in X &\Leftrightarrow 2e \text{ belongs to a block isomorphic to } I_1, \\ e \notin X &\Leftrightarrow 2e \text{ belongs to a block isomorphic to } I_0. \end{aligned} \tag{$\mathcal{R}_e$}$$

Suppose that $|I_0| + |I_1| = q + 1$.

Strategy $\mathcal{R}_e$ in isolation. When $\mathcal{R}_e$ starts working at a stage $s_0 + 1$, we proceed as follows. Assume that $X_{s_0}(e) = 1$ (the other case is treated similarly). We choose q fresh odd numbers $c^e_1, c^e_2, \dots, c^e_q$ and declare them the *companions* of $2e$. We add the chain

$$2e <_{\mathcal{A}} c^e_1 <_{\mathcal{A}} c^e_2 <_{\mathcal{A}} \dots <_{\mathcal{A}} c^e_q$$

to the right of $\mathcal{A}_{s_0}$. If needed, we add finitely many fresh odd numbers in-between $\mathcal{A}_{s_0}$ and $2e$. This procedure ensures that (at the moment) the finite structure $([2e; c^e_q]_{\mathcal{A}}, <_{\mathcal{A}}, f_{\mathcal{A}})$ is isomorphic to the disjoint sum $I_1 + I_0$.

The strategy $\mathcal{R}_e$ *requires attention* at a stage s if inside the current $\mathcal{A}_s$, the number $2e$ belongs to a copy of $I_{1-X_s(e)}$. When $\mathcal{R}_e$ requires attention, we apply a PtR-module.

The PtR-module for $\mathcal{R}_e$. We choose the middle interval C as the set containing $2e$ and all its companions, i.e. $C = \{2e <_{\mathcal{A}} c^e_1 <_{\mathcal{A}} \dots <_{\mathcal{A}} c^e_q\}$. Our target condition is defined as follows: inside the resulting structure $\mathcal{F}$, the structure $(C, <_{\mathcal{F}}, f_{\mathcal{F}} \restriction C)$ is isomorphic to the disjoint sum $I_{X_s(e)} + I_{1-X_s(e)}$. As in Theorem 8, the structure $\mathcal{F}$ is treated as output of the module.

The construction is arranged similarly to that of Theorem 8.

Verification. We need to show that every application of a PtR-module is successful. This follows from two observations:

1. If we want to "transform", say, $I_0 + I_1$ into $I_1 + I_0$, then this can be achieved by an appropriate pushing to the right, since the string $\tau = 10$ occurs infinitely often in α_f.
2. Remark 9 guarantees that one can also safely push the interval D (from the PtR-module): notice that if some block I_r occurs in D, then r occurs infinitely often in α_f.

Since pushing to the right is always successful, every requirement $\mathcal{R}_e$ is satisfied. Note that given $f_{\mathcal{A}}$ as an oracle, one can recover the $f_{\mathcal{A}}$-block of $2e$. This fact (together with $\mathcal{R}_e$-requirements) implies that $X \leq_T f_{\mathcal{A}}$.

Every element $a \in \mathcal{A}$ is pushed to the right only finitely often. Therefore, the structure $\mathcal{A}$ is a computable copy of $(\omega, <)$.

Given an odd number $x = 2k + 1$, one can computably determine which of the following two cases holds:

1. $2k + 1$ is a companion c^e_t of some even number $2e$ (in this case, the indices e and t are also computed effectively), or
2. $2k + 1$ is added as a "filler" by some action of an $\mathcal{R}_e$-strategy (either by its initial actions, or by an application of a PtR-module).

In the second case, the value $f_{\mathcal{A}_s}(x)$ never changes (after being defined for the first time). In the first case, the oracle X can tell us whether $x = c^e_t$ belongs to (a copy of) I_0 or I_1, and X can also compute the image $f_{\mathcal{A}}(x)$. In a similar way, X computes the images $f_{\mathcal{A}}(2e)$, for $e \in \omega$. Hence, we obtain that $f_{\mathcal{A}} \equiv_T X$. This concludes the case when $\sigma = 01$ and $\tau = 10$.

The case of arbitrary σ and τ follows a similar proof outline. We illustrate this by considering $\sigma = 012301$ and $\tau = 013021$. Then our construction will switch between finite structures

$$\mathcal{F}_\sigma = I_0 + I_1 + I_2 + I_3 + I_0 + I_1 \text{ and } \mathcal{F}_\tau = I_0 + I_1 + I_3 + I_0 + I_2 + I_1.$$

Since both σ and τ occur infinitely often in α_f, an appropriate PtR-module can always "transform" $\mathcal{F}_\sigma$ into $\mathcal{F}_\tau$, and vice versa.

During the construction, an even number $2e$ will always belong to the third block from the left inside $\mathcal{F}_\square$ (i.e., either I_2 in $\mathcal{F}_\sigma$, or I_3 in $\mathcal{F}_\tau$). The third block is chosen because it corresponds to the first position, where σ and τ differ. The rest of the corresponding copy of $\mathcal{F}_\square$ consists of companions of $2e$. In the final structure $\mathcal{A}$, we will achieve the following: if $e \in X$, then $2e$ lies in a copy of I_2; otherwise, $2e$ belongs to a copy of I_3. This concludes the discussion of Case (a).

Case (b). Without loss of generality, we assume that $I_k = I_0$. We satisfy the following requirements:

$$\begin{aligned} e \in X \;&\Leftrightarrow\; 2e \text{ lies at the right end of a copy of } I_0, \\ e \notin X \;&\Leftrightarrow\; 2e \text{ lies at the left end of a copy of } I_0. \end{aligned} \tag{$\mathcal{R}_e$}$$

Suppose that $|I_0| = q + 1$. Notice that $q \geq 1$, since f is not almost identity.

Strategy $\mathcal{R}_e$ in isolation. $2e$ will have finitely many odd numbers as its *companions*. In contrast to Case (a), these companions could be added stage-by-stage.

When $\mathcal{R}_e$ starts working at a stage $s_0 + 1$, we proceed as follows. Suppose $X_{s_0}(e) = 1$ (the other case is similar). Then we choose q fresh odd numbers $c_1, \dots, c_q$, and declare that they are companions of $2e$. We set $c_1 <_\mathcal{A} \dots <_\mathcal{A} c_q <_\mathcal{A} 2e$ (these elements are added to the right of $\mathcal{A}_{s_0}$). We ensure that the structure $([c_1; 2e]_\mathcal{A}, <_\mathcal{A}, f_\mathcal{A})$ is isomorphic to I_0 (if needed, one adds fresh odd numbers in-between $\mathcal{A}_{s_0}$ and c_1).

We also ensure that by the end of each stage s, $2e$ and its (current) companions form an interval inside $\mathcal{A}_s$, and this interval can be treated as a sum of blocks (in $\mathcal{A}_s$).

The strategy $\mathcal{R}_e$ *requires attention* at a stage s if inside the current $\mathcal{A}_s$, the corresponding requirement is not satisfied (e.g., if $X_s(e) = 0$ and $2e$ lies at the right end of I_0). When $\mathcal{R}_e$ requires attention, we apply a PtR-module.

The PtR-module for $\mathcal{R}_e$. We choose the middle interval C as the set containing $2e$ and all its current companions. We consider the following two subcases.

Subcase 1. Assume that right now, $X_s(e) = 1$ and $2e$ lies at the left end of a copy of I_0. Then our target condition is defined as follows: inside the resulting output structure $\mathcal{F}$, the number $2e$ should belong to the right end of a copy of I_0.
In order to achieve this condition, we add precisely q fresh odd numbers in-between B and C, and only one fresh odd number in-between C and D. This guarantees that $2e$ "moves" to the right end of a block.

Subcase 2. Otherwise, suppose that $X_s(e) = 0$ and $2e$ lies at the right end of a copy of I_0. Then we pursue the following condition: inside the output $\mathcal{F}$, $2e$ should "move" to the left end of a block I_0.
In order to do this, we add one fresh number in-between B and C, and q fresh numbers in-between C and D.
In both subcases, we declare that the newly added odd numbers belong to the set of companions of $2e$.

The construction is arranged similarly to the previous ones.

Verification. Since almost every block from α_f is isomorphic to I_0, every application of a PtR-module is successful. In addition, the actions of the PtR-module for $\mathcal{R}_e$ does not injure other strategies.

We deduce that all requirements $\mathcal{R}_e$ are satisfied. Given $f_{\mathcal{A}}$ as an oracle, one can recover the position of $2e$ inside its $f_{\mathcal{A}}$-block. This implies that $X \leq_T f_{\mathcal{A}}$. In addition, a standard argument shows that $\mathcal{A}$ is a computable copy of $(\omega, <)$.

Notice the following. Since $2e$ and its companions always stick together as an interval, there are only two possible variants of the final $f_{\mathcal{A}}$-block of $2e$: either it contains q companions of $2e$ added at the very beginning of the work of the $\mathcal{R}_e$-strategy, or it contains q closest (inside $\mathcal{A}$) companions of $2e$ added by the first application of the PtR-module for $\mathcal{R}_e$.

As in the previous case, given an odd number $x = 2k+1$, one can determine which of the following two cases holds:

1. x is a companion of some even number $2e$ (the index e is recovered effectively), or
2. x is added as a "filler" by some action of an $\mathcal{R}_e$-strategy.

In the second case, the value $f_{\mathcal{A}_s}(x)$ never changes. In the first case, the oracle X can tell us the content of the final $f_{\mathcal{A}}$-block containing x: indeed, if $X_{s_0}(e) = X_{s_1}(e)$, then at the stages s_0 and s_1, the blocks of x inside $\mathcal{A}_{s_0}$ and $\mathcal{A}_{s_1}$ contain precisely the same elements. We deduce that $f_{\mathcal{A}} \leq_T X$. This concludes the proof of Case (b).

Case (c). Before describing the construction, we provide a combinatorial analysis of the string α_f.

▶ **Lemma 30.** *If the string α_f satisfies neither Case (a) nor Case (b), then there are symbols $b, d, e \in \Sigma$ such that $d \neq b$, $e \neq b$, and for every natural number n, there exists $m > n$ such that the finite string $db^m e$ occurs in α_f.*

Proof. Without loss of generality, one may assume that every symbol from Σ occurs infinitely often in α_f.

For a finite string σ over the alphabet Σ, we denote

$$\#(\sigma) = |\{a \in \Sigma : a \text{ occurs in } \sigma\}|.$$

We choose a finite string τ such that τ occurs infinitely often in α_f and

$$\#(\tau) = \max\{\#(\sigma) : \sigma \text{ occurs infinitely often in } \alpha_f\}. \tag{1}$$

Let c be the last symbol of the string τ.

There exists a symbol b such that the string $\tau_b = \tau\, b$ occurs infinitely often in α_f. Equation (1) implies that b occurs in τ (indeed, if b does not occur in τ, then $\#(\tau_b) = \#(\tau)+1$).

We prove that $c = b$. Towards a contradiction, assume that $c \neq b$. Then τ can be decomposed as $\tau = \xi\, b\, \delta\, c^k$ for some $k \geq 1$ and finite strings ξ, δ. The string $\tau_b = \xi\, b\, \delta\, c^k\, b$ occurs infinitely often in α_f. In turn, this implies that both $b\, \delta\, c^k$ and $\delta\, c^k\, b$ occur infinitely often in α_f. Therefore, α_f satisfies Case (a), which gives a contradiction.

Hence, we have $\tau = \rho\, b^k$ for some $k \geq 1$ and finite string ρ, and the string $\tau_b = \rho\, b^{k+1}$ occurs infinitely often in α_f. Note that $\#(\tau_b) = \#(\tau)$. This implies that by applying induction, one can show that for *every* $l \geq 1$,

$$\rho\, b^l \text{ occurs infinitely often in } \alpha_f. \tag{2}$$

Since α_f does not satisfy Case (b), there are at least two different symbols occuring infinitely often in α_f. This fact and Equation (2) imply that for every $n \in \omega$, there exist $m > n$ and two symbols d' and e' such that $d' \neq b$, $e' \neq b$, and $d' b^m e'$ occurs in α_f. After that, we apply the pigeonhole principle to finish the proof of the lemma. ◀

By Lemma 30, we may assume that for every $n \in \omega$, there exists $m > n$ such that, say, 10^m2 occurs in α_f. We satisfy the same requirements as in Case (b):

$$\begin{aligned} e \in X &\Leftrightarrow 2e \text{ lies at the right end of a copy of } I_0, \\ e \notin X &\Leftrightarrow 2e \text{ lies at the left end of a copy of } I_0. \end{aligned} \qquad (\mathcal{R}_e)$$

In general, our notations also follow those of Case (b).

Strategy $\mathcal{R}_e$ in isolation. When $\mathcal{R}_e$ starts working at a stage $s_0 + 1$, we proceed as follows. Suppose $X_{s_0}(e) = 0$. We find a large enough number m such that 10^m2 occurs in α_f, and the corresponding sequence of f-blocks $I_1 + I_0 + I_0 + \ldots + I_0 + I_2$ does not intersect with the image of $\mathcal{A}_{s_0}$ inside $(\omega, <)$.

We add $2e$ and fresh odd numbers into $\mathcal{A}$ ensuring that the newly added elements form a sequence of $f_{\mathcal{A}}$-blocks:

$$I_1 + \underbrace{I_0 + \ldots + I_0}_{m \text{ times}} + I_2;$$

if needed, fresh odd numbers are also added in-between $\mathcal{A}_{s_0}$ and this sequence. The number $2e$ lies at the left end of the leftmost block I_0. The elements forming I_1 and I_2 are declared *boundary companions* of $2e$. The odd numbers forming the inner sequence of I_0-s are declared *non-boundary companions* of $2e$.

As usual, $\mathcal{R}_e$ *requires attention* at a stage s if inside the current $\mathcal{A}_s$, the corresponding requirement is not satisfied. When $\mathcal{R}_e$ requires attention, we apply a PtR-module.

The PtR-module for $\mathcal{R}_e$. We choose the middle interval C as the set containing $2e$ and all its companions. Assume that right now, $X_s(e) = 0$ and $2e$ lies at the right end of a copy of I_0 (the other subcase is treated in a similar way). Then the target condition is defined as follows: inside $\mathcal{F}$, the number $2e$ belongs to the left end of a copy of I_0.

Suppose that right now, the companions of $2e$ form a sequence of $f_{\mathcal{A}_s}$-blocks corresponding to a finite string 10^m2.

We always assume the following: if a fresh number x is added between some companions of some $2j$, then it is declared a non-boundary companion of $2j$. In addition, every such x is put between the I_1-block and the I_2-block containing the boundary companions of $2j$. Moreover, we require that inside the resulting structure $\mathcal{F}$, the element x becomes a part of a copy of I_0.

In order to achieve the target condition, we proceed as follows. First, we find a large enough $m' > m$ such that $10^{m'}2$ occurs in α_f, and this occurrence of $10^{m'}2$ lies to the right of the image of $\mathcal{A}_s$ inside $(\omega, <)$. We add fresh odd numbers in such a way that:

- The companions of $2e$ (including newly added companions) form a sequence of $f_{\mathcal{F}}$-blocks corresponding to $10^{m'}2$ (inside α_f). This is achieved by adding numbers in-between B and C, and by adding fresh I_0-blocks between the I_1-block and the I_2-block containing the boundary companions of $2e$.
- Similarly to Case (b), this procedure must ensure that $2e$ moves to the left end of an I_0-block.

Second, we carefully push the companions of $2j$, where $e < j < s$, to the right. Consider each such j (in turn). Suppose that the companions of $2j$ form a sequence of $f_{\mathcal{A}_s}$-blocks corresponding to a finite string $10^{m_j}2$. We choose a large enough $m'_j > m_j$ (again, with $10^{m'_j}2$ occuring in α_f to the right of the image of the current (preliminary) version of $\mathcal{F}$). We add fresh numbers in such a way that:

- The companions of $2j$ (including new ones) form a sequence of $f_{\mathcal{F}}$-blocks corresponding to $10^{m'_j}2$ inside α_f.
- If x is a new companion of $2j$, then it belongs to a new I_0-block which corresponds to one of the underlined zeros in the following decomposition:

$$10^{m'_j}2 = 10^{m_j}\underline{0}\underline{0}\dots\underline{0}2.$$

This careful pushing allows to ensure that the PtR-module does not injure strategies $\mathcal{R}_j$, for $j \neq e$. Indeed, after the pushing, the value $f_{\mathcal{A}}(2j)$ does not change.

The construction is arranged in a similar way as before.

Verification. The fact that α_f contains occurrences of 10^m2 for arbitrarily large m implies that every application of a PtR-module is successful. We deduce that all requirements $\mathcal{R}_e$ are satisfied. The rest of the verification is similar to that of Case (b). This concludes the proof of Theorem 14. ◀