CARPATHIAN J. MATH.
20 (2008), No. 1, 00-00

GAP algorithms for finite abelian groups and applications

SEPTIMIU CRIVEI AND GABRIELA OLTEANU

ABSTRACT. We propose GAP algorithms which determine finite abelian groups or subgroups of them having certain properties of lattice-theoretic nature. We also show their usefulness for obtaining examples and some theoretical results.

1. INTRODUCTION

Establishing structure theorems or finding suitable examples for certain types of algebraic objects may be a very difficult task. A way to achieve this, when assuming some finiteness conditions, is to use efficient computational methods, in a computer system like GAP [10]. As written in its general description [10], "*GAP - Groups, Algorithms, and Programming* is a system for computational discrete algebra, with particular emphasis on Computational Group Theory. GAP provides a programming language, a library of thousands of functions implementing algebraic algorithms written in the GAP language as well as large data libraries of algebraic objects. GAP is used in research and teaching for studying groups and their representations, rings, vector spaces, algebras, combinatorial structures, and more." GAP is a free software, re-distributable and/or modifiable under the terms of the GNU General Public License. So the GAP system allows one to use the objects from the GAP library and to create his/her own objects or functions for personal use or for the official GAP distribution as a package (i.e., a set of user contributed programs).

One of the basic algebraic structures in GAP is that of a group. But even if many GAP packages have been developed throughout its existence since the early 1990s, there are few packages for specialized properties of abelian groups. That reason and a more general interest in module-theoretic properties have been the main motivation for building a collection of GAP algorithms for finite abelian groups by Crivei, Olteanu and Şuteu-Szöllősi [2]. As usual in the case of such algorithms, they are important tools both for easily obtaining examples requiring a considerable amount of calculations as well as for testing conjectures, before proving them rigorously.

In the present paper we shall present some general features of our algorithms, briefly explain the working structures and some methods we have used, and give a couple of applications, which hopefully will illustrate their power.

Received: date; In revised form: date; Accepted:
2000 *Mathematics Subject Classification.* 20K01, 20E15, 68W30.
Key words and phrases. *Finite abelian group, closed subgroup, closure, type subgroup, Hall subgroup.*
Partially supported by the Romanian grants PN-II-ID-PCE-2007-1 project ID_532, contract no. 29/28.09.2007, and PN-II-ID-PCE-2008-2 project ID_2271.

2. Some terminology

Throughout the paper G will be a finite additive abelian group, that is, a $\mathbb{Z}$-module. Now let us give some sample definitions of notions involved in our algorithms. A subgroup H of G is called *essential* in G if $H \cap K \neq 0$ for every non-zero subgroup K of G. A non-zero group G is called *uniform* if every non-zero subgroup of G is essential in G. A subgroup H of G is called a *complement* (respectively *supplement*) if there exists a subgroup K of G such that H is maximal (respectively minimal) in the set of subgroups L of G such that $K \cap L = 0$ (respectively $K + L = G$). Complement subgroups are also called *closed* subgroups. Note that every direct summand of G is a closed subgroup. The group G is called *extending* (or a *CS-group*) if every closed subgroup of G is a direct summand [5]. A subgroup K of G containing a subgroup H of G is called a *closure* of H in G if H is essential in K and K is closed [5]. The group G is called a *UC-group* if every subgroup of G has a unique closure in G [8]. Two abelian groups are called *orthogonal* if they do not have non-zero isomorphic subgroups. A subgroup H of G is called a *type subgroup* if for every subgroup Y of G strictly containing H, H is orthogonal to some non-zero subgroup X of Y [4]. The group G is called a *TS-group* if every type subgroup is a direct summand [4]. Every type subgroup of G is closed, hence if G is extending, then G is a TS-group. Let us point out the lattice-theoretic nature of all these notions. They are basic terminology in module theory, and especially in the very rich theory of extending and lifting modules (see Dung et al. [5] and Clark et al. [1]) as well as in the modern theory of natural classes of modules (see Dauns and Zhou [4]).

3. Algorithms

The name `ELISA` of our collection of algorithms stands for "`E`xtending `LI`fting `S`ubgroup `A`lgorithms", but its development and applications exceed the initial framework of extending and lifting abelian groups and modules to potentially any module-theoretic notions that refer to submodule lattices. The main present features of `ELISA` are the following ones:

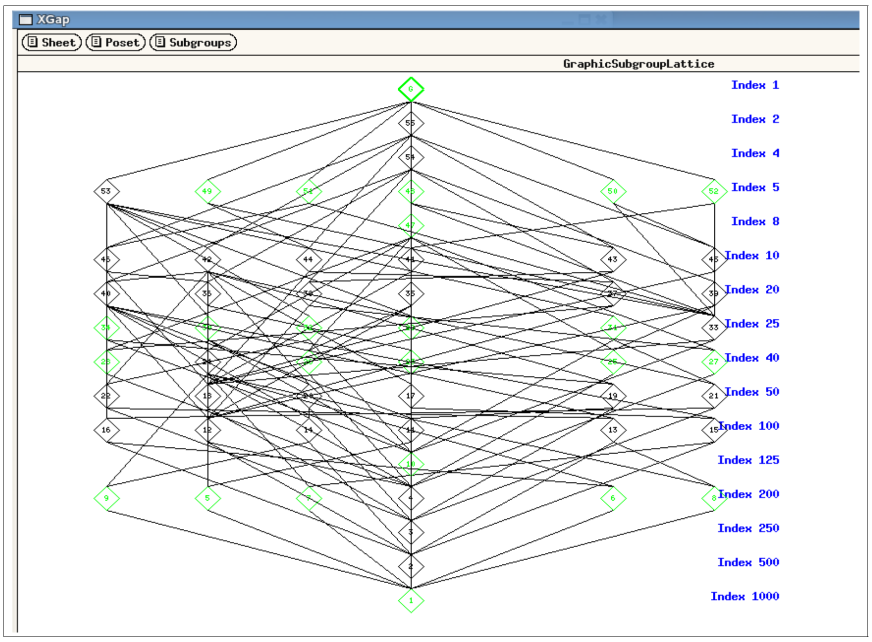
(1) check for a given subgroup the properties of being direct summand, essential, superfluous, coessential, complement (or closed), supplement (or coclosed), type subgroup;
(2) determine for a given group all subgroups with the above properties as well as closures and coclosures;
(3) check for a given group the properties of being extending (or lifting), UC-group (or UCC-group), TS-group.

Some of them have already been described in [3]. The list of functions as well as their source code can be found in the documentation and GAP implementation of `ELISA` [2]. We have used predefined functions from the kernel of GAP, the GAP programming language and suitable mathematical properties for implementation. The main technique is the use of the subgroup lattice of a finite abelian group G, which can be obtained in GAP with the built-in function `LatticeSubgroups(G)`. On one hand, we employ some mathematical properties of the subgroup lattice, such as being self-dual and atomic. On the other hand, we see the subgroup lattice as a directed graph with arcs always pointing "upwards". Some key notions we have used are the neighborhoods of a vertex in a digraph. The point of view of a directed graph allows us to use some classical, but efficient algorithms, such as breadth-first traversal. This combination of methods that work directly on the subgroup lattice makes our algorithms highly efficient. Moreover, GAP has an optional package under Linux, called XGAP, able to visualize the generated subgroup lattice, and the results of our algorithms can be easily spotted out in XGAP.

4. Examples

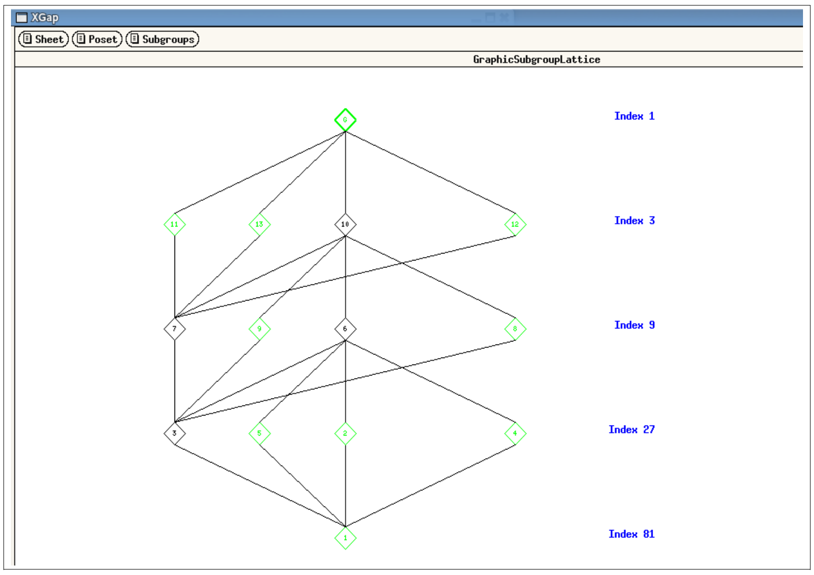
We give some relevant examples to show the usefulness of our algorithms for obtaining examples requiring a considerable amount of calculations and for suggesting theoretical results. They will be used in the last two sections of the paper as a help for proving structure theorems for UC-groups and type subgroups.

Example 4.1. We point out that the results of our functions from `ELISA` are obtained very fast for groups in the vast GAP library, even of considerable size. Consider the group $G = \mathbb{Z}_{200} \oplus \mathbb{Z}_5$ of order 1000, a direct sum of cyclic groups. The XGAP output of the subgroup lattice of G is as follows.



The function `ClosedSubgroups(G)` computes the closed subgroups of G, which are numbered 1, 5, 6, 7, 8, 9, 10, 23, 25, 26, 27, 28, 29, 30, 31, 34, 36, 47, 48, 49, 50, 51, 52 and G, whereas `TypeSubgroups(G)` computes the type subgroups of G, which are numbered 1, 10, 47 and G. Let us note that GAP may provide a description of all these subgroups by generators and relations. The functions `IsExtending(G)` and `IsTS(G)` show that G is extending and a TS-group.

Example 4.2. Let us determine some closures of a subgroup in a group. Consider the group $G = \mathbb{Z}_{27} \oplus \mathbb{Z}_3$. The XGAP output of its subgroup lattice is as follows.



The proper closed subgroups are numbered 2, 4, 5, 8, 9, 11, 12 and 13. If H is the subgroup 7, then the function `Closures(G,H)` gives 3 closures of H in G, which are numbered 11, 12 and 13.

Example 4.3. Using our GAP algorithms we may generate the number of type subgroups and their orders for all abelian groups of order between 51 and 60. The output looks as follows, where CN denotes the cyclic group of order N:

```
C51:             4 type subgroups;   Orders: 1, 3, 17, 51
C52:             4 type subgroups;   Orders: 1, 4, 13, 52
C26 X C2:        4 type subgroups;   Orders: 1, 4, 13, 52
C53:             2 type subgroups;   Orders: 1, 53
C54:             4 type subgroups;   Orders: 1, 2, 27, 54
C18 X C3:        4 type subgroups;   Orders: 1, 2, 27, 54
C6 X C3 X C3:    4 type subgroups;   Orders: 1, 2, 27, 54
C55:             4 type subgroups;   Orders: 1, 5, 11, 55
C56:             4 type subgroups;   Orders: 1, 7, 8, 56
C28 X C2:        4 type subgroups;   Orders: 1, 7, 8, 56
C14 X C2 X C2:   4 type subgroups;   Orders: 1, 7, 8, 56
C57:             4 type subgroups;   Orders: 1, 3, 19, 57
C58:             4 type subgroups;   Orders: 1, 2, 29, 58
C59:             2 type subgroups;   Orders: 1, 59
C60:             8 type subgroups;   Orders: 1, 3, 4, 5, 12, 15, 20, 60
C30 X C2:        8 type subgroups;   Orders: 1, 3, 4, 5, 12, 15, 20, 60
```

For future reference, let us note the following patterns in the above output: (i) the number of type subgroups depends only on the order of the group, and not on its structure (e.g., all groups of order 60, namely $\mathbb{Z}_{60}$ and $\mathbb{Z}_{30} \oplus \mathbb{Z}_2$, have 8 type subgroups); (ii) the number of type subgroups is a power of 2 (2, 4 and 8 in our example); (iii) there is a (unique) type subgroup for any possible subgroup of order relatively prime to the order of the group (e.g., $\mathbb{Z}_{30} \oplus \mathbb{Z}_2$ has type subgroups of order 1, 3, 4, 5, 12, 15, 20, 60, but not of order 2, 6, 10 and 30).

5. UC-GROUPS

As a first theoretical use of our algorithms, we determine the structure of finite abelian UC-groups. It is known that any finite abelian group is a direct sum of its p-components for some primes p. Also, a finite abelian p-group is of the form $G_p \cong (\mathbb{Z}_p)^{k_1} \oplus (\mathbb{Z}_{p^2})^{k_2} \oplus \cdots \oplus (\mathbb{Z}_{p^r})^{k_r}$ for some natural number $r \geq 1$ and some (possible zero) positive natural powers $k_1, \dots, k_r$ [6].

We need the following lemma from [8], which has a straightforward proof.

Lemma 5.1. *Any direct summand of a UC-group is a UC-group.*

Now we are able to prove a structure theorem for finite abelian UC-p-groups. We have Example 4.2 in the background when developing our proof.

Theorem 5.1. *A finite abelian p-group G is a UC-group if and only if G is either semisimple or uniform (i.e., either $G \cong \mathbb{Z}_p^l$ or $G \cong \mathbb{Z}_{p^k}$ for natural numbers $l, k \geq 1$).*

Proof. If G is either semisimple or uniform, then it is clearly a UC-group.

Now suppose that G is neither semisimple nor uniform and show that G is not a UC-group. We proceed in 3 steps. Let $k \geq 2$ be a natural number.

Step 1. We show that $A = \mathbb{Z}_{p^k} \oplus \mathbb{Z}_p$ is not a UC-group. By [9, 21.6], the Jacobson radical $J(A) = \mathbb{Z}_{p^{k-1}}$ of A is superfluous in A. Then $J(A)$ cannot be a supplement, and so it cannot be closed in A by [7, Theorem 4.1.4]. Note that $A/J(A) \cong \mathbb{Z}_p \oplus \mathbb{Z}_p$, hence the subgroup lattice of A has on the level of maximal subgroups the subgroup $\mathbb{Z}_{p^k}$, a subgroup isomorphic to $\mathbb{Z}_{p^{k-1}} \oplus \mathbb{Z}_p$, and $p-1$ subgroups of order p^k isomorphic to $\mathbb{Z}_{p^k}$ which are given by the automorphisms of $\mathbb{Z}_p$. Now all the p subgroups of A of order p^k isomorphic to $\mathbb{Z}_{p^k}$ are closed in A, being complements of $\mathbb{Z}_p$ seen as the second projection of A. They are also uniform, and so $J(A)$ is an essential subgroup of each of them. Hence they are closures of $J(A)$ in A.

Step 2. We show that $B = \mathbb{Z}_{p^k} \oplus \mathbb{Z}_{p^k}$ is not a UC-group. Consider its subgroup $C = \mathbb{Z}_{p^k} \oplus \mathrm{Soc}(\mathbb{Z}_{p^k}) \cong \mathbb{Z}_{p^k} \oplus \mathbb{Z}_p = A$, where $\mathrm{Soc}(\mathbb{Z}_{p^k})$ denotes the socle of $\mathbb{Z}_{p^k}$. As above, $J(C)$ is not closed in C, and so $J(C)$ is not closed in B. As in Step 1 with C instead of A, we deduce the existence of p subgroups of C of order p^k isomorphic to $\mathbb{Z}_{p^k}$. They are closed in B, being complements of $\mathbb{Z}_{p^k}$ seen as the second projection of B, and uniform. Hence they are closures of $J(C)$ in B.

Step 3. Using the general form of a finite abelian p-group, Lemma 5.1 and the first two steps, it follows that G is not a UC-group. $\square$

6. TYPE SUBGROUPS

Now let us show how our algorithms can be used in order to obtain a description of the type subgroups of a finite abelian group. The considerations from Example 4.3 lead us to the following result, which identifies type subgroups and Hall subgroups. Recall that a subgroup H of G is called a *Hall subgroup* if its order and its index are relatively prime.

Theorem 6.2. *A subgroup of G is a type subgroup if and only if it is a Hall subgroup.*

Proof. Let H be a subgroup of G and denote $n = |G|$ and $m = |H|$.

First we suppose that $\gcd(m, \frac{n}{m}) \neq 1$ and show that H is not a type subgroup of G. Let p be a prime dividing $\gcd(m, \frac{n}{m})$. Then there is a subgroup Y of G of order pm. The non-zero maximal subgroups of Y are either subgroups of order m or subgroups of order p^2a for some a. If Z is a subgroup of Y of order m, then Z cannot be orthogonal to H, because Z and H have some isomorphic subgroups of order a prime dividing m. If Z is a subgroup of Y of order p^2a, then Z cannot be orthogonal to H, because Z and H have some isomorphic subgroups of order p. Hence Y has no non-zero maximal subgroup orthogonal to H, and so, no non-zero subgroup orthogonal to H. Therefore, H is not a type subgroup of G.

Conversely, suppose that $\gcd(m, \frac{n}{m}) = 1$. We may assume that $m < n$. Let Y be a subgroup of G strictly containing H. Then $|Y| = mk$ for some k. There is a prime p which divides k and does not divide m (otherwise p would divide both m and $\frac{n}{m}$). Hence Y has a subgroup of order p, say X. Now H is orthogonal to X, because H does not have a subgroup of order p. This shows that H is a type subgroup of G. $\square$

The structure of extending finite abelian groups is well known [5]. Now we are able to discuss this problem for TS-groups.

Corollary 6.1. *Every finite abelian group is a TS-group.*

Proof. Let A be a type subgroup of G. We may assume $A \neq 0$. Let B be a subgroup of G of order $\frac{|G|}{|A|}$. Then by the uniqueness of Hall subgroups of a given order and Theorem 6.2, it follows that $G = A \oplus B$. Hence G is a TS-group. $\square$

References

[1] J. Clark, C. Lomp, N. Vanaja and R. Wisbauer, *Lifting modules. Supplements and projectivity in module theory*, Birkhäuser, Basel, 2006.

[2] S. Crivei, G. Olteanu and Ş. Şuteu-Szöllősi, *ELISA - A collection of GAP algorithms related to extending and lifting abelian groups*. `http://www.gap-system.org/Packages/undep.html` and `http://math.ubbcluj.ro/~crivei/GAP_project.`

[3] S. Crivei and Ş. Şuteu-Szöllősi, *Subgroup lattice algorithms related to extending and lifting abelian groups*, Int. Electron. J. Algebra **2** (2007), 54–70.

[4] J. Dauns and Y. Zhou, *Classes of modules*, Chapman and Hall/CRC, Boca Raton, 2006.

[5] N.V. Dung, D.V. Huynh, P.F. Smith and R. Wisbauer, *Extending modules*, Longman Scientific & Technical, Harlow, 1994.

[6] L. Fuchs, *Infinite abelian groups*, vol. I, Academic Press, New York, London, 1970.

[7] E. Mermut, *Homological approach to complements and supplements*, Ph.D. thesis, Dokuz Eylül University, Izmir, 2004.

[8] P.F. Smith, Modules for which every submodule has a unique closure, in *Proceedings of the Biennial Ohio State – Denison Conference*, May 1992, pp. 302–313.

[9] R. Wisbauer, *Foundations of module and ring theory*, Gordon and Breach, Reading, 1991.

[10] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.4*; 2006. `http://www.gap-system.org.`

Faculty of Mathematics and Computer Science
"Babeş-Bolyai" University
Str. M. Kogălniceanu 1, 400084 Cluj-Napoca, Romania
E-mail address: `crivei@math.ubbcluj.ro`

Department of Mathematics and Computer Science
North University of Baia Mare
Str. Victoriei 76, 430122 Baia Mare, Romania
E-mail address: `golteanu@ubm.ro`