COMPUTABLE CATEGORICITY AND THE ERSHOV HIERARCHY

BAKHADYR KHOUSSAINOV, FRANK STEPHAN AND YUE YANG

Abstract. In this paper, the notions of F_α-categorical and G_α-categorical structures are introduced by choosing the isomorphism such that the function itself or its graph sits on the α-th level of the Ershov hierarchy, respectively. Separations obtained by natural graphs which are the disjoint unions of countably many finite graphs. Furthermore, for size-bounded graphs, an easy criterion is given to say when it is computable-categorical and when it is only G_2-categorical; in the latter case it is not F_α-categorical for any recursive ordinal α.

1. Introduction

The notions of n-r.e. sets and n-r.e. degrees occur naturally in computability theory, as they provide a fine hierarchy of approximations of limit recursive functions. The notion of n-r.e. sets was introduced by Putnam [12] and Gold [7] in the middle of 1960's. Soon after Ershov [3, 4, 5] extended the notion of n-r.e. sets for positive natural numbers n to α-r.e. sets for recursive ordinals α. The resulting general structure is now called the *Ershov hierarchy*. The study of the Ershov hierarchy, at both finite and infinite levels, has applications beyond the traditional local degree structures in recursion theory, for example, in the field of inductive inferences, see Ambainis, Freivalds and Smith [1] and Freivalds and Smith [6]. In this paper, we study the categoricity notion in recursive model theory, by calibrating the isomorphisms in terms of Ershov hierarchy. At first look, this seems less natural; however this paper will demonstrate that the hierarchy arises from some very simple mathematical structures, such as graphs and linear orderings.

First let us recall some basic definitions and fix some notations. We begin with the Ershov hierarchy at finite levels.

1991 *Mathematics Subject Classification.* 03C35, 03D45, 03D80.

The first author is partially supported by Marsden Fund grant of the Royal Society of New Zealand for project *Computability, Complexity, and Randomness*, the second author is partially supported by NUS Academic Research Grants R252-000-212-112 and R252-000-308-112, the last author is partially supported by NUS Academic Research Grants R252-000-212-112 and R146-000-078-112 "Enumerability and Reducibility" (Singapore).

Definition 1.1. *Let n be a positive natural number. A set A is said to be n-r.e. if there is a recursive function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for each x,*

(a) $f(x, 0) = 0$;
(b) $A(x) = \lim_s f(x, s)$;
(c) *the cardinality of the set* $\{s : f(x, s + 1) \neq f(x, s)\}$ *is less than or equal to* n.

A is said to be ω-r.e. if for some recursive function $g(m)$, the same conditions hold, except in (c) the number n is replaced by $g(m)$.

In fact, one can extend the Ershov hierarchy beyond level ω to cover all Δ^0_2-sets, namely, those can be approximately by recursive functions in limit. However, due to the non-uniformity of ordinal notations, it is ambiguous to talk about α-r.e. sets when α is large, say larger than ω^3. Even if one would be more careful, one can move the "point of ambiguity" only up but not get rid of it. More precisely, one defines α-r.e. sets for "notations α of ordinals" and not for "ordinals" as the actual power of the "α-r.e. set" depends much of the notation chosen. Rogers [14] gives an overview on recursive ordinals and their notations. Kleene's $\mathcal{O}$ is given by a partial ordering on $\mathbb{N}$ such that for all $\alpha \in \mathcal{O}$, $\{\beta \in \mathbb{N} : \beta \leq_o \alpha\}$ is well-ordered by $<_o$. From now on, α is always a "notation of a recursive ordinal" and not "an ordinal" as otherwise the notion of an α-r.e. set would be ambiguous and undefined for large α.

We now say that a set A is *α-r.e.* if there is a recursive function $f : \mathbb{N} \times \mathbb{N} \to \mathcal{O}$ such that for each $x \in \mathbb{N}$,

(a) $f(x, 0) = n$ for some $n <_o \alpha$;
(b) $f(x) = \lim_s f(x, s)$ exists and $x \in A$ if and only if $f(x)$ and α have the same parity; here the parity of limit ordinal notations is even and successors of even notations are odd and of odd notations are even;
(c) $f(x, s + 1) \leq_o f(x, s)$ for all $s \in \mathbb{N}$.

For natural numbers n, the "traditional n-r.e. sets" would have to be called "n+1-r.e." according to the definition above, but we keep the traditional name "n-r.e." although it is not compatible with above definition. But ω-r.e. coincides with the traditional definition. The next fact states that there is a uniform enumeration of all α-r.e. sets.

Proposition 1.2. *Fix a recursive ordinal α, there is a recursive enumeration $\{f_e(x, s)\}_{e \in \mathbb{N}}$ such that for each α-r.e. set A, there is an e such that f_e satisfies the conditions* (a), (b) *and* (c) *above.*

Proposition 1.2 enables us to diagonalize all α-r.e. sets.

Corollary 1.3. *For any recursive ordinal α, there is a Δ^0_2 (in fact $\alpha + 1$-r.e.) set A which is not an α-r.e. set.*

Besides the above non-uniformity, when applying the notion of n-r.e. (or α-r.e.) sets in recursive model theory, there is a subtlety of a function being n-r.e. or the graph of the function being an n-r.e. set. We shall see the difference when we talk about categoricity.

Definition 1.4. *Let F_α be the class of all functions $f(x)$ such that there is a recursive approximation $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathcal{O}$ such that for each x, if $g(x, s) = (y_1, m_1)$, $g(x, s + 1) = (y_2, m_2)$ and $y_1 \neq y_2$ then $m_2 <_o m_1$; and the first coordinate of the limit $g(x, s)$ is equal $f(x)$.*

Let G_α be the class of all functions f such that the graph of f is an α-r.e. set.

Proposition 1.5. (a) *The class Δ^0_1 of the recursive functions coincides with F_1 and G_1;*
(b) *F_α is a subclass of G_α and for $\{0, 1\}$-valued functions these two classes coincide.*

We now recall some of the basic notions in recursive model theory, stated in the a restrictive context. More general notions applying to wider contexts can be easily found in survey papers in recursive model theory, for example see Goncharov and Khoussainov [9]. We took the restrictive approach because it is sufficient for our purpose, which is to study computable graphs and computable linear orderings.

Definition 1.6. *A computable presentation of a structure (A, E) is given by a bijection $f : \mathbb{N} \to A$ such that the set*

$$\{\langle x, y\rangle : (f(x), f(y)) \in E\}$$

is recursive, where (x, y) is the ordered pair of x and y and $\langle x, y\rangle$ stands for the (standard) code of the pair (x, y).

Equivalently, $(\mathbb{N}, \tilde{E})$ is a computable presentation of (A, E) if and only if $\tilde{E}$ is a recursive subset of $\mathbb{N} \times \mathbb{N}$ and the structures $(\mathbb{N}, \tilde{E})$ and (A, E) are isomorphic.

Clearly any two computable presentations (A_1, E_1) and (A_2, E_2) of the same structure (A, E) are isomorphic, but not necessarily via a recursive isomorphism. The complexity of such isomorphisms are the central question in this paper: *Given two computable presentations of a given structure (A, E), how difficult is the simplest isomorphism between these two presentations?* This question is formalized as follows:

Definition 1.7. *Given a class C of functions, a structure (A, E) is said to be C-categorical if and only if it has computable presentations and any two computable presentations are isomorphic via a function in C. The Δ^0_1-categorical structures are just called "computable-categorical".*

Note that Cenzer, LaForte and Remmel [2] have obtained the same notion independently. They introduced a bit earlier than us the notions of F_α- and G_α-isomorphism. They showed Theorem 1.8 below. That is, they first observed that for all α, β with $\alpha < \beta$ and every pair of presentations X, Y of some structure: if X and Y are F_α-isomorphic then X and Y are F_β-isomorphic and G_α-isomorphic; if X and Y are G_α-isomorphic then X and Y are G_β-isomorphic; if X and Y are G_1-isomorphic then X and Y are F_1-isomorphic. After this observation they showed that in below Theorem 1.8 that no further implication holds.

Theorem 1.8 (Cenzer, LaForte and Remmel [2]). *For all $\alpha \in \{2, 3, 4, \ldots, \omega\}$ there is a structure A_α with computable presentations X_α, Y_α such that X_α and Y_α are F_α-isomorphic but not G_β-isomorphic for any $\beta < \alpha$.*

There is a structure A with two computable presentations X, Y which are G_2-isomorphic but not F_ω-isomorphic.

We use K to denote a fixed many-one complete r.e. set.

2. Categoricity Results on Graphs

In this section, we focus on graphs (V, E) such that V is countable and every connected component of (V, E) is finite. The choice of graphs simplifies the notions and our real motivation is to reveal the fundamental structure differences even at this simple setting. A quick observation yields the following proposition.

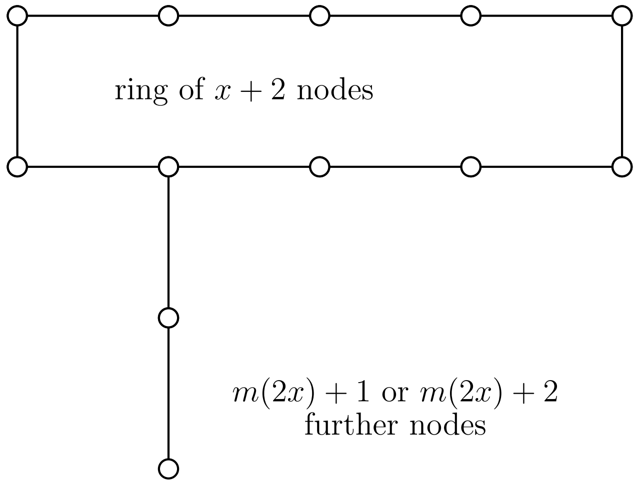
Proposition 2.1. *Let α, β be notations of ordinals with $\alpha < \beta$.*
Every F_α-categorical graph is G_α-categorical.
Every F_α-categorical graph is G_β-categorical.
Every G_α-categorical graph is G_β-categorical.
Every G_α-categorical graph is Δ^0_2-categorical.
Every G_1-categorical graph is computable-categorical.

Now we will show that the separations are witnessed by natural graphs.

Theorem 2.2. *For every ordinal notation α there is a graph $(\mathbb{N}, E_\alpha)$ which is $F_{\alpha+1}$-categorical but not G_α-categorical.*

Proof. Let α be an ordinal notation. Fix a recursive enumeration $\{f_e(\langle x, y\rangle, s)\}_{e\in\mathbb{N}}$ of all graphs of G_α-isomorphism. Fix a recursive function $g(x, s) = (y, m)$, which approximate an $F_{\alpha+1}$-functions such that for any e, $g(2e, s) = (2e, f_e(\langle 2e, 2e\rangle, s) + 1)$. (Remark on notations: We use the usual notation $\alpha + 1$ to denote the successor; thus $f_e(\langle 2e, 2e\rangle, s) + 1$ here means the ordinal notation of the successor of $f_e(\langle 2e, 2e\rangle, s)$.) The intuition is that we will use the number $2e$ to diagonalize f_e which would map $2e$ to $2e$, thus we keep $g(2e, s)$ to be the successor of $f_e(\langle 2e, 2e\rangle, s)$. Let $m(x)$ (not necessarily recursive) be the modulus function of g, that is, $m(x)$ is the least t such that for all $s \geq t$ $g(x, s) = g(x, t)$.

The graph $H = (\mathbb{N}, E_\alpha)$ is defined as follows: For each $x \in \mathbb{N}$, H has two components associated with x: Both components consist of a cycle of $x + 2$ nodes (we need the extra two nodes to form a cycle when $x = 0$) and a chain attached to the cycle; one chain has length $m(2x) + 1$, the other has $m(2x) + 2$. Let us label the one with shorter chain J_x and the other K_x, and label the point where the cycle and chain meet j_x and k_x respectively.



We first argue that the graph is $F_{\alpha+1}$-categorical: Given any two copies L and R of the graph, it suffices to build an $F_{\alpha+1}$-isomorphism from $J_x \cup K_x$ in L to $J_x \cup K_x$ in R for each x. Fix x, wait until the 4 cycles of length $x + 2$ show up, together with four chains attached such that two of which have length at least one, two at least two. Thus we can match them naturally, that gives us a partial isomorphism. We need to change the isomorphism only when the approximation $g(2x, s)$ changes its value. Since $g(2x, s)$ is $F_{\alpha+1}$, the eventually isomorphism is also $F_{\alpha+1}$.

Next we verify that the graph is not G_α-categorical. We define two computable presentations L and R of the graph and satisfy the requirements:

- P_e: The pair J_e and K_e witnesses that f_e is not the isomorphism between L and R.

Construction. We reserve the even numbers $2e$ as j_e in J_e; the rest of the graphs is filled up by odd numbers. Initially we set in L two cycles each of length $e + 2$ and attach a chain of length 1 and 2 to J_e and K_e respectively. Then we calculate $f_e(\langle 2e, 2e\rangle, 0)$; in R attach a length-one chain to K_e and a length-two chain to J_e if $f_e(\langle 2e, 2e\rangle, s)$ and α have the same parity and attach a length-one chain to J_e and a length-two chain to K_e if they have different parity. At stage $s+1$, we act only when $f_e(\langle 2e, 2e\rangle, s+1) \neq f_e(\langle 2e, 2e\rangle, s)$. When we act, we extend in L the chains in both J_e and K_e by one unit (note that the chain in J_e of L remains shorter); however, we do the same in R if $f_e(\langle 2e, 2e\rangle, s+1)$ is of the same parity as $f_e(\langle 2e, 2e\rangle, s)$; we add two units to the shorter chain if $f_e(\langle 2e, 2e\rangle, s+1)$ is of the different parity from $f_e(\langle 2e, 2e\rangle, s)$.

Verification. It is easy to see the construction works. First by the choice of $g(x, s)$, $g(2e, s) \neq g(2e, s+1)$ if and only if $f_e(\langle 2e, 2e\rangle, s+1) \neq f_e(\langle 2e, 2e\rangle, s)$. Thus, we extend the chain in J_e and K_e in L exactly $m(2e)$ times, consequently J_e and K_e in L have length $m(2e)+1$ and $m(2e)+2$ respectively. As $J_e \cup K_e$ in $L[s]$ is isomorphic to $J_e \cup K_e$ in $R[s]$ at every stage s. L and R will be isomorphic copies of the graph. Furthermore, the construction guarantees that the pair $\langle 2e, 2e\rangle$ is in the graph of the limit f_e if and only if J_e in L is isomorphic to K_e in R, because we keep it stage by stage. Therefore, L and R are not G_α-isomorphic. ■

Remark 2.3. When α is finite, there is a more elegant proof based on Kummer's cardinality theorem [10]. Given the number n, one would fix an enumeration of all n-tuples $(x_{m,1}, x_{m,2}, \ldots, x_{m,n})$ and put into the graph for each m two cycles $C_{m,1}, C_{m,2}$ of $m+2$ nodes with a chain added from one node in $C_{m,k}$ which has $k+K(x_{m,1})+K(x_{m,2})+\ldots+K(x_{m,n})$ nodes outside the cycle where $K(y) = 1$ if $y \in K$ and $K(y) = 0$ if $y \notin K$. Now the copies L and R are so arranged that the "root-cycles" (which can be found by exhaustive search) of the components show up in the order $C_{m,1}, C_{m,2}$ in L while the two cycles in the copy R are arranged such that they are alternatingly extended by 2 nodes so that the chain outside the cycle has in the first case an odd number and in the second one an even number of nodes. In other words, if $K(x_{m,1}) + K(x_{m,2}) + \ldots + K(x_{m,n})$ is even then the two cycles are in the order $C_{m,1}, C_{m,2}$ else they are in the order $C_{m,2}, C_{m,1}$. Let $b_L(m)$ and $b_R(m)$ be the branching nodes where the finite chain branches out of the first cycle of length $m+2$ found by exhaustive search. Now an isomorphism from L to R has to map $b_L(m)$ to $b_R(m)$ if and only if $K(x_{m,1}) + K(x_{m,2}) + \ldots + K(x_{m,n})$ is even.

Given any G_{n+1}-isomorphism Φ, one shows that Φ is definitely not a G_n-isomorphism by applying Kummer's cardinality theorem. To see this, consider any n-tuple $(y_1, y_2, \ldots, y_n)$. One can find effectively an m such that $(x_{m,1}, x_{m,2}, \ldots, x_{m,n}) = (y_1, y_2, \ldots, y_n)$ and enumerates now the value $K_s(y_1) + K_s(y_2) + \ldots + K_s(y_n)$ into a set $W_{g(y_1,y_2,\ldots,y_n)}$ if and only if this value is even and $(b_L(m), b_R(m)) \in \Phi_s$ or this value is odd and $(b_L(m), b_R(m)) \notin \Phi_s$. It follows from the correctness of the isomorphism Φ that the value $K(y_1)+K(y_2)+\ldots+K(y_n)$ is enumerated into $W_{g(y_1,y_2,\ldots,y_n)}$. As K is not recursive there is by Kummer's cardinality theorem an n-tuple $(y_1, y_2, \ldots, y_n)$ such that all values $0, 1, 2, \ldots, n$

are enumerated into $W_{g(y_1,y_2,\ldots,y_n)}$. As the values are enumerated in ascending order by the underlying algorithm, this is only possible if $(b_L(m), b_R(m))$ is in Φ_s while an even number is enumerated as stage s and outside Φ_s while an odd number is enumerated at stage s. As $(b_L(m), b_R(m))$ goes into Φ before enumerating 0, it follows that the isomorphism Φ can only be given by an $n+1$-r.e. but not an n-r.e. set. Hence the given class is not G_n-categorical although it is easily seen to be F_{n+1}-categorical.

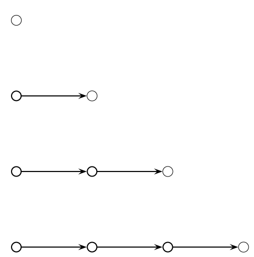
The next result shows that for graphs where all components are finite the oracle K is enough to compute an isomorphism. That is, all such graphs are Δ^0_2-categorical.

Theorem 2.4. *Every graph with a computable presentation and only finite components is Δ^0_2-categorical.*

Proof. Let G be a graph with a computable presentation such that every component of G is finite. Let H be another computable presentation of the same graph. We establish an isomorphism $f \leq_T K$ between G and H component by component by the following back and forth algorithm: Find the smallest $n \in V_G$ such that $f(n)$ is undefined (if no such n exists, then we are done); use the oracle K to identify the whole component containing n, call the component C; then use the same method to search in the graph H an isomorphic component D and define $f : C \to D$ in the obvious way. The finishes the forward direction. Do the same by reversing the roles of G and H to make sure that f is indeed surjective. ■

A natural question is to look for which natural classes there are better isomorphisms, that is, to ask which natural properties of directed graphs give that the corresponding graph is always G_α or F_α-categorical for a suitable α.

We first study the (directed) graph G which is the disjoint union of chains of m nodes for $m = 1, 2, 3, \ldots$; this graph is a natural example of a graph which is not G_α-categorical for any α and provides furthermore a proof that G_α-isomorphisms are not transitive.



This simple example offers us some good insight. First let us fix the notations. For each positive integer m, there is a unique way, up to isomorphism, to linearly order m elements. We will refer it as the chain of m elements. Clearly G has the standard presentation $G_S = (\mathbb{N}, E)$ where $(i, j) \in E$ if and only if for some $k \in \mathbb{N}$

$$\frac{k(k+1)}{2} \leq i, j < \frac{(k+1)(k+2)}{2} \quad \text{and } i < j.$$

Proposition 2.5. *Any computable presentation H of G is G_2-isomorphic to the standard presentation G_S.*

Proof. It suffices to define a G_2-isomorphism between the chain of m elements C_m in G_S and its counterpart D_m in H. We may assume that at any stage s, the chain of m elements in H is unique. Simply match the two chains C_m and D_m in the natural way. Since any new chain, say D'_m in H is necessarily disjoint with D_m, our isomorphism defined this way is indeed G_2. ■

Theorem 2.6. *The graph G is not G_α-categorical for any α; to be more precise, for each fixed ordinal notation α, there are computable presentations L and R which are not G_α-isomorphic.*

Proof. Although one can give a proof by direct diagonalization, we present a proof which pushes up the complexity of the isomorphisms by coding a Δ^0_2-set A into it. To be more precise. Fix a Δ^0_2-set A with recursive approximation $f(x, s)$ such that $\lim_s f(x, s) = A(x)$. We build two recursive presentations L and R of G such that for any isomorphism $h : L \to R$, the set A is many-one reducible to h.

We reserve the even numbers $2x$ for coding purposes. An even number always appears at the starting point of a chain and the rest of the chain is filled out by odd numbers; we may also have chains formed entirely by odd numbers. Let us use $C(x)$ to denote the chain with starting point x. The main idea is to monitor the approximation $f(x, s)$. Whenever $f(x, s) = 1$, we set the chain $C(2x)$ in L the same length as $C(2x)$ in R; when $f(x, s) = 0$, we set the chains to have different length. Thus for any isomorphism h, h maps $2x$ in L to $2x$ in R if and only if $A(x) = 1$.

At each stage $s + 1$, find the least number $x \leq s$ such that either $f(x, s+1) = 1$ and the chains $C(2x)$ in L and R have different length; or $f(x, s+1) = 0$ and the chains have the same length; or there is no chain in L or R containing $2x$. Let y be the least number such that there is no chain of length y in $L[s]$ nor in $R[s]$,

If $f(x, s+1) = 1$, then extend the chains containing $2x$ in L and R to have length y by appending unused odd numbers; for each $z \leq y$ if there is no chain of length z in either L or R, add a chain of length z by using unused odd numbers.

If $f(x, s+1) = 0$, then extend the chain containing $2x$ in L to have length y and the one in R to have length $y + 1$ by appending unused odd numbers; for each $z \leq y+1$ if there is no chain in length z, add a chain of length z in either L or R by using unused odd numbers.

If no chains in $L[s]$ or $R[s]$ contains $2x$, add a chain of length y starting with $2x$ and pending with unused odd numbers.

That finishes our construction.

We now verify that the construction works. First observe that by induction on x and the construction, we have $A(x) = 1$ if and only if the chains containing $2x$ in L and R have the same length. Let h be any isomorphism, since there is a unique way to match two chains of the same length, we have that $x \in A$ if and only if $h(2x) = 2x$. Finally, pick a Δ^0_2-set A which is not in G_α, then the corresponding L and R have no G_α-isomorphism. ■

The combination of Proposition 2.5 and Theorem 2.6 demonstrates some deficiency of the notion of G_2-isomorphism.

Corollary 2.7. *Fix an ordinal notation α. There are structures M_1, M_2 and M_3 such that M_1 is G_2-isomorphic to M_2, M_2 is G_2-isomorphic to M_3 and M_1 is not G_β-isomorphic for any $\beta < \alpha$.*

Proof. Observe that if $f \subset \mathbb{N} \times \mathbb{N}$ is a G_2-isomorphism from G onto H, then f^{-1} is a G_2-isomorphism from H onto G. ■

We also note that it is easy to get the same conclusion on undirected graphs. We now study the case that there is a constant which bounds the number of nodes in any component of the graph. Although the notion of G_2-categorical is not ideal, it turns out that it is closely linked to this natural class of graphs.

Theorem 2.8. *Assume that there is a constant c such that every component of (V, E) has at most c nodes. Then*

(a) *If there are two finite graphs (V_1, E_1) and (V_2, E_2) such that the first is a proper subgraph of the second and both are isomorphic to infinitely many components of (V, E) then (V, E) is G_2-categorical but not F_α-categorical for any α.*

(b) *Otherwise (V, E) is computable-categorical.*

Corollary 2.9. *There is a graph which is G_2-categorical but not F_α-categorical for any notation α.*

Proof. (a) By assumption, there are only finitely many different component graphs $H_k = (V_k, E_k)$, $k = 1, \dots, n$. Define $H_i \prec H_j$ if H_i is a proper subgraph of H_j. Then $\prec$ is a partial order. Without loss of generality, let us assume that $H_1 \prec H_2$ and both occur infinitely often in (V, E).

We first show that (V, E) is G_2-categorical. Given any two computable presentations L and R of (V, E), we define a G_2-isomorphism $h : L \to R$. For those components H_i which occur only finitely often, there is a finite isomorphism h_0 which matches the components H_i in

L to its counterpart in R. By assuming that our h extends the finite function h_0, we may safely ignore those components.

Again we use back and forth argument. Let us assume that L is enumerated in such a way that at any stage s, $L[s] = C_1 < C_2 < \ldots < C_s$ and where C_i is isomorphic to one of the H_i; and $<$ is some fixed canonical ordering of finite sets. Assume the same for $R[s] = D_1 < D_2 < \ldots < D_s$.

At stage s, the "forth" part of the construction goes as follows: Find the least $i < s$ such that there is a node $x \in C_i[s]$ such that $h_s(x)$ is undefined. Search for the R-side, find the least $D_j[s]$ such that $j > i$, $C_i[s]$ is isomorphic to $D_j[s]$, $h_t[C_i[t]] \cap D_j[s] = \emptyset$ for all $t < s$ (here $h_t[C_i[t]]$ stands for the image of the component $C_i[t]$ under the partial isomorphism h_t; this is to ensure that $h_s[C_i]$ has no overlap with its history) and $h_s[C_e[s]] \cap D_j[s] = \emptyset$ for all $e < i$ (this is to respect the higher priority components). Define $h_s[C_i[s]] = D_j[s]$; and for all $i' > i$, let $h_s[C_{i'}]$ be undefined. Similarly do the backward direction.

We show that h is an isomorphism between L and R. Fix a component C in L. Let s_0 be the stage such that for any $t > s_0$, C is $C_i[t]$. Let $s_1 > s_0$ be the least stage after which no C_j and D_j ever acts for $j < i$. Then $h[C_i]$ will be defined by the stage when a new copy of C shows up in R. Furthermore h is in G_2 since h has no overlap with its history.

Next we show that (V, E) is not F_α-categorical for any α.

Fix a computable presentation L of (V, E). We build another presentation R of (V, E). All we need is to use $H_1 \prec H_2$ to diagonalize all potential F_α-isomorphisms h_e, where $\{h_e\}_{e\in\mathbb{N}}$ is a fixed enumeration of all F_α-functions. Again we assume that the graph L was enumerated until some component in H_i shows up. At stage s we say that the function h_e needs attention if

(a) h_e has no component as its witness; or
(b) h_e has C_i as its witness, but $C_i[s]$ is not isomorphic to H_1; or
(c) None of (a) and (b) happens; and h_e is an isomorphism between $C_i[s]$ and $D_j[s]$ for some j.

If no functions need attention, for each component H_i, add a copy in R. Suppose that h_e needs attention. If it is due to (a), find the next (in canonical order) copy of C_i which is isomorphic to H_1, and is not a witness of any h_d for $d < e$, declare C_i is the witness of h_e. If it is due to (b), declare that h_e has no component as its witness. If it is due to (c), extend $D_j[s]$ to H_2.

Verification: We use induction on e to show that each h_e only requires attention finitely often and is not an isomorphism from L onto R.

It remains to show statement (b). Assume that for all i, j, $H_i \not\preccurlyeq H_j$, namely any two components are mutually non-embedible. Given two computable presentations of G, we define a recursive isomorphism as follows. Enumerate both presentations. By assumption, every point is uniquely contained in a component H_i, we can simply wait until the component shows up in both copies, and match them one by one. ■

The next example uses the concept of supersimple sets. It is based on the observation that when constructing an r.e. set whose complement is given by the final positions of markers $x_0, x_1, x_2, \ldots$ one can enforce that for any list $f_0, f_1, f_2, \ldots$ of uniformly K-recursive functions $x_{n+1} > f_m(y)$ for all $y \leq x_m$ and $m \leq n$. This is done by considering approximations $f_{m,t}$ at time t to f_m and to update the markers (initialized as $x_{n,0} = n$ for all n) as follows:

> At time t one determines the least n such that there is a $y \leq x_{n,t}$ and a $m \leq n$ with $f_{m,t}(y) \geq x_{n+1,t}$ and then updates
>
> $$x_{k,t+1} = \begin{cases} x_{k,t}, & \text{if } k \leq n; \\ t+k+1, & \text{if } k > n. \end{cases}$$
>
> If no such n is found then $x_{k,t+1} = x_{k,t}$ for all t.

It is straightforward to verify that the set

$$S = \{y : \exists t > y \, \forall n \leq y \, [x_{n,t} \neq y]\}$$

defines an r.e. set whose complement consists of the numbers $x_n = \lim_t x_{n,t}$ and which satisfies the required properties with respect to the given numbering $f_0, f_1, f_2, \ldots$ of uniformly K-recursive functions. The construction is well-known and supersimple sets have been studied by various authors like McLaughlin [11].

Example 2.10. *Consider the graph X of the form*

$$\circ \qquad \circ \longrightarrow \circ \qquad \circ \qquad \circ \longrightarrow \circ \qquad \circ \qquad \circ \longrightarrow \circ \qquad \circ \qquad \circ \longrightarrow \circ$$

where the nodes $3n+1, 3n+2$ are connected and nothing else. Then this graph is G_2-categorical but not F_α-categorical for any notation α of a recursive ordinal.

Proof. By the preceding theorem, this graph is G_2-categorical but not F_α-categorical for any α. The main motivation for this example is that the negative part can be witnessed by an easy argument.

Consider a uniformly K-recursive list $f_0, f_1, \ldots$ of functions covering all potential F_α-isomorphisms from X to any other potential presentation. Then there is a supersimple set S such that its non-elements $x_0, x_1, x_2, \ldots$ satisfy $f_m(3k) < x_n$ for all $m, k \leq n$. The goal is now to construct a relation such that the x_k are the only once not connected to any other node and then any f_n fails to map $\{0, 3, 6, \ldots, 3n\}$ to $n+1$ different numbers x_k.

The relation is constructed by selecting an infinite recursive subset R of S, taking an ascending enumeration $r_0, r_1, r_2, \ldots$ of R (which is recursive) and any further recursive enumeration $s_0, s_1, s_2, \ldots$ of $S - R$. Then one makes the connections $s_k \to r_k$ for all k and no other ones; the resulting graph is recursive as $r_k \geq k$ for all k and all members of S are connected to some other member of S while $x_0, x_1, x_2, \ldots$ remain unconnected. ■

3. Graphs with infinite components

Infinite components increase the complexity of the isomorphism between computable presentations. The next example is a very natural graph which is G_3-categorical but not G_2-categorical; further examples will separate the corresponding levels relative to the halting problem K.

Theorem 3.1. *The graph consisting of one ω-chain and infinitely many* 0*-chains* (*single points*) *is G_3-categorical but not G_2-categorical.*

Proof. Given two recursive presentations $L = (\mathbb{N}, E_L)$ and $R = (\mathbb{N}, E_R)$, let a_0 and b_0 be the starting points of the ω-chain in L and R respectively. We build an isomorphism h as follows.

At stage 0, define $h(a_0) = b_0$.

We say that a number x is *no longer a singleton* at some stage s, if there is a $y \leq s$ such that the unordered pair $\{x, y\}$ is in E_L or E_R; otherwise x is still a singleton. First observe that if x is no longer a singleton, then x must show up uniquely in the ω-chain, we can wait for the ω-chains on both L and R long enough, then put x in the domain or the range of h naturally.

Thus, we focus on the set of singletons in L and R at stage $s+1$.

Case 1. There is a (least) pair $\langle a, b\rangle \in h$ such that either a or b becomes a non-singleton at stage $s+1$. Then extract $\langle a, b\rangle$ out of h.

Case 2. Case 1 does not happen. Then pick the least singleton a not in the domain of h, and the least singleton $b > a$ on R-side, b is not in the range of h at stage $s+1$, enumerate $\langle a, b\rangle \in h$. Symmetrically, pick the least singleton b not in the range of h, and the least singleton $a > b$ on L-side, enumerate $\langle a, b\rangle \in h$.

Clearly, h is an isomorphism, since singletons are mapped to singletons bijectively. It remains to show that h is G_3. Consider any pair $\langle a, b\rangle$, if one of a, b is in the ω-chain, then $\langle a, b\rangle$ will stay in h forever once it is enumerated in h. Thus, let us monitor $\langle a, b\rangle$ before one of them becomes a non-singleton. Clearly $\langle a, b\rangle$ can only be enumerated in h by Case 2. After that it is extracted out only when one of them become non-singleton. Thus $\langle a, b\rangle$ can only enter h again due to the extension of the ω-chain, and remains in h forever.

We now show that the graph is not G_2-isomorphic. This time we build two computable copies $L = (\mathbb{N}, E_L)$ and $R = (\mathbb{N}, E_R)$ and diagonalize all possible G_2-isomorphisms. We assume that the reader knew the tree construction in recursion theory. Although the proof we present looks like an infinite injury argument, one may have a finite injury proof without using trees. We prefer the tree construction because of its simplicity.

We reserve even numbers for building ω-chains in L and R. At each stage s, either we enumerate the pair $\{2s, 2s+2\}$ in both E_L and E_R; or for some odd numbers x and y, we enumerate $\{2s, x\}$ and $\{x, 2s+2\}$ in E_L, and $\{2s, y\}$ and $\{y, 2s+2\}$ in E_R. We will refer this as *enumerate* x (or y, respectively) *into the ω-chain* in L (or R, respectively).

Claim. The graph built this way is recursive.

Proof of Claim. Given any pair a and b, if both of them are odd, then $\{a, b\} \notin E$; if one of them is even, say $a = s$, then $\{a, b\} \in E$ if and only if $\{a, b\} \in E[s+1]$. This establishes our Claim.

Fix a recursive enumeration of all G_2-functions $\{\varphi_e(\langle a, b\rangle, s)\}_{e\in\mathbb{N}}$. We further assume that at stage s, $\varphi_e[s]$ is injective, that is, for each a there exists at most one b such that $\varphi_e(\langle a, b\rangle, s) = 1$. We need to satisfy the requirements:

- P_e: φ_e is not an isomorphism between L and R.

Let us discuss the strategy to satisfy a single strategy say P_0.

(1) Pick a *witness* x_0 together with a bound $n_0 > x_0$.
(2) Wait for a stage s such that for all singleton $n \le n_0$, there is some m, $\varphi_0(\langle n, m\rangle, s) = 1$. (If no such stage found, then P_0 is satisfied in an obvious way.) Say $\langle x_0, y_0\rangle \in \varphi_0[s]$. We will refer to it as x_0 *is connected to* y_0 by φ_0.
(3) Pick a new odd number say z. Enumerate z and y_0 into the ω-chains of L and R respectively. Now since y_0 is no longer a singleton, φ_0 must connect x_0 to some new element, say y_1. There are two possibilities.
  (3.1) y_1 was connected to some x_1 by φ_0 at some earlier stage. Then the pair $\langle x_1, y_1\rangle$ has finished the "in-out" cycle of φ_0. Enumerate x_1 and y_1 into the ω-chains. We then defeat φ_0, because the unique isomorphism between the ω-chains must send x_1 to y_1, which cannot be achieved by φ_0.
  (3.2) y_1 is a new singleton. Then we update the new bound n_0 to be bigger than y_1, return to (2). If we loop forever between (3.2) and (2), then φ_e is undefined at x_0, again P_e is satisfied.

The outcomes of the strategy are: w indicating that we wait at (2) forever; ∞ indicating that we loop forever between (3.2) and (2); d indicating that we have achieved (3.1). We order them from left to right by $d < \infty < w$.

Let us discuss how to satisfy P_1 with the presence of P_0.

The strategy of P_1 below the outcome d of P_0 simply ignores P_0, as φ_0 is either not G_2 or not an isomorphism.

The strategy below ∞ will work on the singletons which have been connected by φ_0. Under the outcome ∞, all singletons will be connected by φ_0. A P_1 strategy picks its x_1 and work through (2) and (3) as described in P_0. The action of P_1 can send P_0 back to waiting, as P_1 may put the some element y into the ω-chain of R, thus the element which was connected to y by φ_0 needs to be reconnected by φ_0. P_0 can injure P_1 if φ_0 connects x_0 to y_1 which is the one connected to x_1 by φ_0. In this case, we put x_1 and y_1 into the ω-chains in L and R respectively. However, when this happens, P_0 will have outcome d.

The strategy below w will leave the numbers less than or equal to n_0 unchanged, because any change might put P_0 back to outcome ∞. So P_1 selects its witness x_1. If x_1 is connected to some $y_0 < n_0$ by φ_1, then P_1 will pick a new witness x_1', unless x_1 is connected to some element larger than n_0. Since we assume that at any stage s, $\varphi_1[s]$ is injective, eventually P_1 will have its fixed witness x_1. Once the permanent x_1 is chosen, P_1 can act exactly as P_0, since P_0 has no actions.

We organize our construction on a *priority tree* T. Each node α on T has three out-going edges, labeled from left to right $d < \infty < w$; and if α is of height e, then α is working for the requirement P_e. Each α is associated with the following parameters: A witness x_α, a list of nodes β such that $\beta\hat{\ }\infty \subset \alpha$, a number n which plays the role of n_0 in the description earlier, and a number r such that the numbers less than r are reserved for higher priority requirements. When we *initialize* a node, we mean we start over the strategy and set all parameters to be undefined.

Construction. At stage s, we specify a node of length less than or equal to s, called the *accessible string*, and describe the actions along the accessible string.

The root of the priority tree is accessible at any stage s. Suppose that α is the current accessible node. If the length of α is greater than s, then end the stage by initializing all nodes to the right of α.

Suppose that α is of length $e \leq s$. If α was never accessed since it was initialized for the last time, then pick its witness x_α and n_α to be larger than any r set by $\beta\hat{\ }w \subseteq \alpha$ and less than any n set by $\beta\hat{\ }\infty \subseteq \alpha$. Let $\alpha\hat{\ }w$ be accessible.

Thus, let us assume that α was accessible, say at stage s^-, since it was initialized.

- If at stage s^-, we had outcome d, then let $\alpha\hat{\ }d$ be accessible, no other action is taken.
- If at stage s^-, we had outcome ∞. Then if for all $z = \langle x, y\rangle$ such that $x \leq n_0$, $\varphi_e(z, s) = \varphi_e(z, s^-)$, then enumerate the element y_e which is connected to x_e, together with a new odd number u

into the ω-chains to R and L respectively. Set n_0 to be the least number we have not mentions so far, and let w be the outcome.

- If at stage s^-, we had outcome w. Then if x_α is connected to some $y < r$ by φ_e, cancel this x_α, pick a new x_α as described earlier.

  If for some $x < n_0$, $\varphi_e(x)$ is undefined, let w be the outcome.

  If for all $x < n_0$, $\varphi_e(x)$ is defined, then we act based on the following cases:

  Case 1. x_α is connected to some number larger than $n_0[s^-]$, then let $\alpha\hat{}\infty$ be accessible.

  Case 2. x_α is connected to some number less than or equal to $n_0[s^-]$, say y. Find the least x which was connected to y at stage s^-, enumerate x and y into the ω-chain of L and R respectively. Let $\alpha\hat{}d$ be accessible. Initialize all nodes extending and to the right of $\alpha\hat{}d$.

That ends the construction.

Verification. We verify that every requirement P_e is satisfied. First it is routine to check that the truth path exists. Fix a requirement P_e, let α be the node on the true path which is a strategy for P_e. Suppose that $\alpha\hat{}d$ is on the true path, then P_e is clearly satisfied.

Suppose that $\alpha\hat{}\infty$ is on the true path. Let s_0 be the least stage after which $\alpha\hat{}\infty$ is never initialized. By construction, α must have the following sequence of outcomes ∞, w, ∞, $w, \ldots$ which implies that $n_0[s]$ goes to infinity when s goes to infinity. Therefore x_e is connected to larger and larger elements by φ_e. Thus φ_e is not an isomorphism.

Suppose that $\alpha\hat{}w$ is on the true path. Let s_0 be the least stage after which $\alpha\hat{}w$ is never initialized. Then $n_0[s] = n_0[s_0]$ for all $s > s_0$. Then at any stage $s > n_0$ at which α is accessible, there is some $x < n_0[s_0]$, x is not connected by φ_e. Again it shows that φ_e is not an isomorphism. ∎

Note that this principle can be generalized to the case that there are infinitely many copies of two finite graphs X, Y in a graph such that Y is a proper extension of X. The next result deals with the corresponding separation relative to the oracle K. First one deals with a graph consisting of one m-chain for each m and infinitely many ω-chains. The main ideas behind it is that one can enumerate relative to K the indices roots of chains (with one outgoing but without any incoming edge) and then extend an isomorphism of the roots to one of the whole graph. The set of roots of finite chains is K-r.e. and one has to connect two roots of finite chains on the two sides if and only if the corresponding chains have the same length. This permits to relativize the G_3-algorithm from Theorem 3.1 into a G_3^K-algorithm for the next example where the approximations whose mind changes are counted can be K-recursive instead of recursive. Furthermore, one can also translate the negative results into this setting; here note that the construction will only use some but not all m-chains, hence one might have to insert additional m-chains in addition to the roots used to play the K-recursive diagonalization game (by first assuming that they are roots of ω-chains and later updating them to roots of finite chains whenever needed).

Example 3.2. *Consider the graph which consists of an m-chain for every $m \in \mathbb{N}$ and countably many ω-chains. This graph is G_3^K-categorical but not F_α^K-categorical for any recursive ordinal notation α.*

The next example has infinitely many m-chains for every m; hence one can build a G_2^K-isomorphism by choosing a suitable not yet used partner whenever a node turns out to be a root-node of a finite chain. The diagonalization against F_α^K-isomorphisms can be done using the analogue of supersimple sets relative to K. Now one builds a standard copy plus a copy where the root-nodes are the even numbers and where for a given K-supersimple set S diagonalizaing against a desired collection of uniformly K'-recursive functions the chain starting with $2x$ is an ω-chain if and only if $x \notin S$. Note that every K-recursive ordinal is recursive and thus it is sufficient to consider recursive notations of ordinals.

Example 3.3. *The graph consisting of countably many m-chains for each $m \in \{0, 1, 2, \ldots, \omega\}$ is G_2^K-categorical but not F_α^K-categorical for any notation of a recursive ordinal α.*

4. Linear Orders

It is well-known that dense linear orders without end-points are computable-categorical. Indeed, one can generalize this result to the following.

Theorem 4.1 (Goncharov and Dzgoev 1980 [8]; Remmel 1981 [13])**.** *A linear order $(L, <)$ is computable-categorical if and only if L is countable and has only finitely many successive pairs.*

Here (x, y) is a successive pair if and only if $x, y \in L$, $x < y$ and no element of L is between x and y.

Theorem 4.2. *A linear order $(L, <)$ is F_α-categorical if and only if L is countable and has only finitely many successive pairs.*

Proof. As the proof is of the same idea as in Remmel 1981 [13], we only sketch the idea for the non-trivial direction.

Suppose that L has infinitely many successive pairs, we build an isomorphic linear ordering R to diagonalize against all F_α-isomorphisms $f_e(x, s)$. To ensure the isomorphism, at each stage s we have a partial isomorphism $g[s]$ from $L[s]$ to $R[s]$.

The plan for diagonalization is to pick a successive pair (l, l'), wait for f_e mapping it to a successive pair say (r, r'), then put an element in between r and r' while maintaining the partial isomorphism g. There are three main concerns: (1) Being a successive pair is not recursive; (2) For each x, $\lim_s g(x)[s]$ must exist, thus we cannot change $g(x)$ too often; and (3) We need a successive pair (l, l') which allows us to diagonalize against it sufficiently many times.

(1) and (2) are solved exactly as in Remmel's proof. To deal with (1), notice that being a successive pair is a Π^0_1-property, we can recursively approximate and discover the non-successive ones. To deal with (2), our strategy for P_e will not change $g(y)$ for $y \in \{0, 1, \dots, e\} \cup g^{-1}[s]\{0, 1, \dots, e\}$. Suppose those points partition L into finitely many intervals $I_0, I_1, \dots, I_k$ for some k. Since we have infinitely many successive pairs, one of them must appear in an interval which has infinitely many points. Therefore we are able to use this pair to diagonalize against f_e.

(3) is the place where we need to modify, but just a little bit. In Remmel's proof, we only need to act against f_e once, say f_e maps (l, l') to (r, r') and after we successfully put some element between r and r', we are done. In our case, after our action, f_e may map (l, l') to some other pair (q, q'). To make matter worse (q, q') may fall into some I which has only finitely many elements, thus we have to give up (l, l'). However it is easy to see that eventually we are able to find a pair (l_*, l'_*) such that f_e maps it always to some interval which has infinitely many elements. Thus we are able to act against f_e unboundedly many times. That finishes our explanation of proof. ■

Corollary 4.3. *Every F_α-categorical linear order is also computable-categorical.*

We give a natural example of a non-G_α-categorical linear orderings.

Theorem 4.4. *The linearly ordered set $(\mathbb{N}, <)$ is Δ^0_2-categorical but not G_α-categorical for any α.*

Proof. Let $f_e(\langle x, y\rangle)$ be a recursive list of all functions whose graph are G_α-functions with recursive approximation $f_e(\langle x, y\rangle, s)$. We construct two computable presentations L and R of $\mathbb{N}$ and diagonalize against all f_e. For this purpose, we divide $\mathbb{N}$ into disjoint finite intervals I_x, and each interval is used solely by one requirement. We will guarantee that if f_e is an isomorphism from L to R, then f_e must match one interval I_x in L to another one in R of the same length. To distinguish the two intervals, let us use C_x for intervals in L and D_x in R. We need to satisfy the following requirements:

- P_e: There is some $x \in \mathbb{N}$ such that f_e is not an isomorphism between C_x and D_x.

The strategy to satisfy an individual P_e is very simple: Pick x to be the center point of some odd-length interval C in L; wait for f_e to match C with its counterpart D in R. Then $f_e(x)$ must be the center point of D, call it y. We then can force $\langle x, y\rangle$ out of the graph of f_e by adding one element at the beginning of the interval C and one element at the end of D. To force $\langle x, y\rangle$ back into the graph of f_e, we then add one element at the end of C and one element at the beginning of D. This process can go at most finitely many often. There is no conflict between strategies as different requirements work on different intervals. We now give the detailed stage by stage construction.

We define two graphs L and R which are isomorphic to $(\mathbb{N}, <)$. At any stage s, we have finite orderings $L[s] \cong R[s]$ and $L[s] = C_0 \cup \ldots \cup C_s$ and $R[s] = D_0 \cup \ldots \cup D_s$, where C_i and D_i are interval of equal length; and $C_i \cap C_j = \emptyset$ and $D_i \cap D_j = \emptyset$, and if $i < j$ then every element is C_i (and D_i, respectively) is less than every element in C_j (and D_j, respectively). We also have a partial isomorphism $g_s : L[s] \to R[s]$.

At stage $s + 1$, we say that the requirement P_e *requires attention* if P_e has the interval C_i as its witness and $f_e[s]$ maps C_i isomorphically to D_i.

Find the least P_e for $e \leq s$ which needs attention.

If there is no e such that P_e requires attention, then let $C_{s+1} = D_{s+1} = \{z\}$ where z is the least number not in $L[s]$; let $L[s + 1] = L[s] \cup C_{s+1}$ and $R[s + 1] = R[s] \cup D_{s+1}$. If for some (least) $e \leq s$ P_e has not interval as its witness, declare that the interval C_{s+1} is P_e's witness.

Thus let us assume that P_e requires attention and has the interval C_i as its witness. Let y be the least number not in $L[s]$. Form the new linear order $L[s+1]$ and $R[s+1]$ by adding the new point y to C_i and D_i as follows:

- If the length of $C_i[s]$ is odd, append y to the beginning of the interval $C_i[s]$ and append y to the end of $D_i[s]$;
- if the length of $C_i[s]$ is even, append y to the end of $C_i[s]$ and append y to the beginning of $D_i[s]$.

That ends our construction.

It remains to use induction to verify that every requirement P_e requires attention only finitely many times, and its witness intervals have the same length in L and R and is eventually satisfied. Let s_0 be the stage after which no requirements P_d for $d < e$ ever require attention. The intervals having the same length follows from construction. At stage $s_0 + 1$, P_e will have an interval say C as its witness. If P_e requires attention infinitely often, then infinitely often C has odd length and C matches with its counterpart D via f_e, in particular, f_e maps x which is the center of C to x' the center of D, because the starting point of C has to match with the starting point of D (here we have used the induction hypothesis that all intervals before C have the same length with their counterparts). Let us monitor the pair (x, x'): When C has odd length, $(x, y) \in \text{Graph } f_e$; when C has even length, $(x, y) \notin \text{Graph}$

f_e. This implies that (x, y) enters the graph of f_e infinitely often, a contradiction. Thus P_e will eventually stop actions and is satisfied. ■

References

[1] Andris Ambainis, Rūsiņš Freivalds and Carl H. Smith. Inductive inference with procrastination: back to definitions. *Fundamenta Informaticae*, 40(1):1–16, 1999.

[2] Douglas Cenzer, Geoffrey LaForte and Jeffrey Remmel. Equivalence structures and isomorphisms in the difference hierarchy. Manuscript, 2007.

[3] Yuri L. Ershov. A certain hierarchy of sets. I. *Algebra i Logika*, 7(1):47–74, 1968.

[4] Yuri L. Ershov. A certain hierarchy of sets. II. *Algebra i Logika*, 7(4):15–47, 1968.

[5] Yuri L. Ershov. A certain hierarchy of sets. III. *Algebra i Logika*, 9:34–51, 1970.

[6] Rūsiņš Freivalds and Carl H. Smith. On the role of procrastination in machine learning. *Information and Computation*, 107(2):237–271, 1993.

[7] E. Mark Gold. Limiting recursion. *The Journal of Symbolic Logic*, 30(1):28–48, March 1965.

[8] Sergey S. Goncharov and Valeriy D. Dzgoev. Autostability of models. *Algebra i Logika*, 19(1):45–58, 132, 1980.

[9] Sergey S. Goncharov and Bakhadyr Khoussainov. Open problems in the theory of constructive algebraic systems. In *Computability theory and its applications* (*Boulder, CO, 1999*), volume 257 of *Contempary Mathematics*, pages 145–170. American Mathematical Society, Providence, RI, 2000.

[10] Martin Kummer. A proof of Beigel's cardinality conjecture. *The Journal of Symbolic Logic*, 57(2):677–681, 1992.

[11] Thomas G. McLaughlin. Supersimple sets and the problem of extending a retracing function. *Pacific Journal of Mathematics*, 41(2):485–494, 1972.

[12] Hillary Putnam. Trial and error predicates and the solution to a problem of Mostowski. *The Journal of Symbolic Logic*, 30(1):49–57, March 1965.

[13] Jeff Remmel. Recursive categorical linear orderings. *Proceedings of the American Mathematical Society*, 83:383–391, 1981.

[14] Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, New York, 1967.

B. Khoussainov, Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand.

Email address: `bmk@cs.auckland.ac.nz`

F. Stephan, Department of Mathematics, Faculty of Science, National University of Singapore, 2 Science Drive 2, Singapore 117543.

Email address: `fstephan@comp.nus.edu.sg`

Y. Yang, Department of Mathematics, Faculty of Science, National University of Singapore, 2 Science Drive 2, Singapore 117543.

Email address: `matyangy@nus.edu.sg`