Quasiminimal pairs for c.e. degrees of generic and coarse reducibilities

Alexander Rybalov
Sobolev Institute of Mathematics, Pevtsova 13, Omsk 644099, Russia
Omsk State Technical University, prospekt Mira 11, Omsk 644050, Russia

E-mail: alexander.rybalov@gmail.com

Abstract. Generic approach to algorithmic problems in combinatorial group theory was suggested in 2003 by Kapovich, Myasnikov, Schupp and Shpilrain. This approach deals with algorithms, which solves algorithmic problems on "most" of the inputs (i.e., on a generic set) instead of the entire domain and output undefined answer (do not halt) on the rest of inputs (a negligible set). Generic analog of Turing reducibility was introduced by Jockusch and Schupp in 2012. Later Hirschfeldt, Jockusch, Kuyper and Schupp defined coarse reducibility. In this paper we prove that there exist quasiminimal pairs for generic and coarse reducibilities of computably enumerable (c.e.) degrees. The work was supported by Russian Science Foundation, grant number 18-71-10028.

1. Introduction

Generic approach to algorithmic problems in combinatorial group theory [10] was suggested in 2003 by Kapovich, Myasnikov, Schupp and Shpilrain. This approach deals with algorithms, which solves algorithmic problems on "most" of the inputs (i.e., on a generic set) instead of the entire domain and output undefined answer (do not halt) on the rest of inputs (a negligible set). The concept of almost all can be formalized by introducing asymptotic density on input data set. Thus, it may be that the problem is difficult to solve or even undecidable in the classical sense, but easily decidable in the generic sense. In the works of Myasnikov, Remeslennikov, Borovik, Romankov, Kapovich, Schupp, Dickert, Kambites [10, 1, 2, 5, 6, 9, 11, 12] has been proven that many algorithmically unsolvable problems of algebra are generically easy decidable.

At the same time, there is great interest both from a theoretical point of view and from the point of view practical applications, to find algorithmic problems that remain undecidable or intractable in the generic case. For example, in modern cryptography such problems are interesting, which, being (hypothetically) difficult in the classical sense, they remain difficult in the generic sense i.e. for almost all inputs. This is because with random key generation in the cryptographic algorithm, the input of some difficult algorithmic problems underlying the algorithm, is generated. If the problem is generically easy to solve, then for almost all such inputs we can quickly resolve and the keys will almost always be bad. Therefore, the problem must be difficult for almost all inputs. For example, classical algorithmic cryptography problems have this behavior: problem of recognition of quadratic residues, discrete logarithm problem, root extraction problem in residue groups (problem of inversion of the RSA function). Examples of generically undecidable problems were constructed by Myasnikov and Rybalov in [13].

Generic analog of Turing reducibility was introduced by Jockusch and Schupp in [7]. Later Hirschfeldt, Jockusch, Kuyper and Schupp [8] defined coarse reducibility. Classical theorem of Lachlan and Yates states that there exists a minimal pair of computably enumerable (c.e.) degrees [3, 4, 14, 16, 17]. Jockusch and Schupp defined also a notion of quasiminimal pair of generic degrees — an important particular case of minimal pairs. In this paper we prove that there exist quasiminimal pairs for generic and coarse reducibilities of c.e. sets.

2. Preliminaries

Define for a subset $A \subseteq \mathbb{N}$ the following sequence

$$\rho_n(A) = \frac{|\{x : x \leq n,\ x \in A\}|}{n},\ n = 1, 2, 3, \ldots$$

We will call the following limit (if it exists)

$$\rho(A) = \lim_{n\to\infty} \rho_n(A)$$

by *asymptotic density* of set $A \subseteq \mathbb{N}$. We will call the set $A \subseteq \mathbb{N}$ *generic* if $\rho(A) = 1$ and *negligible* if $\rho(A) = 0$. It is obviously that A is generic if and only if $\overline{A}$ is negligible.

We will call an algorithm $\mathcal{A} : \mathbb{N} \to \mathbb{N} \cup \{?\}$ *effective generic* if

(i) $\mathcal{A}$ stops on every input from $\mathbb{N}$,
(ii) set $\{x \in \mathbb{N} : \mathcal{A}(x) =?\}$ is negligible.

Generic algorithm $\mathcal{A}$ computes a function $f : \mathbb{N} \to \mathbb{N}$ if

$$\forall x \in \mathbb{N}\ \mathcal{A}(x) = y \in \mathbb{N} \Rightarrow f(x) = y.$$

We will call a subset $A \subseteq \mathbb{N}$ *effectively generically computable (decidable)* if there is an effective generic algorithm, computing its characteristic function.

A set $A \subseteq \mathbb{N}$ is *effectively generically reducible* to a set $B \subseteq \mathbb{N}$ (we will denote it by $A \leq_{eg} B$), if there is an algorithm $\mathcal{A}$ with call as a subprogram of any function (generic oracle) $\varphi_A : \mathbb{N} \to \{0, 1, ?\}$ such that

(i) Set $\{x : \varphi_A(x) =?\}$ is negligible,
(ii) $\forall x \in \mathbb{N}\ \varphi_A(x) = 1 \Rightarrow x \in A$,
(iii) $\forall x \in \mathbb{N}\ \varphi_A(x) = 0 \Rightarrow x \notin A$,

which is an effective generic algorithm computing the characteristic function of A. We will denote the fact that $A \leq_{eg} B$ and $B \not\leq_{eg} A$ by $A <_{eg} B$.

Let S be a subset of $\mathbb{N}$ with characteristic function χ_S. A partial function $\varphi : \mathbb{N} \to \{0, 1\}$ is called a *generic description* of S if $\varphi(x) = \chi_S(x)$ whenever $\varphi(x)$ is defined and the domain of φ is generic. A set $S \subseteq \mathbb{N}$ is called *generically computable* if there exists a partial computable function φ, which is a generic description of S. Otherwise S is called *generically undecidable*.

An *enumeration operator* is a c.e. set. If W is an enumeration operator, the elements of W are viewed as coding pairs $\langle n, D\rangle$, where $n \in \mathbb{N}$ and D is a finite subset of $\mathbb{N}$ identified with its canonical index $\sum_{k\in D} 2^k$. We view W as the mapping from sets to sets

$$X \to W(X) = \{n : \exists D\ \langle n, D\rangle \in W \& D \subseteq X\}.$$

If Ψ is a partial function, let

$$\gamma(\Psi) = \{\langle a, b\rangle : \Psi(a) = b\},$$

so $\gamma(\Psi)$ is a set of natural numbers coding the graph of Ψ. A set $A \subseteq \mathbb{N}$ is *generically reducible* to a set $B \subseteq \mathbb{N}$ (written $A \leq_g B$) if there is an enumeration operator W such that, for every generic description Ψ of B

$$W(\gamma(\Psi)) = \gamma(\Theta)$$

for some generic description Θ of A. We write $A <_g B$ if $A \leq_g B$ and $B \not\leq_g A$.

A *coarse description* of a set $A \subseteq \mathbb{N}$ is a set $B \subseteq \mathbb{N}$ such that the symmetric difference

$$A \triangle B = (A \setminus B) \cap (B \setminus A)$$

is negligible. A set $A \subseteq \mathbb{N}$ is *coarsely computable* if there is its computable coarse description. A set $A \subseteq \mathbb{N}$ is *coarsely reducible* to a set $B \subseteq \mathbb{N}$ (written $A \leq_c B$) if there is a Turing functional Φ such that if D is a coarse description of B, then Φ^D is a coarse description of A. We write $A <_c B$ if $A \leq_c B$ and $B \not\leq_c A$.

3. Main results

A pair of c.e. sets A and B is called *minimal pair*, if

(i) A and B are not computable,

(ii) for any c.e. set C if $C <_T A$ and $C <_T B$, then C is computable.

Further we need a construction from [7]. For all $k \in \mathbb{N}$ define the following set

$$R_k = \{m \in \mathbb{N} : 2^k \mid m,\ 2^{k+1} \nmid m\}.$$

It is easy to see that

$$\rho(R_k) = 2^{-(k+1)}$$

and

$$R_k \cap R_l = \varnothing$$

for all $k \neq l$. Now for any set $S \subseteq \mathbb{N}$ define

$$\mathcal{R}(S) = \bigcup_{k \in S} R_k.$$

In [7] it was proven (Lemma 4.6), that for all sets A, B it holds

$$A \leq_T B \Leftrightarrow \mathcal{R}(A) \leq_g \mathcal{R}(B).$$

A pair of c.e. sets A and B is called *quasi-minimal pair* for generic reducibility, if

(i) A and B are not generically computable,

(ii) for any c.e. set C if $\mathcal{R}(C) <_g A$ and $\mathcal{R}(C) <_c B$, then C is generically computable.

Theorem 1. *There is a quasiminimal pair of c.e. sets for generic reducibility.*

Proof. Let A and B be a c.e. minimal pair for Turing reducibility. Then it is easy to see that $\mathcal{R}(A)$ and $\mathcal{R}(B)$ is a quasiminimal pair for generic reducibility. □

We will prove an analog of Lemma 4.6 from [7] for effective generic reducibility.

Lemma 1. *For all sets A, B it holds*

$$A \leq_T B \Leftrightarrow \mathcal{R}(A) \leq_{eg} \mathcal{R}(B).$$

Proof. Let $A \leq_T B$ by a machine M_1 with classical oracle B. Then a machine M_2, realizing reducibility $\mathcal{R}(A) \leq_{gT} \mathcal{R}(B)$ with arbitrary generic oracle $\varphi_{\mathcal{R}(B)}$ works on input $x \in \mathbb{N}$ in the following way.

(i) Find k such that $x \in R_k$.

(ii) Run M_1 on k and for every oracle command $y \in B?$, performs the following:
   (a) Find m such that $y \in R_m$.
   (b) Let
   $$R_m = \{r_0, r_1, \ldots\}$$
   be an effective enumeration of set R_m in ascending order.
   (c) $i := 0$.
   (d) Compute $\varphi_{\mathcal{R}(B)}(r_i)$.
   (e) If $\varphi_{\mathcal{R}(B)}(r_i) \neq ?$, then output a correct answer $y \in B?$ and go to the computation of machine M_1.
   (f) If $\varphi_{\mathcal{R}(B)}(r_i) = ?$, then $i := i + 1$ and go to step (d).

Since the set R_m is not negligible, then there is a number j such that $\varphi_{\mathcal{R}(B)}(r_j) \neq ?$ and the described procedure correctly models the oracle command $y \in B?$. So machine $M_2^{\varphi_{\mathcal{R}(B)}}$ realizes a total algorithm for computing of characteristic function of set A, and of set $\mathcal{R}(A)$. That means $\mathcal{R}(A) \leq_{gT} \mathcal{R}(B)$.

Conversely, let $\mathcal{R}(A) \leq_{gT} \mathcal{R}(B)$ by some machine M_1 with arbitrary generic oracle $\varphi_{\mathcal{R}(B)}$. In particular, we can choose total (classical) oracle for set $\mathcal{R}(B)$, and we can model commands $a \in \mathcal{R}(B)?$ by the following procedure. At first, we find k such that $a \in R_k$, then ask $k \in B?$. So we have a machine M_2 with classical oracle B, realizing a generic algorithm for recognition of set $\mathcal{R}(A)$. Now a machine M_3 with oracle B, recognizing set A, works on $x \in \mathbb{N}$ in the following way. Enumerates elements $r_1, r_2, \ldots$ of set R_x in ascending order until it find an element r_i such that $M_2^B(r_i) \neq ?$. Such element exists because set R_x is not negligible. Obviously $x \in A \Leftrightarrow M_2^B(r_i) = 1$. That means $A \leq_T B$. □

A pair of c.e. sets A and B is called *quasi-minimal pair* for effective generic reducibility, if

(i) A and B are not effectively generically computable,

(ii) for any c.e. set C if $\mathcal{R}(C) <_g A$ and $\mathcal{R}(C) <_c B$, then C is effectively generically computable.

Theorem 2. *There is a quasiminimal pair of c.e. sets for effective generic reducibility.*

Proof. Let A and B be a c.e. minimal pair for Turing reducibility. Then by Lemma 1 $\mathcal{R}(A)$ and $\mathcal{R}(B)$ is a quasiminimal pair for effective generic reducibility. □

Further we need a construction from [7]. For all $k \in \mathbb{N}$ define the following set

$$I_k = [k!, (k+1)!).$$

Now for any set $S \subseteq \mathbb{N}$ define

$$\mathcal{I}(S) = \bigcup_{k \in S} I_k.$$

In [8] it was proven (Proposition 2.3), that for all sets A, B it holds

$$A \leq_T B \Leftrightarrow \mathcal{I}(A) \leq_c \mathcal{I}(B).$$

A pair of c.e. sets A and B is called *quasi-minimal pair* for coarse reducibility, if

(i) A and B are not coarsely computable,

(ii) for any c.e. set C if $\mathcal{I}(C) <_g A$ and $\mathcal{I}(C) <_c B$, then C is coarsely computable.

Theorem 3. *There is a quasiminimal pair of c.e. sets for coarse reducibility.*

Proof. Let A and B be a c.e. minimal pair for Turing reducibility. Then it is easy to see that $\mathcal{I}(A)$ and $\mathcal{I}(B)$ is a quasiminimal pair for coarse reducibility. □

[1] Borovik A V and Myasnikov A G and Remeslennikov V N 2007 The conjugacy problem in amalgamated products I: regular elements and black holes *International Journal of Algebra and Compututation* **17 (7)** pp 1299–1333
[2] Borovik A V and Myasnikov A G and Remeslennikov V N 2007 Generic complexity of the conjugacy problem in HNN-extensions and algorithmic stratification of Miller's groups *International Journal of Algebra and Compututation* **17 (963)** pp 963–997
[3] Cooper B 2003 Computability Theory *Chapman and Hall/CRC* pp 424
[4] Cutland N 1980 Computability: An Introduction to Recursive Function Theory *Cambridge University Press* pp 264
[5] Diekert V and Myasnikov A G and WeißA 2016 Conjugacy in Baumslag's group, generic case complexity, and division in power circuits *Algorithmica* **4 (76)** pp 961–988
[6] Diekert V and Myasnikov A G and WeißA 2017 Amenability of Schreier graphs and strongly generic algorithms for the conjugacy problem *Journal of Symbolic Computation* **83** pp 147–165
[7] Jockusch C and Schupp P 2012 Generic computability, Turing degrees, and asymptotic density *Journal of the London Mathematical Society* **85 (2)** pp 472–490
[8] Hirschfeldt D and Jockusch C and Kuyper R and Schupp P 2016 Coarse reducibility and algorithmic randomness *Journal of Symbolic Logic* **81** pp 1028–1046
[9] Kambites M 2011 Generic Complexity of Finitely Presented Monoids and Semigroups *Computational complexity* **20 (1)** pp 21–50
[10] Kapovich I and Myasnikov A and Schupp P and Shpilrain V 2003 Generic-case complexity, decision problems in group theory and random walks *Journal of Algebra* **264 (2)** pp 665–694
[11] Kapovich I and Schupp P 2005 Genericity, the Arzhantseva-Ol'shanskii method and the isomorphism problem for one-relator groups *Mathematische Annalen* **331** pp 1–19
[12] Miasnikov A and Schupp P 2017 Computational complexity and the conjugacy problem *Computability* **4 (6)** pp 307–318
[13] Myasnikov A and Rybalov A 2008 Generic complexity of undecidable problems *Journal of Symbolic Logic* **73 (2)** pp 656–673
[14] Rogers H 1987 Theory of Recursive Functions and Effective Computability *MIT Press* pp 506
[15] Rybalov A 2019 On a generic Turing reducibility of computably enumerable sets *Journal of Physics: Conference Series* **1210** pp 1–5
[16] Soare R 1987 Recursively Enumerable Sets and Degrees *Springer-Verlag Berlin Heidelberg* pp 437
[17] Soare R 2016 Turing Computability: Theory and Applications (Theory and Applications of Computability) *Springer* pp 263