LABELED VERSUS UNLABELED DISTRIBUTED CAYLEY NETWORKS

Evangelos Kranakis and Danny Krizanc

TR-215, NOVEMBER 1992

School of Computer Science, Carleton University
Ottawa, Canada, KIS 5B6

LABELED VERSUS UNLABELED DISTRIBUTED CAYLEY NETWORKS

Evangelos Kranakis*[†] (kranakis@scs.carleton.ca)

Danny Krizanc* (krizanc@scs.carleton.ca)

Abstract

We consider labelings (i.e. assignments of labels to the links that give the network a globally consistent orientation) on anonymous Cayley networks $\mathcal{N}_G$ constructed from a set G of generators of a group $\mathcal{G}$. Such networks can be endowed with a natural labeling $\mathcal{L}_G$ to form the oriented Cayley network, denoted by $\mathcal{N}_G[\mathcal{L}_G]$. We show that in general oriented Cayley networks are more powerful than unoriented Cayley networks, in the sense that the former can compute more Boolean functions than the latter. We also give a characterization of those abelian groups $\mathcal{G}$ which have a canonical set of generators G such that the network $\mathcal{N}_G$ computes more Boolean functions than the network $\mathcal{N}_G[\mathcal{L}_G]$.

1980 Mathematics Subject Classification: 68Q99
CR Categories: C.2.1
Key Words and Phrases: Anonymous Networks, Cayley Networks, Boolean function, Group of automorphisms, Labeled and unlabeled networks.
Carleton University, School of Computer Science: SCS-TR-215

[†] Carleton University, School of Computer Science, Ottawa, Ontario, K1S 5B6, Canada

[†] Research supported in part by NSERC grant # 907002.

1 Introduction

One of the main themes of investigation in distributed computation concerns the design of network topologies which have optimal efficiency characteristics with respect to several selected parameters, like complexity of routing and message transmission, fault tolerance, leader election, etc. Many network topologies have been studied in the literature, ranging from rings and meshes to hypercubes and butterflies. To optimize the complexity characteristics of the resulting algorithms one introduces labelings on (a subset of) the underlying network links in order to give the network a sense of direction (or orientation). However, even within such topologies the efficiency of distributed algorithms may vary widely depending on how the network links are labeled.

Intuitively speaking, by labeling of a network we understand an assignment of different labels (or directions) to some (or all) of the network links in order to give the network a globally consistent sense of direction. There are many different options for choosing a labeling on a subset L of the network links. For example, the links of a hypercube may be left unlabeled (in this case $L = \emptyset$), have a label representing the dimension along which the adjacent links are connected (in this case L is the set of all links of the hypercube), or even be labeled according to a Hamiltonian circuit (in this case L is the set of links constituting the Hamiltonian circuit). Although the underlying topology is the same (i.e. the hypercube), surprisingly the three networks have completely different computational characteristcs. For example, of the three cases previously specified only the second takes full advantage of the hypercube topology; in the first case the hypercube has no sense of direction, while in the third case it behaves like a ring.

The problem of labeling a network in order to achieve a globally consistent direction has been studied in the literature. There has been considerable interest in the problem of orienting a ring [13], [8]. A more general study is [14] which considers algorithms for labeling cliques, tori and hypercubes under the assumptions of (a) existence of a leader, (b) existence of processor identities, or (c) the processors being anonymous. In addition, the availability of a suitable orientation may significantly affect the message complexity of many important network computations [11], like leader election [9], computation of boolean functions [10], etc.

In this paper we are concerned with the impact of the introduction of labelings on Cayley networks. In particular we compare the computational power of labeled versus unlabeled networks by using the size of the corresponding class of boolean functions computable in the network as a comparison tool. The class of boolean functions computable on a network was first considered in [17, 16] for arbitrary networks. Here instead we consider only the class of Cayley networks. It turns out that in this case we can take advantage of the rich underlying group theoretic structure of the network in order to obtain a more systematic study of the corresponding classes of computable functions.

1.1 Anonymous Cayley networks

Cayley networks are connected graphs constructed from a group $\mathcal{G}$ and a set G of generators for $\mathcal{G}$ in the following way. The set of vertices is $\mathcal{G}$, and the set of edges E is defined by $E = \{(u,v) : u^{-1}v \in G\}$. We assume that $G = G^{-1}$, where G^{-1} is the set of g^{-1} such that $g \in G$. To avoid loops in the network $\mathcal{N}_G$ we assume $e \notin G$. Further if $g = g^{-1}$ then we identify the edges g and g^{-1}. This graph is denoted by $\mathcal{N}_G$.

We consider the following natural labeling $\mathcal{L}_G$ on the Cayley graph $\mathcal{N}_G$: the label of the edge (u,v) is $u^{-1}v$. We denote the resulting graph $\mathcal{N}_G[\mathcal{L}_G]$. By an automorphism ϕ of the labeled Cayley graph we mean that the edge-labels are preserved under ϕ, i.e. if $(u,v) \in E$ then

$$\mathcal{L}_G(u,v) = \mathcal{L}_G(\phi(u), \phi(v)). \tag{1}$$

The group of automorphisms of $\mathcal{N}_G$ satisfying (1) is denoted by $Aut(\mathcal{N}_G[\mathcal{L}_G])$. Clearly every Cayley graph is vertex transitive, in the sense that for any nodes $u, v \in \mathcal{G}$ there is a label preserving automorphism ϕ of $\mathcal{N}_G$ such that $\phi(u) = v$. The desired automorphism is $x \rightarrow \phi(x) = vu^{-1}x$. In fact this automorphism is uniquely determined from u and v, which makes the action of $Aut(\mathcal{N}_G[\mathcal{L}_G])$ on the vertices of $\mathcal{N}_G$ regular [15]. The Cayley graph has $|\mathcal{G}|$ nodes and the degree of each node is $|G|$, denoted by $d(G)$.

An anonymous Cayley network is a Cayley graph whose nodes are anonymous (i.e. having no identities), identical, and deterministic processors. We also assume that the processors know the network topology as well as the size of the network and the links are FIFO.

In general we are interested in computing Boolean functions on anonymous Cayley networks. Let B_N be the set of boolean functions on N variables. Let $\mathcal{N}_G = (V, E)$ be a Cayley network of size N, with node set $V = \{0, 1, \ldots, N-1\}$ and edge set $E \subseteq V \times V$. An input to $\mathcal{N}_G$ is an N-tuple $I =< b_v : v \in V >$ of bits $b_v \in \{0,1\}$, where processor v receives as input value the bit b_v. Given a function $f \in B_N$ known to all the processors in the network we compute f on input $I =< b_v : v \in V >$ as follows. During each step of the protocol the processors perform certain computations depending on their input value, their previous history and the messages they receive from their neighbors and then transmit the result of this computation to some or all of their neighbors. After a finite number of steps, predetermined by the initial conditions and the protocol, the processors terminate their computation and output a certain bit. A network computes the function f if for each input I, at the end of the computation each processor computes correctly the value $f(I)$.

1.2 Examples

Many well-known interconnection networks, like tori, hypercubes, n-Star, etc., in fact belong to the class of Cayley graphs [1]. Throughout we assume that

n is arbitrary but fixed. The following table gives a few examples of networks arising from abelian groups.

Network	Group	Generators	Size
Oriented Ring	C_N	$(1, 2, \ldots, N)$	N
d-Torus	$(C_n)^d$	direct product	$N = n^d$
n-Hypercube	$F_n(\cong Z_2^n)$	$\phi_{\{1\}}, \phi_{\{2\}}, \ldots, \phi_{\{n\}}$	$N = 2^n$

The generators of F_n (the automorphism group of the hypercube) we will consider are $\phi_{\{1\}}, \phi_{\{2\}}, \ldots, \phi_{\{n\}}$, where $\phi_{\{i\}}(x_1, x_2, \ldots, x_n)$ is the sequence of bits obtained from $(x_1, x_2, \ldots, x_n)$ by complementing the ith bit, while leaving the other bits unchanged. The next table provides a few examples arising from the symmetric group S_n.

Network	Group	Generators	Size
n-Star	S_n	$(1, k) : 1 < k \leq n$	$N = n!$
n-Bubble-Sort	S_n	$(k-1, k) : 1 < k \leq n$	$N = n!$
n-Pancake-Sort	S_n	$\rho_k, \bar{\rho}_k : 1 < k \leq n$	$N = n!$

Here $\rho_k, \bar{\rho}_k$ are the reflection permutations.

$$\begin{aligned} \rho_k &= \begin{pmatrix} 1 & 2 & \cdots & k \\ k & k-1 & \cdots & 1 \end{pmatrix} \\ \bar{\rho}_k &= \begin{pmatrix} n-k+1 & n-k+2 & \cdots & n \\ n & n-1 & \cdots & n-k+1 \end{pmatrix}. \end{aligned}$$

1.3 Automorphisms

Before proceeding any further we need to mention some results on the automorphism groups of Cayley networks. The first result connects the group $\mathcal{G}$ with the group of automorphisms $Aut(\mathcal{N}_G[\mathcal{L}_G])$. This is easily established from the fact that $Aut(\mathcal{N}_G[\mathcal{L}_G])$ is exactly the set of automorphisms $\{\phi_g : g \in \mathcal{G}\}$, where $\phi_g(u) = gu$. Namely, we have the following theorem.

THEOREM 1 *The group of automorphisms of $\mathcal{N}_G[\mathcal{L}_G]$ is isomorphic to $\mathcal{G}$.* ■

It is important to note that the networks $\mathcal{N}_G[\mathcal{L}_G]$ can be characterized as those transitive networks whose automorphism group has a regular transitive subgroup (see also [3][lemma 16.3]).

THEOREM 2 *If the automorphism group $Aut(\mathcal{N})$ of the transitive network $\mathcal{N}$ has a regular transitive subgroup $\mathcal{G}$ then there is a labeling $\mathcal{L}$ on $\mathcal{N}$ such that $\mathcal{G} = Aut(\mathcal{N}[\mathcal{L}])$. Conversely, the group of automorphisms of every Cayley network has a regular transitive subgroup.* ■

We will not prove these theorems here. Instead we refer the reader to [10] for details.

1.4 Computability

An important result for all our subsequent considerations concerns a characterization of the computability of Boolean functions on Cayley networks. Let $S(f)$ denote the group of permutations in S_N that leave f invariant on all inputs [5]. In general it is true that if a Boolean function f is computable on a network $\mathcal{N}$ then $S(f) \geq Aut(\mathcal{N})$. The converse is not necessarily true, in general. However it is true on oriented Cayley networks. We have the following theorem.

Theorem 3 *For any Boolean function $f \in B_N$, f is computable in the network $\mathcal{N}_G[\mathcal{L}_G]$ if and only if $S(f) \geq Aut(\mathcal{N}_G[\mathcal{L}_G])$.*

Proof (outline) The necessity of $S(f) \geq Aut(\mathcal{N}_G[\mathcal{L}_G])$ follows from the transitivity of the Cayley network. To prove the sufficiency of $S(f) \geq Aut(\mathcal{N}_G[\mathcal{L}_G])$ the processors execute an input collection algorithm across the labeled links. ■

For a complete proof as well as for non-trivial algorithms optimizing the bit complexity of computing Boolean functions the reader should consult [10].

1.5 Results of the paper

In this paper we compare the distributed computational power of labeled versus unlabeled Cayley networks by considering the classes of boolean functions computable in each of these networks. We will consider the following two possibilities on the network $\mathcal{N}_G$:

1. the processors know the topology but are unaware of any labeling (this corresponds to computations in the unlabeled Caley network $\mathcal{N}_G$),

2. in addition to the network topology the processors also know the labeling $\mathcal{L}_G$ (this corresponds to computations in the labeled Caley network $\mathcal{N}_G[\mathcal{L}_G]$),

As in [17, 16] we define $\mathcal{F}(\mathcal{N}_G)$ and $\mathcal{F}(\mathcal{N}_G[\mathcal{L}_G])$ to be the classes of N-ary Boolean functions computable in the two cases above, respectively. Clearly $\mathcal{F}(\mathcal{N}_G) \subseteq \mathcal{F}(\mathcal{N}_G[\mathcal{L}_G])$ (this is easy, because if an algorithm computes a boolean function on the unlabeled the same algorithm will compute the boolean function on the labeled network). More formally, we are interested in studying the computational strength of the labeled and unlabeled Cayley network by comparing the classes $\mathcal{F}(\mathcal{N}_G)$ and $\mathcal{F}(\mathcal{N}_G[\mathcal{L}_G])$. For general groups we will give several sufficient conditions such that $\mathcal{F}(\mathcal{N}_G) \neq \mathcal{F}(\mathcal{N}_G[\mathcal{L}_G])$ holds. We will also give a necessary and sufficient condition on abelian groups with canonical sets of generators such that $\mathcal{F}(\mathcal{N}_G) \neq \mathcal{F}(\mathcal{N}_G[\mathcal{L}_G])$.

In the sequel we use the following notation. We denote by $< G >$ the group generated by the set G of generators, e denotes the identity element of the group, and for $u \in \mathcal{G}$, $|u|$ denotes the order of u in the group $\mathcal{G}$, i.e. the smallest positive integer k such that $u^k = e$. In addition, $\mathcal{G}_1 \otimes \mathcal{G}_2$ denotes the direct product of the groups $\mathcal{G}_1, \mathcal{G}_2$; its elements consist of the pairs (g_1, g_2) such that $g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_2$ with multiplication $(g_1, g_2)(g_1', g_2') = (g_1 g_1', g_2 g_2')$.

2 Arbitrary Groups

We are interested on whether or not the introduction of the labeling $\mathcal{L}_G$ alters the class of functions which are computable in the network. More specifically, we call the labeling $\mathcal{L}_G$ strong if there is a Boolean function on $N = |\mathcal{G}|$ variables which is computable in the network $\mathcal{N}_G[\mathcal{L}_G]$ but not computable in $\mathcal{N}_G$.

If we ignore labels then it is clear that $Aut(\mathcal{N}_G)$ consists of all permutations ϕ of $\mathcal{G}$ such that for all $u, v \in \mathcal{G}$, $u^{-1}v \in G \Leftrightarrow \phi(u)^{-1}\phi(v) \in G$. Two more groups of automorphisms that will be useful in our subsequent study are defined as follows.

$$Aut^*(\mathcal{N}_G) = \{\phi \in Aut(\mathcal{N}_G) : \forall u \in \mathcal{G} \forall g \in G(\phi(u)^{-1}\phi(ug) \in < g >)\}$$

and

$$Aut^{**}(\mathcal{N}_G) = \{\phi \in Aut(\mathcal{N}_G) : \forall u, a \in \mathcal{G}(\phi(u)^{-1}\phi(ua) \in < a >)\}.$$

Now we have the following inequalities.

$$Aut(\mathcal{N}_G[\mathcal{L}_G]) \leq Aut^{**}(\mathcal{N}_G) \leq Aut^*(\mathcal{N}_G) \leq Aut(\mathcal{N}_G). \tag{2}$$

To prove that $\mathcal{L}_G$ is strong it is enough to define a Boolean function f such that

$$Aut(\mathcal{N}_G[\mathcal{L}_G]) \leq S(f) \text{ and } Aut(\mathcal{N}_G) \not\leq S(f). \tag{3}$$

Indeed, in view of Theorem 3, f must be computable on $\mathcal{N}_G[\mathcal{L}_G]$, but it can not be computable on $\mathcal{N}_G$ since f is not invariant under all the automorphisms of $\mathcal{N}_G$.

2.1 Distinguishing by network automorphism

Now we can prove the following theorem which establishes a sufficient condition for the network $\mathcal{N}_G[\mathcal{L}_G]$ to have more computational power than the network $\mathcal{N}_G$.

THEOREM 4 *If $Aut^{**}(\mathcal{N}_G) \neq Aut(\mathcal{N}_G)$ then $\mathcal{L}_G$ is strong.*

PROOF Let $\phi \in Aut(\mathcal{N}_G) \setminus Aut^{**}(\mathcal{N}_G)$. Since $\phi \notin Aut^{**}(\mathcal{N}_G)$ there exists $u, a \in \mathcal{G}$ such that $\phi(ua) \neq \phi(u)a^k$, for all $1 \leq k < |a|$. Define a Boolean function on inputs $< b_x : x \in \mathcal{G} >$ as follows.

$$f(< b_x : x \in \mathcal{G} >) = \begin{cases} 0 & \text{if } \forall x \in \mathcal{G}(b_x = b_{xa}) \\ 1 & \text{otherwise.} \end{cases}$$

It is easy to see that f is kept invariant by all automorphisms of $\mathcal{N}_G[\mathcal{L}_G]$, but this is not true for the above automorphism ϕ. To see this consider an input $< b_x : x \in \mathcal{G} >$ such that $\forall x \in \mathcal{G}(b_x = b_{xa})$ and $b_{\phi(u)} \neq b_{\phi(ua)}$. It follows that

$$0 = f(< b_x : x \in \mathcal{G} >) \neq f(< b_{\phi(x)} : x \in \mathcal{G} >) = 1.$$

This completes the proof of the theorem. ■

In view of Theorem 4 and inequality (2) to prove that $\mathcal{L}_G$ is strong it is enough to prove that $Aut^*(\mathcal{N}_G) \neq Aut(\mathcal{N}_G)$.

Theorem 5 *Assume G_i is a set of generators for the group $\mathcal{G}_i$, with $i = 1, 2$, $G = G_1 \cup G_2$, $G_1 \cap G_2 = \emptyset$ and $\mathcal{G} = \mathcal{G}_1 \otimes \mathcal{G}_2$. Then*

$$\exists i(Aut^*(\mathcal{N}_{G_i}) < Aut(\mathcal{N}_{G_i})) \Rightarrow Aut^*(\mathcal{N}_G) < Aut(\mathcal{N}_G).$$

Proof Assume on the contrary that

$$Aut^*(\mathcal{N}_{G_1}) < Aut(\mathcal{N}_{G_1}) \text{ but } Aut^*(\mathcal{N}_G) = Aut(\mathcal{N}_G).$$

Let $\phi_1 \in Aut(\mathcal{N}_{G_1})$. Define ϕ as follows: $\phi(u_1u_2) = \phi_1(u_1)u_2$, where $u_1 \in \mathcal{G}_1$ and $u_2 \in \mathcal{G}_2$. Then for all $u \in \mathcal{G}$, with $u = u_1u_2$, $u_1 \in \mathcal{G}_1, u_2 \in \mathcal{G}_2$, and all $g \in G$ we have that

$$\phi(u)^{-1}\phi(ug) = \begin{cases} g & \text{if } g \in \mathcal{G}_2 \\ \phi_1(u_1)^{-1}\phi(u_1g) & \text{if } g \in \mathcal{G}_1 . \end{cases} \tag{4}$$

It follows that $\phi \in Aut(\mathcal{N}_G)$. Since $Aut(\mathcal{N}_G) = Aut^*(\mathcal{N}_G)$ equation (4) implies that $\phi_1(u_1)^{-1}\phi_1(u_1g) \in < g >$, for all $u_1 \in \mathcal{G}_1$ and $g \in G_1$. It follows that $\phi_1 \in Aut^*(\mathcal{N}_{G_1})$. Consequently, $Aut^*(\mathcal{N}_{G_1}) = Aut(\mathcal{N}_{G_1})$, which is a contradiction. This completes the proof of the theorem. ■

2.2 Distinguishing by group automorphism

Theorem 6 *If ϕ is an automorphism of the group $\mathcal{G}$ such that $\phi(G) = G$ and $\phi(g) \notin < g >$, for some $g \in G$, then $Aut^*(\mathcal{N}_G) < Aut(\mathcal{N}_G)$.*

Proof Let us define $Aut^*(\mathcal{G})$ as the automorphisms of $\mathcal{N}_G$ satisfying the following condition

$$(u, v) \in E \Rightarrow \phi(\mathcal{L}_G(u, v)) = \mathcal{L}_G(\phi(u), \phi(v)). \tag{5}$$

To prove the theorem we need the following precise characterization of $Aut^*(\mathcal{G})$.

Lemma 7 *The automorphisms of the Cayley network $\mathcal{N}_G$ satisfying condition (5) are exactly the automorphisms ϕ of the group $\mathcal{G}$ satisfying $\phi(G) = G$.*

PROOF of Lemma 7. Let ϕ be an automorphism of the group $\mathcal{G}$ satisfying $\phi(G) = G$. It follows from [3][section 16] that ϕ is an automorphism of the corresponding unlabeled Cayley network and condition (5) is easily verified. For the other direction assume ϕ is an automorphism of the network $\mathcal{N}_G$ satisfying condition (5). Let $u \in \mathcal{G}$, $g \in G$ and put $v = ug$. Clearly, $\mathcal{L}_G(u, v) = g$. Consequently, by (5) $\mathcal{L}_G(\phi(u), \phi(ug)) = \phi(\mathcal{L}_G(u, ug)) = \phi(g)$. This implies that $\phi(ug) = \phi(u)\phi(g)$. Similarly, we can prove $\phi(e) = e$. Since G generates the group $\mathcal{G}$, it is easy to show that ϕ is a group automorphism. This proves the lemma.

It follows from the lemma that if $\phi \in Aut^*(\mathcal{G})$ and $\phi(g) \notin < g >$, for some $g \in G$, then $\phi \notin Aut^*(\mathcal{N}_G)$. This completes the proof of the theorem. ■

Thus using Theorem 6 we can prove that $Aut^*(\mathcal{N}_G) \neq Aut(\mathcal{N}_G)$ for the star, bubble-sort and pancake-sort networks previously considered.

EXAMPLE 8 *$\mathcal{L}_G$ is a strong labeling for the star, bubble sort and pancake-sort networks.*

PROOF For each of the networks listed above we exhibit an automorphism $\phi \in Aut(\mathcal{G})$ such that $\phi(G) = G$ but $\phi(g) \notin < g >$, for some $g \in G$.

CASE 1 n-Star: Consider the automorphism $\phi(\tau) = \sigma^{-1}\tau\sigma$, where $\sigma = (2, 3)$. It is easy to check that that $\phi((1, 2)) = (1, 3) \notin < (1, 2) >$ and $\phi(G) = G$.

CASE 2 n-Bubble-Sort: Consider the automorphism $\phi(\tau) = \sigma^{-1}\tau\sigma$, where $\sigma = (1, 2, \ldots, n)$. It is easy to check that that $\phi((k - 1 \bmod n, k)) = (k, k + 1 \bmod n) \notin < (k - 1 \bmod n, k) >$ and $\phi(G) = G$.

CASE 3 n-Pancake:
Consider the automorphism $\phi(\tau) = \rho_n \tau \rho_n$. It is easy to check that that $\phi(\rho_k) = \bar{\rho}_k \notin < \rho_k >$ and $\phi(G) = G$. ■

For any automorphism $\phi \in Aut(\mathcal{G})$ let

$$G_\phi = G \cup \phi(G) \cup \phi^2(G) \cup \cdots .$$

Then we can prove the following theorem.

THEOREM 9 *For any automorphism $\phi \in Aut(\mathcal{G})$, if $\phi(g) \notin < g >$, for some $g \in G$, then $\mathcal{L}_{G_\phi}$ is strong.*

PROOF Let ϕ be an automorphism in $Aut(\mathcal{G})$. The set G_ϕ generates $\mathcal{G}$ since G does. Moreover, it is trivial to check that $\phi(G_\phi) = G_\phi$. Hence the result of the theorem is immediate from Theorem 6. ■

3 Abelian Groups

Now we consider the case of abelian groups. In the most general case we have arbitrary sets of generators for such groups. A Boolean function $f \in B_N$ represents a group $\mathcal{H} \leq S_N$ if $S(f) = \mathcal{H}$, where $S(f)$ is the set of permutations on N

letters that leave f invariant on all inputs. Such groups are called representable [5].

THEOREM 10 *If the group $Aut(\mathcal{N}_G[\mathcal{L}_G])$ is representable and $Aut(\mathcal{N}_G[\mathcal{L}_G]) < Aut(\mathcal{N}_G)$ then $\mathcal{L}_G$ is strong.* ■

The proof is immediate in view of the represantibility of the group $Aut(\mathcal{N}_G[\mathcal{L}_G])$ and assertion (3). By a result of [4] and [12] (see also the correction in [6] and [7]) we know that $Aut(\mathcal{N}_G)$ is never abelian unless $\mathcal{G} = Z_2^n$ and $n \neq 2,3,4$. Hence in these cases we must have that $Aut(\mathcal{N}_G[\mathcal{L}_G]) < Aut(\mathcal{N}_G)$. Consequently, if $Aut(\mathcal{N}_G[\mathcal{L}_G])$ is also representable then $\mathcal{L}_G$ is strong. For a study of representable groups where Theorem 10 applies the reader should consult [5].

In the sequel we consider abelian groups generated by a canonical set of generators. We call a set G of generators for an abelian group $\mathcal{G}$ *canonical* if it is obtained by one of the following rules:

- $\mathcal{G} =< g >$ is a cyclic group and $G = \{g\}$.
- $\mathcal{G} = \mathcal{G}_1 \otimes \mathcal{G}_2$, $G = G_1 \cup G_2$, and G_1, G_2 are canonical sets of generators for the groups $\mathcal{G}_1$, $\mathcal{G}_2$, respectively.

Clearly all groups obtained via the above rules are abelian and every abelian group is obtained via the above rules. For canonical sets of generators of abelian groups we can give a complete characterization of those Cayley networks for which $\mathcal{L}_G$ is a strong labeling. In fact we can prove the following theorem.

THEOREM 11 *Let $\mathcal{G}$ be a nontrivial abelian group which is not any of the 4 cyclic groups C_2, C_3, C_4, C_5. If G is a canonical set of generators for $\mathcal{G}$ then the labeling $\mathcal{L}_G$ is strong.*

PROOF The proof is in several lemmas. First we take care of cyclic groups.

LEMMA 12 *If G is a canonical set of generators for the cyclic group $\mathcal{G} = C_N$ then $\mathcal{L}_G$ is strong exactly when $N \neq 2,3,4,5$.*

PROOF of the lemma. The automorphism group of $\mathcal{N}_G$ is the dihedral group D_N. Results in [5] show that for any $N = 2,3,4,5$ and any Boolean function f on N variables if $S(f) \geq C_N$ then also $S(f) \geq D_N$. Hence in this case results in [2] show that if f can be computed in $\mathcal{N}_G$ then f can also be computed in $\mathcal{N}_G[\mathcal{L}_G]$. ■

If $\mathcal{G}$ is abelian then a precise characterization of the group $Aut^*(\mathcal{N}_G)$ is possible.

LEMMA 13 *If $G = \{g_1, \ldots, g_k, g_1^{-1}, \ldots, g_k^{-1}\}$ is a canonical set of generators for the abelian group $\mathcal{G}$ then $Aut^*(\mathcal{N}_G) = \bigotimes_{1 \leq i \leq k} D_{|g_i|}$.*

Proof of the lemma. First observe that a canonical set of generators is irreducible, where a set G of generators of a group $\mathcal{G}$ is called irreducible if

$$g, g' \in G \text{ and } g' \neq g, g^{-1} \Rightarrow g' \notin < g > .$$

Let $s = |\{1 \leq i \leq k : g_i^2 \neq e\}|$ and assume that

$$g_1 \neq g_1^{-1}, \ldots, g_s \neq g_s^{-1}, g_{s+1} = g_{s+1}^{-1}, \ldots, g_k = g_k^{-1}.$$

Let $\phi \in Aut^*(\mathcal{N}_G)$. We show that ϕ is uniquely determined from

$$\phi(e), \phi(g_1), \ldots, \phi(g_k).$$

Since G is irreducible there exist $a_1, \ldots, a_s \in \{-1, 1\}$ such that $\phi(g_i) = g_i^{a_i}$, for $i = 1, \ldots, s$. Then we can prove that for all $u \in \mathcal{G}$ if

$$u = g_1^{r_1} \cdots g_s^{r_s} g_{s+1}^{r_{s+1}} \cdots g_k^{r_k}$$

then

$$\phi(u) = \phi(e) g_1^{a_1 r_1} \cdots g_s^{a_s r_s} g_{s+1}^{r_{s+1}} \cdots g_k^{r_k}$$

(the proof is by induction on the exponents). Now the isomorphism claimed in the statement of the theorem is

$$\phi \rightarrow (\phi(e), a_1, \ldots, a_s).$$

This completes the proof of the lemma. ■

Lemma 14 *If G is a canonical set of generators for the group $\mathcal{G}$ such that there exist $g, g' \in G$ with $|g| = |g'|$ and $g' \notin < g >$ then $Aut^*(\mathcal{N}_G) \neq Aut(\mathcal{N}_G)$.*

Proof Easy since we can prove that there is an automorphism ϕ of the group $\mathcal{G}$ permuting g, g' but leaving the other generators fixed. ■

Now we give the proof of the main theorem. Assume that $\mathcal{G} = C_{n_1} \otimes \cdots \otimes C_{n_k}$, with $n_1 \geq \cdots \geq n_k$. If $n_i = n_j$ for some $i \neq j$ then by Lemma 14 and Theorem 4 the labeling $\mathcal{L}_G$ is strong. Hence without loss of generality we may assume that $n_1 > \cdots > n_k$. If $k = 1$ then the theorem follows from Lemma 12. Hence without loss of generality we may assume $k \geq 2$. Assume now that for some i, $n_i \notin \{2, 3, 4, 5\}$. By [5] all dihedral groups are representable and the groups C_n are representable exactly when $n \neq 3, 4, 5$. Hence there is a Boolean function f such that

$$S(f) = D_{n_1} \otimes \cdots \otimes D_{n_{i-1}} \otimes C_{n_i} \otimes D_{n_{i+1}} \otimes \cdots \otimes D_{n_k}.$$

Since $n_i \neq 2$, we have that $C_{n_i} < D_{n_i}$, and hence

$$S(f) < D_{n_1} \otimes \cdots \otimes D_{n_{i-1}} \otimes D_{n_i} \otimes D_{n_{i+1}} \otimes \cdots \otimes D_{n_k} = Aut^*(\mathcal{N}_G) \leq Aut(\mathcal{N}_G).$$

It follows that f is not computable in the network $\mathcal{N}_G$.

The theorem has been proved for all abelian groups except for the following eleven: $\bigoplus_{n\in S} C_n$, where $S \subseteq \{2,3,4,5\}$ and $|S| \geq 2$, which we now consider. For these groups we use the automorhism groups $Aut^{**}(\mathcal{N}_G)$ and prove the following claim.

Claim. $Aut^{**}(\mathcal{N}_G) = Aut(\mathcal{N}_G[\mathcal{L}_G])$, for any of the eleven abelian groups considered.

Proof of the claim. Let $\phi \in Aut^{**}(\mathcal{N}_G)$. Since the groups considered are abelian it is easy to see that there exists an integer $k \geq 0$ such that for all $u, a \in \mathcal{G}$,

$$\phi(ua) = \phi(u)a^k.$$

It is now easy to check that for the eleven groups considered this implies that we can choose $k = 1$. This proves the claim.

Since a theorem of Subidussi [12] implies that $Aut(\mathcal{N}_G)$ is not abelian (the same result also follows directly from the next Theorem 15 without refering to [12]) it follows that $Aut(\mathcal{N}_G) \neq Aut^{**}(\mathcal{N}_G)$. This completes the proof of the theorem. ∎

At this point it is interesting to note two interesting facts without proof. If $Aut_e(\mathcal{N}_G)$ is the set of automorphisms of $\mathcal{N}_G$ fixing the identity element e of the group $\mathcal{G}$ then every $\phi \in Aut(\mathcal{N}_G)$ is of the form $a \cdot \psi$, for some $a \in \mathcal{G}, \psi \in Aut_e(\mathcal{N}_G)$ (the same result holds for any of the groups $Aut^{*}(\mathcal{N}_G), Aut^{**}(\mathcal{N}_G)$). It is also a consequence of the definition of $Aut^{**}(\mathcal{N}_G)$ and the proof of Lemma 13 that for a canonical set of generators of an abelian group $\mathcal{G}$, $Aut_e^{**}(\mathcal{N}_G) = Aut^{*}(\mathcal{G})$. We leave the details to the reader.

Theorem 15 *If G is a canonical set of generators for the group $\bigoplus_{n\in S} C_n$, where $S \subseteq \{2,3,4,5\}$ then*

$$Aut(\mathcal{N}_G) = Aut^{*}(\mathcal{N}_G) = \bigoplus_{n\in S} D_n.$$

Moreover,

$$\{f \in B_N : S(f) \geq Aut(\mathcal{N}_G)\} = \{f \in B_N : S(f) \geq Aut(\mathcal{N}_G[\mathcal{L}_G])\}.$$

Proof In order to prove the theorem we need the following lemma.

Lemma 16 *Assume that $\mathcal{G} = \mathcal{G}' \otimes C_n$ is an abelian group, G' a canonical set of generators for $\mathcal{G}'$, $G = G' \cup \{v\}$ and $|v| = 3$ or $|v| = 5$. Moreover assume that*

1. *if $|v| = 3$ then for all $g \in G'$ $|g| \neq 3$,*
2. *if $|v| = 5$ then for all $g \in G'$ $|g| \neq 3, 5$.*

Then $Aut(\mathcal{N}_{G'\cup\{v\}}) = Aut(\mathcal{N}_{G'}) \otimes D_{|v|}$.

PROOF of the lemma. Let $\phi \in Aut(\mathcal{N}_{G' \cup \{v\}})$ and suppose on the contrary that for some $a \in \mathcal{G}, u \in G'$ $\phi(av) = \phi(a)u$ (the case $\phi(av) = \phi(a)u^{-1}$ is similar). We will derive a contradiction.

First consider the case $|v| = 3$. We have that

$$\phi(a) = \phi(av^3) = \phi(av^2)u_1 = \phi(av)u_2u_1 = \phi(a)u_2u_1u,$$

for some $u_1, u_2 \in G' \cup \{v\}$. But this implies that

$$u_1u_2u = e. \tag{6}$$

If $u_1, u_2 \notin \{u, u^{-1}\}$ then equation (6) implies that $u = e$, which is a contradiction. If $u_1 \in \{u, u^{-1}\}$ while $u_2 \notin \{u, u^{-1}\}$ then either $u_1 = u$ which implies that $u^2u_2 = e$, or else $u_1 = u^{-1}$ which implies that $u_2 = e$; in both cases we get a contradiction. Finally if $u_1, u_2 \in \{u, u^{-1}\}$ then either $u_1 = u_2 = u$, in which case (6) implies $u^3 = e$ (contradicting the fact that v is the unique element of order 3) or $u_1 = u_2 = u^{-1}$, in which case (6) implies $u^{-1} = e$, or $u_1 = u, u_2 = u^{-1}$, in which case (6) implies $u = e$. This proves the lemma in the case $|v| = 3$.

Next consider the case $|v| = 5$. As before there exist $u_1, u_2, u_3, u_4 \in G' \cup \{v\}$ such that

$$u_1u_2u_3u_4u = e. \tag{7}$$

We consider five cases depending on whether or not $0, 1, 2, 3, 4$ generators among the u_1, u_2, u_3, u_4 are in the set $\{u, u^{-1}\}$. As before we use the fact that there is no generator in G' of order 3 or 5 to derive a contradiction. This proves the lemma. ■

In view of Lemma 16 it is enough to consider only groups of the form $C_m \otimes C_2$, with $m > 2$, as well as of the form $C_m \otimes C_4$, with $m > 4$. We show that in these cases as well $Aut(\mathcal{N}_G) = Aut^*(\mathcal{N}_G)$. This would imply that for all 16 abelian groups $\bigotimes_{m \in S} C_m$, where $S \subseteq \{2, 3, 4, 5\}$ and for any canonical set of generators G of that group $Aut(\mathcal{N}_G) = Aut^*(\mathcal{N}_G)$.

First consider the case of the groups $C_m \otimes C_2$, with $m > 2$. Let u be a generator of C_m and v a generator of C_2. Let $\phi \in Aut(\mathcal{N}_{\{u,v\}})$ and suppose on the contrary that $\phi(uv) = \phi(u)u$ (the case $\phi(uv) = \phi(u)u^{-1}$ is similar). We will derive a contradiction. It follows that $\phi(u^2v) = \phi(u)u_1u$, for some $u_1 \in \{u, u^{-1}, v\}$. If $u_1 = u$ then $\phi(u^kv) = \phi(u)u^k$, for all k. If $u_1 = u^{-1}$ then $\phi(u^kv) = \phi(u)u^{-k}$, for all k. If $u_1 = v$ then $\phi(u^kv) = \phi(u)u^{k-1}v$, for all k. But all these statements contradict the injectivity of ϕ.

It remains to examine the case of the abelian groups $\mathcal{G} = C_m \otimes C_4$ when $m > 4$. Let u, v be generators of C_m, C_4, respectively. By contradiction, assume that for some $a \in \mathcal{G}$, and $\phi \in Aut(\mathcal{N}_G)$, $\phi(av) = \phi(a)u$. But arguing as before this would imply that ϕ is not $1-1$. The proof of Theorem 15 is complete. ■

The 11 abelian groups $\bigoplus_{n \in S} C_n$, where $S \subseteq \{2, 3, 4, 5\}$ and $|S| \geq 2$ have a rather interesting behavior. Although Theorem 15 implies that the networks

$\mathcal{N}_G$ and $\mathcal{N}_G[\mathcal{L}_G]$ cannot "distinguish" the Boolean functions they can compute from their automorphism groups alone, Theorem 11 shows that in fact the labeled network $\mathcal{N}_G[\mathcal{L}_G]$ can compute more Boolean functions than the unlabeled network $\mathcal{N}_G$. In particular, for these 11 abelian groups there exist Boolean functions which are computable on $\mathcal{N}_G$ but such that $S(f) \geq Aut(\mathcal{N}_G)$.

4 Conclusion

We studied the labeling problem on anonymous Cayley networks and provided sufficient conditions for the labeling $\mathcal{L}_G$ to be strong. For the case of abelian groups with canonical sets of generators we gave exact characterizations of the groups $\mathcal{G}$ for which the labeling $\mathcal{L}_G$ is strong.

References

[1] S. B. Akers and B. Krishnamurthy. A group theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38(4):555 – 566, 1989.

[2] H. Attiya, M. Snir, and M. Warmuth. Computing on an anonymous ring. *Journal of the ACM*, 35(4):845 – 875, 1988. (Preliminary version has appeared in proceedings of the 4th Annual ACM Symposium on Principles of Distributed Computation, 1985).

[3] N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1974.

[4] C-Y. Chao. On a theorem of Subidussi. *Proceedings American Mathematical Society*, 15:291 – 292, 1964.

[5] P. Clote and E. Kranakis. Boolean functions invariance groups and parallel complexity. *SIAM Journal on Computing*, 20(3):553 – 590, 1991. (Preliminary version has appeared in Proceedings of 4th Annual IEEE Conference on Structure in Complexity Theory, Eugene, pp. 55 - 66).

[6] W. Imrich. Graphen mit transitiver Automorphismgruppe. *Monatshefte für Mathematik*, 73:341 – 347, 1969.

[7] W. Imrich. Graphs with transitive abelian automorphisms groups. In P. Erdös, A. Renyi, and V. Sós, editors, *Combinatorial Theory and its Applications*, volume II. North Holland, 1970.

[8] A. Israeli and M. Jalfon. Uniform self-stabilizing ring orientation. *Information and Computation*, 1992. to appear.

[9] E. Korach, S. Moran, and S. Zaks. Tight upper and lower bounds for some distributed algorithms for a complete network of processors. In *Proceedings of 3th Annual ACM Symposium on Principles of Distributed Computation*, pages 199 – 207, 1984.

[10] E. Kranakis and D. Krizanc. Distributed computing on Cayley networks. In *Proceedings of the 4th IEEE Symposium on Parallel and Distributed Processing, Arlington, Texas, Dec. 1-4*, 1992.

[11] N. Santoro. Sense of direction, topological awareness and communication complexity. *ACM SIGACT News*, (16):50 – 56, 1984.

[12] G. Subidussi. Vertex transitive graphs. *Monatshefte für Mathematik*, 68:426 – 438, 1964.

[13] V. Syrotiuk and J. Pachl. A distributed ring orientation problem. In Jan van Leeuwen, editor, *Proceedings of 2nd International Workshop on Distributed Algorithms, Amsterdam, July 1987*, pages 332 – 336, Heidelberg, 1988. Springer Verlag Lecture Notes in Computer Science. Vol. 312.

[14] G. Tel. Network orientation. Technical Report RUU-CS-91-8, University of Utrecht , Department of Computer Science, 1991.

[15] H. Wielandt. *Finite Permutation Groups*. Academic Press, 1964.

[16] M. Yamashita and T. Kameda. Computing functions on an anonymous network. Technical Report 87-16, Laboratory for Computer and Communication Research, Simon Fraser University, 1987. 27 pages.

[17] M. Yamashita and T. Kameda. Computing on an anonymous network. In *7th Annual ACM Symposium on Principles of Distributed Computation*, pages 117 – 130, 1988.

School of Computer Science, Carleton University
Recent Technical Reports

TR-173 **Self-Organizing Doubly-Linked Lists**
R.S. Valiveti and B.J. Oommen, May 1990

TR-174 **A Presortedness Metric for Ensembles of Data Sequences**
R.S. Valiveti and B.J. Oommen, May 1990

TR-175 **Separation of Graphs of Bounded Genus**
Ljudmil G. Aleksandrov and Hristo N. Djidjev, May 1990

TR-176 **Edge Separators of Planar and Outerplanar Graphs with Applications**
Krzystof Diks, Hristo N. Djidjev, Ondrej Sykora and Imrich Vrto, May 1990

TR-177 **Representing Partial Orders by Polygons and Circles in the Plane**
Jeffrey B. Sidney and Stuart J. Sidney, July 1990

TR-178 **Determining Stochastic Dependence for Normally Distributed Vectors Using the Chi-squared Metric**
R.S. Valiveti and B.J. Oommen, July 1990

TR-179 **Parallel Algorithms for Determining K-width-Connectivity in Binary Images**
Frank Dehne and Susanne E. Hambrusch, September 1990

TR-180 **A Workbench for Computational Geometry (WOCG)**
P. Epstein, A. Knight, J. May, T. Nguyen, and J.-R. Sack, September 1990

TR-181 **Adaptive Linear List Reorganization under a Generalized Query System**
R.S. Valiveti, B.J. Oommen and J.R. Zgierski, October 1990

TR-182 **Breaking Substitution Cyphers using Stochastic Automata**
B.J. Oommen and J.R. Zgierski, October 1990

TR-183 **A New Algorithm for Testing the Regularity of a Permutation Group**
V. Acciaro and M.D. Atkinson, November 1990

TR-184 **Generating Binary Trees at Random**
M.D. Atkinson and J.-R. Sack, December 1990

TR-185 **Uniform Generation of Combinatorial Objects in Parallel**
M.D. Atkinson and J.-R. Sack, January 1991

TR-186 **Reduced Constants for Simple Cycle Graph Separation**
Hristo N. Djidjev and Shankar M. Venkatesan, February 1991

TR-187 **Multisearch Techniques for Implementing Data Structures on a Mesh-Connected Computer**
Mikhail J. Atallah, Frank Dehne, Russ Miller, Andrew Rau-Chaplin, and Jyh-Jong Tsay, February 1991

TR-188 **Generating and Sorting Jordan Sequences**
Alan Knight and Jörg-Rüdiger Sack, March 1991

TR-189 **Probabilistic Estimation of Damage from Fire Spread**
Charles C. Colbourn, Louis D. Nel, T.B. Boffey and D.F. Yates, April 1991

TR-190 **Coordinators: A Mechanism for Monitoring and Controlling Interactions Between Groups of Objects**
Wilf R. LaLonde, Paul White, and Kevin McGuire, April 1991

TR-191 **Towards Decomposable, Reusable Smalltalk Windows**
Kevin McGuire, Paul White, and Wilf R. LaLonde, April 1991

TR-192 **PARASOL: A Simulator for Distributed and/or Parallel Systems**
John E. Neilson, May 1991

TR-193 **Realizing a Spatial Topological Data Model in a Relational Database Management System**
Ekow J. Otoo and M.M. Allam, August 1991

TR-194 **String Editing with Substitution, Insertion, Deletion, Squashing and Expansion Operations**
B John Oommen, September 1991

TR-195 **The Expressiveness of Silence: Optimal Algorithms for Synchronous Communication of Information**
Una-May O'Reilly and Nicola Santoro, October 1991

TR-196 **Lights, Walls and Bricks**
J. Czyzowicz, E. Rivera-Campo, N. Santoro, J. Urrutia and J. Zaks, October 1991

TR-197 **A Brief Survey of Art Gallery Problems in Integer Lattice Systems**
Evangelos Kranakis and Michel Pocchiola, November 1991

TR-198 **On Reconfigurability of Systolic Arrays**
Amiya Nayak, Nicola Santoro, and Richard Tan, November 1991

TR-199 **Constrained Tree Editing**
B. John Oommen and William Lee, December 1991

TR-200 **Industry and Academic Links in Local Economic Development: A Tale of Two Cities**
Helen Lawton Smith and Michael Atkinson, January 1992

TR-201 **Computational Geometry on Analog Neural Circuits**
Frank Dehne, Boris Flach, Jörg-Rüdiger Sack, Natana Valiveti, January 1992

TR-202 **Efficient Construction of Catastrophic Patterns for VLSI Reconfigurable Arrays**
Amiya Nayak, Linda Pagli, Nicola Santoro, February 1992

TR-203 **Numeric Similarity and Dissimilarity Measures Between Two Trees**
B. John Oommen and William Lee, February 1992

TR-204 **Recognition of Catastrophic Faults in Reconfigurable Arrays with Arbitrary Link Redundancy**
Amiya Nayak, Linda Pagli, Nicola Santoro, March 1992

TR-205 **The Permutational Power of a Priority Queue**
M.D. Atkinson and Murali Thiyagarajah, April 1992

TR-206 **Enumeration Problems Relating to Dirichlet's Theorem**
Evangelos Kranakis and Michel Pocchiola, April 1992

TR-207 **Distributed Computing on Anonymous Hypercubes with Faulty Components**
Evangelos Kranakis and Nicola Santoro, April 1992

TR-208 **Fast Learning Automaton-Based Image Examination and Retrieval**
B. John Oommen and Chris Fothergill, June 1992

TR-209 **On Generating Random Intervals and Hyperrectangles**
Luc Devroye, Peter Epstein and Jörg-Rüdiger Sack, July 1992

TR-210 **Sorting Permutations with Networks of Stacks**
M.D. Atkinson, August 1992

TR-211 **Generating Triangulations at Random**
Peter Epstein and Jörg-Rüdiger Sack, August 1992

TR-212 **Algorithms for Asymptotically Optimal Contained Rectangles and Triangles**
Evangelos Kranakis and Emran Rafique, September 1992

TR-213 **Parallel Algorithms for Rectilinear Link Distance Problems**
Andrzej Lingas, Anil Maheshwari and Jörg-Rüdiger Sack, September 1992

TR-214 **Camera Placement in Integer Lattices**
Evangelos Kranakis and Michel Pocchiola, October 1992