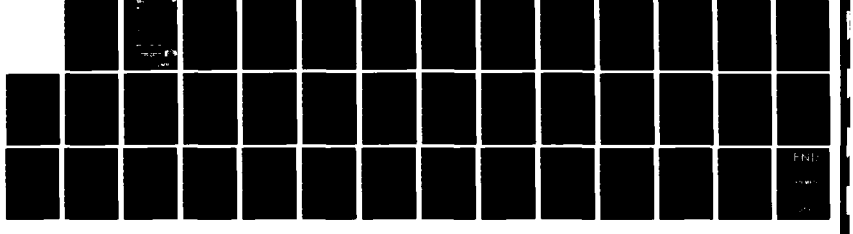
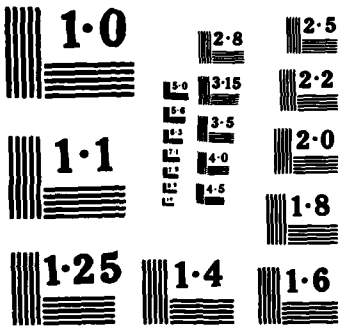
AD-A158 918 ARC COLORINGS PARTIAL PATH GROUPS AND PARALLEL GRAPH CONTRACTIONS..(U) MARYLAND UNIV COLLEGE PARK CENTER FOR AUTOMATION RESEARCH A ROSENFELD JUL 85 CAR-TR-132 1/1

UNCLASSIFIED AFOSR-TR-85-0718 F49620-85-K-0009 F/G 12/1 NL





CAR-TR-132
CS-TR-1524

F49620-85-K-0009
July 1985

ARC COLORINGS, PARTIAL PATH GROUPS, AND PARALLEL GRAPH CONTRACTIONS

Azriel Rosenfeld

Center for Automation Research
University of Maryland
College Park, MD 20742

COMPUTER VISION LABORATORY

CENTER FOR AUTOMATION RESEARCH

UNIVERSITY OF MARYLAND

2

CAR-TR-132
CS-TR-1524

F49620-85-K-0009
July 1985

ARC COLORINGS, PARTIAL PATH GROUPS, AND PARALLEL GRAPH CONTRACTIONS

Azriel Rosenfeld
Center for Automation Research
University of Maryland
College Park, MD 20742

ABSTRACT

We define an algebraic structure on the paths in a graph based on a coloring of the arcs. Using this structure, basic classes of graphs (trees, hypercubes, arrays, cliques, etc.) are characterized by simple algebraic properties. The structure provides a framework for defining parallel contraction operations on a graph, in which many pairs of nodes are simultaneously collapsed into single nodes, but the degree of the graph does not increase. Such operations are useful in defining systematic strategies for simulating large networks of processors by smaller ones, or in building "pyramids" of networks.

The support of the US Air Force Office of Scientific Research under Contract F49620-85-K-0009 is gratefully acknowledged, as is the help of Sandra German in preparing this paper.

1. Introduction

Large, uniformly structured networks of processors are a useful class of models for parallel computation. For example, it has been recognized since the 1950's that mesh-connected processor networks are a very natural medium for image processing, and a number of such networks have actually been built. Ideally, the mesh should be the same size as the image to be processed, so that each pixel (= element of the image) can be assigned to a different processor. Meshes of that size are quite expensive to build; but they can be simulated by smaller meshes, by dividing the image into square blocks of pixels [1] and assigning each block to a processor. Another problem with large meshes is communication time; to compute global properties of an image, information must be transmitted from one side of the mesh to the other, which requires a number of steps on the order of the mesh diameter. This communication problem can be greatly alleviated by building an exponentially tapering "pyramid" on top of the mesh [2]. To do this, we divide the mesh into square blocks; connect the processors in each block to a new processor; and connect all the new processors to form a new, smaller mesh. This process is repeated, resulting in an exponentially tapering stack of meshes.

The process used to simulate large meshes by smaller ones, or to build pyramids of meshes, can be regarded as a "parallel contraction" of the mesh. We divide the mesh into square blocks; represent each block by a new node; and join two of the new nodes by an arc if the corresponding blocks were adjacent in the original mesh. This process is basically one of graph contraction [3], except that

we are contracting many subgraphs into single nodes simultaneously. Because of the regular structure of a mesh, we are able to define a parallel contraction process that does not increase the degree; the contracted graph is still a mesh. The fact that the degree does not increase is important in the simulation and pyramid building applications, since networks of high degree are difficult to realize in hardware.

It would be of interest to define degree-preserving parallel contraction processes for other types of regularly structured graphs. This would provide us with a systematic method of simulating large networks (having graph structures of one of the given types) by smaller ones [4], or of building pyramids of such networks.

This paper defines a class of parallel contraction operations based on a coloring of the arcs in the given graph. The coloring allows us to define an algebraic structure on the paths in the graph, and it turns out that basic classes of regularly structured graphs, including trees, arrays, hypercubes, and cliques, have simple algebraic characterizations. The structure provides a framework for defining parallel contraction operations for these and other classes of graphs, generalizing the standard (blockwise) parallel contraction technique used for arrays.

2. Arc colorings and the partial path group

Let G be a connected graph of degree d, i.e., no more than d arcs emanate from any node. By an *arc coloring* of G we mean an assignment of colors to the arcs of G such that the arcs emanating from any given node all have different colors. It can be shown [5] that this can always be done using at most $d+1$ colors, and for many types of graphs (for example, trees or bipartite graphs) d colors are sufficient. A simple example of a graph requiring $d+1$ colors is a triangle (or in general, a cycle of odd order), where the degree is 2 but 3 colors are needed.

Let the colors be $c_1, \ldots, c_m$; then any string of colors $c_{i_1} \cdots c_{i_t}$ defines a possible *path* from a given starting node P. The interpretation of such a string of c's is as follows: Move from $P = P_0$ to the unique neighbor P_1 joined to P_0 by an arc of color c_{i_1}; move from P_1 to the unique neighbor P_2 joined to P_1 by an arc of color c_{i_2}; and so on. Note that for a given P, this path may not be realizable; there may be *no* arc of color c_{i_1} emanating from P, or no arc of color c_{i_2} emanating from P_1, and so on.

This process of interpreting the colors as moves allows us to define an algebraic structure on the nodes of G, with the colors as "generators". For convenience, let us define a fictitious "blank" color c_0, representing "no move". (This is equivalent to introducing a loop at each node of G and coloring it with c_0; applying c_0 to any P thus leaves us at P.) Let $\alpha = c_{i_1} \cdots c_{i_t}$ be a string of colors; then α defines a partial function on the nodes of G, where $P\alpha$ is the ter-

minal node of the path defined by α starting from P, provided this path is realizable. These partial functions evidently satisfy the following properties:

a) For all P, Pc_0 is defined and $Pc_0 = P$.

This follows from our definition of c_0. We can thus think of c_0 as an identity element.

b) For all P and all c_i, if Pc_i is defined, so is $Pc_i{}^2$, and $Pc_i{}^2 = P$.

Proof: If there is an arc colored c_i at P, then there is an arc colored c_i at P's neighbor Pc_i, and if we move along that arc, we come back to P. Thus each c_i is its own inverse.

c) For all P and all c_i, c_j, if Pc_i and $(Pc_i)c_j$ are defined, then $P(c_i\, c_j)$ is defined, and $P(c_i\, c_j) = (Pc_i)c_j$.

This follows from the definition of $P(c_i\, c_j)$. Thus our algebra is associative, provided both sides are defined.

Note that if $c_i \neq c_0$ and Pc_i is defined, we cannot have $Pc_i = P$; thus the identity element c_0 is unique. Similarly, if $c_i \neq c_j$ and $Pc_i\, c_j$ is defined, we cannot have $Pc_i\, c_j = P$. (The arc joining P and Pc_i has color c_i; the node joined to Pc_i by an arc of color c_j cannot be P.) Thus inverses are also unique.

Using properties $(b-c)$ and induction, we immediately have

Proposition 2.1. Let α^{-1} denote the reversal of the string α, i.e., $(c_{i_1} \cdots c_{i_t})^{-1} \equiv c_{i_t} \cdots c_{i_1}$. Then for all P and all α, if $P\alpha$ is defined, so is $(P\alpha)\alpha^{-1}$, and $(P\alpha)\alpha^{-1} = P$.//

Proposition 2.2. Let $\alpha\beta$ denote the concatenation of the strings α and β, i.e., $(c_{i_1} \cdots c_{i_k})(c_{j_1} \cdots c_{j_h}) = c_{i_1} \cdots c_{i_k} c_{j_1} \cdots c_{j_h}$. Then for all P and all α, β, if $P\,\alpha$ and $(P\,\alpha)\beta$ are defined, so is $P\,(\alpha\beta)$, and $P\,(\alpha\beta) = (P\,\alpha)\beta$.//

These observations show that the structure defined on the nodes of G by the strings of colors resembles a group structure in many ways. It has an identity; any string of colors has an inverse (wherever it is defined); and concatenation of strings is associative (provided both sides are defined). We shall call such a structure a "partial group" or "half-group". From now on we shall refer to this partial group as the *partial path group* of G defined by the given coloring, and denote it by $\Pi(G)$.

3. Free partial path groups and trees

Let $\alpha = c_{i_1} \cdots c_{i_k}$ be a string of colors. The string α' is called an *elementary reduction* of α if one of the following statements is true:

a) $k > 1$; for some $1 \leq j \leq k$ we have $c_{i_j} = c_0$; and $\alpha' = c_{i_1} \cdots c_{i_{j-1}} c_{i_{j+1}} \cdots c_{i_k}$.

b) $k > 2$; for some $1 \leq j < k$ we have $c_{i_j} = c_{i_{j+1}}$; and $\alpha' = c_{i_1} \cdots c_{i_{j-1}} c_{i_{j+2}} \cdots c_{i_k}$.

c) $k = 2$; $c_{i_1} = c_{i_2}$; and $\alpha' = c_0$.

The string α' is called a *reduction* of α if there exist strings $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_m = \alpha'$ such that α_i is an elementary reduction of α_{i-1}, $1 \leq i \leq m$.

Proposition 3.1. If α' is a reduction of α, and $P\alpha$ is defined, then $P\alpha'$ is defined and $P\alpha' = P\alpha$.//

Note that if $P\alpha'$ is defined, $P\alpha$ may not be defined unless the elementary reduction steps involved in obtaining α' from α were all of the form (a).

We call α *irreducible* if it has no (elementary) reductions. (In particular, if $k = 1$, α is irreducible.)

Proposition 3.2. $\alpha \neq c_0$ is irreducible if and only if any two consecutive c's are distinct, and none of them is c_0.//

We call α *fully reducible* if $\alpha = c_0$, or if c_0 is a reduction of α. Evidently, if α is neither irreducible nor fully reducible, it has a reduction that is irreducible; and α is both irreducible and fully reducible only if $\alpha = c_0$.

Corollary 3.3. If α is fully reducible, and $P\alpha$ is defined, then $P\alpha = P$.//

Let $P = P_0, P_1, \ldots, P_k = P$ be a cycle in G, i.e., a sequence of nodes such that $k > 2$; $P_i \neq P_{i-1}$ for all $1 \leq i \leq k$; and $P_{i+1} \neq P_{i-1}$ for all $1 \leq i < k$. Let the color of the arc joining P_{j-1} to P_j be c_{i_j}, $1 \leq j \leq k$. Then the string $\alpha = c_{i_1} \cdots c_{i_k}$ is evidently irreducible. Conversely, let $\alpha = c_{i_1} \cdots c_{i_k}$ be any irreducible string; let $P\alpha$ be defined, and suppose that $P\alpha = P$; and let $P = P_0$, $P_1 = P_0 c_{i_1}$, $P_2 = P_1 c_{i_2}$, $\cdots P_k = P_{k-1} c_{i_k}$. Then $P_0, P_1, \ldots, P_k$ is evidently a cycle in G.

We call $\Pi(G)$ *free* if $P\alpha = P$ implies that α is fully reducible.

Theorem 3.4. If $\Pi(G)$ is free, G is a tree.

Proof: If G had a cycle, say $P = P_0, P_1, \ldots, P_k = P$, then by the remarks in the preceding paragraph there would exist an irreducible string α such that $P\alpha = P$.//

Theorem 3.5. For any arc coloring of a tree G, $\Pi(G)$ is free.

Proof: Suppose we had $P\alpha = P$ with α not fully reducible; then either α itself or some reduction α' of α would be irreducible, and we would have $P\alpha' = P$. Thus in either case we have $P\beta = P$ where $\beta (= \alpha$ or $\alpha')$ is irreducible, and by the remarks in the preceding paragraph, this implies that G has a cycle, contradiction.//

Note that the arcs of a tree of degree d (this is graph degree, not "tree degree") can always be colored with d colors. Indeed, start with any node P, and color the arcs emanating from it with (at most) the d colors. Let Q be a neighbor of P joined to it by an arc of color c_j; then the remaining neighbors of Q can be colored with (at most) the $d-1$ remaining colors $c_1, \cdots c_{j-1}, c_{j+1}, \ldots, c_n$. This process can be repeated for all the neighbors of P, all the neighbors of these neighbors, and so on. Since a tree is connected, every arc eventually receives a color. Since a tree has no cycles, when a node is reached by this coloring process it has not previously had colors assigned to any of its remaining arcs; thus the process never leads to a contradiction (where two arcs emanating from the same node are assigned the same color).

If G is an infinite tree in which every node has degree exactly d, and we color the arcs of G with d colors $c_1, \ldots, c_d$, then for any string of colors α, $P\alpha$ is always defined. Thus in this case we can regard $\Pi(G)$ as a group on d generators $c_1, \ldots, c_d$, with identity c_0, such that $c_i^2 = c_0$ for all i, but the group is otherwise free. Note that if some nodes have degrees $< d$, and in particular if the tree is finite, some $P\alpha$'s will not be defined.

4. Strings, cycles, and cliques

If G has degree 1, it can only consist of a single node or of two nodes joined by an arc (recall that we assume G to be connected). In the first case, the only irreducible string in $\Pi(G)$ is c_0, and in the second case, the only irreducible strings are c_0 and c_1. Thus in the first case we can regard $\Pi(G)$ as the trivial group, while in the second case we can regard it as the cyclic group of order 2 (note that Pc_0 and Pc_1 are both always defined).

Theorem 4.1. $\Pi(G)$ cannot be a cyclic group unless the group has order 1 or 2.

Proof: If there is only one color $c_1 \neq c_0$, G has degree 1.//

If G has degree 2, it must be a (finite or infinite) string or a cycle. If it is a string or a cycle of even order, its arcs can be colored with two colors c_1, c_2; in the string case the colors satisfy the relations $c_1^2 = c_2^2 = c_0$, while for a cycle of length $2m$ we have the additional relations $(c_1c_2)^m = (c_2c_1)^m = c_0$. Conversely, if $\Pi(G)$ is free, by Theorem 3.4 G is a tree, and since it has degree 2, it must be a string.

If G is a cycle of odd order, three colors are needed to color its arcs. In particular, if G is a triangle we have $c_1^2 = c_2^2 = c_3^2 = c_0$; $c_1c_2 = c_2c_1 = c_3$, $c_2c_3 = c_3c_2 = c_1$, $c_3c_1 = c_1c_3 = c_2$. (Actually, at any given node P, there are arcs of only two of the three colors; hence, e.g., c_1c_2 and c_2c_1 are never both defined.) Note that these are the relations that hold among the generators of the Klein 4-group.

We call $\Pi(G)$ *closed* if for all c_i, c_j, and all P such that $Pc_i c_j$ is defined, there exists a c_k such that $Pc_i c_j = Pc_k$.

Proposition 4.2. If G is a clique, then for any arc coloring of G, $\Pi(G)$ is closed.

Proof: Any two nodes P, Q of a clique are joined by an arc; hence $Pc_i c_j$ must be a neighbor of P, say Pc_k.//

Theorem 4.3. If $\Pi(G)$ is closed, G is a clique.

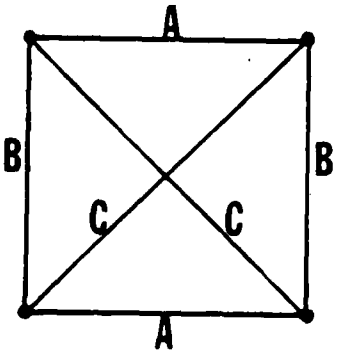
Proof: Let P, Q be any two nodes of G. Since G is connected, there exists a path $P = P_0, P_1, \ldots, P_k = Q$ from P to Q. This path defines a sequence of colors $c_{i_1} \cdots c_{i_k}$. Since $Pc_{i_1}c_{i_2}$ is defined, there exists a color c_{i_2}' such that $Pc_{i_1}c_{i_2} = Pc_{i_2}'$; since $Pc_{i_2}'c_{i_3}$ is defined, there exists a color c_{i_3}' such that $Pc_{i_2}'c_{i_3} = Pc_{i_3}'$; and so on. Hence by induction on the definition of closedness, there exists a color c_{i_k}' such that $Pc_{i_k}' = Q$. Thus Q is a neighbor of P, proving that G is a clique.//

A clique having $d+1$ nodes (degree d) can be colored with d colors if d is odd, but requires $d+1$ colors if d is even. For d even, a coloring with $d+1$ colors can be obtained as follows: Let $d = 2k$; let the nodes be labeled $1,2, \ldots, d+1$; and let the arc between nodes i and j be given color $c_{i+(k+1)(j-i)}$(modulo $2k+1$). Evidently this gives the same color to arc (i,j) as to arc (j,i); indeed, $i+(k+1)(j-i) \equiv j+(k+1)(i-j)$, since $(2k+1)i \equiv (2k+1)j \equiv 0$(modulo $2k+1$). Also, the colors at any node are all distinct: for a given i,

the factors $j-i$ take on all the values $1, 2, \ldots, d = 2k$ (modulo $2k+1$), and since $k+1$ is relatively prime to $2k+1$, the multiples of $k+1$ by $1, 2, \ldots, 2k$ also take on all these values. Note that in this coloring, the arcs at node i have every color *except* c_i.

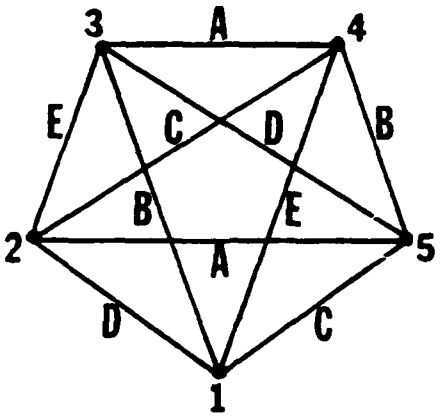
For d odd, a coloring with d colors can now be obtained in a very simple way. Start with a clique having d nodes, and color its arcs with d colors $c_1, \ldots, c_d$ using the construction in the preceding paragraph, so that no arc of color c_i is incident on node i. Now add a $(d+1)$st node, join it with an arc to each of the d nodes (thus yielding a clique having $d+1$ nodes), and give each of these new arcs the previously missing color, i.e., the arc from the new node to old node i gets color c_i. Evidently this is a correct coloring of a $(d+1)$-node clique with d colors.

As we saw earlier in this section, a clique having three nodes (i.e., a triangle) has a partial path group that is abelian "by default"; at any node, if c_1c_2 is defined, c_2c_1 is not, and so on. A clique having four nodes, colored with three colors using the scheme defined in the previous paragraph, turns out to have a partial path group that is actually abelian. For example, if we denote the colors by A, B, C, we might have



Here every color exists at every node, and we always have $AB = BA$, $BC = CB$, and $CA = AC$. In fact, here again we have $AB = BA = C$, $BC = CB = A$, $CA = AC = B$, as in the Klein 4-group.

For a clique having five nodes, we need five colors, e.g.,



Here, if we start at node 1 and move along colors B and E, we reach node 2, but if we move along colors E and B, we reach node 5. Thus this $\Pi(G)$ is not abelian. In the next two sections we will consider classes of graphs for which the partial path group is always (nearly) abelian.

5. Free abelian partial path groups and hypercubes

A partial path group $\Pi(G)$ will be called *abelian* if for all P and all i, j, we have $Pc_i\,c_j = Pc_j\,c_i$ provided both sides are defined. (As we saw in the triangle example, even if one side is defined, the other needn't be.) Evidently, if $\Pi(G)$ is abelian, α is any string of colors, and $\alpha*$ is any permutation of α, then $P\,\alpha = P\,\alpha*$, provided both sides are defined.

Let α be a string of colors. Any reduction α' of a permutation of α will be called a *p-reduction* of α. [Note that $P\,\alpha'$ may not be defined even if $P\,\alpha$ is defined.] We call α *p-irreducible* if it has no p-reductions, and *fully p-reducible* if $\alpha = c_0$ or α has c_0 as a p-reduction.

Proposition 5.1. $\alpha \neq c_0$ is p-irreducible if and only if it consists of *distinct* c's other than c_0 (so that in particular, it has length $\leq d$).//

Proposition 5.2. $\alpha \neq c_0$ is fully p-reducible if and only if each color other than c_0 occurs in α an even number of times.//

A partial path group $\Pi(G)$ will be called *free abelian* if it is abelian, and if $P\,\alpha = P$ implies that α is fully p-reducible. We saw in Section 3 that $\Pi(G)$ free corresponds to G being a tree. In this section we shall show that $\Pi(G)$ free abelian corresponds to G being a subgraph of a hypercube.

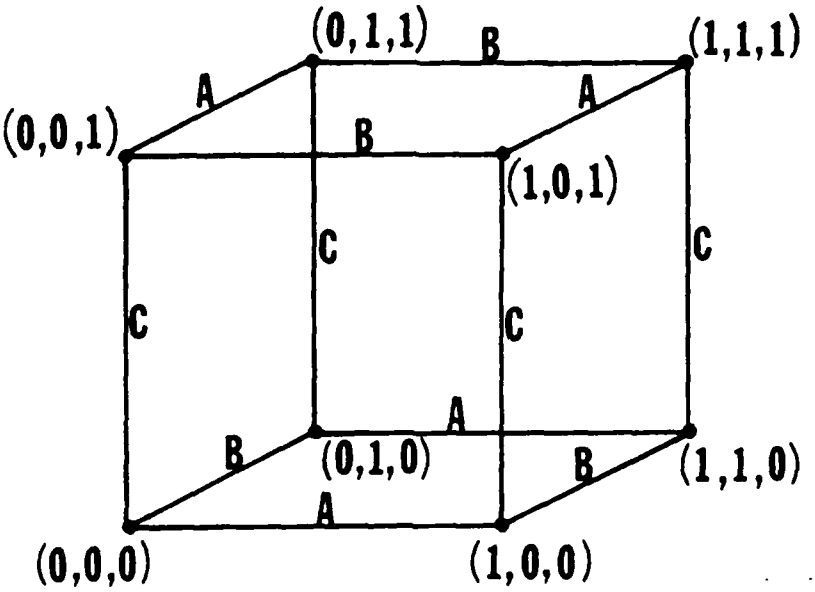
A *d-dimensional hypercube* is a graph G defined as follows: G has 2^d nodes, which we may label with d-tuples $(\epsilon_1, \ldots, \epsilon_d)$ in which each ϵ_i is either 0 or 1. We join nodes $(\epsilon_1, \ldots, \epsilon_d)$ and $(\epsilon_1', \ldots, \epsilon_d')$ with an arc if all but

one of the ϵ's is equal to the corresponding ϵ'. Thus each node P of G has degree d, since there are just d nodes that differ from P in the value of exactly one of the ϵ's. If we think of the ϵ's as coordinates in d-dimensional space, then the nodes of G correspond to the vertices of a d-dimensional unit hypercube located in the first hyperoctant (all coordinates ≥ 0) and with one vertex at the origin.

Theorem 5.3. Let G be a subgraph of a d-dimensional hypercube; then there exists a coloring of the arcs of G with d colors such that $\Pi(G)$ is free abelian.

Proof: Each arc has a "direction" in d-dimensional space, in the sense that along that arc, exactly one coordinate changes. Assign color c_i to the arcs having direction i (i.e., the arcs between nodes $(\epsilon_1, \ldots, \epsilon_d)$ and $(\epsilon_1', \ldots, \epsilon_d')$ such that $\epsilon_i' \neq \epsilon_i$ but $\epsilon_j' = \epsilon_j$ for all $j \neq i$). Since the arcs emanating from a given node all have different directions, this scheme assigns distinct colors to all the arcs at a given node. Evidently, the colors commute wherever they are defined; if we change two coordinates, it doesn't matter in what order we change them. Let $P\alpha = P$; then α must have changed every coordinate of P an even number of times, so that each c_i occurs in d an even number of times. Evidently, any such α is fully p-reducible.//

There exist arc colorings of a hypercube for which $\Pi(G)$ is not abelian. For example, let $d = 3$, and let the edges of the unit cube be colored with colors A, B, C as follows:



This is an admissible coloring, but if we start at (0,0,0), moving along colors A and C gives us (1,0,1), while moving along colors C and A gives us (0,1,1).

Theorem 5.4. Let $\Pi(G)$ be free abelian; then G is a subgraph of a hypercube.

Proof: Let the colors ($\neq c_0$) be $c_1, \ldots, c_d$. Pick any node P of G and label it with the d-tuple $(0, \ldots, 0)$. Let Q be the neighbor of P joined to it by an arc of color c_j; then label Q with the d-tuple $(0, \ldots, 0,1,0, \ldots, 0)$, where the 1 is on the jth place. In general, if any node has been labeled $(\epsilon_1, \ldots, \epsilon_d)$, where each ϵ_i is 0 or 1, and it has a neighbor joined to it by an arc of color c_j, label it $(\epsilon_1, \ldots, \epsilon_{j-1}, \epsilon_j', \epsilon_{j+1}, \ldots, \epsilon_d)$, where $\epsilon_j' \neq \epsilon_j$ (i.e., if ϵ_j is 0, ϵ_j' is 1, and vice versa). Since G is connected, this process assigns a label to every node. Suppose it assigns two labels to the same node R; then there are two paths from P to R, corresponding to two strings of colors α and β, and we have $P\alpha = P\beta = R$. By Propositions 2.1 and 2.2, it follows that $P(\alpha\beta^{-1})$ is defined

and equal to P. Since $\Pi(G)$ is free abelian, $\alpha\beta^{-1}$ must be fully p-reducible. By Proposition 5.2, this means that $\alpha\beta^{-1}$ contains each color other than c_0 an even number of times. Thus if r_i is the number of times α contains c_i, and s_i is the number of times β contains c_i, r_i and s_i must have the same parity, $1 \leq i \leq d$. It follows that the labels assigned to R when it is reached from P along the two paths α and β are the same. Thus the labeling process assigns a unique label to each node of G. If we regard the labels as coordinates, this shows that G is (isomorphic to) a subgraph of a hypercube.//

6. Near-abelian partial path groups and arrays

A partial path group $\Pi(G)$ will be called *near-abelian* if each $c_i \neq c_0$ commutes with all but one of the other c's, and does not commute with the remaining one. It follows that there must be an even number of c_i's (other than c_0). Indeed, let the colors be $c_0, c_1, \ldots, c_m$. Let c_1 commute with all of the c's except (say) c_2; then c_2 must commute with all of the c's except c_1. Thus c_3 must commute with c_1, c_2, and all of the other c's except (say) c_4; hence c_4 commutes with all of the c's except c_3, and continuing in this way, we see that m must be even. In the remainder of this section, we shall denote the colors by $c_0, c_1, \ldots, c_k, c_1', \ldots, c_k'$, where c_i commutes with all the colors except c_i', $1 \leq i \leq k$. If $k = 1$, $\Pi(G)$ is near-abelian by default, since c_1 and c_1' need not commute; we shall assume from now on that $k \geq 2$.

A permutation of a string of colors will be called *admissible* if it never reverses the order of any c_i and any c_i' $(1 \leq i \leq k)$. Any reduction α' of an admissible permutation of α will be called an *a-reduction* of α. [Note that $P\alpha'$ need not be defined even if $P\alpha$ is defined.] We call α *a-irreducible* if it has no *a*-reductions, and *fully a-reducible* if $\alpha = c_0$ or α has c_0 as an *a*-reduction.

Proposition 6.1. $\alpha \neq c_0$ is *a*-irreducible if and only if c_0 does not occur in α, and for each $i \neq 0$, c_i and c_i' occur in α alternately (not necessarily adjacent to one another).//

Proposition 6.2. α is fully *a*-reducible if and only if, for each $j \neq 0$, the (sparse) substring of α composed of c_j's and c_j''s (ignoring all the other colors) is

fully reducible.//

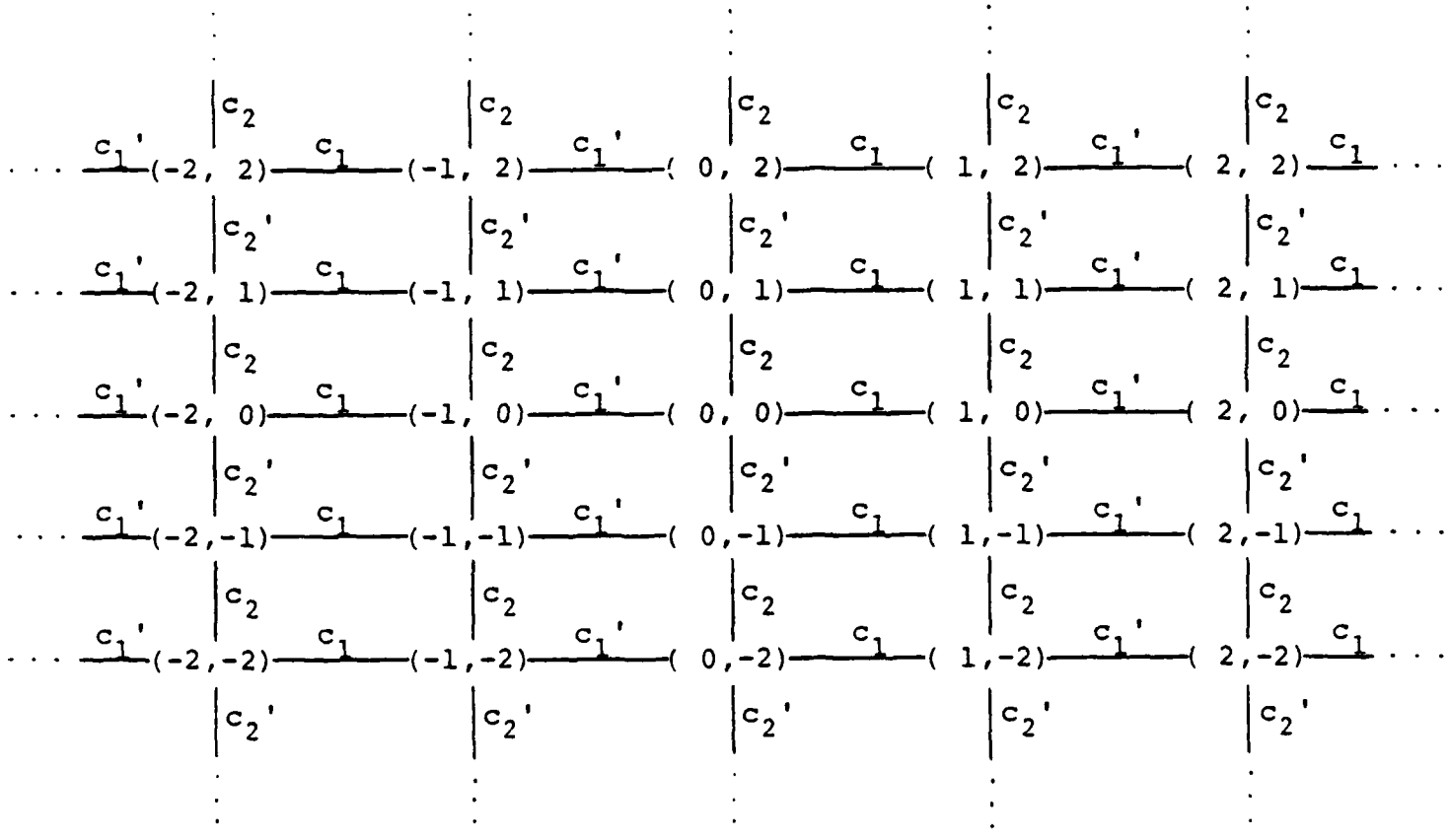
A partial path group $\Pi(G)$ will be called *free near-abelian* if it is near-abelian, and if $P\alpha = P$ implies that α is fully a-reducible. In this section we shall show that $\Pi(G)$ free near-abelian corresponds to G being a subgraph of an array.

An infinite *k-dimensional array* is a graph G defined as follows. The nodes of G are labeled with k-tuples $(a_1, \ldots, a_k)$, where each a_i is an integer (positive, negative, or zero). Nodes $(a_1, \ldots, a_k)$ and $(a_1', \ldots, a_k')$ are joined by an arc if and only if all but one of the a's is equal to the corresponding a', and the pair that are unequal differ by exactly 1. Thus each node P of G has degree $2k$, since there are just two nodes that differ from P by exactly 1 in the value of exactly one coordinate. If we think of the a's as coordinates in k-dimensional space, then the nodes of G correspond to lattice points (i.e., points with integer coordinates).

Theorem 6.3. Let G be a subgraph of a k-dimensional array; then there exists a coloring of the arcs of G with $2k$ colors such that $\Pi(G)$ is free near-abelian.

Proof: We define the coloring as follows: Let P, Q be two nodes joined by an arc; thus all but one of the coordinates of P is the same as that of Q. Let the coordinates that differ be a_j and a_j'; since they differ by 1, exactly one of them is even, say a_j'. If $a_j = a_j'+1$, assign color c_j to the arc between P and Q; if $a_j = a_j'-1$, assign color c_j' to it. Thus if we move along a "row" of the array, allowing only the jth coordinate to change, the colors c_j and c_j' alternate. The

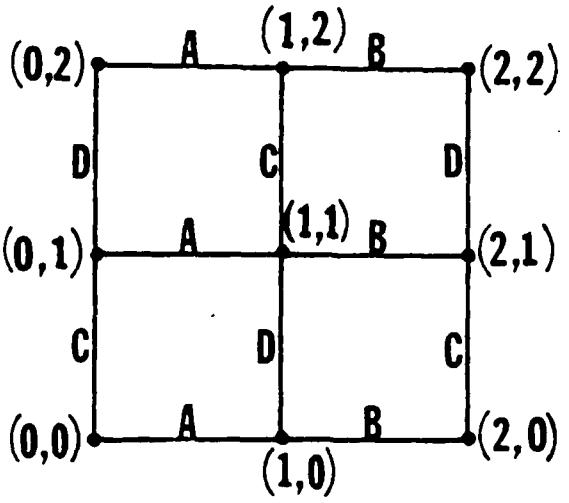
arc between $a_j' = 0$ and $a_j = 1$ has color c_j, and that between $a_j = 0$ and $a_j = -1$ has color c_j', and this completely determines the colors of the arcs along the row, no matter what the values of the other coordinates. A portion of such a coloring for $k = 2$ is shown below.



Evidently this scheme uses only $2k$ colors, and the colors assigned to the arcs emanating from a given node are all different, since a different pair of colors is used when each of the k coordinates changes. Changes in the values of different coordinates evidently commute; hence each c_i or c_i' commutes with all the c_j's and c_j''s for $j \neq i$. Let $P\alpha = P$; then α must give rise to as many positive as

negative changes in each coordinate of P. Each succession of positive changes in the jth coordinate corresponds to an alternating sequence of c_j's and c_j''s (not necessarily adjacent to each other), and similarly for each succession of negative changes. Let a succession of positive changes, corresponding to the alternating sequence β, be followed by a succession of negative changes, corresponding to the alternating sequence γ (or vice versa). The β must end and γ begin with the same color (c_j or c_j'), so that α can be a-reduced. If β and γ have the same length, this a-reduction cancels them completely; if one is longer than the other, the shorter one is cancelled, and the length of the remaining one is the difference of the original lengths. Since the number of positive changes is still equal to the number of negative changes, this argument can be repeated until all the c_j's and c_j''s are eliminated. This can be done for every j; thus α can be fully a-reduced. [Note that this proof does not depend on $P\alpha'$ being defined for any of the intermediate a-reduced strings α'.] Thus we have proved that when G is colored in this way, $\Pi(G)$ is near-abelian. [The proof is longer than that of Theorem 5.2, because we have no simple characterization of fully a-reduced strings; and it is longer than that of Theorem 3.5, because we cannot assume that α is irreducible.]//

There exist colorings of an array for which $\Pi(G)$ is not free near-abelian. For example, let $k = 2$, and let the arcs be colored with colors, A, B, C, D as follows:



Here, at node (1,1), A and B commute with neither C nor D.

Theorem 6.4. Let $\Pi(G)$ be free near-abelian; then G is a subgraph of an array.

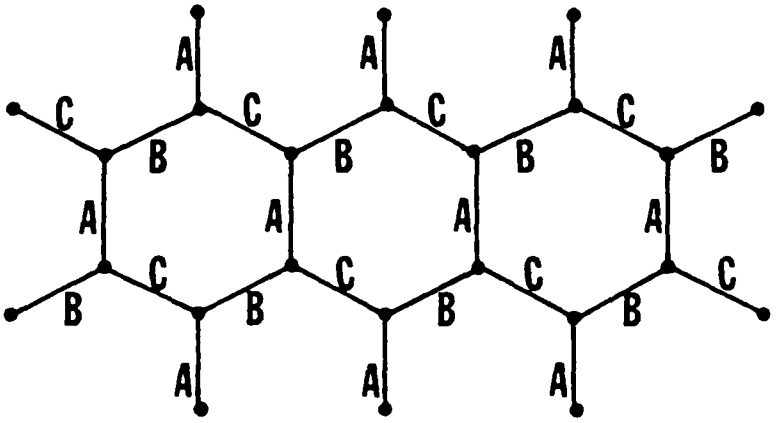
Proof: Let the colors ($\neq c_0$) be $c_1, \ldots, c_k, c_1', \ldots, c_k'$. Pick any node P of G and label it with the k-tuple $(0, \ldots, 0)$. Let Q be the neighbor of P joined to it by an arc of color c_j (or c_j'); then label Q with the k-tuple $(0, \ldots, 0,1,0, \ldots, 0)$ or $(0, \ldots, 0,-1,0, \ldots, 0)$, where the 1 or -1 is in the j th place. In general, if any node has been labeled $(a_1, \ldots, a_k)$, where the a's are integers, and it has a neighbor joined to it by an arc of color c_j, then if a_j is even give that neighbor the same label except that a_j is replaced by a_j+1, and if a_j is odd give the neighbor the same label except that a_j is replaced by a_j-1; and similarly if the color of the arc is c_j', with the roles of odd and even reversed. Since G is connected, this process assigns a label to every node of G.

Suppose it assigns two labels to the same node R; then there are two paths from P to R, corresponding to two strings of colors α and β, and we have $P\alpha = P\beta = R$. As in the proof of Theorem 5.4, it follows that $P(\alpha\beta^{-1})$ is defined and equal to P, and since $\Pi(G)$ is free near-abelian, $\alpha\beta^{-1}$ must be fully a-reducible. By Proposition 6.2, this means that the substrings of $\alpha\beta^{-1}$ composed of c_i's and c_j''s, for each j, are reducible. It follows from this that if we start at P and make the sequence of moves defined by $\alpha\beta^{-1}$, the net change in each component of the label must be zero, so that $P(\alpha\beta^{-1})$ is given the same label $(0, \ldots, 0)$ that P had originally. It is not hard to see that when we start at $P\alpha = P\beta$ and make the sequence of moves defined by β^{-1}, we produce the same set of changes in the label components, in reverse, that we produced by starting at P and make the moves defined by β. Since the net effect of the label component changes defined by $P\beta$ and by $(P\alpha)\beta^{-1}$ is zero, it follows that the set of label component changes defined by $P\alpha$ and that defined by $P\beta$ must be the same, in some order. Thus the labeling process assigns a unique label to each node of G. If we now regard the labels as coordinates, this proves that G is (isomorphic to) a subgraph of an array.//

7. Some other types of arrays

a) *The triangular array*

Let G be a subgraph of the graph obtained by tessellating the plane into equilateral triangles, regarding each triangle as a node, and joining two nodes by an arc if the corresponding triangles share a side. Thus G has degree 3, and its arcs can be colored with three colors A, B, C, e.g., as illustrated below:

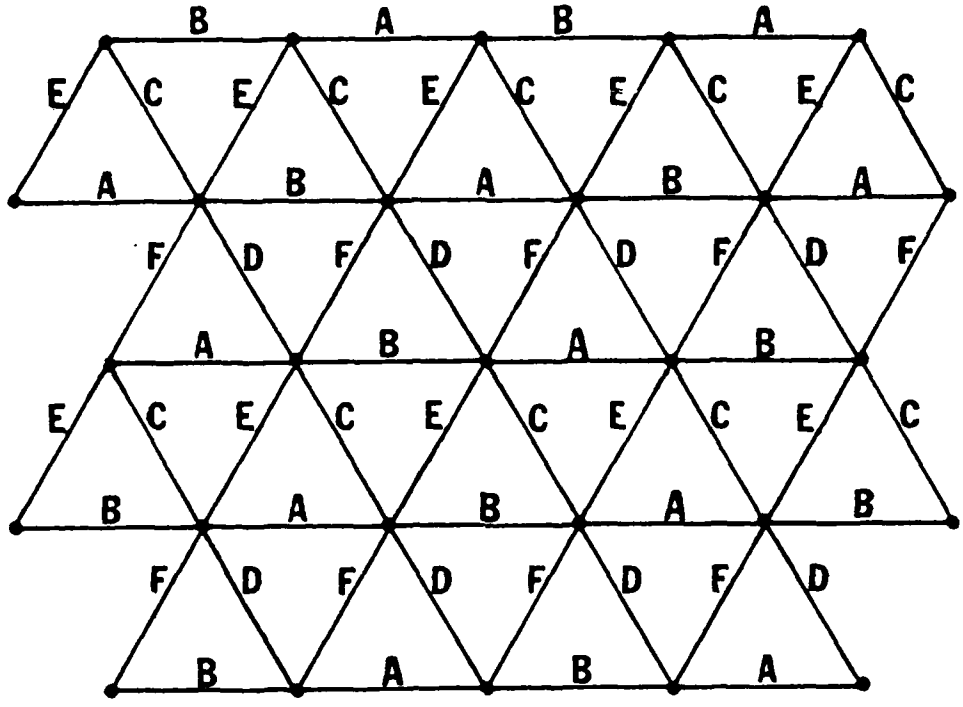


Note that in this coloring no two of the colors commute, but they satisfy other constraints, namely $ABC = CBA$, $ACB = BCA$, $BAC = CAB$.

b) *The hexagonal array*

Let G be defined analogously to case (a), but using a hexagonal instead of a triangular tessellation. [Using a square tessellation evidently gives us a two-dimensional array, which we have already studied.] Then G has degree 6, and its arcs can be colored with six colors A, B, C, D, E, F.

e.g., as illustrated below.



In this coloring, A commutes with C and D, but not with B, E, or F, and similarly for other combinations.

In [6], Mylopoulos defined a different type of group structure on an array; in this structure a generator corresponded to a move in a given direction, and moves in opposite directions were inverses of one another. A k-dimensional (square) array gives rise in this way to a free abelian group on $2k$ generators (k generators and their inverses); and other types of arrays (triangular, hexagonal, etc.) give rise to other types of abelian groups whose generators satisfy additional relations. In our scheme, square arrays correspond to free near-abelian groups, and other types of arrays have more complicated characterizations; but our scheme

also allows us to treat other classes of graphs, such as trees and hypercubes, in addition to arrays, and they turn out to have very simple characterizations.

8. Parallel contraction

We now show how parallel contraction operations can be defined for arc-colored graphs. We define a parallel contraction of a graph G of degree d as follows:

(1) G is partitioned into a set of node-disjoint subgraphs $G_1, \ldots, G_n$, where the number of nodes in each G_i does not exceed some constant m.

(2) The contracted graph G' has nodes $P_1, \ldots, P_n$, each representing one of the G_i's. P_i and P_j are joined by an arc if and only if some node of G_i was joined to some node of G_j by an arc.

A parallel contraction is called *degree-preserving* if the degree of G' does not exceed d. Recall that we are interested in parallel contractions as a method of simulating large processor networks by smaller ones, or building "pyramids" of networks. A parallel contraction of G that preserves degree allows us to simulate G with a network G' of bounded degree d, with each processor in the new network simulating at most m of the original processors. Similarly, if we use G' as the first step in building a pyramid on top of G, the degrees of the nodes of the pyramid do not exceed $d+m+1$ (d connections to neighbors at the same level, m to "children" on the level below, and 1 to a "parent" on the level above). Note that if (most of) the G_i's have $m>1$ nodes, G' has (close to) $\frac{1}{m}$ as many nodes as G, so a pyramid built in this way will taper (nearly) exponentially.

For simplicity, we shall consider here only the case $m = 2$, i.e., we contract G by "merging" pairs of nodes into single nodes. We shall first assume that the pairs of nodes are neighbors in G.

8.1. Merging pairs of neighboring nodes

Arc coloring provides a natural basis for parallel contraction of a graph by merging pairs of neighboring nodes. We simply choose a color, say c_i, and merge all pairs of nodes that are joined by an arc of that color. (We shall refer to this process below as "collapsing a color".) Evidently this process can only merge P with one other node, since at most one arc colored c_i can emanate from P; hence the subgraphs that are merged by collapsing c_i are guaranteed not to have more than two nodes each.

Since G has degree d, when we merge a pair of neighboring nodes P and Q (ignoring for the moment the other merges that are occurring simultaneously), the resulting new node R has degree at most $2d-2$, since P had at most $d-1$ neighbors other than Q, and Q had at most $d-1$ neighbors other than P. In order to insure that our parallel contraction is degree-preserving, we must be able to guarantee that the degree of R does not exceed d. There are two possibilities for reducing the degree of R:

(1) If P and Q have h common neighbors, the degree of R will be reduced by h.

(2) If k pairs of neighbors of P and Q are themselves merged by the parallel contraction (i.e., they are joined by arcs colored c_i), the degree of R will be

reduced by k.

We can express possibilities (1) and (2) in terms of arc colors as follows:

(1′) If $c_i\, c_j = c_k$ for other colors c_j (this is the "closure" property of Section 4), P and Q will have common neighbors; in fact, the neighbor of Q joined to it by an arc of color c_j (if it exists) will be joined to P by an arc of color c_k, and vice versa.

(2′) If $c_i\, c_j = c_j\, c_i$ for other colors c_j (i.e., if c_i commutes with other colors), then other pairs of neighbors of P and Q must be joined by arcs of color c_i and so will be merged by the parallel contraction. In fact, $c_i\, c_j = c_j\, c_i$ implies that the neighbors of P and Q joined to them by arc of color c_j (if they exist) are themselves joined by an arc of color c_i.

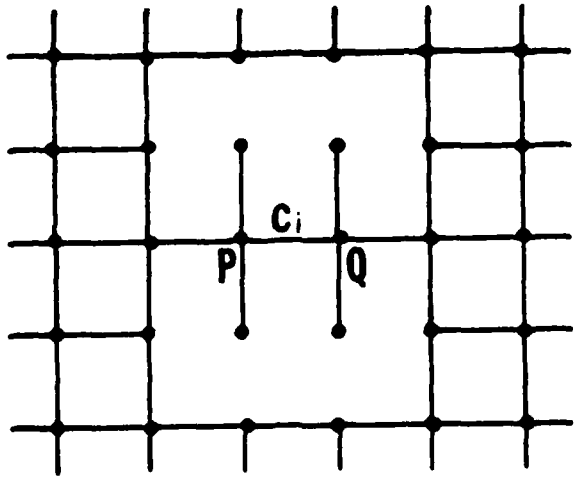
These observations tell us that collapsing a color will give us a degree-preserving parallel contraction in the following cases:

a) Let G be a clique with its arcs colored as described in Section 4. If G has an even number of nodes, say $2k$, collapsing a color gives us a clique G' with k nodes. If G has an odd number of nodes $2k-1$, collapsing a color also gives us a clique with k nodes, because in this case, for any color c_i, there is one node at which there is no arc having color c_i. Note that in these cases the degree of G decreases (from $2k-1$ or $2k-2$ to $k-1$); not only does a pair of nodes in a clique have many common neighbors (in fact, all the neighbors are common!), because of the closure property, but many pairs of these neighbors will also merge.

b) Let G be a hypercube of dimension d with its arcs colored as in Section 5. Then collapsing a color gives us a hypercube of dimension $d-1$. Here too the degree of G decreases, from d to $d-1$. In fact, because of the abelian (=commutativity) property, when we merge neighboring nodes P and Q in a hypercube, each of the other $d-1$ neighbors of P merges with one of the other $d-1$ neighbors of Q.

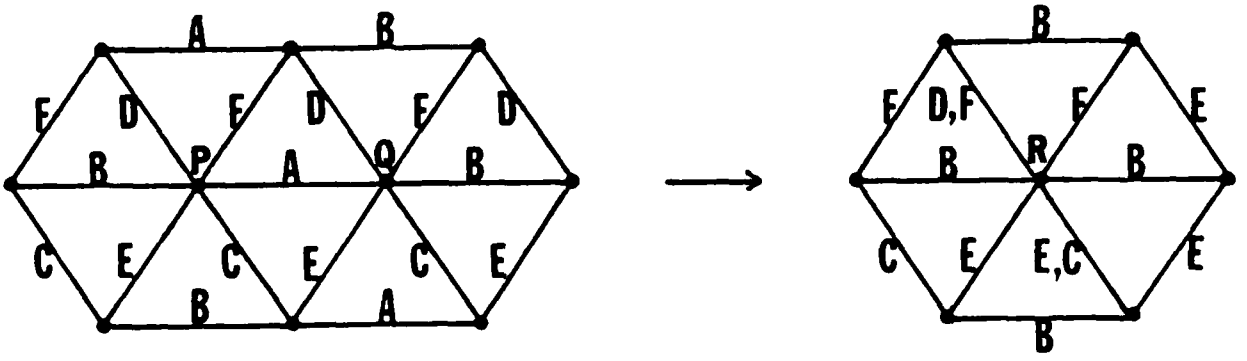
c) Let G be an array of dimension k (and degree $d = 2k$) with its arcs colored as in Section 6. Here collapsing a color gives us an array of the same dimension (but with half as many nodes). In fact, because of the near-abelian property, when we merge neighboring nodes P and Q in an array, *all but one* of the other $d-1$ neighbors of P merges with all but one of the other $d-1$ neighbors of Q; the number of neighbors of the new node R is thus $(d-2)+2 = d$.

In these examples we have assumed that G is a complete clique, hypercube, or array. If G is only a subgraph of a clique, hypercube, or array, collapsing a color may leave too many neighbors unmerged at some node of G. For example, let G be a subgraph of a two-dimensional array, and let part of G be as shown below:



Then when we collapse color c_i, node R resulting from the merge of nodes P and Q will have six neighbors. This type of problem arises when arcs are missing from G; it does not arise when nodes are missing. Thus collapsing a color yields a degree-preserving parallel contraction if the array is finite, if it is a complete finite subgraph of an infinite array; problems arise only when the array is "ragged", as in the example above.

Collapsing a color will yield a degree-preserving parallel contraction for any arc-colored graph having sufficient numbers of common neighbors and commuting pairs of colors. For example, if G is the hexagonal array shown (in part) in Section 7(b), collapsing color A gives the following result:



Here each node originally had six neighbors, and a pair of neighboring nodes had two common neighbors. In addition, each color commutes with two other colors; thus when we collapse a color, two other pairs of neighbors merge. In the case shown above, nodes P and Q have a total of eight neighbors (because of the two common neighbors), but when we merge them into R, it has only six neighbors (because of the two additional merges of pairs). Thus collapsing a color in a

hexagonal array does yield a degree-preserving parallel contraction, because of a combination of the closure and commutativity properties. On the other hand, collapsing a color in a triangular array (Section 7(a)) is not degree-preserving; here there are no common neighbors, and no two colors commute, so when we collapse a color the degree of a node goes up from $d = 3$ to $2d-2 = 4$.

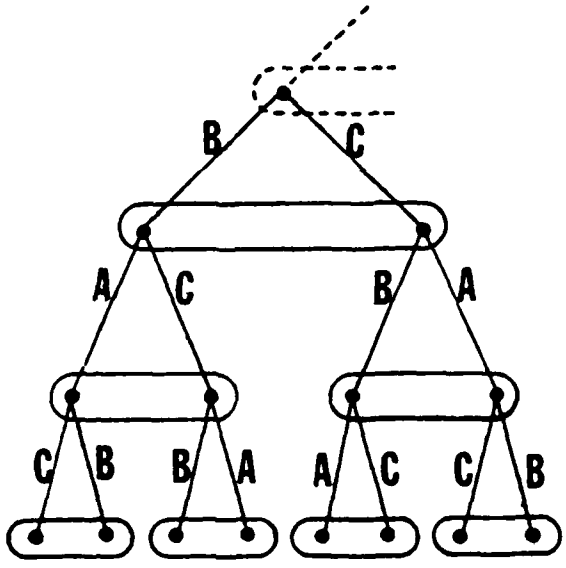
It should be pointed out that when we collapse a color, the arcs of the new graph G' will in general need to be recolored (if we want to use the colors as a basis for repeating the contraction process). For example, in the hexagonal array case shown above, node R now has two arcs colored B emanating from it; we need to reintroduce color A and assign it to half of these arcs. Recoloring is not necessary for a clique or hypercube, where the degree goes down when we collapse a color; but it *is* necessary for arrays.

8.2. Merging nodes at distance 2

Collapsing a color will not yield a degree-preserving parallel contraction in the case of a tree, because there are no common neighbors and no two colors commute. Thus when we merge two neighboring nodes in a tree of degree d, the resulting node has degree $2d-2$, no matter what other disjoint pairs of nodes we merge at the same time. Note that if $d = 2$ we have $2d-2 = 2$, so that the parallel contraction *is* degree-preserving; a tree of degree 2 is a string, which can also be regarded as a one-dimensional array (it is near-abelian "by default").

We can obtain a degree-preserving parallel contraction of a tree by merging disjoint pairs of nodes that are distance 2 apart (i.e., that have a common neigh-

bor), rather than pairs of neighboring nodes. For example, let us merge the circled pairs of nodes in the following (piece of a) tree of degree 3:

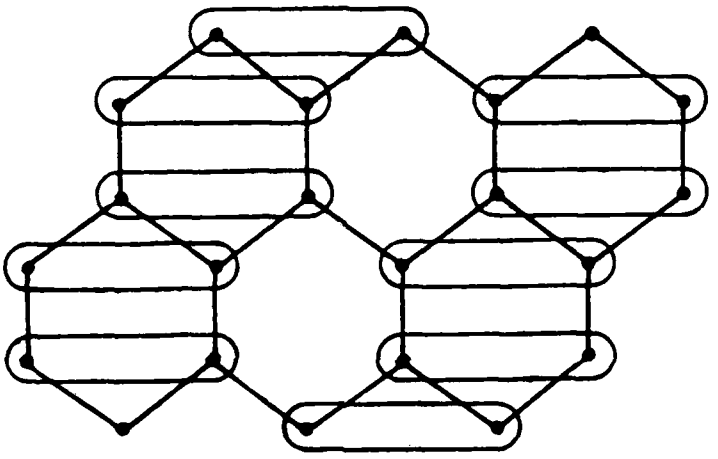


Then we obtain a tree of degree 3.

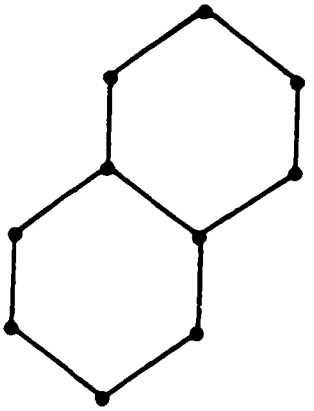
There is no obvious way of defining this type of merge in terms of arc colors; if we color the arcs A, B, C, say as shown above, we see that the merged pairs of nodes have their common neighbors at the other ends of arcs labeled with all possible pairs of colors. In fact, suppose we wanted to define a parallel contraction based on merges of pairs of nodes having a common neighbor to which they are joined by arcs having a given pair of colors c_1 and c_2. Given a node P, it (usually) has a neighbor Q to which it is joined by an arc of color c_1, and Q in turn (usually) has a neighbor R to which it is joined by an arc of color c_2, so we can merge P with R. However, P will usually also have a neighbor Q' joined to it by an arc of color c_2, and Q' in turn will usually have a neighbor joined to it by

an arc of color c_1, so we can also merge P with R'. In other words, arc colors do not provide a basis for defining parallel contractions based on merging *disjoint* pairs of nodes that are distance 2 apart.*

Merging pairs of nodes at distance 2 also yields a degree-preserving parallel contraction in other cases. For example, in the triangular array of Section 7(a), let us merge the circled pairs of nodes as shown below:



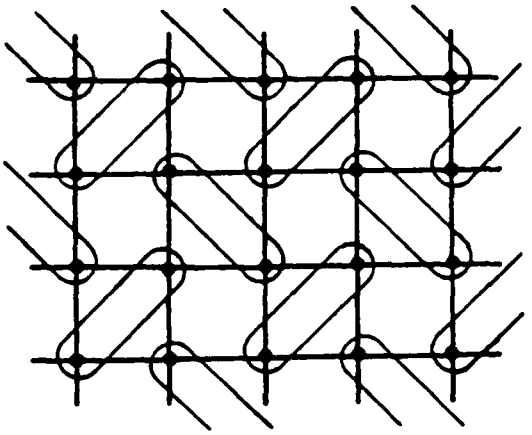
Then we again obtain a triangular array:



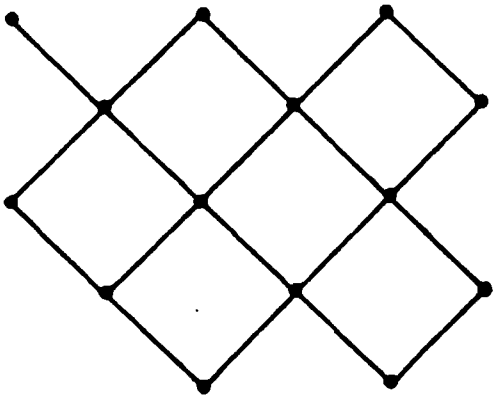
*We could provide such a basis by considering the graph G^2 in which two nodes are joined by an arc if and only if they are at distance ≤ 2 in G, and introducing an arc coloring on G^2. We could then define a parallel contraction of G by collapsing a color in G^2. Note that the degree of G^2 will generally be considerably higher than that of G; for example, if G is a tree of degree d, G^2 has degree d^2. We will not pursue this possibility further here.

In this case the pairs of nodes that we merge are all joined to their common neighbors by the same pair of colors (B and C in Section 7(a)), but we are not merging all such pairs.

As a final example, in the case of a two-dimensional (square) array, let us merged the circled pairs of nodes



Then we obtain a "diagonally oriented" square array:



Note that here the merged pairs of nodes have *two* common neighbors. Here again, there is no simple interpretation in terms of arc colors; for example, we could have equally well merged the opposite diagonal pairs.

9. Concluding remarks

Arc colorings provide a basis for characterizing parallel contractions of a graph based on merging pairs of neighboring nodes. By introducing arc colors, we can define such contractions uniquely; merging all pairs of nodes joined by an arc of a given color will never cause more than two nodes at a time to merge, since a node can have only one incident arc of the given color. At the same time, by introducing an algebraic structure, the partial path group, defined by the arc colors, we can define conditions under which such contractions will not increase the degree of the graph.

Merging pairs of nodes that have a common neighbor (i.e., that are at distance 2) seems to be a more powerful method of defining degree-preserving parallel contractions; it works even for trees, which are a "worst case" from the standpoint of merging pairs of neighbors. It would be of interest to characterize the conditions (perhaps defined in terms of the partial path group of arc colors on G^2) under which such contractions are guaranteed to preserve degree.

References

1. T. Kushner, A.Y. Wu, and A. Rosenfeld, Image processing on ZMOB, *IEEE Transactions on Computers 31,* 1982, 943–951.

2. A. Rosenfeld, ed., *Multiresolution Image Processing and Analysis,* Springer, Berlin, 1984.

3. F. Harary, *Graph Theory,* Addison-Wesley, Reading, MA, 1968, pp. 112–113.

4. S.K. Bhaskar, A.Y. Wu, and A. Rosenfeld, Simulation of large networks of processors by smaller ones, University of Maryland Computer Science Technical Report 1401, May 1984.

5. S. Fiorini and R.J. Wilson, *Edge-Colourings of Graphs,* Pitman, London, 1977.

6. J.P. Mylopoulos and T. Pavlidis, On the topological properties of quantized spaces, *J. ACM 18,* 1971, 239–254.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS: N/A
2a. SECURITY CLASSIFICATION AUTHORITY: N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): CAR-TR-132 CS-TR-1524	5. MONITORING ORGANIZATION REPORT NUMBER(S): N/A AFOSR-TR- 0718
6a. NAME OF PERFORMING ORGANIZATION: University of Maryland	6b. OFFICE SYMBOL (If applicable): N/A
7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research	
6c. ADDRESS (City, State and ZIP Code): Center for Automation Research College Park, MD 20742	7b. ADDRESS (City, State and ZIP Code): Bolling Air Force Base Washington, D.C. 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR	8b. OFFICE SYMBOL (If applicable): -
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: F49620-85-K-0009	
8c. ADDRESS (City, State and ZIP Code): - Bolling AFB D.C. 20332	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO
61102F	2304	A7	

11. TITLE (Include Security Classification): Arc colorings, partial path groups, and parallel graph contractions

12. PERSONAL AUTHOR(S): Azriel Rosenfeld

13a. TYPE	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Technical	FROM _____ TO _____	July 1985	36

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

We define an algebraic structure on the paths in a graph based on a coloring of the arcs. Using this structure, basic classes of graphs (trees, hypercubes, arrays, cliques, etc.) are characterized by simple algebraic properties. The structure provides a framework for defining parallel contraction operations on a graph, in which many pairs of nodes are simultaneously collapsed into single nodes, but the degree of the graph does not increase. Such operations are useful in defining systematic strategies for simulating large networks of processors by smaller ones, or in building "pyramids" of networks.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Dr Robert Buchal	(202) 767-4939	NM

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

END

FILMED

10-85

DTIC