The University of Warwick

THEORY OF COMPUTATION

REPORT NO. 40

ON THE COMPUTATION OF THE STRUCTURE OF AN ABELIAN GROUP REPRESENTED BY A SET OF DEFINING RELATIONS

BY

COSTAS S. ILIOPOULOS

University of Warwick
Department of Computer Science
COVENTRY CV4 7AL.

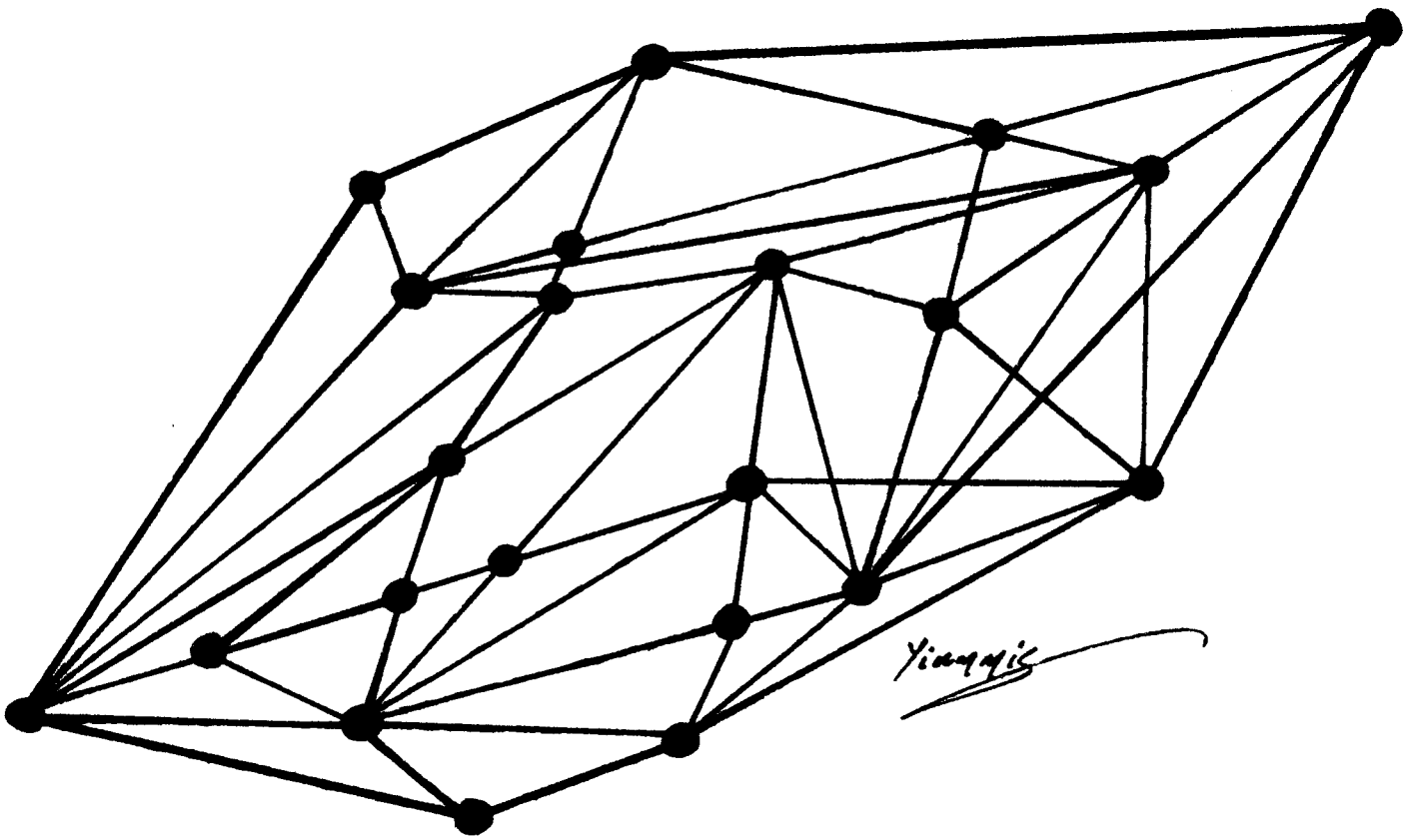
May 1982

ON THE COMPUTATION OF THE STRUCTURE OF AN ABELIAN GROUP REPRESENTED BY A SET OF DEFINING RELATIONS

Costas S. Iliopoulos,
University of Warwick,
Department of Computer Science,
Coventry CV4 7AL,
United Kingdom.

February 1982

To My Parents



1. INTRODUCTION

Suppose that the set

$$R = \{x_1^{c_{1i}} x_2^{c_{2i}} \dots x_n^{c_{ni}} = 1,\ 1 \leq i \leq m,\ c_{ij} \in \mathbb{Z},\ x_j = x_j x_i,\ \forall i,j\}$$

is a set of defining relations for the abelian group G and let C be an $m \times n$ matrix with $C = (c_{ij})$. In this paper, the computational complexity of the problem of computing the order and the structure of a group G, given R, is examined.

The complexity of an algorithm is measured in elementary operations. An elementary operation is a Boolean operation on a single binary bit or pairs of bits. The size s of an $m \times n$ matrix $C = (c_{ij})$, is the number

$$m + n + \log\|C\|$$

where $\|C\| = \max_{i,j}\{|c_{ij}|\}$.

It is shown that in the case of a finite group G, represented by a matrix C of size s, the order and the structure of G can be computed in polynomial time. This computation requires $O(s^5\ M(s^2))$ elementary operations, where M(n) denotes the number of elementary operations required for the multiplication of two integers of length at most n bits.

Also it is given an upper bound for the time required to compute the structure of an infinite group G is also given. This computation requires $O(2^{r+\varepsilon})$ elementary operations, where r is the rank of the matrix representing the group G.

Sims in [4] formulated the classical algorithm for the above problem without analysing its computational complexity.

NOTATION

$Row_C(i)$ and $COL_C(i)$ will be used to denote the i-th row and the i-th column of a matrix C respectively.

An integer row-column operation (IRC operation) is said to be:

(i) The multiplication of all the elements of a row(column) by -1

or (ii) The interchange of two rows(columns)

or (iii) The addition of an integer multiple of row(column) to a different row(column).

2. THE COMPUTATION OF THE DETERMINANT

Suppose that an abelian group G is represented by an m×n matrix C with integer entries, where m is the number of defining relations and n is the number of generators. Then it is not difficult to show that the group G is finite if and only if

$$\text{rank}_{\mathbb{Q}}(C) = n$$

where $\mathbb{Q}$ denotes the field of rationals.

Hence one can decide whether or not G is finite by means of Gaussian elimination on C.

2.1 PROCEDURE DET

```
Procedure    DET(C)
Comment  This procedure computes the determinant (for square matrix only)
and the rank of an m×n matrix C(W.l.o.g. m ≥ n is assumed)
Begin
   i ← 0;
1. while < C is not in echelon form > do
      Begin
         i ← i+1;
           if c_li = 0 then < interchange ROW(i) and ROW(j) where c_ji≠0, j>i >
         μ_j ← c_ji / c_ii for i < j ≤ m;
3.       ROW(j) ← ROW(j) - μ_j ROW(i) for i < j ≤ m;
4.    end
   Return    DET(C) = Π_i c_ii (if C is square), RANK(C) = i;
end □
```

PROPOSITION 2.2 The procedure DET correctly computes the determinant and the rank of a matrix C in $O(s^3M(s^2))$ elementary operations, where s is the size of the matrix C.

Proof

Let $C^{(i)}$ denote C at the beginning of the i-th iteration of loop 1-4. Step 3 requires at most mn multiplications or $O(mn\ M(\log\|C^{(i)}\|))$ elementary operations. Hence the procedure requires at most $O(\sum_{i=1}^{m} mn\ M(\log\|C^{(i)}\|))$ elementary operations.

Now using the formula (12) in [2] (p.26), one can see that

$$\|C^{(i)}\| = O((i\|C\|)^i)$$

Hence the procedure terminates in $O(m^2n\ M(\log(m\|c\|^m))) = O(s^3\ M(s^2))$.

REMARK Step 3 of the procedure DET can be expressed as the matrix multiplication R.C, where

$$R = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & \mu_{i+1} & \ddots & \\ 0 & & \vdots & & 1 \\ & & \mu_m & & \end{pmatrix}$$

i-th column

Hence it may be helpful to use the fast matrix multiplication algorithms (see [3] for the computation of step 3. It is worth mentioning here that Strassen's algorithm for matrix multiplication (see [1]) is not suitable for matrix multiplication over the rationals, since it uses a large number of additions and an addition over the rationals costs three integer multiplications.

COROLLARY 2.3 There exists an algorithm which decides whether or not an abelian group G represented by a matrix C, is finite or infinite in $O(s^3\ M(s^2))$ elementary operations. □

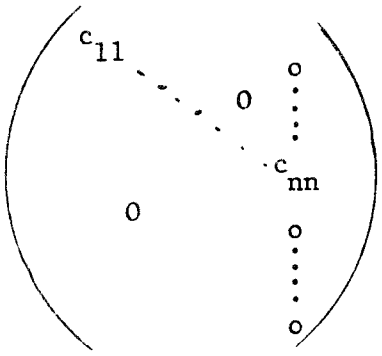
3. THE ALGORITHM FOR FINITE GROUPS

3.1 THE I.R.C. OPERATIONS

The I.R.C. operation transforms the matrix C of the set of defining relations of an abelian group G to a matrix C* which represents the same group. The interchange of rows yields the set of defining relations reordered, the multiplication of a row by -1 substitutes the relation $\prod_i x_i^{c_{ij}}=1$ for the relation $\prod_i x_i^{c_{ij}}=1$ and the addition of an integer multiple of a row to a different row substitutes the relations $\{\prod_i x_i^{c_{ji}}=1, \prod_i x_i^{c_{ki}}=1\}$. The interchange of columns corresponds to the renaming of the generators, the multiplication of a column by -1 is the substitution of the generator x^{-1} for the generator x and the addition of an integer multiple of a column to a different column substitutes $\{xy^p, y\}$ for $\{x, y\}$.

3.2 BACKGROUND FOR THE ALGORITHM

A. Given an m×n matrix C representing a finite abelian group G, the algorithm transforms C via IRC operations to the matrix of the form



where $c_{ii} \mid c_{i+1,\,i+1}$, $1 \leq i \leq n$. The form of the matrix implies that

$$G \approx \ell_{c_{11}} \times \ldots \times \ell_{c_{nn}}$$

where $\ell_{c_{ii}}$ is a cyclic group of order c_{ii}, for $1 \leq i \leq n$.

B. In the case m = n = rank(C), the order $|G|$ of the group G can be computed by means of Gaussian elimination, since

$$|G| = \det(C).$$

In the case $m > n = \text{rank}(C)$, there exist n linearly independent rows of C over the rationals, say, $\text{ROW}(j_i)$, $1 \leq i \leq n$. If $C' = (\text{ROW}(j_1), \ldots, \text{ROW}(j_n))^T$ represents an abelian group F, then the order of the group F is

$$|F| = \det(C')$$

Moreover, since G is a subgroup of F, it follows that

$$|F| \equiv 0 \pmod{|G|} \qquad (3.1)$$

Hence, if one knows the order $|G|$ of group G represented by C, then it is not difficult to see that the matrix $C'' = (c''_{ij})$, with $c''_{ij} = c_{ij} \pmod{|G|}$ representing the same group. And similarly, if one knows a multiple of $|G|$ (e.g. $|F|$ above), then the matrix $C''' = (c'''_{ij})$ with $C'''_j = c_{ij} \pmod{|F|}$ represents the same group.

3.3 THE ALGORITHM

INPUT : An m×n matrix C representing a finite abelian group G

OUTPUT : The canonical structure and the order of the group G

PROCEDURE ELIMINATEROW(C,p,D)

Comment The procedure transforms C via IRC operations to a matrix having $\text{ROW}(p) = (a_1, a_2, \ldots, a_p, 0, \ldots, 0)$. The parameter D is a multiple of the order of the group if one is known, else $D = \infty (x \bmod \infty := x, \forall\, x \in \mathbb{Z})$

Begin

while $c_{pi} \neq 0$ for some $p < i \leq n$ do

Begin

$\lambda \leftarrow \text{index}\{|c_{p\lambda}| = \min_{p \leq j \leq n} \{|c_{pj}| > 0\}\}$

< Interchange COL(λ) and COL(p) >;

< Compute e_k, f_k, $p < k \leq n$: $c_{pk} = e_k\, c_{pp} + f_k$ with $|f_k| < c_{pp}$ 2 >;

COL(k) ← COL(k) − e_k COL(p) (mod D) for $p < k \leq n$;

end

Return C;

end .

In a similar way the procedure ELIMINATECOL(C ,p,D) is defined.

PROCEDURE DIAG(C,D)

Comment This procedure transforms the matrix C via IRC operations to a matrix $C' = (c'_{ij})$ with $c'_{ij} = 0$ for $i \neq j$.

Begin

p ← 0;

(i) while $c_{ij} \neq 0$ for some $i \neq j$ do

Begin

p ← p+1;

(ii) while $< c_{pi} \neq 0$ for some $p < i \leq n >$ do

Begin

(iii) ELIMINATEROW(C,p,D);

(iv) ELIMINATECOL(C,p,D);

(v) end (step (ii))

(vi) end (step (i))

Return C;

end .

The main algorithm is the following:

Begin

1. D ← det(C) (If C non-square, D ← 0);
   If D = 0 then
2. D ← < a minor determinant of C of rank r = rank(C) >;
3. G ← DIAG(C,D);
4. < Transform C via IRC operations to a matrix C such that the sequence $\{c_{11},\ldots,c_{rr}\}$ is sorted by increasing order > ;
5. $D \leftarrow \prod_{i=1}^{r} c_{ii}$;
6. while $< c_{ii} \nmid c_{i+1,i+1}$ for some $1 \leq i \leq r >$ do

Begin

< let α: $c_{\alpha\alpha} \nmid c_{\alpha+1,\alpha+1}$ for some $1 \leq \alpha \leq r$ and $c_{ii} | c_{i+1,i+1}$ $i<\alpha >$

7. < compute v,t : $c_{\alpha+1,\alpha+1} = t\, c_{\alpha\alpha} + v$ with $|v| < |c\alpha\alpha|/2 >$;
8. $\begin{pmatrix} c_{\alpha\alpha} & 0 \\ 0 & c_{\alpha+1,\alpha+1} \end{pmatrix} \leftarrow DIAG\left(\begin{pmatrix} c_{\alpha\alpha} & c_{\alpha\alpha} \\ -t\, c_{\alpha\alpha} & v \end{pmatrix}\right)$
9. end (step 6)

end. □

PROPOSITION 3.4 Algorithm 3.3 correctly computes the order and the structure of the finite abelian group G in $O(s^5\, M(s^2))$, where s is the size of the matrix C.

Proof

The correctness of the algorithm follows from observations in §3.1 and §3.2.

Steps 1-2 require $O(s^3 M(s^2))$ elementary operations by Proposition 2.2 and the remarks of §3.2B. Moreover

$$\log|D| \leq \log(m!\, \|c\|^m) \leq s^2 \qquad (3.2)$$

Since $D \neq 0$, it is not difficult to show that the procedure ELIMINATEROW (respectively ELIMINATECOL) requires $O(\mu s^2 M(\log|D|)) = O(\mu s^2 M(s^2))$ elementary operations, where μ is the number of iterations required by the loop of the procedure. Moreover

$$|c_{pp}^{(i)}| \leq |c_{pp}^{(i-1)}|/2 \qquad (3.3)$$

where $c_{pp}^{(i)}$ denotes c_{pp} at the i-th iteration of the loop of the procedure ELIMINATEROW (respectively ELIMINATECOL).

Suppose now that in the procedure DIAG the loop (ii)-(v) requires g iterations, and that μ_j, ν_j, $1 \leq j \leq g$, denotes the number of iterations of the loop of the procedure ELIMINATEROW and ELIMINATECOL respectively at the j-th iteration of the loop (ii)-(v). Then using (3.3) we have

$$\sum_{j=1}^{g} (\mu_j + \nu_j) \leq \log|D| \leq s^2$$

Hence loop (ii)-(v) requires $O(s^4\, M(s^2))$ elementary operations and moreover procedure DIAG terminates in $O(s^5\, M(s^2))$ elementary operations.

Now step 4 requires $O(s^3 \log s)$ elementary operations by using an a sorting algorithm which requires $O(s \log s)$ comparisons (see [1]).

In a similar way, as above, one can show that steps 7-8 require $((s^2\, M(s^2))$ elementary operations. Moreover, loop 6-9 requires at most

$\sum_{j=1}^{n} \log D = O(s^3)$ iterations and thus loop 6-9 requires $O(s^5\, M(s^2))$ elementary operations.

From the above analysis the proposition follows. □

4. THE ALGORITHM FOR INFINITE GROUPS

Suppose that an infinite abelian group G is represented by an m×n matrix C, with r = rank(C) < n. In this case, the matrix C is transformed via IRC operations to an m×n matrix of the form

$$\left(C^* \;\middle|\; 0 \right) \qquad (4.1)$$

where C* is an m×r matrix with rank(C*) = r. Then one can apply algorithm 3.3 to C* to obtain the structure of G.

The transformation can be done in a similar way as in algorithm 3.3 using the following algorithm:

ALGORITHM 4.1

INPUT : an m×n matrix C representing the infinite abelian group G

OUTPUT : The order and the canonical structure of G.

```
Begin
    r←rank(C);
1.  For p = 1 to r do
      Begin
        if < c_pj = 0, for all p ≤ j ≤ n > then
        Begin
          λ ← index{ROW(λ) : c_λj ≠ 0 for some p ≤ j ≤ n};
          < Interchange ROW(λ) and ROW(p) >;
        end
        ELIMINATEROW(C,p,∞);
2.    end
      C* ← as it is in (4.1)
      < Apply algorithm 3.3 on C* >
end. □
```

PROPOSITION 4.2 Algorithm 4.1 correctly computes the structure of the group G in $O(2^{\tau+\epsilon})$ elementary operations.

Proof

It is not difficult to conclude its correctness.

Let $C^{(i)}$ denote the matrix C at the i-th iteration of the loop of the procedure ELIMINATEROW. Then one can show

$$\|C^{(i)}\| \leq i \|C\|^2$$

Now since

$$|c_{pp}^{(i)}| < |c_{pp}^{(i-1)}| /2$$

the number of iterations of the loop of the procedure ELIMINATEROW is at most $\log\|C\|$.

Hence one can show that after r iterations of loop 1-2 the size of the matrix C is bounded by

$$m + n + \log(\|C\|^{2^r})$$

and thus the proposition follows. □

ACKNOWLEDGEMENTS

I am indebted to Dr. Meurig Beynon for his help. Also I would like to thank Prof. Christos Papadimitriou for pointing the reference [2]. Thanks also to Mrs. Josie Lloyd and Mrs. Valerie Clayton for typing this paper.

REFERENCES

[1] AHO,A., HOPCROFT,J., ULLMAN,J., "The Design and Analysis of Computer Algorithms", Addison-Wesley (1974).

[2] GANTMACHER "Matrix theory", Vol.I, Chelsea (1960).

[3] PAN,V., "Field extension and trilinear aggregating, uniting and cancelling for acceleration of matrix multiplications" in 20th Annual Symposium on Foundations of Computer Sc. (1979).

[4] SIMS,C.C., "The Influence of Computers in Algebra", Proceedings of Symposia in Applied Mathematics 20, 13-30, (1974).