# Enumerations and Torsion Free Abelian Groups*

Alexander G. Melnikov

Sobolev Institute of Mathematics, Novosibirsk, Russia
vokinlem@bk.ru

**Abstract.** We study possible spectrums of torsion free Abelian groups. We code families of finite sets into group and set up the correspondence between their algorithmic complexities.

## 1 Introduction

Studying model theory and theory of algorithms gives us another branch of science - computable model theory. We say that the model is computable if it's main set, predicates and functions are recursive, and all functions and predicates are effectively enumerated. We may think these models as "the only ones that can be applied in computer science and that can be presented on some computer" or "the ones we can exactly imagine" etc. Note that we don't think about time or space complexity of algorithms - this is another subject for studying.

Starting with abstract computable models, we try to apply some results or their variations to effective algebra. In particular, computable fields, Boolean algebras and groups are widely studying.

We generalize the notion of computable model replacing in it's definition all words "recursive" by "X-recursive", where $X$ is some countable set. That means that we can ask someone on some steps of given program whether $x \in X$ or not, for arbitrary $x$. It's not easy to imagine, how can we apply it in computer science. We can think that this "oracle" is some physical experiment - but is there any physicist who knows everything about halting problem?.. That means that we need some methods to answer the question:

**Question.** *Let $\mathcal{A}$ be a model, and suppose that $\mathcal{A}$ has copies, computable in a fixed family of Turing degrees respectively. Does it necessarily follow that $\mathcal{A}$ has a computable copy?*

If $\mathcal{A}$ is a Boolean algebra, and it has a low copy, then the answer is "yes"[1]. There is another related question:

**Question.** *Given a structure $\mathcal{A}$ what can we say about $\{deg(\hat{\mathcal{A}}) : \hat{\mathcal{A}} \simeq \mathcal{A}\}$?*

For an arbitrary structure, this family of degrees (called *degree spectrum*) can be enough complicated and reach, but does not contain **0**-degree or low degrees. Wehner [7] built a graph that has presentations exactly in non-recursive degrees

* Partially supported by President grant of Scientific School NSh-4413.2006.1.

S.B. Cooper, B. Löwe, and A. Sorbi (Eds.): CiE 2007, LNCS 4497, pp. 566–574, 2007.

(see also [4] for alternative proof). Miller R. [3] built a linear order, that has all noncomputable $\Delta_2^0$ - copies, but does not have a computable one. These results give us examples of algorithmic "anomaly" and shows the variety and richness of pure and applied computable algebra.

We study possible spectrums of torsion free Abelian groups. For torsion free Abelian group the key notion is its rank. In this paper we study algorithmic properties of groups in the case of infinite rank, and we obtain the following:

**Theorem.** *For any family $R$ of finite sets there exists a torsion free Abelian group $G_R$ of infinite rank, such that $G_R$ has X-computable copy iff $R$ has $\Sigma_2^X$-computable enumeration.*

Interpretation of a family, that can be obtained by relativization of Wehner's result, gives us the corollary:

**Theorem.** *There exists a torsion free Abelian group $G$ of infinite rank, such that $G$ has X-computable copy iff $X' >_T 0'$, i.e. has exactly nonlow copies.*

## 2 Basic Notions

We need some basic notions and facts from computability theory, theory of groups and computable model theory. For better background see also [5], [6] and [2]. We suppose that the reader knows the elementary properties of recursive functions and recursively enumerable sets.

**Definition 1.** *A set $A$ is recursive with respect to a set $B$ ($A \leq_T B$), if its characteristic function is $B$-recursive. That means that it can be computed by Turing machine with "oracle" $B$. If $A \leq_T B$ and $B \leq_T A$ then $A \equiv_T B$. It's obvious that $\equiv_T$ is the relation of equivalence. The equivalence classes for $\equiv_T$ are called degrees.*

**Definition 2.** *Let $K = \{x \colon \Phi_x^A(x) \downarrow\} = \{x \colon x \in W_x^A\}$. This set is denoted by $A'$ and called the jump of a set $A$.*

Index $A$ in $\Phi_x^A(x)$ means that $\Phi$ is (partially) recursive with respect to a set $A$. It's clear how to define the n-th jump of A ($A^{(n)}$) using the same construction for $A^{(n-1)}$. Jump is well-defined on degrees, and iteration of jumps induces hierarchy, that is called arithmetical:

$$X \in \Sigma_n^Y \leftrightarrow \text{X is r.e. in } Y^{(n-1)}.$$

**Definition 3.** *We say that a set $A \leq_T \emptyset'$ is low if $A' \equiv_T \emptyset'$, and it is n-low if $A^{(n)} \equiv_T \emptyset^{(n)}$.*

**Definition 4.** *Let $R = \{R_i | i \in \omega\}$ and $\nu : \omega \to^{on} R$. The set $S_\nu \leftrightharpoons \{\langle n, i\rangle | n \in \nu(i)\}$ is called enumeration of $R$. We also refer to $\nu$ as an enumeration of $R$, using a simple fact, that having $S_\nu$ we can recollect the map $\nu$ and vice versa.*

*We will follow the tradition of enumeration theory in defining computable enumeration:*

*Enumeration* $\nu$ *is called* $\Sigma_n^X$*-computable if* $S_\nu \in \Sigma_n^X$ *(and X-computable if* $S_\nu \in \Sigma_1^X$*).*

Now let G be a countable group.

**Definition 5.** *A group* $\langle G, \cdot \rangle$ *is called computable group if* $| G | \subseteq N$ *is a recursive set and the operation* $\cdot$ *is presented by some recursive function.*

We define $A$-computable groups by substitution of the word "recursive" by "$A$-recursive" in the definition above.

**Definition 6.** *Let* $\langle G, +, 0 \rangle$ *be a torsion free Abelian group (i.e for all* $a \neq 0$, $0 \neq n \cdot a \leftrightharpoons \underbrace{a + a + ... + a}_{n}$*). The elements* $g_0, ..., g_n \in G$ *are linearly independent if, for all* $c_0, ..., c_n \in Z$*, the equality* $c_0 g_0 + c_1 g_1 + ... + c_n g_n = 0$ *implies that* $c_i = 0$ *for all i. An infinite set is linearly independent if every finite subset is linearly independent. A maximal linearly independent set is called a basis, and the cardinality of any basis is called the rank of G.*

As for vector spaces, it can be proved that the notion of rank is proper, i.e. all maximal linearly independent sets have the same cardinality.

Fix a canonical listing of prime numbers:

$$p_1, p_2, ..., p_n, ...$$

**Definition 7.** *Let* $g \in G$*. Then* $p^k | g \leftrightharpoons (\exists h \in G)(p^k h = g)$ *and*

$$h_p(g) = \begin{cases} max\{k : p^k | g\}, \text{if this maximum exists,} \\ \infty, \text{else.} \end{cases}$$

*The infinite sequence* $\chi(g) = (h_{p_1}(g), ..., h_{p_n}(g), ...)$ *is called the characteristic of element g.*

Now we are ready to define one of the basic notions in Abelian groups theory.

**Definition 8.** *Given two characteristics,* $(k_1, ..., k_n, ...)$ *and* $(l_1, ..., l_n, ...)$*, we say that they are equivalent,* $(k_1, ..., k_n, ...) \simeq (l_1, ..., l_n, ...)$*, if* $k_n \neq l_n$ *only for finite different n, and only if these* $k_n$ *and* $l_n$ *are finite. This relation is obviously an equivalence relation, and the corresponding equivalence classes are called types.*

It can be easily proved that linear dependant elements has the same type. That means that we can give a proper definition of *type of group* in the case of *rank 1*. The following theorem is the key result for torsion free Abelian groups of rank 1:

**Theorem 1 (Baer, see [6]).** *Let G and H be torsion free Abelian groups of rank 1. Then G is isomorphic to H iff they have the same type.*

*Proof (sketch).*
We can choose any nonzero $g \in G$ and $h \in H$, and it will be necessarily $\chi(g) \simeq \chi(h)$. Then we extract the finite number of roots receiving $g' \in G$ and $h' \in H$ with identical characteristics. We define isomorphism $\varphi : G \to H$ starting with $g' \to h'$.

## 3 Constructing The Group

**Theorem 2.** *For any family $R$ of finite sets there exists a torsion free Abelian group $G_R$ of infinite rank, such that $G_R$ has $X$-computable copy iff $R$ has $\Sigma_2^X$-computable enumeration.*

*Proof.*
**Notation.** *We fix the family of finite sets denoted by $R$ and it's $\Sigma_2^X$-computable enumeration $\nu^X$ with corresponding $S_\nu^X \leftrightharpoons \{\langle n,i\rangle | n \in \nu^X(i)\} \in \Sigma_2^X$. Without loss of generality, we can assume that $\emptyset \in R$.*

*We will use the fact, that every $\Sigma_2^X$-relation can be presented as $\{\langle i,k\rangle : (\exists^{<\infty} x) P^X(x,\langle i,k\rangle)\}$, where $P^X$ is some recursive in $X$ relation. We fix $T^X$ such that*

$$S_\nu^X = \{\langle i,k\rangle : (\exists^{<\infty} x) T^X(x,\langle i,k\rangle)\}.$$

The **scheme of proof** is the following:

Given a $\Sigma_2^X$-computable enumeration $\nu^X$ of $R$ we build a r.e. in $X$ presentation $G^X \subseteq Q^\omega$ of group $G_R = \bigoplus_{k\in\omega}\bigoplus_{m\in\omega} G_{k,m}$, where $rank(G_{k,m}) = 1$. Since $Q^\omega$ is computable, then $G^X$ must have a computable copy as a r.e. subgroup (see [2]).

Then, to make inverse step, we need to construct some $\Sigma_2^X$-computable enumeration if we have some $X$-computable presentation of $G_R$.

This is the **idea** of building a group:

1. Fix a computable listing of prime numbers $\{p_n\}_{n\in\omega}$ and canonical enumeration of (nonempty) finite sets $\{D_n\}_{n\in\omega}$.

2. Build a group such that any $\nu^X(k)$ corresponds to $\omega$ linearly independent elements $g_{k,m}$, and for all $m$, $n$, $k$:

$$\neg(p_n^\infty | g_{k,m}) \Longleftrightarrow (D_n \subseteq \nu^X(k)).$$

3. To build a group, enumerate $T^X(x,\langle i,k\rangle)$ until a *new* $x$ for some pair $\langle i,k\rangle$ appeared. If we have found such $x$, add $p_n$-roots to elements $g_{k,m}$ for all $n$, such that $i \in D_n$.

First we define a procedure **Root(**$\langle i,k\rangle, t^n_{\langle x,i,k\rangle}, Y^n_{\langle x,i,k\rangle}$**)**, that adds prime roots to elements $g_{k,m}$. $Y^n_{\langle x,i,k\rangle}$ is the "memory" of this procedure, and $t^n_{\langle x,i,k\rangle}$ is it's "counter of steps".

**Root(**$\langle i,k\rangle, t^n_{\langle x,i,k\rangle}, Y^n_{\langle x,i,k\rangle}$**)** :
For all $n' \in Y^n_{\langle x,i,k\rangle}$, add $p_{n'}$-root to $g_{k,t^n_{\langle x,i,k\rangle}}$. If $i \in D_{t^n_{\langle x,i,k\rangle}}$, then $Y^{n+1}_{\langle x,i,k\rangle} := Y^n_{\langle x,i,k\rangle} \cup \{t^n_{\langle x,i,k\rangle}\}$ and add $p_{t^n_{\langle x,i,k\rangle}}$-root to $g_{k,m}$, $m \leq t^n_{\langle x,i,k\rangle}$. If $i \notin D_{t^n_{\langle x,i,k\rangle}}$, then $Y^{n+1}_{\langle x,i,k\rangle} := Y^n_{\langle x,i,k\rangle}$.

Finally, let $t^{n+1}_{\langle x,i,k\rangle} := t^n_{\langle x,i,k\rangle} + 1$.

**End of procedure.**

Let **Search(**$\langle i,k\rangle, l$**)** $\leftrightharpoons \mu_x(x \geq l \wedge T^X(x,\langle i,k\rangle))$.

**Construction.**

**Step 0.** Fix a computable presentation of $Q^\omega$ and numbers for $g_{k,m}$ (we can suppose that $g_{k,m}$ is an element of a form $(\underbrace{0,0,...,0,1}_{p_k^m},0,0,...))$.

For all $i,k,x$, let $l^0_{\langle i,k\rangle}=0$, $t^0_{\langle x,i,k\rangle}=0$, $Y^0_{\langle x,i,k\rangle}=\emptyset$.

**Step s.** We denote by $G^X_s$ the part of $G^X$ that has been constructed by the step $s$. For all $\{(g_1,...,g_n): g_i\in G^X_s, g_i\le s, n\le s\}$, add to $G^X_s$ linear combinations $\{m_1g_1+...+m_ng_n: m_i\le s\}$ (if they were not already added).

Make $s$ steps in computation of $Search(\langle i,k\rangle, l^s_{\langle i,k\rangle})$, $\langle i,k\rangle\le s$.

If $Search^s(\langle i,k\rangle, l^s_{\langle i,k\rangle})\downarrow = x$ for some $i,k,x$, then $l^{s+1}_{\langle i,k\rangle}:=x+1$, and $\langle x,i,k\rangle$ *gets attention.* Suppose $R^s_{\langle x,i,k\rangle}\leftrightharpoons Root(\langle i,k\rangle, t^s_{\langle x,i,k\rangle}, Y^s_{\langle x,i,k\rangle})$, and

$$\langle x_1,i_1,k_1\rangle,...,\langle x_j,i_j,k_j\rangle$$

be the listing of all triples, that have got attention by this moment. Perform $R^{x_1,s}_{\langle i_1,k_1\rangle}$, then perform the next, ..., and finally $R^{x_j,s}_{\langle i_j,k_j\rangle}$[1].

**End of construction.**

**Lemma 1.** *$G^X$, built by construction, is torsion free Abelian group and has computable in $X$ copy.*

*Proof.* The first statement is clear: $G\subseteq Q^\omega$ by construction. The second is true because the algorithm of building $G^X$ is effective with oracle $X$, i.e. $G$ is $X$-r.e., and $G\subseteq Q^\omega$, that is computable (again see [2]).

**Lemma 2.** *For any $k$ and $m$, $\neg(p_n^\infty | g_{k,m})$ in $G^X$ iff $(D_n\subseteq \nu^X(k))$.*

*Proof.* $\nu^X(k)$ is finite, and $i\in\nu^X(k)$ iff

$$(\exists^{<\infty}x)T^X(x,\langle i,k\rangle),$$

That means that in the procedure **Search** *after some moment* no "new" $x$ for $\langle i,k\rangle$ will appear **for all** $i\in\nu^X(k)$.

We notice that the existence of such step follows from two key properties: $R$ contains only finite sets and $i\in\nu^X(k)$ iff *"there is only finitely many $x$, such that $T^X(x,\langle i,k\rangle)$."*

We can make a conclusion that after this step, roots that correspond to subsets of $\nu^X(k)$, will not be added by procedures **Root** to elements of a form $g_{m,k}$.

Now let $n\in\{l: D_l\nsubseteq \nu^X(k)\}$. That means that $D_n$ contains $i\notin\nu^X(k)$ and

$$(\exists^\infty x)T^X(x,\langle i,k\rangle),$$

i.e. infinitely many triples of a form $\langle x,i,k\rangle$ will get attantion. Therefore procedures **Root** will add infinetly many $p_n$-roots to elements $g_{m,k}$. This completes the proof of lemma.

[1] Remember that the procedure $Root(\langle i,k\rangle, s, Y^s_{\langle x,i,k\rangle})$ defines the value of $Y^{s+1}_{\langle x,i,k\rangle}$ and $t^{s+1}_{\langle x,i,k\rangle}$.

**Lemma 3.** *Let $\nu^X$ and $\nu^Y$ be enumerations of $R$. Then two groups $G^X$ and $G^Y$ (built using construction for $\nu^X$ and $\nu^Y$ respectively) are isomorphic.*

*Proof.* First fix enumeration $\nu^X$. Notice that $\omega$ identic elements $\{g_{k,m}\}_{m\in\omega}$ (in $G^X$), corresponds to one $\nu^X(k)$, and $g_{k_1,m_1}$ and $g_{k_2,m_2}$ are linearly independent for $\langle k_1, m_1\rangle \neq \langle k_2, m_2\rangle$.

We add roots to $g_{k,m}$ in such a way that $G^X$ is a direct sum:

$$G^X = \bigoplus_{k\in\omega} \bigoplus_{m\in\omega} G^X_{k,m},$$

where $G^X_{k,m}$ corresponds to element $g_{k,m}$ (and therefore codes $\nu^X(k)$, i.e. $G^X_{k,m} \simeq G^X_{k,n}$ for all $m, n$ and fixed $k$), and $rank(G^X_{k,m}) = 1$.

Now we fix $\nu^Y$, and build $G^Y$. We receive a group of the similar form

$$G^Y = \bigoplus_{k\in\omega} \bigoplus_{m\in\omega} G^Y_{k,m}.$$

All sets from $R_i \in R$ are finite, therefore $\nu^X(k)$ and $\nu^Y(t)$, coding the same $R_i \in R$ in enumerations, give us elements of the same type: these elements have only finitely many finite roots, and these roots correspond to the same prime numbers. By Baer Theorem we have the isomorphism of groups of rank 1.

The last problem is to show that the direct sum has the same structure. But by construction we always have exactly $\omega$ subgroups, coding the same $R_i \in R$, even if there are repetitions in coding of $R_i$ in enumeration.

We showed that both $G^X$ and $G^Y$ are isomorphic to

$$G_R = \bigoplus_{k\in\omega} G_k,$$

where $R_k \in R$ is coded by $G_k = \bigoplus_{m\in\omega} G_{k,m}$, $G_{k,m} \cong G_{k,m'}$

Given $\Sigma^X_2$ enumeration of $R$ we can build a r.e. group $G^X$ that has a computable copy. It is a presentation of

$$G_R = \bigoplus_{k\in\omega} \bigoplus_{m\in\omega} G_{k,m}.$$

Now we need an inverse step, i.e. to construct some $\Sigma^X_2$-computable enumeration if we have some $X$-computable presentation of $G_R$.

**Proposition 1.** *There exists an algorithm, that for any computable in $X$ presentation $G^X$ of $G_R$ (defined above for $R$) gives $\Sigma^X_2$-enumeration of $R$.*

*Proof.*

We define $X$-p.r.f. $r$ as follows:

$$r(g, n, k) = \begin{cases} 1, & \text{if } G^X \models p_n^k | g, \\ r(g, n, k)\uparrow, & \text{else.} \end{cases}$$

We define also $X'$-recursive function $\hat{r}$:

$$\hat{r}(g,n,k) = \begin{cases} 1, & \text{if } r(g,n,k) \downarrow = 1, \\ 0, & \text{if } r(g,n,k) \uparrow . \end{cases}$$

Using $\hat{r}$ we can check (with oracle $X'$) the existence of prime roots for any $g \in G^X$. If there is a pair $\langle n, k\rangle$, such that $\hat{r}(g,n,k) = 0$, then $g$ has only finitely many $p_n$-roots.

We identify elements from $G$ with there codes in $G^X$.

**Construction.**

**Step 0:** Let all $m_t^0$ be undefined.

**Step s:**
**Substep s,1:** For $g \in G^X$, such that $g \le s$ and $m_g^{s-1}$ is undefined, compute $\hat{r}(g,m,k)$ for $m, k \le s$. If there exist $g_i, m_i, k_i \le s$, such that $\hat{r}(g_i, m_i, k_i) = 0 \wedge (\forall n < k_i)(\hat{r}(g_i, m_i, n) = 1)$, then suppose $m_{g_i}^s = m_i$ [2]. For every such $g_i$ add to the enumeration all pairs

$$\{\langle j, g_i\rangle : j \in D_{m_{g_i}^s}\}.$$

**Substep s,1:** For $g \in G^X$, such that $g \le s$ and $m_g^{s-1}$ is defined, compute $\hat{r}(g,m,k)$ for $m, k \le s$. If there exist $g_i, m_i, k_i \le s$, such that $\hat{r}(g_i, m_i, k_i) = 0 \wedge (\forall n < k_i)(\hat{r}(g_i, m_i, n) = 1 \wedge D_{m_{g_i}^{s-1}} \subset D_{m_i}$, then for every such $g_i$, add to the enumeration pairs

$$\{\langle j, g_i\rangle : j \in D_{m_i} \setminus D_{m_{g_i}^{s-1}}\},$$

and then suppose $m_{g_i}^s = m_i$.

**End of Construction.**

**Lemma 4.** *Described algorithm builds the enumeration of R.*

*Proof.* Remember that

$$G_R = \bigoplus_{k\in\omega} \bigoplus_{m\in\omega} G_{k,m},$$

where $rank(G_{k,m}) = 1$ and $G_{k,m} \cong G_{k,m'}$ (for any $m, m'$). For every $g_{k,m} \in G_{k,m}$ we have the following:

$$\neg(p_n^\infty | g_{k,m}) \iff (D_n \subseteq R_k^X).$$

Let $g \in G$. Then $g = r_{k_1,m_1} g_{k_1,m_1} + ... + r_{k_t,m_t} g_{k_t,m_t}$ for some $g_{k_1,m_1} \in G_{k_1,m_1}, ..., g_{k_t,m_t} \in G_{k_t,m_t}$. But $G$ is the direct sum, and $g$ is the linear combinayion of linear independent elements. It is easy to see that

$$(\forall k)((\neg p_k^\infty | g) \iff \bigvee_{j=1,\dots,t} (D_k \subseteq R_{k_j})),$$

[2] We can suggest (for every $i$) $m_i$ be the minimal one with this property.

i.e. the characteristic of $g$ is the g.l.b. of characteristics of components. That means that $g$ codes the *union* of all subsets, coded by it's components.

Algorithm let us to move higher and higher along the subsets of some $R_{k_j}$, until we reach this $R_{k_j}$. After we reach it, there will be no new pairs of a form $\langle l, g \rangle$ added in enumeration. That means that we enumerate $R_{k_j}$, and we do it for all elements of $R$, and only this elements could be enumerated.

**Lemma 5.** *Enumeration built by algorithm (for $G^X$) is $\Sigma_2^X$.*

*Proof.* The function $\hat{r}(g, n, k)$ is recursive in $X'$. That means that the Procedure is effective in $X'$, and enumeration is $\Sigma_2^X$.

We set up a correspondence between $\Sigma_2^X$-enumerations of $R$ and computable in $X$ presentations of $G_R$. This completes the proof of theorem.

The following result is one of the possible applications of the previous theorem:

**Theorem 3.** *There exists a torsion free Abelian group $G$ of infinite rank, such that $G$ has $X$-computable copy iff $X' >_T 0'$, i.e. has exactly nonlow copies.*

*Proof (sketch).* First we relativize the result of Wehner [7]. This gives us the family of finite sets that has $\Sigma_2^X$ $(X' >_T 0')$ enumerations, but has no $\Sigma_2$ enumeration. Then we apply construction from previous theorem for this family of sets.

## 4 Questions

We suggest some related problems.

*Question 1.* Is it possible to build a torsion free Abelian group with copies exactly in none-recursive degrees? Can we generalize our second theorem for the case of $low_n$-degrees, for arbitrary $n$?

At the case of finite rank the answer for the first part is "no" (for the second part it is also natural to get "no"). But in general case the question is open - there is no uniform procedure for coding of any given property into torsion free Abelian groups, especially coding respecting effectiveness.

*Question 2.* What can we say about Abelian p-groups? Can we build a p-group with any of these properties?

Studying p-groups from these point of view is very interesting and needs some new ideas and methods to be developed.

## References

1. Downey, R., Jockusch, G.: Every low Boolean algebra is isomorphic to a recursive one. Proc. Amer. Math. Soc 122, 871–880 (1994)
2. Ershov, Y., Goncharov, S.: Constuctive models. Novosibirsk, Nauchnaya kniga (1999)

3. Miller, R.: The $\Delta_2^0$-spectrum of a linear order, preprint.
4. Slaman, T.: Relative to ani non-recursive set. Proc. of the Amer. Math. Soc 126, 2117–2122 (1998)
5. Soare, R.I.: Recursively Enumerable Sets and Degrees. Springer, Berlin Heidelberg (1987)
6. Fuchs, L.: Infinite Abelian Groups. vol. II. Academic Press, San Diego (1973)
7. Wehner, S.: Enumerations, countable structures and Turing degrees. Proc. of the Amer. Math. Soc 126, 2131–2139 (1998)