Parallel Computation of Sylow Subgroups in Solvable Groups

Peter D. Mark

CIS-TR-93-06
March 1993

Department of Computer and Information Science
University of Oregon

to appear in
GROUPS AND COMPUTATION
DIMACS Series in Discrete Mathematics and Theoretical Computer Science

Parallel Computation of Sylow Subgroups in Solvable Groups

PETER D. MARK

ABSTRACT. Sylow's theorem is a fundamental tool in group-theoretic investigations. In computational group theory there is an important role for efficient constructive analogs of Sylow's theorem. For computational purposes, we assume that a group is given by a set of permutations that generate it. This leads to the following problems:

SYLFIND(p, G)

GIVEN: a prime p and generators for a group G,
FIND: generators for a Sylow p-subgroup $P \leq G$.

SYLCONJ(P_1, P_2, G)

GIVEN: generators for G and two of its Sylow p-subgroups P_1 and P_2,
FIND: an element $g \in G$ for which $P_1^g = P_2$.

SYLEMBED(P, G)

GIVEN: generators for G and for a p-subgroup $P \leq G$,
FIND: generators for a Sylow p-subgroup of G containing P.

This paper shows SYLFIND, SYLCONJ, and SYLEMBED for solvable groups are in the complexity class NC; namely, they are solvable in polylogarithmic time ($O(\log^c n)$ steps) using a polynomial number of processors working in parallel. A future paper by Kantor, Luks, and Mark extends these results to general groups.

1. Introduction

Over the last two decades, sequential, polynomial-time algorithms have been found for a variety of basic problems concerning permutation groups, including determining the order, testing membership, and finding some of the important

1991 *Mathematics Subject Classification.* Primary 20B40,68Q25; Secondary 20D20,68Q22.
The author was supported in part by NSF Grant CCR-9013410 while he was a research assistant at the University of Oregon.
This is the final version of this paper.

CIS-TR-93-06
University of Oregon
March 1993

subgroups such as centers, commutator subgroups, and Sylow subgroups. Sylow subgroups are a crucial tool in [6] for finding centers in quotient groups and for various other problems (see also [11]). Also, a restricted version of SYLFIND played a role in Luks's polynomial time algorithm for bounded valence graph isomorphism testing [8]. In a recent series of papers, Kantor gave polynomial-time solutions for the Sylow problems SYLFIND and SYLCONJ [3, 4, 5]. His methods exploit a well-developed library of polynomial-time procedures for permutation groups and consequences of the classification of finite simple groups.

Inspired by new generations of machines, new theoretical models were developed to describe and analyze parallel computation. In particular, the class NC gives a useful framework in which to study the inherent parallelizability and logical structure of computations independent of any interprocessor connection network. In the case of permutation group algorithms, many of the known techniques appeared to depend on inherently sequential methods. In a series of papers [1, 10, 12, 14], entirely new machinery was developed for permutation group computation. It ultimately brought a sizable portion of the collection of polynomial-time problems, including finding composition factors, into NC. However, a number of critical questions remained open. As pointed out in [1], a leading one of these was the parallelization of Sylow subgroup computations. This paper presents efficient parallelizations of these computations for solvable groups. These are special cases of methods that solve the same problems for general permutation groups [7, 13].

Most of the sequential permutation group algorithms exploit a tower of subgroups, $G = G_0 \geq G_1 \geq \cdots \geq 1$, that is either a tower of pointwise set stabilizers or a composition series for G [2, 3, 4, 19]. Membership testing, for example, uses a tower of point stabilizers and reduces membership testing for G_i to membership testing for G_{i+1} [19]. Kantor's polynomial-time sequential Sylow algorithms [4] use a composition series [9] for G.

Both of these towers can have length linear in the degree n of G, and hence may be too long for computation in NC. The parallelization of order and membership testing as well as the new NC Sylow algorithms utilize a different (normal) tower $G = K_0 \rhd K_1 \rhd \cdots \rhd K_r = 1$ for G whose length r is polylogarithmic in n. The quotients K_i/K_{i+1} of successive groups in the tower are semisimple (direct products of simple groups). The existence and NC-constructibility of such a normal tower are demonstrated in [1] and [10].

Groups in this tower arise from two different divide and conquer strategies. The first involves only a naive construction of a *structure forest* [12], a data structure based on the orbits and imprimitivity blocks of the group (see also [11]). This reduces the tower construction to the primitive group case, which requires a second divide and conquer approach based on the internal structure of primitive groups, Luks's composition factors algorithm, and other consequences of the classification of finite simple groups. The tower may be refined so that successive quotients K_i/K_{i+1} are direct products of simple groups that are either

all abelian or all nonabelian. In the abelian levels (and hence for solvable groups, in which all levels are abelian) the Sylow problems reduce to the problem of solving systems of linear equations over finite fields, which has been shown to be in NC by Mulmuley [15]. In nonabelian levels, parallelism arises for many problems in a natural way by working within each simple group independently.

In this paper, we focus on the Sylow algorithms for solvable groups. However, we follow an approach that works for Sylow subgroup computations in solvable quotient groups as well. This plays an important role in the extension of the present results to general permutation groups [7, 13]. Moreover, although the nonsolvable case demands deeper group theory, including case analyses based on the classification of the finite simple groups, it exhibits much of the logical structure of the present paper.

One can describe the algorithm for SYLFIND as the computation of a tower of subgroups $G = P_0 \geq P_1 \geq \cdots \geq P_r$ where P_i/K_i is a Sylow p-subgroup of G/K_i and $G = K_0 \rhd K_1 \rhd \cdots \rhd K_r = 1$ is a semisimple tower of polylogarithmic length as described above. Such a semisimple tower is available, indeed it is intrinsic to the basic machinery for NC computation, including membership-testing [1, 12]. However, it simplifies our exposition to avoid repeated and explicit reference to a precomputed tower. Thus, we compute suitable K_i as they are needed.

Let $K \lhd H$ where H/K is abelian semisimple. An important ingredient in parallel permutation group computations is the ability to represent H/K in a computationally efficient manner as a product of vector spaces. Once effective representations for such quotients are available, two particular problems emerge: finding a Sylow p-subgroup P of a group H where $K \lhd H \lhd G$, K is an elementary abelian p'-group, and H/K is an elementary abelian p-group, and finding a subgroup $G^* \leq G$ that normalizes P and contains a Sylow p-subgroup of G. The latter problem is an algorithmic form of the Frattini argument. These permutation group problems are transformed into solving systems of linear equations over finite fields.

2. Preliminaries

2.1. NC Concepts and Results. The complexity class NC is the set of problems which can be solved using a polynomial ($O(n^k)$) number of processors in polylogarithmic time ($O(\log^c n)$ steps) where k and c are constants and n is the input size. Processors communicate via shared memory. This model of computation, known as a PRAM and introduced in [16], has become a standard tool for analyzing a problem's inherent parallelizability.

Permutation-group problems require three basic permutation operations: computing the product of two permutations, computing inverses of permutations, and computing large powers of permutations. The first two are straightforward. To form a power α^b of a permutation α, where b is represented in

binary (in practice, b may be $O(n!)$ where n is the degree of α), we form α^b independently on each cycle of α by reducing b modulo the cycle length. Note that, for sequential computation, one does not need to emphasize such powering as a "primitive" operation, for it is accomplished by repeated-squaring, therefore by a polynomial number of multiplications. However, the number of successive squarings would be prohibitive for an NC result.

We make use of NC algorithms for the following permutation group problems that are described in [12]. In all of the following problems, we assume G is given by a set of generating permutations: $G = \langle S \rangle \leq \mathrm{Sym}(\Omega)$.

PROBLEM 2.1.1. MEMBERSHIP(G, x)

GIVEN: a permutation group $G \leq \mathrm{Sym}(\Omega)$ and an element $x \in \mathrm{Sym}(\Omega)$,
DETERMINE: whether or not $x \in G$.

PROBLEM 2.1.2. COMMUTATOR(G)

GIVEN: a permutation group $G = \langle S \rangle \leq \mathrm{Sym}(\Omega)$,
FIND: the commutator subgroup G' of G.

G' is the normal closure of the set $\{[s,t] \mid s,t \in S\}^G$; an NC algorithm for computing normal closures in solvable groups is given in [12].

PROBLEM 2.1.3. FACTOR(x, H, K)

GIVEN: $H, K \leq \mathrm{Sym}(\Omega)$ where $K = \langle Y \rangle$ is normalized by $H = \langle X \rangle$, and $x \in HK$,
FIND: elements $h \in H$ and $k \in K$ such that $x = hk$.

SKETCH OF THE ALGORITHM: The NC membership testing algorithm given in [1] constructs x from the set $X \cup Y$ of generators of HK. Following the same construction but substituting the identity for the elements of Y yields h.

2.2. Definitions and Basic Concepts. A group is *semisimple* if it is the direct product of simple groups; it is *abelian semisimple* if each of its simple factors is abelian. Semisimple groups play an important role in the NC algorithms for SYLFIND and SYLCONJ in general groups in [7]; in the present context of solvable groups, we concern ourselves only with abelian semisimple groups. Since an abelian semisimple group is a direct product of elementary abelian groups for various primes, we may view it as a direct product of vector spaces over different prime fields. For convenience, we refer to such a group as a *generalized vector space*.

A *generalized basis* for a generalized vector space $H = P_1 \times \cdots \times P_l$ (assume P_i is an elementary abelian p_i-group with $p_i \neq p_j$ for $i \neq j$) is the union $\cup_i \mathcal{B}_i$ where $\mathcal{B}_i$ is a basis for P_i. Similarly, if the quotient H/K is abelian semisimple, a subset $\mathcal{B} \subset H$ is called a *generalized basis for H modulo K* if $\{Kb \mid b \in \mathcal{B}\}$ is a generalized basis for H/K.

LEMMA 2.2.1. *Given groups $K \lhd H = \langle T \rangle$ with H/K abelian semisimple, a generalized basis $\mathcal{B}$ for H modulo K can be found in NC.*

PROOF. Let $H/K = P_1/K \times \cdots \times P_l/K$ where each P_i/K is an elementary abelian p_i-group and $p_i \neq p_j$ if $i \neq j$. Let $\pi = \prod_{i=1}^{l} p_i$, let $\pi_i = \pi/p_i$, and let $\mathcal{C}_i = \{t^{\pi_i} \mid t \in T\}$ for each $i = 1, \ldots, l$. Then $P_i = \langle \mathcal{C}_i, K \rangle$ To obtain a basis $\mathcal{B}_i$ of P_i modulo K for all $i = 1, \ldots, l$ in parallel, suppose $\mathcal{C}_i = \{t_1, \ldots, t_m\}$, and let $\mathcal{B}_i = \{t_j \in \mathcal{C}_i \mid t_j \notin \langle t_1, \ldots, t_{j-1}, K \rangle\}$. These membership tests are each in NC (Problem 2.1.1) and may be performed in parallel. $\mathcal{B} = \cup_{i=1}^{l} \mathcal{B}_i$ is a generalized basis for H modulo K. □

We use F_q^d to denote the d-dimensional vector space over F_q, the field of q elements, and $\{e_1, \ldots, e_d\}$ to denote the standard basis. If H/K is an abelian semisimple group, a *generalized vector space representation* for H/K is a pair (V, ϕ) consisting of a generalized vector space V together with an epimorphism $\phi : H \rightarrow F_{p_1}^{d_1} \times \cdots \times F_{p_l}^{d_l}$ with kernel K, where the primes p_i are all distinct. A generalized vector space representation (V, ϕ) for H/K is *NC-effective* if ϕ is an *NC-computable function*, i.e. there is an NC procedure that can compute $\phi(h)$ for any given $h \in H$.

LEMMA 2.2.2. *Given groups $K \lhd H$ where H/K is abelian semisimple, an NC-effective generalized vector space representation (V, ϕ) for H/K can be found in NC. Moreover, for any $v \in V$, a preimage of v in H, i.e. an element $h \in H$ for which $\phi(h) = v$, can be found in NC.*

PROOF. Suppose $H/K = P_1/K \times \cdots \times P_l/K$ where $P_i/K \cong V_i = F_{p_i}^{d_i}$, for each $i = 1, \ldots, l$. By Lemma 2.2.1, a generalized basis $\mathcal{B} = \cup_{i=1}^{l} \mathcal{B}_i$ for H modulo K may be found in NC, where $\mathcal{B}_i = \{b_{i1}, \ldots, b_{id_i}\}$ is a basis for P_i modulo K. Let $\{e_{i1}, \ldots, e_{id_i}\}$ be the standard basis of V_i. We specify an NC-computable function $\phi : H \rightarrow V$ as follows. Let h be an element of H. For all b_{ij}, $1 \leq i \leq l$, $1 \leq j \leq d_i$, in parallel: test, for all a, $1 \leq a \leq p_i - 1$, in parallel, whether $h^{-1} \cdot b_{ij}^a \in \langle \mathcal{B} \setminus \{b_{ij}\}, K \rangle$ (tested using Problem 2.1.1) and let a_{ij} be the unique a satisfying this condition. Then $h \equiv \prod_{i,j} b_{ij}^{a_{ij}} \pmod{K}$. Define $\phi(h) = \sum_{ij} a_{ij} e_{ij}$. Hence the map $\phi : H \rightarrow V$ is an NC-computable function with kernel K. The pair (V, ϕ) is an NC-effective representation for H/K.

Furthermore, for any element $v = \sum_{ij} a_{ij} e_{ij} \in V$, the element $h = \prod_{i,j} b_{ij}^{a_{ij}} \in H$ satisfies $\phi(h) = v$. Hence we may compute preimages in H of elements of V in NC. □

We also require the ability to perform basic operations of linear algebra within generalized vector spaces in NC. A *generalized linear transformation* of a generalized vector space $V = V_1 \times \cdots \times V_d$ is a direct product of linear transformations $L = L_1 \times \cdots \times L_d$, where each L_i is a linear transformation of V_i. Hence $Lv = (L_1 v_1, \cdots, L_d v_d)$, where $v = (v_1, \cdots, v_d) \in V$.

LEMMA 2.2.3. *Given a generalized linear transformation $L = L_1 \times \cdots \times L_d$ of a generalized vector space $V = V_1 \times \cdots \times V_d$ and an element $b = (b_1, \ldots, b_d) \in V$, the set of solutions of $Lx = b$ can be found in NC.*

PROOF. Each set $X_i = \{x_i \mid L_i x_i = b_i\}$ may be found in NC using [15]. Then the set $X_1 \times \cdots \times X_d \subseteq V$ is the set we seek. □

A set of equations of the form described in Lemma 2.2.3 is called a system of *generalized linear equations.*

For a prime p and a group G, let $R_p(G)$ denote the smallest normal subgroup $H \trianglelefteq G$ for which G/H is an elementary abelian p-group, and let $R_{\mathcal{A}}(G)$ denote the smallest normal subgroup $H \trianglelefteq G$ for which G/H is abelian semisimple. Note that $R_{\mathcal{A}}(G) < G$ for any solvable $G \neq 1$. Let $O^p(G)$ denote the smallest normal subgroup $H \trianglelefteq G$ such that G/H is a p-group. Let $R_p^0(G) = G$ and $R_p^{i+1}(G) = R_p(R_p^i(G))$. Let $d_p(G)$ denote the smallest integer r for which $R_p^r(G) = R_p^{r+1}(G)$. Note that $O^p(G) = R_p^r(G)$, where $r = d_p(G)$. Similarly, let $R_{\mathcal{A}}^0(G) = G$ and $R_{\mathcal{A}}^{i+1}(G) = R_{\mathcal{A}}(R_{\mathcal{A}}^i(G))$. Let $d_{\mathcal{A}}(G)$ denote the smallest integer r for which $R_{\mathcal{A}}^r(G) = R_{\mathcal{A}}^{r+1}(G)$. Note that if G is solvable, $R_{\mathcal{A}}^r(G) = 1$, where $r = d_{\mathcal{A}}(G)$.

PROPOSITION 2.2.4. *For a solvable group $G \leq \mathrm{Sym}(\Omega)$, the tower $G > R_{\mathcal{A}}(G) > R_{\mathcal{A}}^2(G) > \cdots > R_{\mathcal{A}}^s(G) = 1$ has length s* logarithmic *in $|\Omega|$, i.e. $s \leq c \log |\Omega|$ for some constant c.*

PROOF. Since $d_{\mathcal{A}}(G) \leq \max_{\Delta \in \mathcal{O}} \{d_{\mathcal{A}}(G^\Delta)\}$ where $\mathcal{O}$ is the set of G-orbits, we may assume without loss of generality that G is transitive. If G is primitive, then $|G| < |\Omega|^{3.25}$, by [17, 20], hence $d_{\mathcal{A}}(G)$ is logarithmic in $|\Omega|$. If G is imprimitive, let G^* be the primitive action on $m > 1$ blocks of size n/m and let K be the kernel of this action. Inductively assuming the result is true for groups of degree smaller than $|\Omega|$, we have $d_{\mathcal{A}}(G) \leq d_{\mathcal{A}}(G^*) + d_{\mathcal{A}}(K) \leq c \log m + c \log(n/m) = c \log(n)$. □

COROLLARY 2.2.5. *For a solvable group $G \leq \mathrm{Sym}(\Omega)$, the tower*

$$G \rhd R_{\mathcal{A}}O^p(G) \rhd (R_{\mathcal{A}}O^p)^2(G) \rhd \cdots \rhd (R_{\mathcal{A}}O^p)^s(G) = 1$$

has length s logarithmic in $|\Omega|$.

PROOF. One shows that $(R_{\mathcal{A}}O^p)^i(G) \leq R_{\mathcal{A}}^i(G)$. □

COROLLARY 2.2.6. *For a solvable group $G \leq \mathrm{Sym}(\Omega)$, $d_p(G)$ is logarithmic in $|\Omega|$.*

PROOF. The quotient $G/O^p(G)$ is isomorphic to a homomorphic image $\overline{P}$ of a Sylow p-subgroup P of G. Hence $d_p(G) = d_p(G/O^p(G)) = d_p(\overline{P}) \leq d_p(P)$. But $d_p(P) = d_{\mathcal{A}}(P)$, which is logarithmic in $|\Omega|$ by Proposition 2.2.4. □

LEMMA 2.2.7. *Given a permutation group $G = \langle S \rangle \leq \mathrm{Sym}(\Omega)$, the subgroups $R_p(G), R_{\mathcal{A}}(G)$, and $O^p(G)$ can each be computed in NC.*

PROOF. $R_p(G) = \langle G', \{s^p \mid s \in \mathcal{S}\}\rangle$ and $R_{\mathcal{A}}(G) = \langle G', \{s^q \mid s \in \mathcal{S}\}\rangle$, where q is the product of the primes that divide $|G|$, so each of these groups may be found in NC (see Problem 2.1.2). Since $O^p(G) = R_p^r(G)$ where $r = d_p(G)$, $O^p(G)$ may be found by computing $R_p^i(G)$ for $i = 1, \ldots, r$ sequentially. (This is an NC computation since r is logarithmic in $|\Omega|$ by Corollary 2.2.6.) □

3. The Sylow Algorithms for Solvable Groups

3.1. A Base Case and the Frattini Argument. Finding and conjugating Sylow subgroups of solvable groups give rise to two particular subproblems: finding a Sylow p-subgroup P of a group H where $K \lhd H \lhd G$, K is an elementary abelian p'-group, and H/K is an elementary abelian p-group, and finding a subgroup $G^* \leq G$ that normalizes P and contains a Sylow p-subgroup of G. The latter problem is an algorithmic form of the "Frattini argument" ([18, p. 61]). This section describes procedures BASECASE1 and FRATTINI for these problems.

The essential technique in both these procedures is to transform a group theoretic condition into the problem of solving systems of generalized linear equations over finite fields, for which there exist NC algorithms (Lemma 2.2.3). Linear transformations arise where K and H are both normal subgroups of another group G, H/K is abelian semisimple, and (V, ϕ) is an NC-effective generalized vector space representation for H/K. Then each element $g \in G$ induces a generalized linear transformation on V denoted T_g, given by $\phi(x) \mapsto \phi(g^{-1}xg)$.

PROBLEM 3.1.1. *BASECASE1(H, K, L)*

GIVEN: $L \trianglelefteq K \trianglelefteq H \leq \mathrm{Sym}(\Omega)$, $L \trianglelefteq H$, H/K is an elementary abelian p-group, and K/L is an abelian semisimple p'-group,

FIND: a group P for which $L \leq P \leq H$ and P/L is a Sylow p-subgroup of H/L.

PROPOSITION 3.1.2. *BASECASE1 is in NC.*

PROOF. If P is such a group then $H = PK$ and, since $P \cap K = L$, $H/K \cong P/L$. In particular, P/L is elementary abelian. Suppose $H = \langle \mathcal{S}\rangle$. Then for each $s \in \mathcal{S}$, there exists $x_s \in K$ such that $sx_s \in P$. Hence,

(i) for all $s \in \mathcal{S}$, $(sx_s)^p \in L$

(ii) for all $s, t \in \mathcal{S}$, $[sx_s, tx_t] \in L$.

Conversely, if $\{x_s \mid s \in \mathcal{S}\} \subseteq K$ satisfies (i) and (ii), then we can take $P = \langle \{sx_s \mid s \in \mathcal{S}\}\rangle$.

Let $\phi : K \to V$ induce a generalized vector space representation of K/L (see Lemma 2.2.2) and for $g \in G$, let T_g be defined as above. Since $[sx_s, tx_t] = x_s^{-1}(x_t^{-1})^s[s,t]x_s^t x_t$,

$$\phi([sx_s, tx_t]) = -\phi(x_s) - T_s\phi(x_t) + \phi([s,t]) + T_t\phi(x_s) + \phi(x_t).$$

Thus $[sx_s, tx_t] \in L$ if and only if each pair of elements in $\{\phi(x_s) \mid s \in \mathcal{S}\}$ satisfies the following system of $|\mathcal{S}|^2$ generalized linear equations:

$$\forall s, t \in \mathcal{S},\ (T_t - I)X_s - (T_s - I)X_t = -\phi([s,t]).$$

Hence, a set $\{x_s \mid s \in \mathcal{S}\}$ satisfying (ii) is obtained by solving this system for $\{X_s\} \subseteq V$ (see Lemma 2.2.3) and letting x_s be a preimage of X_s in K (see Lemma 2.2.2).

We can modify this set to satisfy (i), while maintaining (ii), by replacing each x_s by $s^{-1}(sx_s)^m$ where m is chosen so that $|K/L|$ divides m and $m \equiv 1 \pmod{p}$. □

Recall the *Frattini argument* (see [18, p. 61]): if $P \leq K \trianglelefteq G$ with P a Sylow p-subgroup of K then $G = N_G(P)K$, where $N_G(P)$ is the normalizer of P in G.

PROBLEM 3.1.3. *FRATTINI*(G, H, K, L, P)

GIVEN: $L \lhd K \lhd H \trianglelefteq G \leq \mathrm{Sym}(\Omega)$, each of L, K is also normal in G, K/L is an abelian semisimple p'-group, H/K is a p-group, and P/L is a Sylow p-subgroup of H/L,

FIND: a subgroup $G^* \leq G$ that normalizes P and contains P for which G^*/L contains a Sylow p-subgroup of G/L.

PROPOSITION 3.1.4. *FRATTINI is in NC.*

PROOF. We are given $G = \langle \mathcal{S} \rangle$, $P = \langle T \rangle$. By the Frattini argument, for any $s \in \mathcal{S}$ there exists $x_s \in K$ such that $sx_s \in N_G(P)$. For any such collection, $\{x_s \mid s \in \mathcal{S}\}$, we can take $G^* = \langle P, \{sx_s \mid s \in \mathcal{S}\}\rangle$; to see that G^*/L contains a Sylow p-subgroup of G/L, we observe that $|G^*K/L| = |G/L|$, but the p-part of $|G^*K/L|$ equals the p-part of $|G^*/L|$ since $(|K/L|, p) = 1$.

To find such x_s (in parallel for each $s \in \mathcal{S}$) it suffices to ensure that $t^{sx_s} \in P$ for each $t \in T$. For each $t \in T$, write $t^s = a_t k_t$, with $a_t \in P, k_t \in K$ (see Problem 2.1.3). The required x_s must therefore satisfy $a_t^{x_s} k_t \in P$. But, $a_t^{x_s} k_t = a_t[a_t, x_s]k_t$ and $[a_t, x_s]k_t \in K$. Since $P \cap K \leq L$, the condition on x_s is equivalent to $[a_t, x_s]k_t \in L$, or, if $\phi : K \to V$ induces a vector-space representation of K/L, to

$$\phi(x_s) - \phi(x_s^{a_t}) + \phi(k_t) = 0.$$

Therefore, $\phi(x_s)$ is a solution to the system of $|T|$ generalized linear equations

$$\forall t \in T,\ (I - T_{a_t})X + \phi(k_t) = 0,$$

where T_{a_t} is defined just before Problem 3.1.1. Hence, x_s is obtained by solving this system for $X \in V$ (see Lemma 2.2.3) and letting x_s be a preimage of X in K (see Lemma 2.2.2). □

3.2. SYLFIND. Before giving a procedure for SYLFIND, we describe another special case.

PROBLEM 3.2.1. *BASECASE2*(G, K, L, p)

GIVEN: a solvable group G and $L \trianglelefteq K \trianglelefteq G \leq \mathrm{Sym}(\Omega)$, where $L \triangleleft G$, G/K is a p-group, and K/L is an abelian semisimple p'-group,
FIND: a subgroup $P \leq G$ containing L for which P/L is a Sylow p-subgroup of G/L.

PROPOSITION 3.2.2. *BASECASE2 is in NC.*

PROOF. Let $G_1 = R_p(G)L$ (see Lemma 2.2.7). Note that $K \leq G_1$. If $G_1 = G$, then $G = K$, and $|G/L| = |K/L|$ is relatively prime to p so G/L has no nontrivial Sylow p-subgroup. In this case, return $P = L$.

Otherwise, recursively find $P_1 = \mathrm{BASECASE2}(G_1, K, L)$. The group $G^* = \mathrm{FRATTINI}(G, G_1, K, L, P_1)$ normalizes P_1 and contains a Sylow p-subgroup of G. We seek a group P for which P/P_1 is a Sylow p-subgroup of G^*/P_1. Find the group $U = R_p(G^*)L$ (see Lemma 2.2.7) . By definition, G^*/U is an elementary abelian p-group. Also note that U/P_1 is an elementary abelian p'-group since, letting $W = U \cap K$, we have $U/P_1 = P_1W/P_1 \cong W/(P_1 \cap W) = W/L \leq K/L$, which is an elementary abelian p'-group. Hence BASECASE1 (see Problem 3.1.1) applies, and $P = \mathrm{BASECASE1}(G^*, U, P_1)$ is the group we seek.

In each recursive call, $L \leq G_1 < G$, so the number of recursive calls is bounded by $d_p(G/L)$. Since $d_p(G/L) \leq d_p(G)$, which is logarithmic in $|\Omega|$ by Lemma 2.2.6, the algorithm for BASECASE2 is in NC. □

PROBLEM 3.2.3. *SYLFIND*(G, p)

GIVEN: a solvable group $G \leq \mathrm{Sym}(\Omega)$ and a prime p that divides $|G|$,
FIND: a Sylow p-subgroup P of G.

THEOREM 3.2.4. *SYLFIND is in NC.*

PROOF. Let $K = O^p(G)$ (see Lemma 2.2.7) and $L = R_{\mathcal{A}}(K)$ (see Lemma 2.2.7). If $L = K$ (tested using Problem 2.1.1) then $R_p(K) = R_{\mathcal{A}}(K)$, which implies $K = 1$, so return G. Otherwise let $P = \mathrm{BASECASE2}(G, K, L, p)$ and recurse by returning $\mathrm{SYLFIND}(P, p)$.

To analyze the running time of SYLFIND, note that $L = R_{\mathcal{A}}O^p(G)$ so the group P that is passed in the recursive call satisfies $R_{\mathcal{A}}O^p(P) = R_{\mathcal{A}}O^p(L) = (R_{\mathcal{A}}O^p)^2(G)$. Hence the depth of the recursion is logarithmic by Lemma 2.2.5.

Moreover, the procedures BASECASE1 and FRATTINI are in NC. Hence SYLFIND is in NC. □

3.3. SYLCONJ and SYLEMBED. We now describe a procedure SYLCONJ-EMBED which can be used for conjugating Sylow p-subgroups and embedding a p-subgroup into a Sylow p-subgroup. Specifically, we obtain SYLCONJ as a special case of SYLCONJ-EMBED by letting P_1 and P_2 both be Sylow p-subgroups of G. Similarly, SYLEMBED can be implemented by first letting $P_2 = \text{SYLFIND}(G, p)$, then setting $g = \text{SYLCONJ-EMBED}(G, P_1, P_2)$ and returning $P_2^{g^{-1}}$, a Sylow p-subgroup of G containing P_1.

PROBLEM 3.3.1. *SYLCONJ-EMBED*(G, P_1, P_2)

GIVEN: a solvable group $G \leq \text{Sym}(\Omega)$, a p-subgroup P_1 of G, and a Sylow p-subgroup P_2 of G,

FIND: an element $x \in G$ for which $P_1^x \leq P_2$.

THEOREM 3.3.2. *SYLCONJ-EMBED is in NC.*

PROOF. Find $H = O^p(G)$ (see Lemma 2.2.7), so that $P_1H/H \leq G/H = P_2H/H$. Find $K = R_A(H)$ (see Lemma 2.2.7). If $K = H$, then $H = 1$ and $P_1 \leq P_2 = G$. In this case, return $x = 1$.

We may assume $K < H$. We first show that we can find $x \in H$ such that $P_1^x \leq P_2K$. Suppose $P_1 = \langle \mathcal{S} \rangle$. Since $G = P_2H$, for each $s \in \mathcal{S}$, we can factor $s = bh$ with $b \in P_2, h \in H$ (see Problem 2.1.3). For $x \in H$, $s^x \in P_2K$ if and only if $b^{-1}s^x = (x^{-1})^b x h^x \in P_2K$; since $h^x = h[h, x] \in hK$, this happens if and only if $(x^{-1})^b x h \in P_2K \cap H \leq K$. Thus, if $\phi : H \to V$ induces a generalized vector space representation of H/K (see Lemma 2.2.2), then $s^x \in P_2K$ if and only if $\phi(x)$ is a solution to the system of $|\mathcal{S}|$ linear equations

$$\forall s \in \mathcal{S},\ (I - T_{b_s})X + \phi(h_s) = 0,$$

where $s = b_s h_s$ with $b_s \in P_2, h_s \in H$, and where T_{b_s} is defined just before Problem 3.1.1.

Hence, x is obtained by solving this system for $X \in V$ and taking a preimage of X in H (see Lemma 2.2.2).

Let $G^* = \langle P_1^x, P_2 \rangle$ and recursively solve SYLCONJ-EMBED(G^*, P_1^x, P_2) for $y \in G^*$ such that $(P_1^x)^y \leq P_2$. Return the element xy.

To analyze the running time of SYLCONJ-EMBED, note that the group K computed in SYLCONJ-EMBED is equal to $R_A O^p(G)$. Hence the group G^* that is passed in the recursive call satifies $R_A O^p(G^*) = R_A O^p(K) = (R_A O^p)^2(G)$. Thus, the depth of the recursion is logarithmic by Lemma 2.2.5. Therefore, SYLCONJ-EMBED is in NC. □

4. Conclusion

A natural continuation of this work is the search for efficient parallelizations of other group-theoretic sequential algorithms, for example, the problem of finding Sylow normalizers. This problem has been shown to be in polynomial time

by Kantor [5]. The ability to compute Sylow normalizers may extend the applicability of the Sylow theorems to parallel computation in ways analogous to its use in sequential algorithms [6]. These algorithms may also give rise to more efficient sequential algorithms than those initially given by Kantor.

REFERENCES

1. Babai, L., Luks, E. M., Seress, Á. *Permutation groups in NC*, Proc. 19th ACM STOC, 1987, 409-420.
2. Furst, M. L., Hopcroft, J. and Luks, E. M. *Polynomial time algorithms for permutation groups*, Proc. 21th IEEE FOCS, 1980, 36-41.
3. Kantor, W. M. *Polynomial time algorithms for finding elements of prime order and Sylow subgroups*, J. Algorithms 6, 1985, 478-514.
4. Kantor, W. M. *Sylow's theorem in polynomial time*, J. Comp. Sys. Sci. 30, 1985, 359-394.
5. Kantor, W. M. *Finding Sylow normalizers in polynomial time*, J. Algorithms 11, 1990, 523-563.
6. Kantor, W. M. and Luks, E. M. *Computing in quotient groups*, Proc. 22nd ACM STOC, 1990, 524-534.
7. Kantor, W.M, Luks, E.M., Mark, P.D., In preparation.
8. Luks, E. M., *Isomorphism of graphs of bounded valence can be tested in polynomial time*, J. Comp. Sys. Sci., 25 1982, 42-65.
9. Luks, E. M., *Computing the composition factors of a permutation group in polynomial time*, Combinatorica 7 (1987), 87-99.
10. Luks, E. M., *Parallel algorithms for permutation groups and graph isomorphism*, Proc. 27th IEEE FOCS, 1986, 292-302.
11. Luks, E.M., *Permutation groups and polynomial-time computation*, in these Proceedings.
12. Luks, E. M. and McKenzie, P. *Parallel algorithms solvable permutation groups*, J. Comp. Sys. Sci., 37 1988, 39-62.
13. Mark, P. D. *Sylow's Theorem and Parallel Computation* Ph.D. Dissertation, University of Oregon, 1993.
14. McKenzie, P. and Cook, S. A. *The parallel complexity of the abelian permutation group membership problem*, Proc. 24th IEEE FOCS, 1983, 1-20.
15. Mulmuley, K. *A fast parallel algorithm to compute the rank of a matrix over an arbitrary field*, Combinatorica 7, 1987, 101-104.
16. Pippinger, N. *On simultaneous resource bounds*, Proc. 20th IEEE FOCS 1979, 307-311.
17. Pálfy, P. P., *A polynomial bound for the orders of primitive solvable groups*, J. Algebra 77, 1982, 127-137.
18. Rotman, J. J. *The theory of groups*, 3rd ed.Allyn and Bacon, 1984.
19. Sims, C. C., *Some group-theoretic algorithms*, in Springer Lecture Notes in Math. Vol 697, 1978, 108-124.
20. Wolf, T. R., *Solvable and nilpotent subgroups of* $GL(n, q^m)$, Can. J. Math. 34, 1982, 1097-1111.

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE, UNIVERSITY OF OREGON, EUGENE, OR 97403

E-mail address: pdm@cs.uoregon.edu