We study the invariance groups $S(f)$ of boolean functions $f \in B_n$ on $n$ variables, i.e. the set of all permutations on $n$ elements which leave $f$ invariant. We give necessary and sufficient conditions via Pólya's cycle index for a general permutation group to be of the form $S(f)$, for some $f \in B_n$. This leads to an almost optimal algorithm for deciding the representability of an arbitrary permutation group. For cyclic groups $G \leq S_n$ we give an $NC$-algorithm for determining whether the given group is of the form $S(f)$, for some $f \in B_n$. Further, it is proved that asymptotically "almost all" boolean functions have trivial invariance groups. For any formal language $L$ let $L_n$ be the characteristic function of the set of all strings in $L$ which have length exactly $n$ and let $S_n(L_n)$ be the invariance group of $L_n$. We use the classification results on maximal permutation groups to show that any language satisfying $|S_n(S_n(L_n))| = n^{O(1)}$ is in $NC^1$. As a consequence we show that the problem of "weight-swapping" (modulo a sequence of groups of polynomial index) is in $NC^1$.

1. Introduction

The aim of this paper is to study the invariance groups of boolean functions, provide "efficient" parallel algorithms for determining the representability of a given group as the invariance group of a boolean function, and use group-theoretic techniques in order to deduce results about the complexity of formal languages.

For any finite alphabets $A, B$, consider a function $f : A^n \to B$. The invariance group $S(f)$ consists of those permutations on $n$ letters which do not change the value of $f$ on any input, i.e. for all $(x_1, ..., x_n) \in A^n$,

$$f(x_1, ..., x_n) = f(x_{\sigma(1)}, ..., x_{\sigma(n)}).$$

Since $S(f)$ is a subgroup of the symmetric group $S_n$, one might hope that there is a relation between the "complexity" of $f$ and the size and/or group theoretic structure of $S(f)$. Of particular interest is the case where $f$ is a boolean function on $n$ variables, denoted by $f \in B_n$. If $f$ is a symmetric boolean function, then clearly the invariance group $S(f)$ equals the full symmetric group $S_n$.

Using a counting argument Lupanov-Shannon-Strassen have shown that "most" boolean functions have exponential size circuit complexity. Despite this result, very little is known concerning specific languages or families of boolean functions. Our interest in the present study arose from attempting to use group theoretic techniques in order to generalize the simple observation that any family $\{f_n : n \geq 0\}$, $f_n : 2^n \to 2$ of symmetric boolean functions has polynomial size circuits. Probabilistic techniques have been successfully used by several authors [Yao, Furst-Saxe-Sipser], etc., in order to obtain lower bounds on families of certain symmetric boolean functions. However, there are few results giving tight upper bounds, apart from the well-known fact that any family of symmetric boolean functions has log depth polynomial size circuits (formula size bounds have been obtained by various authors in this case). In this paper we indicate the applicability of group theory in obtaining upper bounds for the parallel
complexity of families of boolean functions. Our work is different from, but somewhat related to studies on the automorphism groups of error-correcting codes (e.g. k-th order Reed-Muller codes, which are specific k-dimensional subspaces of $2^n$ [MacWilliams et al.]), as well as to work in [Harrison, 1964] where group theoretic methods are used to calculate the number of non-equivalent boolean functions. Related work is also dimensional subspaces of $2^n$ [MacWilliams et al.], complexity of families of boolean functions.

Related work is also dimensional subspaces of $2^n$ [MacWilliams et al.], complexity of families of boolean functions. Related work is also dimensional subspaces of $2^n$ [MacWilliams et al.], complexity of families of boolean functions.

Automorphism groups of error-correcting codes (e.g. $k$ the group of primings of a boolean function, as well as theoretic methods are used to calculate the number of invariance groups of boolean functions, and on the other hand which permutation groups are isomorphic to invariance groups of boolean functions, (ii) to determine the complexity of deciding the representability of a permutation group, and (iii) derive results about the complexity of a given language by studying the size of its invariance group.

1.1. Scope of the Paper

In summary the goals of the paper are: (i) to determine the invariance groups of boolean functions, (ii) to determine (iii) derive results about the complexity of a given language by studying the size of its invariance group.

1.2. Results of the Paper

Following is an outline of the main results and contents of the paper. We begin by reminding the reader of the essential parts of Polya's beautiful enumeration theory that will be used in the present study and give some examples of important representable groups.

In section 4 we study the representation problem for general permutation groups. We distinguish between groups which are "(equal to a) representable" and groups which are "isomorphic to a representable" group. In the case of "isomorphic to a representable group", we show that every permutation group $\leq S_n$ is isomorphic to a representable group $S(f)$, for some $f : 2^n \rightarrow 2$; but as stated, this isomorphism is at the expense of increasing space from $n$ to $n(\log n + 1)$.

The problem is more interesting in the case of "equal to a representable group". We give a necessary and sufficient condition in terms of the Polya index for a general permutation group $\leq S_n$ to be of the form $S(f)$, for some boolean function $f : 2^n \rightarrow 2$. This leads to an almost optimal algorithm for determining the representability of general permutation groups. We also give an NC-algorithm which on input a cyclic group $G \leq S_n$ decides whether $G$ is representable, in which case it outputs a boolean function $f : 2^n \rightarrow 2$ such that $G = S(f)$. Finally we prove that asymptotically "almost all" boolean functions have invariance group which is equal to the identity (permutation group).

Given a language $L \subseteq \{0,1\}^n$, let $L_n$ be the characteristic function of the set of words of $L$ of length exactly $n$. Section 5 is concerned with the complexity of languages of polynomial index, i.e. languages $L$ for which $S_n : S_n(L) = n^{O(1)}$, where $S_n(L)$ denotes the invariance group of the boolean function $L_n$. We study the closure properties of the class of these languages and use classification results on maximal permutation groups in order to show that any language of polynomial index is in non uniform-NC$^1$, by showing that any such language is "almost symmetric".

As an immediate consequence of our methods we generalize the "almost necessary and sufficient property for a family of symmetric boolean functions to be in AC$^0$" given in [Fagin et al.] to all families of languages of polynomial index. As a further application of our techniques, we also show that for any sequence of permutation groups $G = \langle G_n \rangle \leq S_n : n \in \mathbb{N}$ of polynomial index the problem $SWAP(G)$ is in NC$^1$, where by $SWAP(G)$ we understand the following problem:

**Input.** $n \in \mathbb{N}, a_1, \ldots, a_n \in \mathbb{Q}^*$. 

**Output.** A permutation $\sigma \in G_n$ such that for all $1 \leq i < n$, $a_{\sigma(i)} + a_{\sigma(i+1)} \leq 2$, if such a permutation exists, and the response "NO" otherwise.

2. Preliminaries

Here we give some introductory definitions and results regarding permutation groups and complexity of circuits that will be used in our subsequent investigations. The three topics we will discuss include: (1) the group index, (2) the cycle index and its computation via Polya's formula, and (3) complexity of boolean functions via the size of the circuits realizing them; the circuits concerned are constructed with "or", "and", and "not" gates.

2.1. Index of a Permutation Group

In the sequel it will be convenient to think of permutations on the set $\{1,2,\ldots,n\}$ as bijective mappings on positive integers such that for all $k > n$, $O(k) = k$. Part of this paper is primarily concerned with "large" permutation subgroups of the full symmetric group. Let $S_n$ denote the group of permutations on $n$ elements, and $A_n$ be the subgroup of even permutations (also known as the alternating group on $n$ letters). In general, for any non-empty set $\Omega$ let $S_\Omega$ denote the set of permutations on $\Omega$. For any group $G$ the symbol $H \leq G$ means that $H$ is a subgroup of $G$. Regarding the sizes of permutation groups the following theorem summarizes some of the known results.

**Theorem 1.** Let $H \leq S_n$ be a permutation group.

1. If $H \neq S_n$ and $H \neq A_n$ then $|S_n : H| \geq n$.
2. If the order of $H \neq A_n$ is maximal then $|S_n : H| = n$. In fact, for $n \neq 6$ the only subgroups $H$ of $S_n$ with $|S_n : H| = n$ are exactly the one point stabilizers of $S_n$.
3. (Bochert) $|S_n : H| \leq (n+1)/2$.
4. If $H$ is a "known" group or else $|H| < n^{10\log n}$.

2.2. Cycle Index of a Permutation Group

The main objects of study of this paper are boolean functions and their invariance groups. Let $B_{n,k}$ be the set of all $k$-valued functions $f : 2^n \rightarrow k$ on $n$ boolean variables. If $k = 2$ then we abbreviate $B_{n,2}$ by $B_n$. For $x = (x_1, \ldots, x_n) \in 2^n$ and any permutation $\sigma \in S_n$, let
The invariance group of \( f \) is defined by

\[
\text{S}(f) = \{ \sigma \in \text{S}_n : f = f^\sigma \}.
\]

(If \( K \subseteq \{0,1\}^n \) is a set of words of length \( n \) then by abuse of notation we shall write \( \text{S}(K) \) for the invariance group of the characteristic function of the set \( K \).) If \( L \subseteq \{0,1\}^n \) is a set of finite words and \( n \geq 1 \) then \( \text{S}_n(L) \) denotes the invariance group of the \( n \)-ary boolean function \( L_n \). Study of these groups leads very naturally to the cycle index of a permutation group, which we now proceed to define.

Let \( G \) be a permutation group on \( n \) elements. Define an equivalence relation \( i \sim j \) if and only if for some \( \sigma \in G \), \( \sigma(i) = j \). The equivalence classes under this equivalence relation are called orbits. Let \( G_i = \{ \sigma \in G : \sigma(i) = i \} \) be the stabilizer of \( i \), and let \( i^G \) be the orbit of \( i \). The stabilizer theorem asserts that \( |G_i| | i^G | = 1\). Using this last theorem we can obtain the well known theorem of Burnside and Frobenius, which states that for any permutation group \( G \) on \( n \) elements, the number of orbits of \( G \) is equal to the average number of fixed points of a permutation \( \sigma \in G \),

\[
\omega_n(G) = \frac{1}{|G|} \sum_{\sigma \in G} |i : \sigma(i) = i|,
\]

where \( \omega_n(G) \) is the number of orbits of \( G \). Any permutation \( \sigma \in \text{S}_n \) can be identified with a permutation on \( 2^n \) defined as follows:

\[
x = (x_1, ..., x_n) \rightarrow x^\sigma = (x_{\sigma(1)}, ..., x_{\sigma(n)}).
\]

Hence, any permutation group \( G \) on \( n \) elements can also be thought of as a permutation group on the set \( 2^n \). If we define \( x^G = \{ x^\sigma : \sigma \in G \} \) and call \( o(\sigma) \) the number of orbits of (the group generated by) \( \sigma \) then we obtain Pólya's formula:

\[
|\{ x^G : x \in 2^n \}| = \frac{1}{|G|} \sum_{\sigma \in G} o(\sigma).
\]

The number \( |\{ x^G : x \in 2^n \}| \) is called the cycle index of the permutation group \( G \) and will be denoted by \( \Theta(G) \). If we want to stress the fact that \( G \) is a permutation group on \( n \) letters then we write \( \Theta_n(G) \), instead of \( \Theta(G) \). For more information on Pólya's enumeration theory the reader should consult [Berge] and [Pólya et al.].

2.3. Circuits

We consider \( n \)-ary, 2-valued, fan-in circuits \( \alpha \) constructed from the gates \( \land, \lor, \land \). For such a circuit \( \alpha \) let \( c(\alpha) \) denote the number of gates of \( \alpha \) and \( d(\alpha) \) denote the depth of \( \alpha \), i.e. the maximal length from an input to the output node. Intuitively speaking, for a circuit \( \alpha \) \( d(\alpha) \) represents "time", while \( c(\alpha) \) represents the "number of processors". Any boolean function \( f : 2^n \rightarrow 2 \) can be realized by such a circuit. We define

\[
c(f) = \min(c(\alpha) : \alpha \text{ realizes } f).
\]

Regarding the size of \( c(f) \) the following results are known [Yablonsky]: for any symmetric \( f \in \text{B}_n \), \( c(f) = O(n) \).

(Lupanov-Shannon-Strassen)

\[
1 \{ f \in \text{B}_n : c(f) < q \} = O(q^{n+1}).
\]

A language \( L \subseteq \{0,1\}^n \) is said to have (or be solvable by) polynomial size circuits, denoted \( L \in \text{SIZE}(n^O(1)) \), if there is a circuit family \( \langle \alpha_n : n \in \mathbb{N} \rangle \) where \( \alpha_n \) computes the characteristic function of \( L_n = L \cap \{0,1\}^n \) and \( c(\alpha_n) \leq p(n) \), for some polynomial \( p \). A language \( L \subseteq \{0,1\}^n \) is in non-uniform \( \text{NC}^k \) if there is a circuit family \( \langle \alpha_n : n \in \mathbb{N} \rangle \) where \( \alpha_n \) recognizes \( L_n \) and in addition it is true that

\[
d(\alpha_n) = O((\log n)^k), \quad \text{and} \quad c(\alpha_n) = n^{O(1)}.
\]

We also define \( \text{NC} = \bigcup_k \text{NC}^k \), the class of problems (languages) solvable in polylog \((\log n)^O(1)\) time using non-uniform polynomial size circuits ([Pippenger]). Similarly, \( \text{AC}^k \) is defined like \( \text{NC}^k \) but with unbounded fan-in \( \lor, \land \)-gates. Also \( \text{AC} = \bigcup_k \text{AC}^k \) and \( \text{NC}^k \subseteq \text{AC}^k \subseteq \text{NC}^{k+1} \).

3. Examples of Invariance Groups and Languages

To give some intuition behind the concepts before presenting our results, we exhibit some illustrative examples.

The palindrome language is defined as the set of all strings \((\text{in the alphabet } \Sigma, \text{with at least two elements}) \ u = u_1 \cdots u_n \text{ such that } \forall i (u_i = u_{n-i+1}) \). If \( L \) is the palindrome language then

\[
\sigma \in \text{S}_n(L) \iff (\forall i \leq n)(\sigma(n-i+1) = n-\sigma(i)+1).
\]

Moreover, \( \text{S}_n(L) \) is isomorphic to \( \text{S}_n(2) \times (\mathbb{Z}_2)^{n/2} \).

The Dyck language \( D \) ([Harrison, 1978]) is defined as the least set of strings in the alphabet \( \{0,1\} \) such that \( A \in D \) and \( (\forall x, y \in D)(xy \in D \text{ and } 0x \in D) \). If \( D \) is the Dyck language then

\[
\text{S}_n(D) = \begin{cases}
\{id_n\} & \text{if } n \text{ is odd} \\
\langle (i, i+1) : i < n \text{ is even} \rangle & \text{if } n \text{ is even}
\end{cases}
\]

A language \( L \) is said to realize a sequence \( G = \langle G_n : n \geq 1 \rangle \) of permutation groups \( G_n \leq S_n \) if it is true that \( S_n(L) = G_n \), for all \( n \). We consider the following types of groups:

\text{Reflection.} \ R_n = \langle p \rangle, \text{ where } p(i) = n + 1 - i \text{ is the reflection permutation,} \\
\text{Cyclic.} \ C_n = \langle (1, 2, ..., n) \rangle, \\
\text{Dihedral.} \ D_n = C_n \times R_n, \\
\text{Hyperoctahedral.} \ O_n = \langle (i, i+1) : i \text{ is even } \leq n \rangle.

Let \( L = 0^*1^* \): then \( L \) realizes the family of identity permutation groups. Let \( L = 0^*1^*0^* \): then \( L \) realizes the family of reflection groups. Let \( L = 0^*1^*0^* \cup 1^*0^*1^* \): then \( L \) realizes the family of dihedral groups. If the regular language \( L = L_1 \cup L_2 \), where \( L_1 \) is the union of the languages \( 1^*0^*1^* \), \( 0^*1^*0^* \), \( 101000^*1, \ 0^*1101000^*1, \ 0^*011010^* \).
0\,001101, 10\,001100, 010\,00111, and \( L^2 \) is the complement of the language 10\,001101 then \( L \) realizes the family of cyclic groups for \( n \geq 6 \). It follows from our analysis that that for \( 3 \leq n \leq 5 \) the groups \( C_n \) are not representable as the invariance groups of a boolean function. Let \( L \) consist of the set of all finite strings \( x = (x_1,\ldots,x_k) \) such that for some \( i \leq k/2, x_{2i-1} = x_{2i} \). Then \( L \) realizes the family of hyperoctohedral groups.

There is no regular language \( L \) such that \( S_2n(L) = (S_2n(1,2,\ldots,n)) \), for all but a finite number of \( n \). There is a regular language \( L \) such that for all \( n \) we have that \( S_2n(L) = (S_2n(2i : i \leq n/2)) \), where \( G_\Delta \) is the pointwise stabilizer of the permutation group \( G \) on the set \( \Delta \).

4. Representations of Permutation Groups

The aim of this section is to give general results on permutation groups \( G \leq S_n \) which can be represented as the invariance groups of boolean functions, i.e. \( G = S(f) \) for some \( f \in B_n \). It will be seen in the sequel that there is a "rich" class of permutation groups which are representable thus.

A useful observation is that the alternating group \( A_n \) is not the invariance group of any boolean function \( f \in B_n \) provided that \( n \geq 3 \). Although this will follow directly from our representation theorem it will be instructive to give a direct proof. We claim that for any boolean function \( f \in B_n \) if \( A_n \subseteq S(f) \) then \( S_n = S(f) \). Indeed, assume that \( A_n \subseteq S(f) \) and let \( \sigma \in S_n \) be an arbitrary permutation. We must prove that also \( \sigma S(f) \). If \( \sigma \) is the product of an even number of transpositions then by assumption \( \sigma S(f) \). If \( \sigma \) is the product of an odd number of transpositions then we claim that for all \( x \in 2^n(f(x)^2 = f(x)) \). To prove this let \( x \in 2^n \) be arbitrary. Since \( n \geq 3 \) there exist \( i < j \) such that \( x_i = x_j \). It follows that \( x^2 = x \), for the transposition \( t = (i,j) \). Hence \( f(x^2) = f(x^0) = f(x) \), as desired. In either case we have that \( \sigma S_n(f) \). which is a contradiction. As a matter of fact it is clear, using part (1) of theorem 1 of the previous section, that \( A_n \) is not even isomorphic to the invariance group \( S(f) \) of any \( f \in B_n \). However, \( A_n \) is isomorphic to the invariance group \( S(f) \) for some boolean function \( f \in B_n(\log n + 1) \) (see theorem 2, below).

4.1. Elementary Properties

Before we proceed with the general results we will prove several simple observations that will be used frequently in the sequel.

Theorem 1.

1. If \( f \in B_n \) is symmetric then \( S(f) = S_n \).
2. \( S(f) = S(\Theta \varphi) \), for all \( f \in B_n \).
3. For any permutation \( \sigma, S(f)^\sigma = S(f)^\varphi \).
4. For each \( f \in B_n, S(f) = S(f) \).
5. If \( f_1,\ldots,f_k \in B_n \), \( f \in B_k \), \( g = f(f_1,\ldots,f_k) \in B_n \) then \( S(f_1) \cdots S(f_k) \subseteq S(g) \).
6. \( \forall k \leq n \forall f \in B_n, S(f) = S_k \).

A permutation group \( G \leq S_n \) is called representable (respectively, strongly representable) if there exists an integer \( k \) and a function \( f \in B_{n,k} \) (respectively, with \( k = 2 \)) such that \( G = S(f) \). It will be seen in the sequel (representability theorem) that the distinction representable and strongly representable is superfluous since these two notions coincide.

An important issue concerns the number of variables allowed in a boolean function in order to represent a permutation group \( G \leq S_n \).

Theorem 2. (Isomorphism Theorem) Every finite permutation group \( G \leq S_n \) is isomorphic to the invariance group of a boolean function \( f \in B_n(\log n + 1) \).

Proof. Fix \( n \) and let \( s = \log n + 1 \). View each word \( w \in \{0,1\}^{sn} \) (of length \( ns \)) as consisting of \( n \) many blocks \( w(1), w(2), \ldots, w(n) \) each of length \( s \). For a given permutation group \( G \leq S_n \) let \( L_G \) be the set of all words \( w \in \{0,1\}^{sn} \) such that either (i) exactly one of the blocks of \( w \) consists of 1s, while the rest of the blocks consist only of 0s; or (ii) \( w \) has less than \( s \) 1s and for each \( 1 \leq i \leq n \), the complement of the \( i \)th block of \( w \) is monotone (this implies that each \( w(i) \) consists of a sequence of 1s concatenated with a sequence of 0s); or (iii) \( w \) has at least \( n \) 1s and for each \( 1 \leq i \leq n \) the first bit of \( w(i) \) is 0 and the binary representations of the words \( w(i) \), say \( \text{bin}(w,i) \), are pairwise distinct and \( \sigma_n \in G \) where \( \sigma_n : \{1,\ldots,n\} \to \{1,\ldots,n\} \) is the permutation defined by \( \sigma_n(i) = \text{bin}(w,i) \).

The intuition for items (i) and (ii) above is the following. The words with exactly \( s \)-many 1s have all these 1s in exactly one block. This guarantees that any permutation "respecting" the language \( L_G \) must map blocks to blocks. By considering words with a single 1 (which by monotonicity must be located at the first position of a block) we guarantee that each permutation "respecting" \( L_G \) must map the first bit of a block to the first bit of some other block. Inductively, by considering the word with exactly \( (r-1)\)-many 1s all located at the beginning of a single block, while all other bits of the word are 0s, we guarantee that each permutation "respecting" \( L_G \) must map the \( (r-1)\)st bit of each block to the \( (r-1)\)st bit of some other block. It follows that any permutation respecting \( L_G \) must respect blocks as well as the order of elements in the blocks. Using this we can complete the proof of the theorem.

We conclude this section by comparing the different definitions of representability given above.
Theorem 3. For any permutation group \( G \leq S_n \) the following statements are equivalent: 
1. \( G \) is representable.
2. \( G \) is the intersection of a finite family of strongly representable permutation groups.
3. For some \( m \), \( G \) is a pointwise stabilizer of a strongly representable group over \( S_{n+m} \), i.e. 
   \[ G = (S_{n+m}(f))(a_{n+1}, \ldots, a_m) \] 
   for some \( f \in B_{n+m} \) and \( m \leq n \).

4. Representation Theorems for General Permutation Groups

Here we study the representability problem for general permutation groups, giving a necessary and sufficient condition via Pólya’s cycle index for a permutation group to be representable and show that the notions of representable and strongly representable coincide. In order to state the first general representation theorem we define for any \( n+1 \leq 0 \leq 2^n \) and any permutation group \( G \leq S_n \) the set \( G_k^{(0)} = \{ M \leq G : \Theta_k(M) = \emptyset \} \). Also, for any \( H \leq S_n \), any \( g \in S_n \), the notation \( <H, g> \) denotes the least subgroup of \( S_n \), containing the set \( H \cup \{ g \} \).

Theorem 4. (Representation Theorem) The following statements are equivalent for any permutation groups \( H < G \leq S_n \):
1. \( H = G \cap K \), for some strongly representable permutation group \( K \leq S_n \).
2. \( H = G \cap K \), for some strongly representable permutation group \( K \leq S_n \).
3. \( \forall g \in G-H \) \( \Theta_n(<H,g>) < \Theta_n(H) \).
4. \( H \) is maximal in \( G_k^{(0)} \), where \( \Theta_n(H) = \emptyset \).

Proof. We prove the equivalence of the above statements by showing the following sequence of implications: \( (1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (1) \) and \( (4) \Rightarrow (3) \Rightarrow (4) \). The proof of \( (1) \Rightarrow (2) \) is trivial. First we prove \( (2) \Rightarrow (3) \). By results of the previous section \( K \) is the intersection of a family strongly representable groups. Hence by assumption let \( \mathcal{S}(f_i) \) such that \( \{ f_i \} \subseteq \mathcal{B}_n \), be a finite family of invariance groups such that

\[ H = \bigcap_i \mathcal{S}(f_i) \cap G. \]

Assume on the contrary that there exists an \( H < K \leq G \) such that \( \Theta(K) = \Theta(H) \). This last statement is equivalent to the statement

\[ \forall x \in 2^n (x^K = x^H). \]

We show that in fact

\[ K \subseteq \bigcap_i \mathcal{S}(f_i) \cap G, \]

which is a contradiction since the right-hand side of the above inequality is equal to \( H \). Indeed, let \( \sigma \in K \) and \( x \in 2^n \). Then we know that 

\[ x^K = (x^\sigma)^K = (x^\sigma)^H. \]

It follows that \( x = (x^\sigma)^2 \), for some \( \tau \in H \). Consequently, \( f_i(x) \tau = f_i((x^\sigma)^2) = f_i(x^\sigma) \), as desired.

Next we prove that \( (3) \Rightarrow (1) \). By assumption for all \( g \in G-H \), 

\[ \Theta_n(<H,g>) < \Theta_n(H). \]

In particular, for all \( g \in G-H \), there exists a boolean function \( f_g \in B_n \) such that \( H \leq S_n(f_g) \), but \( <H,g> \) is not a subset of \( S_n(f_g) \). Consider the representable group \( K \) defined by

\[ K = \bigcap_{g \in G-H} S(f_g). \]

It is now trivial to check that \( H = K \cap G \). Moreover, as in the implication \( (2) \Rightarrow (3) \) above, it follows that the permutation group \( K \) satisfies property \( P \), i.e. \( \forall L > K \) \( \Theta_n(L) < \Theta_n(K) \). We want to show that \( K \) is strongly representable. Assume on the contrary that this is false and let \( K \) be of maximal size satisfying \( P \), but is not strongly representable. It follows that \( \forall L > K \) \( L \) satisfies \( P \Rightarrow L \) is strongly representable.

Since the full symmetric group \( S_n \) is strongly representable we can assume without loss of generality that \( K < S_n \). In particular, there is strongly representable group \( L > K \) of minimal size. Let \( h \in B_n \) be such that \( L = \mathcal{S}(h) \). For the groups \( K < L \) chosen as before we have that

\[ \forall M (K < M < L \Rightarrow M \text{ does not satisfy } P \). (*) \]

In the sequel we derive a contradiction from (*) by showing that in fact \( K \) is strongly representable. Since \( K \) satisfies property \( P \), we have that \( \Theta_n(L) < \Theta_n(K) \). It follows that there exist \( x, y \in 2^n \) such that

\[ x \equiv y \mod L, \quad x \neq y \mod K, \]

where for \( H \leq S_n \) and \( x, y \in 2^n \) the symbol \( x \equiv y \mod H \) means that \( y = x^\sigma \), for some \( \sigma \in H \). Define a boolean function \( g \in B_n \) as follows, for \( w \in 2^n \),

\[ g(w) = \begin{cases} 
  h(w) & \text{if } w \equiv x \mod K, \quad w \neq y \mod K \\
  0 & \text{if } w = x \mod K \\
  1 & \text{if } w = y \mod K.
\end{cases} \]

It follows from the definition of the boolean function \( g \) that \( K \leq S(g) < \mathcal{S}(h) = L \). Since every strongly representable group satisfies \( P \), an immediate consequence of (*) is that \( K = \mathcal{S}(h) \). This completes the proof \( (3) \Rightarrow (1) \).

It remains to prove the equivalence of the last statement of the theorem. First we prove \( (4) \Rightarrow (3) \). Assume that \( H \) is a maximal element of \( G_k^{(0)} \), but that for some \( g \in G-H \), we have that \( \Theta_n(<H,g>) = \Theta_n(H) \). But then \( H < <H,g> \leq G \), contradicting the maximality of \( H \). Finally we prove \( (3) \Rightarrow (4) \). Assume on the contrary that \( (3) \) is true but that \( H \) is not maximal in \( G_k^{(0)} \). This means there exists \( H < K \leq G \) such that \( \Theta_n(K) = \Theta_n(H) \). Take any \( g \in G-H \) and notice that

\[ \Theta_n(<H,g>) \geq \Theta_n(K) = \Theta_n(H) \geq \Theta_n(<H,g>). \]

Hence, \( \Theta_n(H) = \Theta_n(<H,g>) \), contradicting \( (3) \) \( \circ \).

A "naive" algorithm for testing the representability of a general permutation group \( G \leq S_n \) to test all
Algorithm for Deciding the Representability of Permutation Groups

Input
A permutation group \( G \leq S_n \).
for each \( \sigma \in S_n - G \) do
if \( \Theta_\sigma(<G, \sigma>) = \Theta_\sigma(G) \) then output \( G \) is not representable.
else output \( G \) is representable.
\end{algorithm}

As a matter of fact we can polynomially reduce the negation of the well-known graph isomorphism problem (NGIP) to the above group representation problem. Indeed, let

\[ G = (\{u_1, \ldots, u_n\}, E_G), H = (\{v_1, \ldots, v_n\}, E_H) \]

be two graphs on \( n \) vertices each. Consider the permutation group \( ISO(G, H) \leq S_{n+3} \) whose generators \( \sigma \) satisfy:

\[ \forall i, j \in \{0, 1\}. (E_G(u_i, u_j) \Leftrightarrow E_H(v_{\sigma(i)}, v_{\sigma(j)})) \]

and in addition the permutation \( n+i \mapsto \sigma(n+i), i = 1, 2, 3, \) belongs to the cyclic group \( C_3 \). It is easy to see that if \( G, H \) are isomorphic then there exists a group \( K \leq S_n \) such that \( ISO(G, H) = K \times C_3 \). On the other hand, if \( G, H \) are not isomorphic then \( ISO(G, H) \neq \langle id_n, \tau \rangle \). As a consequence of the non-representability of \( C_3 \) and the representability theorem of direct products, it follows that \( G, H \) are not isomorphic if and only if \( ISO(G, H) \) is representable. (See [Schöning], page 574, for a discussion on the complexity of NGIP.)

The previous theorem also has a consequence concerning the representation of "maximal" permutation groups.

**Theorem 5.** (Maximality Theorem)

1. If \( H \) is a maximal proper subgroup of \( G \leq S_n \) then \( \Theta_\sigma(G) \neq \Theta_\sigma(H) \Leftrightarrow (\exists f \in B_n) [H = G \cap S(f)] \).
2. All maximal subgroups of \( S_n \) are strongly representable, the only exceptions being: (a) the alternating group \( A_n \), for all \( n \geq 3 \); (b) the 1-dimensional, linear, affine group \( AGL_1(5) \) over the field of 5 elements, for \( n = 5 \); (c) the group of linear transformations \( PGL_2(5) \) of the projective line over the field of 5 elements, for \( n = 6 \); (d) the group of semi-linear transformations \( PGL_2(8) \) of the projective line over the field of 8 elements, for \( n = 9 \).

As noted above all maximal permutation groups with the exception of \( A_n \) are of the form \( S(f) \), provided that \( n \geq 10 \). Such maximal permutation groups include: the cartesian products \( S_k \times S_{n-k} \) for \( k \leq n/2 \), the wreath products \( S_k \wr S_j \) for \( k = jl, k,j > 1 \), the affine groups \( AGL_d(p) \), for \( n = p^d \), etc. The interested reader will find a complete survey of classification results for maximal permutation groups in [Kleidman et al.]. It should also be pointed out that there are plenty of non maximal permutation groups which are not representable. In fact it can be verified that examples of such groups are the wreath products \( G \wr A_n \). In general we can prove the following theorem for any permutation groups \( G \leq S_m, H \leq S_n \).

**Theorem 6.**

1. \( G \) and \( H \) representable \( \Rightarrow G \cap H \) is representable.
2. \( G \cap H \) is representable \( \Rightarrow H \) is representable.
3. For \( p \) prime, a \( p \)-Sylow subgroup \( P \) of \( S_n \) is representable \( \Leftrightarrow p \not\in \{3, 4, 5\} \).

**Proof.** The proof of the above theorem is not difficult. The proof of (3) uses the fact that \( p \)-Sylow subgroups of \( S_n \) can be obtained as conjugates of wreath product iterations of \( C_p \) [Passman].

The converse of part (1) of the above theorem is not necessarily true. This is easy to see from the following example. We show that the wreath product \( A_3 \wr S_2 \) is representable, however \( A_3 \) is not. Indeed, consider the language \( L \) consisting of the six strings

\[ 001101, 010011, 110100, 001110, 100011, 111000 \]

We already proved that \( A_3 \) is not representable. We claim that \( A_3 | S_2 = S_3(L) \). Consider the three-cycle \( \tau = (1, 2, 3) \) like in the three-cycles \( \tau, \tau^2, \tau^3 \). A straightforward (but tedious) computation shows that \( S_3(L) \) also consists of exactly the above 24 permutations.

4.3. An NC Algorithm for the Representability of Cyclic Groups

The main result of the present section is the following theorem.

**Theorem 7.** (Cyclic Group Representability)

There is an NC algorithm which when given as input a cyclic group \( G \leq S_n \) decides whether the group is representable, in which case it outputs a function \( f \in B_n \) such that \( G = S(f) \).

The representability of general abelian groups which can be decomposed into disjoint factors can be decided with the following NC algorithm: (1) use an NC algorithm [Luks et al., McKenzie, McKenzie et al., Mulmuley] to "factor" the given abelian group into its cyclic factors and then (2) use the "cyclic-group" algorithm given in the box to each of the cyclic factors of the given permutation group \( G \). In view of the first lemma below the group \( G \) is representable exactly when each of its disjoint, cyclic factors is.

**Lemma 1.** For \( G \leq S_m, H \leq S_n \) permutation groups, \( G \times H \leq S_{m+n} \) is representable \( \Leftrightarrow \) both \( G, H \) are representable.
Algorithm for Representing Cyclic Groups

Input
G = <\sigma> cyclic group.

Step 1
Decompose \( \sigma = \sigma_1 \sigma_2 \cdots \sigma_k \), where \( \sigma_1, \sigma_2, \ldots, \sigma_k \) are disjoint cycles of lengths \( l_1, l_2, \ldots, l_k \geq 2 \), respectively.

Step 2
if for all \( 1 \leq i \leq k \),
\[ l_i = 3 \Rightarrow (\exists j \neq i)(3 \mid l_j) \text{ and } l_i = 4 \Rightarrow (\exists j \neq i)(\text{gcd}(4, l_j) \neq 1) \]
and
\[ l_i = 5 \Rightarrow (\exists j \neq i)(5 \mid l_j) \]
then output G is representable.
else output G is not representable.

The rest of this section is dedicated to the proof (sketch) of correctness of the above algorithm. The proof is in a series of lemmas. At first we will need two definitions. A boolean function \( f \in B_n \) is called special if for all words \( w \) of length \( n \),
\[ |w|_1 = 1 \Rightarrow f(w) = 1. \]
Let \( \sigma_1, \ldots, \sigma_k \) be a collection of cycles. We say that the group \( G = <\sigma_1, \ldots, \sigma_k> \) generated by the permutations \( \sigma_1, \ldots, \sigma_k \) is specially representable if there exists a special boolean function \( f : 2^\Omega \to 2 \) (where \( \Omega \) is the union of the supports of the different \( \sigma_i \)) such that \( G = \text{Supp}(f) \). The support of a permutation \( \sigma \), denoted by \( \text{Supp}(\sigma) \), is the set of \( i \) such that \( \sigma(i) \neq i \). The support of a permutation group \( G \), denoted \( \text{Supp}(G) \), is the union of the supports of the elements of \( G \).

4.3.1. Main ideas of the Proof
Before proceeding with the details of the proof it will be instructive to give an outline of the main ideas needed for the correctness proof. We are given a cyclic group \( G \) generated by a permutation \( \sigma \). Decompose \( \sigma \) into disjoint cycles \( \sigma_1, \sigma_2, \ldots, \sigma_k \) of lengths \( l_1, l_2, \ldots, l_k \geq 2 \), respectively.

If \( k = 1 \) then we know that \( G \) is specially representable exactly when \( l_1 \neq 3, 4, 5 \). (The representability of the cyclic group \( C_s \), for \( s \neq 3, 4, 5 \), was discussed in section 3; for \( s = 3, 4, 5 \) observe that for any \( f \in B_s \), if \( C_s \subseteq \text{Supp}(f) \) then \( D_s \subseteq \text{Supp}(f) \).)

If \( k = 2 \) then the result will follow by considering several possibilities for the pairs \((l_1, l_2)\):
if \( \text{gcd}(l_1, l_2) = 1 \) then \( G = <\sigma_1 \sigma_2> \) is the direct product of \( <\sigma_1> \) and \( <\sigma_2> \). Hence, \( G \) is specially representable exactly when both factors are specially representable.
if \((l_1, l_2) = (3, 3) \) or \((4, 4) \) or \((5, 5) \) then \( G \) is specially representable.
if \((l_1, l_2) = (3, m) \) (with \( 3 \mid m \)) or \((4, m) \) (with \( \text{gcd}(4, m) \neq 1 \)) or \((5, m) \) (with \( 5 \mid m \)) then \( G \) is specially representable.

This will take care of deciding the representability of \( G \) for all possible pairs \((l_1, l_2)\). A similar argument will work for \( k \geq 3 \). This concludes the outline of the proof of correctness.

4.3.2. Outline of the Proof
The details of the above constructions are rather tedious but a sufficient outline is given in the sequel.

Lemma 2. Suppose that \( \sigma_1, \ldots, \sigma_{n+1} \) is a collection of cycles such that both \( <\sigma_1, \ldots, \sigma_n> \) and \( <\sigma_{n+1}> \) are specially representable and have disjoint supports. Then \( <\sigma_1, \ldots, \sigma_{n+1}> \) is specially representable.

An immediate consequence of the previous lemma is the following

Lemma 3. If \( G, H \) have disjoint support and are specially representable then \( GH \) is specially representable.

Next we will be concerned with the problem of representing cyclic groups. In view of the results of section 3 we know that the cyclic group \( <(1, 2, \ldots, n)> \) is representable exactly when \( n \neq 3, 4, 5 \). In particular, the groups \( <(1, 2, 3)>, <(1, 2, 3, 4)>, <(1, 2, 3, 4, 5)> \) are not representable. The following lemma may therefore come as a surprise.

Lemma 4. Let the cyclic group \( G \) be generated by a permutation \( \sigma \) which is the product of two disjoint cycles of lengths \( l_1, l_2 \), respectively. Then \( G \) is specially representable exactly when the following conditions are satisfied:
\( l_1 = 3 \Rightarrow 3 \mid l_2 \) and \( l_2 = 3 \Rightarrow 3 \mid l_1 \),
\( l_1 = 4 \Rightarrow \text{gcd}(4, l_2) \neq 1 \) and \( l_2 = 4 \Rightarrow \text{gcd}(4, l_1) \neq 1 \),
\( l_1 = 5 \Rightarrow 5 \mid l_2 \) and \( l_2 = 5 \Rightarrow 5 \mid l_1 \).

Proof. (Sketch) It is clear that the assertion of the lemma will follow if we can prove that the three assertions below are true.
(1) The groups \( <(1, 2, \ldots, n)(n+1, n+2, \ldots, kn)> \) are specially representable when \( n \neq 3, 4, 5 \).
(2) The groups \( <(1, 2, 3, 4)(5, \ldots, m+4)> \) are specially representable when \( \text{gcd}(4, m) \neq 1 \).
(3) Let \( m, n \) be given integers such that either \( m = n = 2 \) or \( m = 2 \) and \( n \geq 6 \) or \( n = 2 \) and \( m \geq 6 \) or \( m, n \geq 6 \). Then \( <(1, 2, \ldots, m)(m+1, m+2, \ldots, m+n)> \) is specially representable.

Proof of (1). We give the proof only for the case \( n = 5 \) and \( k = 1 \). The other cases are \( n = 3, n = 4 \) and \( k \geq 3 \) are treated similarly. Details of these constructions are left to the reader. Let \( \sigma = \sigma_0 \sigma_1 \), where \( \sigma_0 = (1, 2, 3, 4, 5) \) and \( \sigma_1 = (6, 7, 8, 9, 10) \). From the results of section 3 we know that \( D_3 = \text{Supp}(L') = \text{Supp}(L'') \), where \( L' = 0^1 0^0 1^1 0^1 0^0 \) and \( L'' = \{ w \in E' : |w|_1 \geq 5 \} \). Let \( L \) consist of all words \( w \) of length 10 such that
\( \text{ either } |w|_1 = 1 \)
\( \text{ or } |w|_1 = 2 \) and for some \( 1 \leq i \leq 5 \).
exists an \( \tau \) and suppose that it is enough to show that none of the permutations
\( x = 1 \) and \( w = 0 \) respectively. The group \( G \) is specifically representable generated by a permutation \( \sigma \) which can be decomposed into \( k \)-many disjoint cycles of lengths \( l_1, l_2, \ldots, l_k \), respectively. The group \( G \) is specially representable exactly when the following conditions are satisfied for all \( 1 \leq i \leq k \):
\[
\begin{align*}
l_1 &= 3 & \Rightarrow & \exists j \neq i (3 \not| l_j) & \text{and} \\
l_2 &= 4 & \Rightarrow & \exists j \neq i (gcd(4, l_j) \neq 1) & \text{and} \\
l_3 &= 5 & \Rightarrow & \exists j \neq i (5 \not| l_j). \end{align*}
\]

4.4. Asymptotic Behavior

Finally, we prove an asymptotic result, which indicates how difficult it is to represent permutation groups as the invariance groups of boolean functions. In fact, we prove that a "0–1 law" holds for sequences of groups.

**Theorem 8.** (0–1 Law for Sequences of Groups)

For any family \( \{G_n: n \geq 1\} \) of permutation groups such that each \( G_n \leq S_n \) we have that
\[
\lim_{n \to \infty} \frac{|\{f \in B_n : S(f) = \{id_n\} \}|}{2^n} = 1.
\]

Moreover, if \( \lim_{n \to \infty} |G_n| > 1 \) then
\[
\lim_{n \to \infty} \frac{|\{f \in B_n : S(f) \geq G_n\}|}{2^n} = 0.
\]

**Proof.** During the course of this proof we use the abbreviation \( \Theta(m) := \Theta_{\sigma_n}(<(1,2,\ldots,m)>) \). First we prove the second part of the theorem. By assumption there exists an \( n_0 \) such that for all \( n \geq n_0 \), \( |G_n| > 1 \). Hence, for each \( n \geq n_0 \), \( G_n \) contains a permutation of order \( k(n) \geq 2 \), say \( \sigma_n \). Without loss of generality we can assume that each \( k(n) \) is a prime number. Since \( k(n) \) is prime, \( \sigma_n \) is a product of \( k \)-cycles. If \( (i_1, \ldots, i_{k(n)}) \) is the first \( k \)-cycle in this product then it is easy to see that
\[
\Theta_{\sigma_n}(<(i_1)>) \leq \Theta_{\sigma_n}(<(i_1, \ldots, i_{k(n)})>).
\]

It follows that
\[
\frac{|\{f \in B_n : S(f) \geq G_n\}|}{2^n} \leq \frac{2^{\Theta_{\sigma_n}(<(i_1, \ldots, i_{k(n)})>)}}{2^{2n}}.
\]

Recall from [Berge] that the formula
\[
\Theta(m) = \frac{1}{m} \sum_{k \mid m} \phi(k) \cdot 2^{m/k}
\]
gives the Polya cycle index of the group \(<(1,2,\ldots,m)\rangle\) acting on \(\{1,2,\ldots,m\}\), where \(\phi(k)\) is Euler's totient function. However it is easy to see that for \( k \) prime
\[
\Theta(k) = \frac{1}{k} + \frac{2}{k} - \frac{2}{k^2}
\]
In fact the function in the right-hand side of the above equation is decreasing in \( k \). Hence, for \( k \) prime,
\[
\Theta(k) \leq \Theta(2) = \frac{3}{2}\n
It follows that
\[
\frac{|\{f \in B_n : S(f) \geq G_n\}|}{2^n} \leq \frac{2^{\Theta(k(n))}}{2^{2n}} \leq 2^{\Theta(k(n)) - 1} \leq 2^{-2n}.
\]

Since the right-hand side of the above inequality converges to 0 the proof of the second part of the theorem is complete. To prove the first part notice that
\[
\{f \in B_n : S(f) \neq id_n\} \subseteq \bigcup_{\sigma \in S_n} \{f \in B_n : \sigma \in S(f)\},
\]
where \(\sigma\) ranges over cyclic permutations of order a prime number \( \leq n \). Since there are at most \( n \) permutations on \( n \) letters we obtain from the last inequality that
\[
\frac{|\{f \in B_n : S(f) \neq id_n\}|}{2^n} \leq n \cdot 2^{-2n} = 2^{O(n \log n)} \cdot 2^{-2n/2} \to 0,
\]
as desired.

As a consequence of the above theorem we obtain that asymptotically almost all boolean functions have trivial invariance group.
5. Invariance Groups of Languages and Circuits

In this section we are interested in providing results which determine the complexity of languages according to the size of their invariance groups. Furthermore we consider questions concerning their structural properties and complexity. Recall that for each \( L \subseteq \{0,1\}^* \) and \( n \), \( L_n \) is the set of strings in \( L \) of length exactly \( n \). By abuse of notation we also denote the characteristic function of \( L_n \) with the same symbol. Let \( S_n(L) \) denote the invariance group of the \( n \)-ary boolean function \( L_n \). For any language \( L \) and any sequence \( \sigma = \langle \sigma_n : n \geq 1 \rangle \) of permutations such that each \( \sigma_n \in S_n \) we define the language \( L_\sigma = \{ x \in 2^n : x^{\sigma_n} \in L_n \} \). For each \( n \) let \( G_n \leq S_n \) and put \( G = \langle G_n : n \geq 1 \rangle \). Define

\[
L_G = \bigcup_{\sigma_n \in G_n} L_\sigma.
\]

Let \( L(P) \) be the set of languages \( L \subseteq \{0,1\}^* \) such that there exists a polynomial \( p \) satisfying \( \forall n (|S_n(L)| \leq p(n)) \). Occasionally, we will be referring to a language \( L \in L(P) \) as a language which has polynomial index. Following is a list of the structural properties of the class of languages \( L(P) \). For any \( L \in L(P) \),

1. \( L \subseteq L(P) \) is closed under boolean operations and homomorphisms.
2. \( (L, \Sigma) \in L(P) \).
3. \( L_\sigma \in L(P) \), where \( \sigma = \langle \sigma_n : n \geq 1 \rangle \), with each \( \sigma_n \in S_n \).
4. \( |S_n \cap S_{n-1}(G_n)| = n^{O(1)} \Rightarrow L_G \in L(P) \).

5.1. Circuit Complexity of Formal Languages

In the sequel we study the complexity of languages \( L \subseteq L(P) \). Our main result (theorem 4) will follow from the fact that any such language is "almost symmetric" (see claim in the proof of theorem 4). A simple upper bound proof on the complexity of languages in \( L(P) \) is given below.

**Theorem 2.** If \( L \in L(P) \) then \( L \) is in non-uniform \( NC^1 \).

**Proof.** As a first step in the proof we will need the following claim.

**Claim.** There is an \( NC^1 \) algorithm which, when given \( x \in \{0,1\}^* \), it outputs \( \sigma \in S_n \) such that \( x^{\sigma} = 1^{m}0^{n-m} \), for some \( m \).

**Proof of the claim.** Before giving the proof of the claim, we illustrate the idea by citing an example. Suppose that \( x = 1011001111. \) By simultaneously going from left to right and from right to left, we swap an "out-of-place" 0 with an "out-of-place" 1, keeping track of the respective positions.* This gives rise to the desired permutation \( \sigma \). We find \( \sigma = (2,9)(5,8)(6,7) \) and \( x^{\sigma} = 1^{9}0^{3} \).

Now we proceed with the proof of the main claim. By work of [Buss] the predicates \( E_{k,b}(u) \), which hold when there are exactly \( k \) occurrences of \( b \) in the word \( u \) \( (b = 0,1) \) are in \( NC^1 \). For \( k = 1, \ldots, \lfloor n/2 \rfloor \) and \( 1 \leq i < j \leq n \), let \( \alpha_{i,j,k} \) be a log depth circuit which outputs 1 exactly when the \( k \)th "out-of-place" 0 is in position \( i \) and the \( k \)th "out-of-place" 1 is in position \( j \). It follows that \( \alpha_{i,j,k}(x) = 1 \) if and only if there exist \( k-1 \) zeroes to the left of position \( i \), the \( i \)th bit of \( x \) is zero and there exist \( k \) ones to the right of position \( i \); and there exist \( k-1 \) ones to the right of position \( j \), the \( j \)th bit of \( x \) is one and there exist \( k \) zeroes to the left of position \( j \). This in turn is equivalent to

\[
E_{k-1,0}(x_{1} \cdots x_{i-1}) \text{ and } x_{i} = 0 \text{ and } E_{k,1}(x_{i+1} \cdots x_{n}) \text{ and }
E_{k-1,1}(x_{j+1} \cdots x_{n}) \text{ and } x_{j} = 1 \text{ and } E_{k,0}(x_{1} \cdots x_{j-1}).
\]

This implies that the required permutation can be defined by

\[
\sigma = \prod_{1 \leq i < j \leq \lfloor n/2 \rfloor} \alpha_{i,j,k}.
\]

Converting the fan-in, \( \lfloor n/2 \rfloor \)-v-gate into a \( \log(\lfloor n/2 \rfloor) \) depth tree of fan-in, \( 2 \)-v-gates, we have an \( NC^1 \) procedure for computing \( \sigma \). This completes the proof of the claim.

Next we continue with the proof of the main theorem. Put \( G_n = S_n(L) \) and let \( R_n = \{ h_1, \ldots, h_q \} \) be a complete set of representatives for the left cosets of \( G_n \), with \( q \leq p(n) \) and \( p(n) \) is a polynomial such that \( |S_n : G_n| \leq p(n) \). Fix \( x \in \{0,1\}^n \). By the previous claim there is a permutation \( \sigma \) which is the product of disjoint transpositions and an integer \( 0 \leq k \leq n \) such that \( x^{\sigma} = 1^{k}0^{n-k} \). So \( x = (1^{k}0^{n-k})^0 \). In parallel for \( i = 1, \ldots, q \) test whether \( h_i^{-1} \sigma \in G_n \) by using the principal result of [Babai et al.], thus determining \( i \) such that \( \sigma = h_i g \). For some \( g \in G_n \). Then we obtain that

\[
L_n(x) = L_n((1^{k}0^{n-k})^0) = L_n((1^{k}0^{n-k})^0). \]

By hardwiring the polynomially many values \( L_n((1^{k}0^{n-k})^0) \) for \( 0 \leq k \leq n \) and \( 1 \leq i \leq q \), we produce a polynomial size polylogarithmic depth circuit family for \( L \).

**Theorem 2** involves a straightforward application of the beautiful \( NC \) algorithm of Babai, Lukihs and Seress [Babai et al.] for testing membership in a finite permutation group. By using the deep structure consequences of the O'Nan-Scott theorem below, together with Bochert's result on the size of the index of primitive permutation groups (see theorem 1 (3) in section 2), we can improve the \( NC \) algorithm of theorem 2 to an optimal \( NC^1 \) algorithm. First, we take the following discussion and statement of the O'Nan-Scott theorem from [Kleidman et al.], page 376.

Let \( I = \{1,2, \ldots, n\} \) and let \( S_n \) act naturally on \( I \). Consider all subgroups of the following five classes of subgroups of \( S_n \):

\[\alpha_n: S_k \times S_n-k, \text{ where } 1 \leq k \leq n/2,\]
\[ \Omega_2 : S \alpha_4 \] \( S \alpha_5 \), where either \( n = ab \) and \( a, b > 1 \),

\( \Omega_3 : \text{the affine groups } AGL_{d}(p) \), where \( n = p^d \),

\( \Omega_4 : T^k(Out(T) \times S_{\alpha_4}) \), where \( T \) is a non-abelian simple group, \( k \geq 2 \) and \( n = |T|^k-1 \),
as well as all groups in the class

\( \Omega_5 : \text{almost simple groups acting primitively on } I \).

**Theorem 3.** (O'Nan-Scott) Every subgroup of \( S \alpha_n \) not containing \( A \alpha_n \) is a member of \( \Omega_1 \cup \cdots \cup \Omega_5 \).

Now we can improve the result of theorem 2 in the following way.

**Theorem 4.** (Parallel Complexity of Languages of Polynomial Index) If \( L \in L(P) \) then \( L \) is in non-uniform NC.!

**Proof.** The proof requires the following consequence of the O'Nan-Scott theorem.

**Claim.** Suppose that \( <G_n : S \alpha_n : n \geq 1 > \) is a family of permutation groups such that for all \( n \), \( |S_n : G_n| = n^k \), for some \( k \). Then there exists an integer \( N \) such that for all \( n \geq N \) there exists an \( i_n \leq k \) for which \( G_n = U \alpha_n \times V \alpha_n \) with the supports of the groups \( U \alpha_n \), \( V \alpha_n \) disjoint and \( U \alpha_n \leq S \alpha_{n-i_n} \).

Before proving the claim we complete the details of the proof of theorem 4. Apply the claim to \( G_n = S \alpha_n(L) \) and notice that given \( x \in 2^n \), the question of whether \( x \) belongs to \( L \) is decided completely by the number of \( 1 \)s in the support of \( K \alpha_n = S \alpha_{n-i_n} \), together with information about the action of a finite group \( H \alpha_n \leq S \alpha_{i_n} \), for \( i_n \leq k \). Using the counting predicates as in the proof of theorem 2, it is clear that this is an NC algorithm. Hence, the proof of the theorem is complete assuming the claim.

**Proof of the claim.** We have already observed at the beginning of section 3 that \( G_n \neq A \alpha_n \). By the O'Nan-Scott theorem, \( G_n \) is a member of \( \Omega_1 \cup \cdots \cup \Omega_5 \). Using Bochert's theorem on the size of the index of primitive permutation groups (section 2, theorem 1 (3)), the observations of [1, liebeck et al.] concerning the primitivity of the maximal groups in \( \Omega_3 \cup \Omega_4 \cup \Omega_5 \) and the fact that \( G_n \) has polynomial index with respect to \( S \alpha_n \), we conclude that the subgroup \( G_n \) cannot be a member of the class \( \Omega_3 \cup \Omega_4 \cup \Omega_5 \). It follows that \( G_n \in \Omega_1 \cup \Omega_2 \). We show that in fact \( G_n \in \Omega_2 \). Assume on the contrary that \( G_n \leq H \alpha_n = S \alpha_{i_n} \). It follows that \( |H \alpha_n| = a \alpha(b)!^a \).

**Case 1.** \( n = ab \), for \( a, b > 1 \).

In this case it is easy to verify using Stirling's interpolation formula

\[ (n/e)^{n/e} < n! < (n/e)^{n/e} \sqrt{n} \]

that

\[ |S_n : H_n| = \frac{n!}{a \alpha(b)!^a} - \frac{a^{n-a}}{3b \alpha \alpha(3a)^{2/3}a^{\alpha}} \]

Moreover it is clear that the right-hand side of this last inequality cannot be asymptotically polynomial in \( n \), since \( a \leq n \) is a proper divisor of \( n \), which is a contradiction.

**Case 2.** \( n = a^b \), for \( a \geq 5, b \geq 2 \).

A similar calculation shows that asymptotically

\[ |S_n : H_n| = \frac{n!}{a \alpha(b)!^a} \approx \frac{n!}{a \alpha(b)!^a} \approx \frac{n!}{a \alpha(b)!^a} \]

where \( b' = a^{b-1} \). It follows from the argument of case 1 that this last quantity cannot be asymptotically polynomial in \( n \), which is a contradiction. It follows that

\[ G_n \in \Omega_1 \].

Let \( G_n \leq S \alpha_{n-i_n} \), for some \( 1 \leq i_n < n/2 \).

We claim that in fact \( i_n \leq k \), for all but a finite number of \( n \)'s. Indeed, put \( i_n = i \) and notice that

\[ |S_n : S \alpha_{n-i} | = \frac{n!}{(n-i)!} = \Omega(n^i) \leq |S_n : G_n| = n^k, \]

which proves that \( i \leq k \). It follows that \( G_n = U \alpha_n \times V \alpha_n \), where \( U \alpha_n \leq S \alpha_{i_n} \) and \( V \alpha_n \leq S \alpha_{2-i_n} \). Since \( i_n \leq k \) and \( |S_n : G_n| = n^k \) it follows that for \( n \) large enough \( V \alpha_n = S \alpha_{2-i_n} \). This completes the proof of the claim and hence of the theorem.

5.2. Applications

An immediate consequence of our analysis is that if \( <G_n \leq S \alpha_n : n \geq 1 > \) is a family of transitive permutation groups such that \( |S_n : G_n| = n \alpha(1) \) then \( G_n \) is not for all but a finite number of \( n \)'s (this answers a conjecture of D. Perrin). It is also possible to give a more algebraic formulation of the main consequence of theorem 4. A family \( <p_n : n \geq 1 > \) of multivariate polynomials in \( Z \alpha[x_1, ... x_n] \) is of polynomial index if \( |S_n : S(p_n)| = n \alpha(1) \).

**Theorem 5.** If \( <p_n : n \geq 1 > \) is family of multivariate polynomials (in \( Z \alpha[x_1, ... x_n] \)) of polynomial index then there is a family \( <q_n : n \geq 1 > \) of multivariate polynomials (in \( Z \alpha[x_1, ... x_n] \)) of polynomial length such that \( p_n = q_n \).

Because of the limitations of families of groups of polynomial index proved in the claim above, we obtain a generalization of the principal results of [Fagin et al.]. Namely, for \( L \subseteq \{0, 1\}^n \) let \( \mu_L(n) \) be the least number of input bits which must be set to a constant in order for the resulting language \( L \alpha = \{0, 1\}^n \) to be constant (see [Fagin et al.] for more details). Then we can prove the following theorem.

**Theorem 6.** If \( L \in L(P) \) (i.e. \( L \) is a language of polynomial index) then

\[ \mu_L(n) \leq (\log n) \alpha(1) \Rightarrow L \in AC^0. \]

\[ \mu_L(n) \geq n \alpha(1) \Rightarrow L \notin AC^0. \]

Our characterization of permutation groups of polynomial index given during the proof of theorem 4 can also be used to determine the parallel complexity of the "weight-swapping'' problem mentioned in the introduction.
Theorem 7. For any sequence \( G \) of permutation groups of polynomial index, the problem \( SWAP(G) \) is in non-uniform NC\(^1\).

Proof. By the characterization of sequences of groups of polynomial index, there exist integers \( k, N \) such that for all \( n \geq N \), \( G_n = H_n \times K_n \), where \( H_n \leq S_{i_n} \) and \( K_n = S_{i_n} - k \), with \( i_n \leq k \). Given \( n \geq N \), and \( n \) positive rational weights \( a_1, \ldots, a_n \) test whether there exist permutations \( \sigma \in H_n \) and \( \tau \in K_n \) such that for \( 0 \leq i < n \), \( a_{\sigma(i)} + a_{\sigma(i+1)} \leq 2 \), as follows. For \( \tau \), sort the set of weights \( \{a_i : i \in \text{Supp}(K_n)\} \) in decreasing order. Let \( \rho \in K_n \) be a “sorting” permutation such that \( a_{\rho(1)} \geq a_{\rho(2)} \geq \cdots \geq a_{\rho(n-1)} \). Test in parallel whether
\[
\begin{align*}
\rho(1) + \rho(n-1) & \leq 2, \\
\rho(2) + \rho(n-2) & \leq 2, \\
& \quad \vdots \\
\rho(n-1) + \rho(2) & \leq 2 
\end{align*}
\]
If so, then let \( \tau \) be the appropriate permutation such that
\[
\begin{align*}
j_1 & = \tau(1), \\
j_2 & = \tau(n-i_n), \\
& \quad \vdots \\
j_{n-i_n-1} & = \rho(n-i_n-1), \\
j_{n-i_n} & = \rho(n-i_n),
\end{align*}
\]
if \( n - i_n \) is even, and a variant of this, if \( n - i_n \) is odd.

Since sorting is in \( NC^1 \), computing \( \tau \) is in \( NC^1 \). Since \( H_n \leq S_{i_n} \), where \( i_n \leq k \), there are only a finite number of possibilities to test for \( \sigma \). These are hardwired (by non-uniformity) into the circuit.

The following conjecture might relate the cycle index of a sequence \( G = \langle G_n : n \geq 1 \rangle \) of groups with the circuit complexity of the language \( L \).

Conjecture 8. For any language \( L \subseteq \{0,1\}^* \), if \( L \in L_{\text{log}}(P) \) then \( L \) is in non-uniform NC.

This conjecture appears somewhat plausible, since it follows from the next theorem that if \( G = \langle G_n \leq S_n : n \geq 1 \rangle \) is a sequence of groups whose cycle index \( \Theta_n(G_n) \), as a function of \( n \), majorizes all polynomials, then there is a language \( L \) with \( S_n(L) \supseteq G_n \) and \( L \notin SIZE(n^{O(1)}) \).

Theorem 9. For any sequence of permutation groups \( G_n \leq S_n \) it is possible to find a language \( L \) such that\( L \notin SIZE(\sqrt{\Theta(G_n)}) \), and \( \forall n (S(L_n) \supseteq G_n) \).

6. Discussion and Open Problems

Three of the main questions we tried to answer in the present paper are (1) which permutation groups arise as the invariance groups of boolean functions, (2) which permutation groups are isomorphic to invariance groups of boolean functions, and (3) determine the complexity of deciding the representability of a permutation group. Concerning question (1), we saw that most (i.e. with a few exceptions) maximal permutation subgroups of \( S_n \) are representable. In the case of question (2), we have shown that every permutation group \( G \leq S_n \) is isomorphic to the invariance group of a boolean function \( f \in B_{\Omega(n)} \). However, we do not know if this last “upper bound” can be improved to \( f \in B_{\Omega(n)} \), for some constant \( c \) independent of \( n \). Concerning question (3), we gave an \( NC \)-algorithm for deciding the representability of cyclic groups. In general, however, we do not know of any efficient algorithm for deciding the representability of any “class” of permutation groups other than cyclic ones (e.g. abelian, nilpotent, solvable, etc.), aside from the fact that the negation of the graph isomorphism problem is reducible to the group representability problem.

In section 4 we studied the relation between the size of the index of the invariance group of a formal language and its complexity. By using results on maximal permutation groups we were able to show that essentially every language of polynomial size index is almost symmetric. In turn, this last statement implies that any language of “polynomial size index” is in non-uniform \( NC^1 \). It is possible that a finer analysis of the structure results for maximal permutation groups will yield a similar result for other classes of languages, like the ones with subexponential or even exponential size index. We also conjecture that a similar result is true for any language of “polynomial size Pólya index”. In the last part of section 5 theorem 9 we provide some evidence for this conjecture.

Acknowledgements

Discussions with Peter van Emde Boas, Danny Krizanc, Dominique Perrin, Paul Schupp, and Paul Vitányi are gratefully acknowledged. A. M. Cohen was extremely helpful with the literature on maximal permutation groups. Lambert Meertens made comments that significantly improved the presentation and pointed out that essentially our original, \( n^{2}\)-upper-bound proof of the isomorphism theorem could yield the improved, \( n(\log n + 1)\)-upper-bound.

REFERENCES

[6] Comtet, L., Analyse Combinatoire, 2me Tome,


Tuesday Morning Session

Chair: N. Immerman