Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Archive for Mathematical Logic (2025) 64:819–841
https://doi.org/10.1007/s00153-024-00958-x

Mathematical Logic

Constructivity conditions on immune sets

John Case[1]

Received: 3 June 2023 / Accepted: 17 December 2024 / Published online: 29 January 2025
© The Author(s) 2025

Abstract

Definitionally: *strongly effectively immune* sets are infinite and their c.e. subsets have *maximums* effectively bounded in their c.e. indices; whereas, for *effectively immune* sets, their c.e. subsets' *cardinalities* are what're effectively bounded. This definitional difference between these two kinds of sets is very nicely paralleled by the following difference between their *complements*. McLaughlin: *strongly* effectively immune sets can*not* have *immune complements*; whereas, the main theorem herein: *effectively* immune sets can*not* have *hyperimmune complements*. Ullian: *effectively* immune sets *can* have *effectively* immune complements. The main theorem *improves* Arslanov's, effectively hyperimmune sets can*not* have *effectively* hyperimmune complements: the *improvement* omits the second '*effectively*'. Two *natural* examples of *strongly effectively immune* sets are presented with new cases of the first proved herein. The first is the set of minimal-Blum-size programs for the partial computable functions; the second, the set of Kolmogorov-random strings. A proved, *natural* example is presented of an *effectively dense simple*, *not* strongly effectively simple set; its complement is a set of maximal run-times. Further motivations for this study are presented. *Kleene* recursion theorem proofs herein emphasize how to conceptualize them. Finally, is suggested, future, related work—illustrated by a first, *natural*, *strongly effectively* Σ_2^0*-immune set*—included: solution of an open problem from Rogers' book.

Keywords Immunities in computability · Constructivities

Mathematics Subject Classification 03D99 · 03F99

Contents

1 Introduction 820
2 Author's motivations 825
3 Results 827
3.1 Constructivity for immunity 827

✉ John Case
case@udel.edu

[1] Professor Emeritus, CIS, University of Delaware, 700 W. Harbor Drive, Unit 505, San Diego, CA 92101, USA

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

3.1.1 Three theorems . . . 827
3.1.2 Two natural examples . . . 829
3.2 Constructivity for hyperimmunity and dense immunity . . . 834
4 Suggested future, related work . . . 836
References . . . 839

1 Introduction

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$, the set of natural numbers [1]. We use A to range over subsets of $\mathbb{N}$, and $i, j, k, n, p, q, t, x, y, z$ with or without decorations to range over $\mathbb{N}$. We use $\overline{A}$ to denote the complement of A with respect to $\mathbb{N}$.

We use *c.e.* herein to abbreviate *computably enumerable*, a.k.a., recursively enumerable [1].

We consider the *acceptable programming systems* (synonym: *acceptable numberings*) [1–6] to be those programming systems for the 1-argument partial computable (synonym: partial recursive) functions: $\mathbb{N} \to \mathbb{N}$ which are intercompilable with naturally occurring general purpose programming formalisms such as a full Turing Machine formalism or a LISP programming language. Rogers [1, 2] characterized the acceptable systems as those with universality and S-m-n. The present author in [6] characterized the acceptable systems as those with universality and an implementation of every control structure. Typically, in computability theory, one works with *numerical* names for the programs in these systems—whence the term 'numbering'. Let $\varphi = \lambda p, x.(\varphi_p(x))$ be any fixed acceptable programming system. φ_p denotes the 1-argument partial computable function: $\mathbb{N} \to \mathbb{N}$ computed by program (number) p in the φ-system. $\delta(\cdot)$ maps any partial function into its domain. We write W_p for $\delta(\varphi_p)$, that is, the c.e. set accepted by φ-program p. Since such a p can easily be algorithmically transformed into a program p' for *enumerating* W_p, we sometimes think of the p in W_p as also enumerating W_p.

Total partial computable functions are called *computable functions*, a.k.a. recursive functions. Re computations, $\downarrow$ means converges or is defined, and $\uparrow$ means diverges or is undefined. We let $\max(\emptyset) = 0$. $\mathrm{card}(\cdot)$ denotes the cardinality function on sets. We write $\equiv$ between sets to denote they are *recursively isomorphic* [1, P. 52]—we'll call $\equiv$, *computably isomorphic*. Herein most unexplained notation or terminology,[1] is from [1, 7, 8] or some other reference given on the spot.

Definition 1.1 A is *immune*[2] iff [A is infinite $\&(\forall x)[W_x \subseteq A \Rightarrow W_x$ is finite]].

Immune sets additionally satisfying constructivity conditions have been studied, and herein we *begin* with

[1] For *example* this will occur in some cases where the present paper does not *directly* prove any results featuring such.

[2] This concept is implicitly introduced in Post's seminal [9] &named by Dekker.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Definition 1.2

1. A is *constructively immune*[3] iff [A is infinite &there exists a partial computable function ψ such that $(\forall x)[W_x$ is infinite $\Rightarrow [\psi(x)\downarrow$ to a counterexample to $W_x \subseteq A$, that is, to a member of $(W_x \cap \overline{A})]]]$.
2. A is *strongly effectively immune*[4] iff [A is infinite &there is a partial computable function ψ such that $(\forall x)[W_x \subseteq A \Rightarrow \max(W_x) \leq \psi(x)\downarrow]]$.
3. A is *effectively immune*[5] iff [A is infinite &there is a partial computable function ψ such that $(\forall x)[W_x \subseteq A \Rightarrow \text{card}(W_x) \leq \psi(x)\downarrow]]$.
4. A is X *simple*, where X is the empty string or one of constructively, strongly effectively, or effectively, iff [A is c.e. &its complement is X immune].

As noted in [10] (see also [7]), if $\overline{A}$ is simple, then A is constructively immune: given x, search for the first, if any, y found in a computable enumeration of $(W_x \cap \overline{A})$, then, if such y is found, use $\psi(x) = y$.

By a remark of Odifreddi [7, P. 264] ψ witnessing the constructive immunity of co-simple A just above can*not* be taken to be total. Moreover, by an expansion of his argument, we have

Proposition 1 *Suppose A is* any *constructively immune set as witnessed by (awb) some partial computable ψ. Then there is an e such that $W_e = \emptyset$ and $\psi(e)\uparrow$.*

Proof By *Kleene's* (intensional) recursion theorem [1, Top of P. 214],[6] there is a φ-program e such that e first looks in a mirror or makes a copy of itself (a copy external to itself)[7] so as to have a copy of e available as data, essentially *self-knowledge* for the original e *and* for further algorithmic processing,[8] and, for this original e, that further processing involves its trying to compute $\psi(e)$ (employing its self-copy as the argument to ψ) and setting

$$W_e = \begin{cases} \overline{\{\psi(e)\}}, & \text{if } \psi(e)\downarrow; \\ \emptyset, & \text{otherwise.} \end{cases} \tag{1}$$

Clearly, $\psi(e)\downarrow$ is contradictory.

□

By contrast with Proposition 1 above, it is well known that, for the *other* two immunity concepts from Definition 1.2 just above, the ψ's *can*, without loss of generality

[3] This concept is introduced in X. Li's [10], and herein we use the terminology from [7] for it.

[4] This concept is explicitly introduced in McLaughlin's [11].

[5] This concept is implicitly introduced in Smullyan's [12].

[6] In [13] is a discussion of Kleene's version versus Rogers' fixed point version—complete with relevant theorems.

[7] *External* to e itself avoids unneeded infinite regress.

[8] In [14, 15] are discussed these ways of thinking about Kleene's recursion theorem—including Kleene's proof thereof. The self-copying part of Kleene's proof, as analyzed in [15], involves a mechanism quite similar to the top level replication program of a single-celled organism and also the top level programming of von Neumann's self-reproducing automaton [16]. In general the present author views this theorem as a vehicle to study what use is self-knowledge [17] (see also [18, 19]).

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

(wolog), be taken to be total: for proofs one can use twice more the Rogers' fixed point recursion theorem argument on [1, P. 184] that any partial computable function witnessing a set is *productive* can be taken to be total.

How do these various immunities compare? It is well known that, while obviously strong effective immunity implies effective immunity implies immunity, none of the converses hold—even when the complements are c.e. [7, 20, 21]. Furthermore, Ullian [7, 11] showed there is an A such that A and $\overline{A}$ are both effectively immune (see also [22, Lemma 2.1]); whereas, by contrasts, from McLaughlin's [11], strongly effectively immune sets always have *non*-immune complements *and*, from X. Li's [10], constructively immune sets also always have non-immune complements—each with associated witnessing infinite c.e. sets available from their respective proofs; furthermore, Theorem 3.2 in Sect. 3.1.1 below entails that strong effective immunity implies constructive immunity, a new result.

Theorem 3.3 in Sect. 3.1.1 below says, for any A strongly effectively immune awb computable f, for W_x infinite (this includes $W_x = \mathbb{N}$), one can uniformly algorithmically find from a program for f and from x, a program to enumerate algorithmically, in increasing order, an infinite subset of $(W_x - A)$. This is a tiny bit stronger than the McLaughlin result mentioned just above.[9]

For any x, we let D_x denote Rogers' [1] finite set with *canonical index* x. Sans details, but importantly, from any x, one can algorithmically find *both* a y such that y enumerates D_x (i.e., $W_y = D_x$) *and* card(D_x). The latter tells us when y has finished enumerating D_x.

The reverse translation is algorithmic too. (2)

(2) just above will be employed in the proof of Theorem 3.4 further below in Sect. 3.1.1.

Definition 1.3 1. A *disjoint strong array* is an infinite c.e. sequence of canonical indices for a corresponding sequence of pairwise disjoint non-empty finite sets.
2. A disjoint strong array *meets* A iff *each* of the corresponding finite sets has non-empty intersection with A.
3. A is *hyperimmune*[10] iff [A is infinite *&no* disjoint strong array meets A].
4. A is *hypersimple* [9] iff [A is c.e. $\&\overline{A}$ is hyperimmune].

It is well known that hyperimmune sets are immune but not conversely—even when they have c.e. complements [1, 9].

The main theorem proved herein, Theorem 3.4, says, for any A effectively immune awb computable f, for W_x infinite (including $W_x = \mathbb{N}$), $(W_x - A)$, while infinite, is

[9] In effect, McLaughlin's result features only the case of $W_x = \mathbb{N}$. The present author obtained Theorem 3.3 before he learned by literature search of the McLaughlin result. Even another proof of the McLaughlin result comes from Theorem 3.2 mentioned in the paragraph just before the paragraph referencing the present footnote and X. Li's [10] result mentioned just before that that his constructively immune sets cannot have immune complements.

[10] This concept is also implicitly introduced in Post's [9].

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

not hyperimmune (where a witness to non-hyperimmunity is algorithmic in a program for f and x).[11] This theorem is proved in Sect. 3.1.1 below.[12]

With the above mentioned theorems and others further below we study the mixture of constructivities with Post's and other pathological sets. In Sect. 2 below we provide intrinsic motivation for the study of these mixtures and motivations re the components so mixed; we note that constructivity (intuitionism) is a point of view in philosophy of mathematics which requires more general, strict, and pervasive constructivity conditions on all of mathematics.[13]

In Sect. 3.1.2 below we will present two *natural* examples of immune sets satisfying some of the above constructivity conditions (the presentation will involve definitions, theorems, and corollaries): these examples will not be created by such means as diagonalization, and we will present in detail and prove correct the first example but merely discuss in detail the other—since the latter example is now handled in referenced literature.

It is well known that hyperimmunity is characterized as follows (see also [1, 7]).

Theorem 1.4 (Medvedev [29], Rice [30], Uspenskii [31], and Kuznekov)
A is hyperimmune iff [*A is infinite* $= \{a_0 < a_1 < a_2 < \cdots\}$ *&*$(\forall p \mid \varphi_p$ *is computable*$)(\exists k)[a_k > \varphi_p(k)]$].

Next we look at a constructivity condition on hyperimmune sets.

[11] This is a tiny bit stronger than the version of the main theorem announced in the Abstract above—including that the *latter* is only the $W_x = \mathbb{N}$ case.

[12] One of the reviewers of my prior draft of the present paper kindly pointed out [23], a followup paper on constructively immune sets, and, in effect, expressed interest in seeing if the *effectively* immune sets might satisfy a variant of results from [23]. We see above that effectively immune sets are more removed from each of constructively immune and strongly effectively immune vis-à-vis their complements, so, perhaps, the *strongly* effectively immune sets might satisfy a even simpler variant of results from [23]. Some readers may wish to pursue these proposed studies.

[13] We let PA be First Order Peano Arithmetic [1, 24]. Heyting Arithmetic (HA) [25] is just PA with the general Law of Excluded Middle (LEM) carefully *excised* from the underlying classical first order predicate logic. Re the paragraphs just below, which concern intuitionism/constructivity, including in philosophy of mathematics, see [26, 27] *and their references and links*. The Brouwer–Heyting–Kolmogorov (BHK) interpretation of intuitionistic logic features *functions* which, e.g., convert one proof into another. Importantly, the Kleene 1945 notion of *realizability* for HA, which notion essentially is a special case of BHK, interestingly restricts such functions to the algorithmic. This realizability *and its consequences for* HA provided an interesting clue that *some* of Brouwer's mathematical intuitions reduce to, among other related things, the *algorithmicity of finding examples of things proved to exist*. That, at least some of Brouwer's intuitions feature just *algorithmic constructions*, intellectually intrigues the present author—in part since he believes [17, 19] the macroscopic (physical) universe (including us) to have algorithmic next states and the quantum universe to have algorithmic statistical distribution of next states—this is not-necessarily-constructive algorithmic, since we mere humans know a bit only of these algorithms (and sometimes, even then, approximate versions). The present author is not a constructivist philosophically—instead he supports what he terms, *neo-fictionalism*, regarding the philosophy of mathematics—he has, as of this writing, a draft only of a paper explaining this (as well as his beliefs about the universe); in that draft he notes that his favoring this neo-fictionalism over constructivism is mostly because the latter unnecessarily limits the wonderful, applicable, precise, fantasy world of mathematics. Of course, some constructivity in proofs is important to achieve application-necessary proof-mining, and some classical existence proofs can have nothing to mine, e.g., Rogers on [1, P. 9] presents a wonderful example of a trivially (non-constructively) provably computable function g which has *no currently known* algorithm; see also [28]. Some material in the present footnote is important for the third & fourth paragraphs in Sect. 2 below.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Definition 1.5 (*M. Arslanov* [32])

1. A is *effectively hyperimmune* iff [A is infinite $= \{a_0 < a_1 < a_2 < \cdots\}$ &for some computable f, $(\forall p \mid \varphi_p \text{ is computable})(\exists k \leq f(p))[a_k > \varphi_p(k)]$].[14]
2. A is *effectively hypersimple* [A is c.e. &$\overline{A}$ is effectively hyperimmune].

M. Arslanov [32] shows effectively hyperimmune sets exist. In Sect. 3.2 below we will present with proof a *natural* example, not created by such means as diagonalization (the presentation will involve definitions, theorem, and corollaries). M. Arslanov's [32, Theorem 2] says that effectively hyperimmune sets are also effectively immune. While, there are A such that both A and $\overline{A}$ are hyperimmune [1, P. 156], M. Arslanov's [33, Theorem 3] says that an *effectively* hyperimmune set can*not* have an *effectively* hyperimmune complement. The corollary immediately below, then, improves this latter M. Arslanov result.

Corollary 1.6 *Any effectively hyperimmune set can*not *have a just plain hyperimmune complement.*

Proof This corollary follows from M. Arslanov's [32, Theorem 2] mentioned just above and our main Theorem 3.4 also mentioned above (and restated and proved below in Sect. 3.1.1). □

The immediately above Corollary 1.6 will not be restated below.

One can ask if the condition $(\exists k \leq f(p))$ in the definition of effective hyperimmunity above can, wolog, be improved to (for $k = f(p)$). After the next two definitions a much stronger and surprising situation will be seen to hold: Theorem 1.9, an observation of M. Arslanov's from [32].

We define next a known to be more extreme notion than hyperimmunity and, after that, consider M. Arslanov's [32] effectivization of it.

We write the quantifier, $(\forall^\infty k)$, to mean: for all but finitely many $k \in \mathbb{N}$.

Definition 1.7 ([34–36])

1. A is *dense immune* iff [A is infinite $= \{a_0 < a_1 < a_2 < \cdots\}$ &$(\forall p \mid \varphi_p \text{ is computable})(\forall^\infty k)[a_k > \varphi_p(k)]$].
2. A is *dense simple* iff [A is c.e. and $\overline{A}$ is dense immune].

It is well known that dense immunity is strictly intermediate in strength between hyperimmunity and hyperhyperimmunity—including for sets with c.e. complements [34–36].

Definition 1.8 (*M. Arslanov* [32])

1. A is *effectively dense immune* iff [A is infinite $= \{a_0 < a_1 < a_2 < \cdots\}$ &for some computable f, $(\forall p \mid \varphi_p \text{ is computable})(\forall k \geq f(p))[a_k > \varphi_p(k)]$].

[14] M. Arslanov [33] shows that any infinite A is effectively hyperimmune iff, for some computable f, for each p, where φ_p is computable *and* φ_p's successive values form a disjoint strong array, there is an $k \leq f(p)$ such that $D_{\varphi_p(k)}$ *fails* to meet A, i.e., $D_{\varphi_p(k)} \subseteq \overline{A}$.

M. Arslanov [33] also notes that, for $\overline{A}$ c.e., replacing computable f right after the 'iff' in Definition 1.5 above by a *partial* computable function would yield a condition true of any co-hypersimple set.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

2. A is *effectively dense simple* iff [A is c.e. and $\overline{A}$ is effectively dense immune].

Next is the promised surprising result of M. Arslanov.

Theorem 1.9 (M. Arslanov [32]) *The* effectively *dense immune sets are exactly the* effectively *hyperimmune sets.*

Proof This result directly follows from unpacking definitions in [32] and M. Arslanov's [32, Theorem 3]. □

Hence, the natural example of an effectively hyperimmune set in Sect. 3.2 below is also a natural example of an effectively dense immune set: Corollary 3.17 in that same section. Some refinements from Daley [37] of the just mentioned natural example lead to an also natural example so that its complement is effectively dense simple but *not* strongly effectively simple: Corollary 3.19 at the end of Sect. 3.2.[15]

2 Author's motivations

M. Arslanov's interest (see, for example, [32]) in effectiveness added to immune and hyperimmune sets was degree-theoretic, for example, that, for co-c.e. such sets, it forces Turing-completeness [21]. Such examples are nicely amazing; however, the present author finds the adding of constructivities to Post's and other wonderful pathological sets an *intrinsically conceptually* interesting, subtle mixing of the algorithmic with the non-algorithmic.

We note that Post's simple and hypersimple sets have more applications than those Post [9] designed them for: solving Post's Problem for some reducibilities *stronger* than Turing reducibility. Rabin has applied simple sets to obtain a nice game-theoretic result [1, P. 121]. Hypersimple (and hyperimmune) sets seem to capture an essence of many separation and independence results in a wide diversity of contexts. They are crucial to a surprising characterization of computably (a.k.a. recursively) axiomatizable theories that have no independent computable axiomatization [38] (see also [39]). They have been intimately involved in a separation result characterization in computability-theoretic learning theory [40]. Furthermore, such sets have played a role in independence results for both complexity theory [41] and for control structures in effective programming systems (effective numberings) [6].

Immune not hyperimmune sets interact subtly/interestingly with intuitionistic/constructivist mathematics.[16] Suppose A is immune but not hyperimmune, the latter witnessed by a computable function f such that the c.e. sequence $f(0), f(1), f(2), \ldots$ is a disjoint strong array which meets A. For each j, we write $D_{f(j)}$ as $\{k_1^j, k_2^j, \ldots, k_{k_j}^j\}$. Then, for each j, the disjunction

$$\left[k_1^j \in A \ \vee \ k_2^j \in A \ \vee \cdots \vee \ k_{k_j}^j \in A\right] \tag{3}$$

[15] Martin [21] first indicated a priority construction of an effectively simple set not strongly so—based on Sacks' priority construction in [20] of a simple set not effectively so. Our proof of Corollary 3.19 does not employ priorities—this avoidance of priorities is mostly in the set up work from Daley [37].

[16] For this and the next paragraph, the reader can, as desired, review relevant material in Footnote 13 in Sect. 1 above.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

is *true*. To prove (3) intuitionistically/constructively, *uniformly in* j, would require [42], again, *uniformly*, for each j, ones proving *one* of the disjuncts in (3). Success would yield a proof that A is *not* immune.[17] Hence, the disjunctions above do not admit *uniform* intuitionist/constructivist proofs. Herein, we won't go further with this example. The point was interaction between Post's pathologies and constructivity is, again, interesting.

We next describe another little excursion into non-constructivity in another part of computability theory—*but* where we begin to measure the degree of non-constructivity. The simple computability theory result

$$(\forall q)[\{p \mid \varphi_p = \varphi_q\} \text{ is } not\text{c.e.}]. \tag{4}$$

is expressible and *non*-constructively provable in PA; furthermore, from [44] this non-constructivity is *provably inherent* for (4). In [44] (4) is analyzed as to how high in the Arithmetical Hierarchy one must add instances of the Law of Excluded Middle (LEM) [45] to Heyting Arithmetic (HA) to prove it.[18] The *partial* answer in [44] was an adequate upper-bound of (HA + Π^0_2−LEM) and an inadequacy lower bound of (HA + Σ^0_1−LEM)—*but* the latter worked only for excluded middle disjunctions of the form being or not being in Σ^0_1 index sets.[19] This study constituted a bare beginning to a reverse mathematics analysis of how much non-constructivity one must add to intuitionism to get non-constructive classical results—in computability theory.

The present author likes very much the study of productive sets,,[20] another case of some constructivity added to being non-c.e. He also likes the "completion" of productive sets to the *completely productive sets* which are (constructively) co-extensive with the productive sets (due to [1, Myhill, Theorem VI, pp. 183–184]), and that they constitute the *constructively non-c.e. sets* [1, P. 92–93].[21] Part of the study [44] treated in the just prior paragraph extended this latter to a new concept of *uniformly c-productive sequences*.

[17] A very nice example illustrating the role of constructively proving disjunctions and its relevance to proof-mining is found in [43].

[18] [44] also considers examples besides (4).

[19] For prenex normal form arithmetical closed formulas with a non-empty string of quantifiers in front, the predicate inside can be taken, wolog, to be primitive recursive [1]; hence, all our example theories in the paragraph referencing this footnote, examples such as (HA + Π^0_2−LEM), admit of computable (recursive) axiomatizations, where Π^0_2−LEM is, then, the set of all cwffs of the form

$$(\forall u)(\exists v)R(u, v) \vee \neg(\forall u)(\exists v)R(u, v), \tag{5}$$

each for some primitive recursive predicate R.

[20] Implicitly introduced in Post's [9] and named by Dekker. The present author likes the iteration of a 1-1 productive function up to any constructive ordinal [1, Parikh, pp. 212–213] the concept of *full* productive set [1, Dekker, p. 102], and that these latter are ≡ to Kleene's O (this due to [1, Parikh, P. 222]).

[21] In [46] is provided a similar "completion" of Smullyan's *effective inseparability* [1, 7, 8, 47] (for disjoint pairs of sets), but *this* completion is shown, still in [46], to be *strictly* more refined in the sense of Baire Category Theory than Smullyan's notion.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

3 Results

In this section we will present and prove results. In Sect. 3.1, the results are about constructivities on immune sets, and in Sect. 3.2 are some remaining results about effectively hyperimmune, equivalently, effectively dense immune sets.

3.1 Constructivity for immunity

For A (strongly) effectively immune, the present study began with its author thinking about what can then be said about $(W_x - A)$ in the case W_x is infinite (N.B. this includes $W_x = \mathbb{N}$).

3.1.1 Three theorems

The proofs of Theorems 3.2, 3.3, and 3.4 below in the present section (Sect. 3.1.1) employ the notation and conclusion of the following

Lemma 3.1 *Uniformly in x, one can (try*[22] *to) obtain an* increasing order, algorithmic *enumeration of* positive *elements from W_x:*

$$n_0^x < n_1^x < n_2^x < \cdots . \tag{6}$$

Clearly, for each j,

$$\text{if } n_j^x\!\downarrow,\ \text{it's positive and, hence, it's } > j;\ \text{moreover, for each } i < j,\ n_i^x \text{ also } \downarrow. \tag{7}$$

The present author obtained the next theorem *sans its last sentence* before he learned about X. Li's [10]; this next theorem was originally just a precursor to the next two theorems thereafter (Theorems 3.3 and 3.4).

Theorem 3.2 *Suppose A is strongly effectively immune awb computable f. Then there is a partial computable ψ such that, for each x where W_x is infinite, $\psi(x)\!\downarrow \in (W_x - A)$. Hence, A is constructively immune.*

Proof Here is the idea of this proof. From x, we want to find an element in W_x (at least when it's infinite) which is not in A. For an element of W_x, we'll choose an element from (6) above. We'll employ Kleene's Parametric Recursion Theorem (KPRT) to obtain a computable function g such that, for each x, φ-program $g(x)$ has x stored (algorithmically retrievably) inside and $g(x)$ looks in a mirror or makes an external self-copy so as to have self-knowledge.[23] $g(x)$ employs its self-knowledge to compute $f(g(x))$, an upper-bound on the elements of $W_{g(x)}$ for this set to be contained in A.

[22] The trying will eventually fail iff W_x is finite.

[23] Basically KPRT is a combination of S-m-n and Kleene's Recursion Theorem, where the S-m-n part is the storing of data inside (algorithmically retrievably).

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

$g(x)$ also retrieves x from storage,[24] and it seeks one of the elements of W_x from (6) *not* in A; more specifically, it chooses (justified by Lemma 3.1 and (7) above) an element from (6) to put in $W_{g(x)}$ which is bigger than the upper-bound $f(g(x))$, namely, $n^x_{f(g(x))}$—by the $j = f(g(x))$ case of (7). We need to explain what happens more generally, including the case where $n^x_{f(g(x))}\uparrow$. Generally, for each x,

$$W_{g(x)} = \begin{cases} \left\{n^x_{f(g(x))}\right\}, & \text{if } n^x_{f(g(x))}\downarrow; \\ \emptyset, & \text{otherwise.} \end{cases} \tag{8}$$

Let

$$\psi(x) = n^x_{f(g(x))}. \tag{9}$$

ψ is obviously partial computable.

Clearly, by Lemma 3.1 and (7),

$$n^x_{f(g(x))}\downarrow \Rightarrow \psi(x)\downarrow > f(g(x)). \tag{10}$$

Suppose now W_x is infinite. Hence, $n^x_{f(g(x))}\downarrow$. Therefore, by (10) just above, $\psi(x)\downarrow > f(g(x))$, and $W_{g(x)} = \{\psi(x)\downarrow\}$. Since, then, $\max(W_{g(x)})\downarrow = \psi(x)\downarrow$, and this is $> f(g(x))$, we have $\psi(x)\downarrow \in (W_x - A)$.

Therefore, A is constructively immune. □

This latter proof can be extended employing an extra parameter and padding for the KPRT to prove

Theorem 3.3 *Suppose A is strongly effectively immune awb computable f.*

For W_x infinite (this includes $W_x = \mathbb{N}$), one can uniformly algorithmically find from a program for f and from x, a program to enumerate algorithmically, in increasing order, an infinite subset of $(W_x - A)$.

Proof Suppose wolog f is *strictly monotone increasing*. By a Padded KPRT, there is a computable g which, for each value of its second argument, is *strictly monotone increasing in its first argument*, and it is such that, for each k, x, $g(k, x)$ behaves as follows

$$W_{g(k,x)} = \begin{cases} \left\{n^x_{f(g(k,x))}\right\}, & \text{if } n^x_{f(g(k,x))}\downarrow; \\ \emptyset, & \text{otherwise.} \end{cases} \tag{11}$$

Clearly, then, for each k,

$$n^x_{f(g(k,x))}\downarrow \Rightarrow n^x_{f(g(k,x))}\downarrow > f(g(k,x)). \tag{12}$$

[24] For later proofs, we'll consider it understood that stored data is retrievable and will no longer mention it. Furthermore, for each self-knowing program g-of-its-arguments we'll now just use a copy of this program in describing its behavior without mentioning mirrors or self-copying.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Suppose now W_x is infinite. Hence, for each k, $n^x_{f(g(k,x))}\downarrow$. Therefore, for each k, $n^x_{f(g(k,x))}\downarrow > f(g(k,x))$, and $W_{g(k,x)} = \left\{n^x_{f(g(k,x))}\downarrow\right\}$. Since, then,

$$\max(W_{g(k,x)})\downarrow = n^x_{f(g(k,x))}\downarrow, \tag{13}$$

and this is $> f(g(k,x))$, we have $n^x_{f(g(k,x))}\downarrow \in (W_x - A)$. Since, clearly, $n^x_{f(g(k,x))}$ strictly increases in k, $n^x_{f(g(k,x))}$, $k = 0, 1, 2, \ldots$ is the desired increasing c.e. sequence. □

Next is our main theorem of the present paper.

Theorem 3.4 *Suppose A is effectively immune awb computable f.*

Then, for W_x infinite (including $W_x = \mathbb{N}$), $(W_x - A)$, while infinite, is not *hyperimmune (where a witness to non-hyperimmunity is algorithmic in a program for f and x).*

Proof Suppose wolog f is strictly monotone increasing. By Padded KPRT there is a computable g strictly monotone increasing in its first argument such that, for each k, x, $g(k,x)$ behaves as described further below.

Unlike in the just previous proof, $g(k,x)$ does *not* seek an element of (6) *larger than* $f(g(k,x))$—since in the present proof $f(g(k,x))$ upper-bounds the *cardinality* of $W_{g(k,x)}$, not its max—were it to be a subset of A. Instead it needs to put *too many elements into* $W_{g(k,x)}$—*compared to* the number $f(g(k,x))$. Also, for each x with W_x infinite, we will make the $W_{g(k,x)}$, $k \in \mathbb{N}$, finite and pairwise disjoint.

For each k, x, program $g(k,x)$, then, behaves as follows

$$W_{g(k,x)} = \begin{cases} \left\{n^x_{kf(g(k,x))}, n^x_{kf(g(k,x))+1}, \cdots, n^x_{(k+1)f(g(k,x))}\right\}, & \text{if } n^x_{(k+1)f(g(k,x))}\downarrow; \\ \emptyset, & \text{otherwise.} \end{cases} \tag{14}$$

Suppose now W_x is infinite. Hence, for each k, $n^x_{(k+1)f(g(k,x))}\downarrow$. Clearly, then, since $(k+1)f(g(k,x)) < (k+1)f(g(k+1,x))$, the finite $W_{g(k,x)}$, $k \in \mathbb{N}$, are pairwise disjoint, and, for each $k \in \mathbb{N}$,

$$\mathrm{card}(W_{g(k,x)}) = 1 + f(g(k,x)) > f(g(k,x)). \tag{15}$$

Hence, for this x, by (2) in Sect. 1 above, there is a computable function h such that, for each k, $D_{h(k)} = W_{g(k,x)}$. Each $D_{h(k)} = W_{g(k,x)}$, while a finite $\subseteq W_x$, is, by (15) above, too large to be contained in A, so each *meets* $(W_x - A)$. The sequence, $h(0), h(1), h(2), \ldots$, then witnesses that $(W_x - A)$ is *not* hyperimmune.

□

3.1.2 Two natural examples

Next we begin to introduce our first natural example of a strongly effectively immune set.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Definition 3.5 (*Blum* [48]) A *Blum program size measure* is a finite-to-one computable function s such that the function, $c_s = \lambda n.$(the canonical index of $s^{-1}(n)$), is also computable.

Here are two *example* such size measures s_1, s_2.

$$\text{Let } s_1 = \lambda p.(p). \tag{16}$$

The *reduced binary* representation of p omits any *redundant* leading 0s; for *example*, in reduced binary, 0 is written as 0, not as 000, and the base 10 number 42 is written in reduced binary as 101010, not as 0101010.[25]

$$\text{Let } s_2 = \lambda p.(\text{length of the reduced binary representation of } p). \tag{17}$$

So, $s_2(p)$ is the number of computer *bits* for storing $p \in \mathbb{N}$.

Now suppose s is *any* Blum size measure with associated function c_s.

Definition 3.6 $\text{Min}_s = \{q \mid (\forall p \mid s(p) < s(q))[\varphi_p \neq \varphi_q]\}$.

Then, Min_s is the set of *s-minimal* φ-programs for the class of 1-argument partial computable functions: $\mathbb{N} \to \mathbb{N}$.

Meyer [49] notes that Blum first showed Min_{s_1} is immune; Meyer shows it is *not* hyperimmune. Bagchi [50] showed Min_s more generally is immune and not hyperimmune. None of them considered constructivity conditions on immunities. Meyer and Bagchi's [51, Theorem 3] says $\emptyset^{(2)}$ is Turing-equivalent to $\text{Min}_s \in (\Sigma^0_2 - \Pi^0_2)$—with proofs in [49, 50]. Kinber [52] shows the btt-degree of Min_{s_1} depends on the choice of acceptable system φ and that, for *some* acceptable system φ, Min_{s_1} has tt-degree $\emptyset^{(2)}$. Even for $s = s_1$, it is open whether the tt-degree of Min_s is $\emptyset^{(2)}$.

Next we have a result providing a *natural* strongly effectively immune set. The case of $s = s_1$ was shown by the present author a number of years ago, he told people about it, and a proof of this case appears on [53, P. 524].[26] The proof herein of the next theorem adds a bit of reasoning to a close variant of that proof found on [53, P. 524].[27]

Theorem 3.7 Min_s *is strongly effectively immune.*

Proof By the KPRT there is a computable g such that, for each x, y, $g(x)$ achieves the following input/output behavior

$$\varphi_{g(x)}(y) = \begin{cases} \varphi_z(y), & \text{if } z \text{ is } 1st \text{ number found in } W_x \text{ for which } s(z) > s(g(x)); \\ \uparrow, & \text{if no such } z \text{ is found.} \end{cases} \tag{18}$$

[25] Herein the empty string is not a binary number; however, in the *dyadic* representation of $\mathbb{N}$ (defined further below), it is the only representation of 0.

[26] Owings [54] nicely generalizes this case in another direction.

[27] For completeness, we do include our proof, and we need to refer to it in Sect. 4 below. These proofs share some ideas from proofs in Blum's [48].

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Suppose for now $W_x \subseteq \text{Min}_s$.

Suppose for contradiction there is a $z \in W_x$ such that $s(z) > s(g(x))$. Then from (18) above, for the 1st such $z \in W_x$ found by φ-program $g(x)$,

$$\varphi_{g(x)} = \varphi_z. \tag{19}$$

Since $z \in W_x \subseteq \text{Min}_s$ and since $s(z) > s(g(x))$, from the definition above of Min_s (Definition 3.6), we must have

$$\varphi_{g(x)} \neq \varphi_z, \tag{20}$$

a contradiction to (19) above.

Hence, for each $z \in$ this W_x, $s(z) \leq s(g(x))$. Therefore, $W_x \subseteq \{z \mid s(z) \leq s(g(x))\}$.

Hence, for *any* x, if we let

$$h(x) = \max(D_{c_s(0)} \cup \cdots \cup D_{c_s(s(g(x)))}), \tag{21}$$

h is computable, and, for each x such that $W_x \subseteq \text{Min}_s$, $\max(W_x) \leq h(x)$.

□

Corollary 3.8 $\overline{\text{Min}_s}$ *is* not *immune, and an infinite, c.e. list of programs in it can be found.*

Proof This corollary follows from the multiply-proved McLaughlin result above that the complement of a strongly effectively immune set is not immune—where an infinite, c.e. list of programs in it can be found.[28] □

Convention 3.9 *Again, we let* PA *be First Order Peano Arithmetic [1, 24].*

Let T be an arbitrary extension of PA *with a* c.e. set of axioms *such that all T's theorems expressible in* PA *are* true *in the standard model of* PA *[1, 24].*

Example such Ts include PA, SOA [1, 55],[29] and ZFC [24].

Corollary 3.10 *There is a computable function f so that, for any x, if W_x is the c.e. set of (numerically named) axioms of T, then $f(x)$ upper-bounds the* max *of the* finitely *many numerically named programs that are T-*provably in Min_s.

However, *even* PA *can prove of infinitely many φ-programs that each is in* $\overline{\text{Min}_s}$.

X. Li's [10, Theorem 4] provides that Min_{s_1} is what we call herein constructively immune. By Theorems 3.2 and 3.7 above, more generally, Min_s is constructively immune.[30]

[28] From Theorem 3.3 above we get the slight improvement that, for any infinite W_x, $W_x \cap \overline{\text{Min}_s}$, while infinite, is not immune. So, e.g., the infinite set of even numbers in $\overline{\text{Min}_s}$ as well as the infinite set of odd numbers in $\overline{\text{Min}_s}$ are each not immune.

[29] SOA stands for Second Order Arithmetic, and it is, of course, a (two-sorted) *first order* theory.

[30] Schaefer's Corollary 3.11 *in* [53] generalizes this conclusion.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Stephan and Teutsch [56] insightfully point out the interest in minimal size programs (as above) parallels the interest in shortest programs for single integers or finite strings as in Kolmogorov-complexity (K-complexity) (re latter, see, for example, [7, 57]). So next we'll look at the latter subject vis-à-vis the concerns of the present paper—eventually presenting the second natural example.

First, since some relevant sources do K-complexity for (non-negative) integers and others for the *dyadic* (finite) strings (over the alphabet $\{0, 1\}$),[31] we set up the standard correspondence between them as follows.

For the *dyadic representations* of the elements of $\mathbb{N}$,

$$0, 1, 2, 3, 4, 5, 6, 7, 8, \ldots, \tag{22}$$

we use the *respective* finite strings over $\{0, 1\}$,

$$\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots, \tag{23}$$

where ϵ is the empty such string.

$$\text{Let } |\cdot| = \lambda n\,.(\text{the } \mathit{length} \text{ of the dyadic representation of } n). \tag{24}$$

While $|\cdot|$ is one more example Blum measure of program size, we'll *also* use it (as well as s_1) to measure the size of dyadic strings which are not necessarily interpreted as program numbers.

Different sources employ slightly different definitions of K-complexity and K-randomness—differences depending on choice of $\Xi = (s, \varphi)$.[32]

Definition 3.11

1. The *K-complexity function* (in dependence on Ξ) is given by

$$K_\Xi = \lambda x\,.(\min(\{s(p) \mid \varphi_p(0) = x\}). \tag{25}$$

2. The *set of K-random numbers or strings* (in dependence on Ξ) is given by

$$R_\Xi = \{x \mid K_\Xi(x) \geq s(x)\}). \tag{26}$$

The intuition behind (26) above is that strings x which have s-minimal φ-programs computing x (on the trivial input, 0) not smaller than the s-size of x itself ostensibly have *no algorithmically exploitable pattern in them* enabling their being computed by

[31] The term 'dyadic', essentially as we employ it, first appeared in [47, § 5]; *however*, this reference (as well as others) employed the alphabet $\{1, 2\}$ (instead of $\{0, 1\}$); the term is used later, e.g., in [58, Sect. 2.4]—*but* with the alphabet $\{0, 1\}$ as in the present paper. Some real world applications of K-complexity to genomics, e.g., [59], employ application-relevant larger alphabets. Interestingly [59] nicely employs the modern compression algorithm `zpaq` to *approximate* uncomputable K-complexity.

[32] Note that herein, for K-complexity, we do *not need* to employ (nor define) a prefix-free domain—as *is* usefully employed in [57, Chapter 3].

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

an s-shorter φ-program, and we can imagine s-minimal φ-programs for such x may just have to store x itself inside,[33]

Odifreddi [7, P. 263], employing $\Xi = (s_1, \varphi)$, shows that the set of natural numbers that are *not* Kolmogorov-random is effectively simple. A. Arslanov's [60, Corollary 2.2], *essentially* employing $\Xi = (|\cdot|, \varphi)$, says that the set of K-random strings is an effectively immune, non-hyperimmune, co-c.e. set; his [60, Theorem 2.3] says the set of non-K-random strings is wtt-complete. He mentions that Kummer [61] shows that tt-completeness of the set of *non*-K-random strings depends on the choice of φ.

The present author wondered if, in this context, effectively could be replaced by strongly effectively. He found Buhrman and Orponen's [62], and in there, between their Definition 3.3 and Theorem 3.4 there is an un-numbered claim that the proof of their Lemma 3.1 about *time-bounded* K-randomness can be modified, by removing the time-boundedness, to show the set of K-random strings is *strongly* effectively immune. *However*, the definition of strongly effectively immune in [62] is *not* the definition in [11]—which is used in the present paper,[34] The one in [62] is also ostensibly weaker. The present author talked with Royer about rewriting the proof of the just above mentioned Lemma 3.1 and seeing if a modification could achieve the strong effective immunity *from* [11]. Royer was successful for K-randomness measured by $|\cdot|$ and some, important acceptable φs. Next we explain Royer's result [63, Theorem 8]. Royer employs an *additively optimal (partial computable) universal function* from [57, Definition 2.0.1] which functions are shown to exist by [57, Lemma 2.1.1]. The example in the proof of this lemma (applied to the set of 1-argument partial computable functions) happens to constitute an acceptable system, and we'll temporarily call an arbitrary fixed such example, φ^{AO}.[35]

$$\text{Royer in [63]uses the notation } U \text{ for our } \varphi^{\mathrm{AO}}. \tag{27}$$

We have

Theorem 3.12 (Royer's Theorem 8 [63]) *$R_{(|\cdot|,U)}$ is strongly effectively immune in the sense of [11] awb the function $\lambda x\,.(k_1(x+1))$, where k_1 is a suitably large positive constant.*

Royer's proof involves neither recursion theorems nor even S-m-n and is mostly in the style of M. Li and Vitányi's [57, Chapter 2] and, of course, is partly based on [62, Proof of Lemma 3.1].[36]

[33] For example, let x_0 be the dyadic string consisting of 10, 000 copies of 10001 one right after the other. In many programming systems, one can store 10001 and, with *little program size overhead* program a loop to iterate 10, 000 times the printing of the stored string one time after the other. The size of such a program printing x_0 can be much smaller than the length of x_0. x_0 will not, then, be a K-random string: it has an easily, algorithmically exploitable pattern in it.

[34] *In effect* Buhrman and Orponen's concept of a strongly effectively immune set A in [62] is that there is a computable function f which, given any *total* program p for *deciding the computable (a.k.a. recursive) set* $\varphi_p^{-1}(1)$, when this set is a subset of A, $f(p)$ upper-bounds $\max(\varphi_p^{-1}(1))$.

[35] The superscript is not an oracle, but is short for Additively Optimal.

[36] Interestingly, his $\{W_p \mid p \in \mathbb{N}\}$ need be neither acceptable nor even based on $U = \varphi^{\mathrm{AO}}$. As noted, though, his set of K-random dyadic strings *is* based on $|\cdot|$ and his additively optimal U.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Corollary 3.13 *There is a computable function f so that, for any x, if W_x is the c.e. set of (numerically named) axioms of T, then $f(x)$ upper-bounds the* max *of the* finitely *many numbers represented by the strings T-*provably *in $R_{(|\cdot|,U)}$.*

3.2 Constructivity for hyperimmunity and dense immunity

As promised near at the end of Sect. 1 above, we proceed to present and prove a natural example of an effectively dense simple, not strongly effectively simple set.

First a little history related to the example: Rado [64] gave two examples of very fast growing, non-computable functions based on a concrete Turing Machine (TM) formalism. His TM model featured the requirement that single tape-head moves could only be one square left or right (and not stay on the same square for one move). His first function, on n, output the maximum number of 1s printed by the *halting* TMs on one fixed, trivial input *and with n states*; his second, on n, output the maximum number of tape-head moves by the *halting* TMs on the same one fixed, trivial input *and with n states*. His motive was to achieve non-computable functions not created by diagonalization. Two kinds of follow up papers resulted. For his functions, one kind involved computing or upper-bounding a few values on smallish arguments or proving that evaluation on a few thousand (see, for example, Yedidia and Aaronson's [65]) to a few hundred states is independent of ZF + other assumptions (for all this see [66] including its bibliography). The other kind was to employ (sometimes with extra but natural restrictions) Blum program size measures [48] and Blum dynamic complexity [67] to provide more abstract forms of Rado's second function and their range sets, where, in some papers, (variants of) these range sets are used to simplify constructions in computability theory. This second kind is due to Daley (see [37, 68, 69] and their bibliographies). Our example is of the latter kind and essentially originates from Daley's [37].

In Sect. 3.1.2 above we already presented and used the Blum axioms for program size measures [48] (see Definition 3.5 above and discussion thereafter).

Next we will present the Blum dynamic complexity axioms [67].

Definition 3.14 We associate with φ another partial computable function $\Phi = \lambda p, x\,.(\Phi_p(x))$. Φ is a *Blum complexity measure* for φ iff it satisfies the following

1. $\delta(\Phi) = \delta(\varphi)$;
2. $\{(p, x, t) \mid \Phi_p(x)\!\downarrow \leq t\}$ is a computable set.

Two example Blum complexity measures Φ^1_p, Φ^2_p follow, each for the same acceptable TM system φ^{TM}:

1. Let $\Phi^1_p(x) = \begin{cases} \text{number of steps } p \text{ executes on } x, & \text{if } \varphi^{\mathrm{TM}}_p(x)\!\downarrow; \\ \uparrow, & \text{otherwise.} \end{cases}$
2. Let $\Phi^2_p(x) = \begin{cases} \text{number of distinct tape squares } p \text{ visits on } x, & \text{if } \varphi^{\mathrm{TM}}_p(x)\!\downarrow; \\ \uparrow, & \text{otherwise.} \end{cases}$

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Φ^1 is a *time* measure. Φ^2 is a *space* measure. We'll sometimes speak of the output of Blum complexity measures as *steps* or *run-times*—even if the $\downarrow$ output actually is a count of distinct tape squares visited.

For now let Φ be an *arbitrary* Blum complexity measure associated with φ.

Definition 3.15 Let $\Omega = (s, \Phi)$, where, for now and as above in Sect. 3.1.2, s is an arbitrary Blum measure of program size.

$$\text{Let } b(n) = \max(\{\Phi_k(0)\downarrow \mid s(k) \leq n\}). \tag{28}$$

$$\text{Let } B_\Omega = \{b(0), b(1), b(2), \ldots\}, \text{ which is, of course, infinite.} \tag{29}$$

$$\text{Write } B_\Omega = \{b_0 < b_1 < b_2 < \cdots\}. \text{ Clearly, } (\forall n)[b_n \geq b(n)]. \tag{30}$$

Theorem 3.16 *B_Ω is* effectively *hyperimmune.*[37]

Proof We'll create a computable g so that, for φ_p total, program $g(p)$ will have $\Phi_{g(p)}$ on any z (but we care about $z = 0$ only) come out $\downarrow$ and larger than the value of φ_p on some convenient, helpful program size. We'll use size: $s(g(p))$.

By the KPRT there is a computable function g such that, for each p, z, φ-program $g(p)$ computes $s(g(p))$ and, then, achieves the following input/output behavior

$$\varphi_{g(p)}(z) = \begin{cases} \uparrow, & \text{if } [\varphi_p(s(g(p)))\uparrow \ \vee \ \Phi_{g(p)}(z)\downarrow \leq \varphi_p(s(g(p)))\downarrow]; \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

The intuitive idea of (31) just above is that, when $\varphi_p(s(g(p)))\downarrow$, program $g(p)$ (on z) *also* uses its self-knowledge to see if it halts too soon, that is, within $\varphi_p(s(g(p)))$ steps; and, if so, it takes infinitely many steps ($\uparrow$), contradicting its halting too soon.[38] Hence, when $\varphi_p(s(g(p)))\downarrow$, program $g(p)$ (on z) halts (with output 0), but *not* too soon, that is, for each z (again we care about $z = 0$ only),

$$\Phi_{g(p)}(z)\downarrow > \varphi_p(s(g(p)))\downarrow. \tag{32}$$

Let $f(p) = s(g(p))$. *Suppose φ_p is total.* Of course $k = g(p)$ is such that $s(k) = s(g(p)) \leq s(g(p)) = f(p)$. Then, by (28), (29), (30), and (32), for $z = 0$,

$$b_{f(p)} \geq b(f(p)) = \max(\{\Phi_k(0)\downarrow \mid s(k) \leq f(p)\}) \geq \Phi_{g(p)}(0)\downarrow > \varphi_p(f(p))\downarrow \tag{33}$$

Hence, by (33), B_Ω is effectively hyperimmune awb f.

□

Corollary 3.17 *B_Ω is* effectively *dense immune.*

Proof This corollary follows from Theorem 3.16 above in the present section (Sect. 3.2) and M. Arslanov's Theorem 1.9 in Sect. 1 further above. □

[37] Evaluation of Rado's functions at a few values are called, *Busy Beaver Games*; this is why, besides our following loosely Daley's notation in [37], we used upper and lower case b's in our notations in Definition 3.15 just above.

[38] This trick is somewhat reminiscent of Blum's proof of his [67, Theorem 4].

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Based on B_Ω being just effectively immune, we have the following corollary *analogous* to the first paragraph of Corollary 3.10 and to Corollary 3.13—each in Sect. 3.1.2 above (see also Convention 3.9 in Sect. 3.1.2 above).

Corollary 3.18 *There is a computable function f so that, for any x, if W_x is the c.e. set of (numerically named) axioms of T, then $f(x)$ upper-bounds the* cardinality *of the* finitely *many step-counts that are T-*provably in B_Ω,[39]

In [37] Daley repeatedly adds new, natural axioms to the Blum Axioms for Ω. This was to help him prove many many computability and complexity theoretically interesting properties of the resultant B_Ωs. He did not, however, consider constructivity conditions on immune sets re the B_Ωs. We will not detail nor justify any of his extra axioms here (since that is nicely done in [37]), but, shortly below, we will state (but not reprove) just two of his large number of properties which follow from many of his extra axioms—which axioms we now assume too (without restating them). We now write $B_{\Omega'}$ for any one of the *resultant* B_Ωs—satisfying these extra axioms. The two properties are: $\overline{B_{\Omega'}}$ is c.e. and $B_{\Omega'}$ is retraceable.[40] From these two properties and Yates' [71] characterization of hyperhypersimple sets: A is hyperhypersimple iff [A is c.e., $\overline{A}$ is infinite, and $\overline{A}$ has no infinite retraceable subsets]; we have that $\overline{B_{\Omega'}}$ is not hyperhypersimple.[41]

From our Theorem 3.16 above in the present section (Sect. 3.2) and M. Arslanov's [32, Theorem 2] used in Sect. 1 above, $\overline{B_{\Omega'}}$ is effectively simple, and, then, from [21], it is, as announced in [37], Turing-complete; also, since it is dense simple, from Cohen and Jockusch's surprising [72, Theorem 1], we have

Corollary 3.19 *The effectively dense simple set, $\overline{B_{\Omega'}}$, is* not *strongly effectively simple.*

4 Suggested future, related work

$=^*$ between two partial functions: $\mathbb{N} \to \mathbb{N}$ stands for equals except on finitely many arguments.

Definition 4.1 $\mathrm{Min}^*_s = \{q \mid (\forall p \mid s(p) < s(q))[\varphi_p \neq^* \varphi_q]\}$.

Hence, Min^*_s is the set of *all* programs each s-minimal among those for *the finite variants of* some (1-argument) partial computable function.

Theorem 4.2 Min^*_s *is infinite but free from infinite Σ^0_2 subsets.*

The case of $s = s_1$ was shown by the present author a number of years ago, he told people about it, and a hint for a proof (unlike the present author's) of this case is on [53, P. 541]. The above Theorem 4.2 is strengthened by us in Theorem 4.4 below.[42]

[39] N.B. Corollary 3.18 features an upper-bound on *cardinality* not on max.

[40] Re retraceable sets see [1, 70].

[41] Odifreddi on [7, P. 288] introduces *a particular* notion of *effectively hyperhypersimple set* and shows no such set exists; he suggests this may explain why, historically, it took some time to prove hyperhypersimple (hhs) sets existed at all. In the future, it may be interesting to pursue the question: what about *other* notions of effectively hhs sets? We do not explore this further in the present paper.

[42] A number of other nice results about Min^*_s, re its position in the Arithmetical Hierarchy and degree of unsolvability, appear in Schaefer's [53, Sect. 5]—with the degree of Min^*_s not yet completely nailed down.

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

We use χ_A for the characteristic function of A.

The Σ_2^0 sets can be captured by an oracular, acceptable programming system (for the equivalent sets c.e. in K), W_p^K, $p \in \mathbb{N}$, where K is the diagonal halting set [1].

We can, then, define such notions as (strongly) effectively Σ_2^0-immune, *where* the witnessing upper-bounding functions are still *computable* (details omitted).

We write the quantifier, $(\exists^\infty t)$, to mean: there exists infinitely many $t \in \mathbb{N}$ such that.

For convenience in working algorithmically with the system, W_p^K, $p \in \mathbb{N}$, we can prove (see Footnote 44 referenced a little further below) the following

Lemma 4.3 *There is a* computable, $\{0, 1\}$*-valued function* F^2 *such that*[43]

$$(\forall p, z)[\chi_{W_p^K}(z) = \liminf_{t\to\infty} F^2(p, z, t)]. \tag{34}$$

Hence, Lemma 4.3 provides that, for some computable F^2, for each p, z: if $z \in W_p^K$, then $(\forall^\infty t)[F^2(p, z.t) = 1]$; if $z \notin W_p^K$, then $(\exists^\infty t)[F^2(p, z.t) = 0]$.[44]

This lemma motivates the present author to call the Σ_2^0 sets, the *lim-c.e.* sets.[45]

Theorem 4.4 Min_s^* *is strongly effectively* Σ_2^0*-immune.*

We can prove Theorem 4.4 just above by Lemma 4.3 above and the KPRT *for* φ by an (omitted) argument a bit more complicated than the proof of Theorem 3.7 above in Sect. 3.1.2. We note that Theorem 4.4 just above provides the first known, as well as a *natural* example of, a strongly effectively Σ_2^0-immune set.

We have, then, Corollary 4.6 further below. This corollary is analogous to, for example, the first paragraph of Corollary 3.10 above in Sect. 3.1.2.

Convention 4.5 PA *is still First Order Peano Arithmetic.*

Let T' *be an arbitrary extension of* PA *with a* lim-c.e. set of axioms *such that all the theorems of* T' *expressible in* PA *are* true *in the standard model of* PA.[46]

For each $n \geq 1$, V_n is the set of all the Σ_n^0*-sentences of* PA *true* in PA's standard model [1, P. 319]. Rogers' [1, Corollary X, p. 319] says, for each $n \geq 1$, $V_n \leq_1 \emptyset^{(n)}$. For example, then, V_2 *is* lim-c.e. Footnote 47 referenced at the end of this sentence proves: for each $n \geq 1$, $V_n \equiv \emptyset^{(n)}$—thereby solving the open question on [1, P. 319],[47]

Stephan and Teutsch [56] nicely provide immunities, more generally, for the Arithmetical Hierarchy—so far without added constructivities.

[43] F^2 could be made primititve recursive or even lineartime computable ([58]).

[44] This assertion, provided by Lemma 4.3 and equivalent to Lemma 4.3, can be proved easily from Markwald's [73, Satz X] and also [1, Theorem XVII, p. 328].

[45] Of course [1] the c.e. sets themselves are the Σ_1^0 sets.

[46] Craig's trick from [74] relativizes, e.g., to oracle K, so, for T': we can replace the lim-c.e. axiom set by an *equivalent* one that's Δ_2^0 (which is, then, *limiting-computable* thanks to Shoenfield's Limit Lemma [75]—according to [76], this lemma was also previously known to Post—but ostensibly not promulgated by him). Proof-checking in T' would, then, come down to *deciding* axiomhood. With a Δ_2^0, but not computably decidable axiom set, axiomhood can, nevertheless, be decided correctly *past some point in the limiting process—but* we cannot, in general, find *algorithmically* where that point is.

[47] Strictly after Rogers' book [1] was created, Hilbert's 10th Problem was solved. We can, *now* improve Rogers' Corollary X to: for each $n \geq 1$, $V_n \equiv \emptyset^{(n)}$; hence, for example, V_2 is Σ_2^0*-complete*, so we

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Examples of T' from Convention 4.5 just above include the result of adding the (Σ^0_2-*complete*—by Footnote 47) lim-c.e. set V_2 to the regular axioms of each of PA, SOA, and ZFC.[48]

Corollary 4.6 *There is a* computable *function* f*, so that, for all* x *such that* W^K_x *is the* lim-c.e. *set of (numerically named) axioms of* T'*, then* $f(x)$ *upper-bounds the* max *of the finitely many numerically named programs that are* T'-provably in Min^*_s.

Finally in this Sect. 4: the present author suggests it may be fun for others to carry out further work on (strong) effective Σ^0_2-immunity *and the like* for W^K_p, $p \in \mathbb{N}$, ...[49]

get, then, a direct proof that ZFC alone cannot derive all the sentences of V_2 (indirectly: the standard Gödel sentences for ZFC are Π^0_1-assertions expressible in the language of PA, and we can add a spurious, numerical, existential quantifier in front of them). To get this improvement just above to Corollary X, we'll first indicate how to prove: for each $n \geq 1$, $\emptyset^{(n)} \leq_{\mathrm{m}} V_n$. Then, we'll directly lift this $\leq_{\mathrm{m}}$-case to the desired $\leq_1$-case. This lift can be done, from slightly to a lot less succinctly, employing any one of multiple routes to showing that each V_n is a cylinder (proofs omitted; re cylinders see, e.g., Rogers' [1, Section 7.6], including [1, Theorem VIII (c), p. 89] and [1, Theorem IX, p. 90]). Anyhow, once we have the $\leq_1$-case, from that together with Rogers' original Corollary X and [1, Myhill, Theorem VI, pp. 85–86], we get the $\equiv$-conclusion for the improvement of Corollary X. From Sect. 1: the variables x, y, and each z_i in a fixed length vector $\vec{z}$ range over $\mathbb{N}$. So, to sketch the $\leq_{\mathrm{m}}$-case, we begin with a universal Diophantine polynomial $\mathbf{U}$ having the property: $(\forall x, y)[y \in W_x \Leftrightarrow (\exists \vec{z})[\mathbf{U}(x, y, \vec{z}) = 0]]$ (see, e.g., Matijasevič [77] (reprinted with an addendum in [82]) and Jones [78, 79]). $\mathbf{U}$ can be completely multiplied out and the resultant terms, those with and those without minus signs, can be separated so that, for Diophantine polynomials $\mathbf{U}^1$, $\mathbf{U}^2$ with *no* terms having minus signs we have:

$$(\forall x, y)[y \in W_x \Leftrightarrow (\exists \vec{z})[\mathbf{U}^1(x, y, \vec{z}) = \mathbf{U}^2(x, y, \vec{z})]]. \tag{35}$$

Importantly: the right-hand side of the $\Leftrightarrow$-statement within (35) just above *can be directly written in the language of* PA, where, then, each $\mathbf{U}^i(x, y, \vec{z})$, $i = 1, 2$, *becomes a term in that language*. In the next paragraph, for the cases of $n \geq 2$: one has to translate simple inequalities between variables into *the language of* PA—easily done employing a single existential (or universal) quantifier; also, as the need arises, it is straightforward to put judiciously into prenex form sentences *of* PA. The $n = 1$ case is easy from (35) and the remark about it right after it: we can show K, the Σ^0_1-*complete* set $\{k \mid k \in W_k\}$, is $\leq_{\mathrm{m}} V_1$. To handle the $n = 2$ case, with (35) and the remark about it right after it, we can show: the Σ^0_2-*complete* set, $\{k \mid W_k \text{ is finite}\}$, is $\leq_{\mathrm{m}} V_2$. For each $n \geq 3$, we can similarly employ (35) *more than once* and the remark about it right after it to show: considering n even and n odd separately, the Σ^0_n-*complete* set easily extracted from Rogers' [1, Corollary XVIII, p. 330] is $\leq_{\mathrm{m}} V_n$. Lastly, we'll directly lift the $\leq_{\mathrm{m}}$-case to the desired $\leq_1$-case. Let f_n be a computable function mapping $\mathbb{N}$ into the set of Σ^0_n-*sentences of* PA and witnessing $\emptyset^{(n)} \leq_{\mathrm{m}} V_n$. We define a 1-1 computable function f'_n witnessing the $\leq_1$-case as follows. We write the Σ^0_n-*sentence of* PA, $f_n(k)$, as $\vec{Q}^n_k[u^n_k]$, where $\vec{Q}^n_k$ is its prefix of quantifiers and $[u^n_k]$ is its matrix (free of quantifiers). Let $\overline{k}$ be the result of a fixed, effective way of making a *numeral of* PA which represents $k \in \mathbb{N}$. Finally, we let $f'_n(k) = \vec{Q}^n_k[\overline{k} = \overline{k} \wedge [u^n_k]]$. We note that our improvement of Rogers' Corollary X also implies each V_n is a cylinder—thanks to Rogers' [1, P. 319] observation that each $\emptyset^{(n)}$ is a cylinder. □

[48] [80, 81] feature axiom sets above-c.e. but low in the Arithmetical Hierarchy.

[49] For some results would we have to resort to lim-effectively, etc?

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

Acknowledgements This paper is dedicated to the memory of Tom McLaughlin who was the present author's Ph.D. advisor many years ago. Furthermore, we'd like to thank Jim Royer for his valuable contribution [63] and for numerous helpful discussions re the present paper. Thanks also to Jim as well as to Greg Haase for typo corrections and nice rhetoric change suggestions.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Rogers, H.: Theory of Recursive Functions and Effective Computability. McGraw Hill, New York(1967). Reprinted, MIT Press, (1987)
2. Rogers, H.: Gödel numberings of partial recursive functions. J. Symb. Log. **23**, 331–341 (1958)
3. Machtey, M., Young, P.: An Introduction to the General Theory of Algorithms. North Holland, New York (1978)
4. Riccardi, G.: The independence of control structures in abstract programming systems. PhD thesis, SUNY Buffalo (1980)
5. Riccardi, G.: The independence of control structures in abstract programming systems. J. Comput. Syst. Sci. **22**, 107–143 (1981)
6. Royer, J.: A Connotational Theory of Program Structure. Lecture Notes in Computer Science, vol. 273. Springer, Berlin, Germany (1987)
7. Odifreddi, P.: Classical Recursion Theory. North-Holland, Amsterdam (1989)
8. Odifreddi, P.: Classical Recursion Theory II. Elsevier, Amsterdam (1999)
9. Post, E.: Recursively enumerable sets of positive integers and their decision problems. Bull. Am. Math. Soc. **50**, 284–316 (1944)
10. Li, X.: Effective immune sets, program index sets and effectively simple sets—generalizations and applications of the recursion theorem. In: Chong, C.T., Wicks, W.J. (eds.) Southeast Asian Conference on Logic, pp. 97–106 (1983)
11. McLaughlin, T.: On a class of complete sets. Can. Math. Bull. **8**, 33–37 (1965)
12. Smullyan, R.: Effectively simple sets. Proc. Am. Math. Soc. **15**, 893–895 (1964)
13. Case, J., Moelius, S.: Program self-reference in constructive Scott subdomains. Theory Comput. Syst. **51**, 22–49 (2012)
14. Case, J.: Infinitary self-reference in learning theory. J. Exp. Theor. Artif. Intell. **6**(1), 3–16 (1994)
15. Case, J.: Periodicity in generations of automata. Math. Syst. Theory **8**(1), 15–32 (1974)
16. von Neumann, J.: Theory of Self–Reproducing Automata. University of Illinois Press, Illinois (1966). Edited and completed by A. W. Burks
17. Case, J.: Directions for computability theory beyond pure mathematical. In: Gabbay, D., Goncharov, S., Zakharyaschev, M. (eds.) Mathematical Problems from Applied Logic II. New Logics for the XXIst Century. , International Mathematical Series, Vol. 5, pp. 53–98. Springer, New York, NY (2007). Invited book chapter
18. Case, J.: Machine self-reference and consciousness. In: proceedings and abstracts of the 3rd annual meeting of the association for the scientific study of consciousness (London, Ontario, 1999). www.eecis.udel.edu/~case/slides/krt-consc-slides.pdf

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

19. Case, J.: Algorithmic scientific inference: Within our computable expected reality. International Journal of Unconventional Computing **8**(3), 192–206 (2012). Invited journal expansion of an invited talk and paper at the *3rd International Workshop on Physics and Computation 2010*
20. Sacks, G.: A simple set which is not effectively simple. Proc. Am. Math. Soc. **15**, 51–55 (1964)
21. Martin, D.A.: Completeness, the recursion theorem, and effectively simple sets. Proc. Am. Math. Soc. **17**(4), 838–842 (1966)
22. Hirschfeldt, D., Jockusch, C., Kjos-Hanssen, B., Lempp, S., Slaman, T.: The strength of some combinatorial principles related to Ramsey's theorem for pairs. In: Computational Prospects of Infinity, Singapore, pp. 143–161 (2008). Chap. 8
23. Birns, S., Kjos-Hanssen, B.: On the degrees of constructively immune sets. In: Connecting with Computability - 17th Conference on Computability in Europe (CiE 2021), Proceedings. LNCS, vol. 12813, pp. 50–59. Springer, Berlin-Heidelberg, Germany (2021)
24. Mendelson, E.: Introduction to Mathematical Logic, 6th edn. Chapman & Hall, London (2015)
25. Heyting, A.: Intuitionism: An Introduction, 3rd edn. North-Holland, Amsterdam (1971)
26. Brouwer-Heyting-Kolmogorov interpretation. en.wikipedia.org/wiki/Brouwer-Heyting-Kolmogorov_interpretation
27. Realizability. en.wikipedia.org/wiki/Realizability
28. Case, J., Ralston, M.: Beyond Rogers' non-constructively computable function. In: Bonizzoni, P., Brattka, V., Löwe, B. (eds.) The Nature of Computation - Ninth Conference of Computability in Europe (CiE 2013), Proceedings. Lecture Notes In Computer Science, vol. 7921, pp. 45–54. Springer, Berlin (2013)
29. Medvedev, Y.: On non-isomorphic recursively enumerable sets. Dokl. Akad. Nauk SSSR **102**, 211–214 (1955)
30. Rice, H.G.: Recursive and recursively enumerable orders. Trans. Am. Math. Soc. **74**, 277–300 (1956)
31. Uspenskii, V.: Some remarks on re sets. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **3**, 157–170 (1957)
32. Arslanov, M.: Effectively hyperimmune sets and majorants. Matematicheskie Zametki **38**, 302–309 (1985)
33. Arslanov, M.: On effectively hypersimple sets. Algebra I Logic **8**, 143–153 (1969)
34. Martin, D.A.: A theorem on hyperhypersimple sets. J. Symb. Log. **28**, 273–278 (1963)
35. Martin, D.A.: Classes of recursively enumerable sets and degrees of unsolvability. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **12**, 295–310 (1966)
36. Robinson, R.W.: Simplicity of recursively enumerable sets. J. Symbol Logic **32**, 162–172 (1967)
37. Daley, R.: On the simplicity of busy beaver sets. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **24**, 207–224 (1978)
38. Pour-El, M.: Independent axiomatization and its relation to the hypersimple set. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **14**, 449–456 (1968)
39. Kreisel, G.: Independent recursive axiomatizability. J. Symb. Log. **22**, 109 (1957)
40. Baliga, G., Case, J.: Learnability: Admissible, co-finite, and hypersimple sets. J. Comput. Syst. Sci. **53**, 26–32 (1996)
41. Joseph, D., Young, P.: Independence results in computer science. J. Comput Syst Sci, 205–222 (1981)
42. Brouwer, L.E.J.: In: van Dalen, D. (ed.) Brouwer's Cambridge Lectures on Intuitionism. Cambridge University Press (1981)
43. Bauer, A.: Constructive gem: irrational to the power of irrational that is rational. math.andrej.com/2009/ (2009)
44. Case, J., Ralston, M., Akama, Y.: A non-uniformly c-productive sequence & non-constructive disjunctions. In: Zhao, X., Feng, Q., Kim, B., Yu, L. (eds.) Proceedings of the 13th Asian Logic Conference, pp. 29–52. World Scientific, Singapore (2015)
45. Akama, Y., Berardi, S., Hayashi, S., Kohlenbach, U.: An arithmetical hierarchy of the law of excluded middle and related principles. 19th annual symposium on logic in computer science (LICS'04), 192–201 (2004)
46. Case, J.: Effectivizing inseparability. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **37**, 97–111 (1991). www.eecis.udel.edu/~case/papers/mkdelta.pdf
47. Smullyan, R.: Theory of formal systems. Annals of Mathematics Studies No. 47 (1961)
48. Blum, M.: On the size of machines. Inf. Control **11**, 257–265 (1967)
49. Meyer, A.: Program size in restricted programming languages. Inf. Control **21**, 382–394 (1972)
50. Bagchi, A.: Topics in complexity theory and recursion theory. PhD thesis, MIT (1972)

Version of record at: https://doi.org/10.1007/s00153-024-00958-x

51. Meyer, A., Bagchi, A.: Program size and economy of description. In: symposium on the theory of computation (1972)
52. Kinber, J.: On btt-degrees of sets of minimal numbers in Gödel numberings. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **23**, 201–212 (1977)
53. Schaefer, M.: A guided tour of minimal indices and shortest descriptions. Arch Math Logic **18**, 521–548 (1998)
54. Owings, J.: Effective choice functions and index sets. J. Comput. Syst. Sci. **32**, 370–373 (1986)
55. Simpson, S.: Subsystems of Second Order Arithmetic, 2nd edn. Cambridge University Press, Cambridge (2010)
56. Stephan, F., Teutsch, J.: Immunity and hyperimmunity for sets of minimal indices. Notre Dame J Form Logic **49**, 107–125 (2008)
57. Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and Its Applications, 3rd edn. Springer, New York, NY (2008)
58. Royer, J., Case, J.: Subrecursive programming systems: complexity and succinctness. Research monograph in Progress in theoretical computer science. Birkhäuser, Boston (1994). See www.eecis.udel.edu/~case/RC94Errata.pdf for corrections
59. Cilibrasia, R., Vitányi, P.: Fast whole-genome phylogeny by compression: the COVID-19 case. Proceedings of the National Academy of Sciences of the United States of America, 1–7 (To appear)
60. Arslanov, A.: On elementary computability-theoretic properties of algorithmic randomness. Electron Notes Theoret Comput Sci **42**, 41–51 (2001)
61. Kummer, M.: On the complexity of random strings. In: Proceedings of the symposium on theoretical aspects of computer science (STACS'96). Lecture Notes in Computer Science, vol. 1845, pp. 25–36. Springer, Berlin (1996)
62. Buhrman, H., Orponen, P.: Random strings make hard instances. J. Comput. Syst. Sci. **8**, 261–266 (1996)
63. Royer, J.: The Set of Kolmogorov Random Strings is Strongly Effectively Immune. www.cis.syr.edu/~royer/seimmune.pdf (2020)
64. Rado, T.: On non-computable functions. Bell Syst Tech J **41**, 877–884 (1962)
65. Yedidia, A., Aaronson, S.: A relatively small Turing machine whose behavior is independent of set theory. Complex Syst **25**(4), 297–327 (2016)
66. Busy Beaver. en.wikipedia.org/wiki/Busy_beaver
67. Blum, M.: A machine independent theory of the complexity of recursive functions. J. ACM **14**, 322–336 (1967)
68. Daley, R.: Busy beaver sets and the degrees of unsolvability. J. Symb. Log. **46**, 468–474 (1981)
69. Daley, R.: Busy beaver sets: characterizations and applications. Inf. Control **52**, 52–67 (1981)
70. Dekker, J., Myhill, J.: Retraceable sets. Can. J. Math. **10**, 357–373 (1958)
71. Yates, C.E.M.: Recursively enumerable sets and retracing functions. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik **8**, 331–345 (1962)
72. Cohen, P., Jockusch, C.: A lattice property of Post's simple set. Ill. J. Math. **19**(3), 450–453 (1975)
73. Markwald, W.: Zur theorie der konstruktiven wohlordnugen. Math. Ann. **127**, 35–149 (1954)
74. Craig, W.: On axiomatizability within a system. J. Symb. Log. **18**, 30–32 (1953)
75. Shoenfield, J.: On degrees of unsolvability. Ann. Math. **69**, 644–653 (1959)
76. Shapiro, N.: Review of "Limiting recursion" by E.M. Gold and "Trial and error predicates and the solution to a problem of Mostowski" by H. Putnam. Journal of Symbolic Logic **36**, 342 (1971)
77. Matijasevič, Y.: Enumerable sets are diophantine. Soviet Mathematics. Doklady **11**(2), 354–358 (1970). Reprinted with an addendum as a book chapter in [82]
78. Jones, J.: Three universal representations of recursively enumerable sets. J. Symb. Log. **43**, 335–351 (1978)
79. Jones, J.: Universal diophantine equation. J. Symb. Log. **47**, 549–571 (1982)
80. Magari, R.: Su certe teorie non enumerabili (Sulle Limitazioni dei sistemi formali, I). Annali di Matematica Pura ed Applicata **98**, 119–152 (1974)
81. Jeroslow, R.: Experimental logics and Δ_2^0-theories. J. Philos. Log. **4**, 253–266 (1975)
82. Sacks, G.: Mathematical Logic in the 20th Century. Singapore University Press and World Scientific, Singapore (2003)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.