PUC

Series: Monographs in Computer Science
and Computer Applications

Nº 3/73

THE CHARACTERISTIC POLYNOMIAL OF GRAPH PRODUCTS

by

Larry Kerschberg

Computer Science Department - Rio Datacenter

Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente, 209 — ZC-20
Rio de Janeiro — Brasil

THE CHARACTERISTIC POLYNOMIAL OF GRAPH PRODUCTS†

L. Kerschberg
Associate Professor

Computer Science Department
PUC/RJ

Series Editor: S. M. dos Santos 1973

†This research was sponsored in part by the BNDE
Published in the Proceedings, Seventh Asilomar Conference on
Circuits, Systems and Computers
Pacific Grove, California - 1973

ABSTRACT

An algebraic method is presented which calculates the characteristic polynomial of the product of graphs (boolean operations and expressions on graphs) in terms of the polynomials of the factor graphs.

1. INTRODUCTION

Recent work by Mowshowitz [8] and Clarke [1] has shown that the characteristic polynomial of a graph may be used to classify the graph with respect to isomorphism, coverings, 1-factors, and graphical reconstructions. Both authors show that the coefficients of the characteristic polynomial can be computed by counting the number of collections of disjoint directed (or undirected) cycles of specified length.

The approach utilized herein is algebraic rather than combinatorial; we take advantage of the multilinear structure of the graph products to obtain necessary and sufficient conditions under which the characteristic polynomial of a class of graph products can be expressed in terms of the polynomials of the factor graphs. Extensive use is made of the eigenvalues of a graph, and to the author's knowledge, this treatment is new.

2. GRAPHS AND GRAPH PRODUCTS

A *digraph* $D = (V,E)$ is an irreflexive binary relation on a finite set $V = V(D)$ of *vertices* of D; the collection $E = E(D)$ of ordered pairs of vertices are called the *edges* of D. Let $|V|$ denote the cardinality of $V(D)$. A *graph* is a symmetric digraph.

The *adjacency matrix* $A = A(D)$ of a digraph is a binary - valued, $|V| \times |V|$ dimensional matrix defined by its i,j-th entry as:

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E(D) \\ 0, & \text{otherwise} \end{cases}$$

for $1 \leq i,j \leq |V|$.

Let $\underline{n} = \{1,2,\ldots,n\}$. The complete graph $K_n = (\underline{n}, E)$ is a graph such that

$$E(K_n) = \{(i,j) \mid i,j \in \underline{n} \ \& \ i \neq j\}$$

The *zero graph* $0_n = (\underline{n}, E)$ has an adjacency matrix which is the $n \times n$ dimensional zero matrix. The *identity graph* $I_n = (\underline{n},E)$ is the identity relation on $\underline{n} \times \underline{n}$. Notice that the identity graph is a reflexive relation, contradicting the above definition of a digraph. We permit this special case because of its usefulness in the sequel.

Let $D_1 = (V, E_1)$ and $D_2 = (V, E_2)$ be digraphs. The *sum* $D_1 + D_2$ is defined as the sum modulo 2 of their adjacency matrices. The sum $D_1 + D_2$ is said to be *edge - disjoint* if for no $1 \leq i,j \leq |V|$, $a_{ij}(D_1) = 1$ and $a_{ij}(D_2) = 1$. The *complement* $\bar{D}$ of a digraph D is the sum $\bar{D} = D + K_{|V(D)|}$.

Harary and Wilcox [4] have made a thorough study of graph products, calling them boolean operations on graphs. A *boolean operation*, o, on an ordered pair of disjoint digraphs D_1 and D_2 yields a digraph $D = D_1 \circ D_2$ such that $V(D) = V(D_1) \times V(D_2)$ and $E(D)$ is expressed in terms of $E(D_1)$ and $E(D_2)$. We define three basic operations and express all others in terms of these basic ones.

The *Kronecker product* $D = D_1 \otimes D_2$ (conjunction [4]) is a digraph such that for any two vertices $u = (u_1, u_2)$ and $v = (v_1,v_2)$ in $V(D) = V(D_1) \times V(D_2)$, the edge $(u,v) \in E(D)$ if $(u_1,v_1) \in E(D_1)$ *and* $(u_2,v_2) \in E(D_2)$. In terms of the adjacency matrices we have

$$A(D) = A(D_1) \otimes A(D_2)$$

where $\otimes$ denotes the Kronecker (or tensor) product and is defined as follows: let A and B be $p_1 \times p_1$ and $p_2 \times p_2$ dimensional, binary-valued matrices, respectively. The Kronecker product $A \otimes B$ is the $p_1p_2 \times p_1p_2$ dimensional matrix of the form

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1p_1}B \\ \vdots & & & \vdots \\ a_{p_11}B & a_{p_12}B & \cdots & a_{p_1p_1}B \end{bmatrix}$$

The *Kronecker sum* $D = D_1 \times D_2$ (cartesian product [9]) is a digraph such that for any two vertices $u = (u_1,u_2)$ and $v = (v_1,v_2)$ in $V(D)$, the edge $(u,v) \in E(D)$ if

$\left[(u_1,v_1) \in E(D_1) \text{ and } u_2 = v_2\right]$ *or* $\left[u_1 = v_1 \text{ and } (u_2,v_2) \in E(D_2)\right]$

It is clear that if $p_1 = |V_1|$ and $p_2 = |V_2|$, then

$$D_1 \times D_2 = D_1 \otimes I_{p_2} + I_{p_1} \otimes D_2 \tag{1}$$

$$A(D_1 \times D_2) = A(D_1) \otimes I_{p_2} + I_{p_1} \otimes A(D_2) \tag{2}$$

The similarity between (1) and (2) is due to our notation. It should be noted that the equality of (1) is isomorphism while that of (2) denotes equality of matrices. Henceforth the adjacency matrix notation will be used only when necessary.

The *complete Kronecker sum* $D = D_1 * D_2$ is the digraph

$$D_1 * D_2 = D_1 \otimes K_{p_2} + K_{p_1} \otimes D_2$$

where $p_1 = |V_1|$ and $p_2 = |V_2|$.

Table 1 below lists the important boolean operations of Harary and Wilcox in terms of the basic operations.

Table 1

Boolean Operation	Definition
Composition [5] (Lexicographic Product [10])	$D_1[D_2] = D_1 \otimes K_{p_2} + D_1 \times D_2$
Symmetric Difference	$D_1 \oplus D_2 = D_1 * D_2 + D_1 \times D_2$
Disjunction	$D_1 \vee D_2 = D_1 * D_2 + D_1 \times D_2 + D_1 \otimes D_2$
Rejection	$D_1 \mid D_2 = \bar{D}_1 \otimes \bar{D}_2$
γ-product	$\overline{(\bar{D}_1 \vee \bar{D}_2)} = D_1 \times D_2 + D_1 \otimes D_2$

Lemma 1 The boolean operations $D_1 \times D_2$, $\overline{(\bar{D}_1 \vee \bar{D}_2)}$ and $D_1[D_2]$ are composed of edge-disjoint sums of digraphs.

Proof For the first two operations, it is sufficient to consider the γ-product since the Kronecker sum is contained therein. The adjacency matrix for the γ-product is of the form:

$$A\overline{(\bar{D}_1 \vee \bar{D}_2)} = A(D_1) \otimes I_{p_2} + I_{p_1} \otimes A(D_2) + A(D_1) \otimes A(D_2) \quad (3)$$

Without loss of generality, suppose that $p_1 = 4$ and $A(D_1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.

The partitioned form of the right hand side of (3) is

$$\left[\begin{array}{c|c|c|c} 0 & 0 & I_{p_2} & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & I_{p_2} & 0 & I_{p_2} \\ \hline 0 & 0 & 0 & 0 \end{array}\right] + \left[\begin{array}{c|c|c|c} A(D_2) & 0 & 0 & 0 \\ \hline 0 & A(D2) & 0 & 0 \\ \hline 0 & 0 & A(D_2) & 0 \\ \hline 0 & 0 & 0 & A(D_2) \end{array}\right] + \left[\begin{array}{c|c|c|c} 0 & 0 & A(D_2) & 0 \\ \hline 0 & 0 & 0 & 0 \\ \hline 0 & A(D_2) & 0 & A(D_2) \\ \hline 0 & 0 & 0 & 0 \end{array}\right]$$

Performing the sums block by block we note that they are edge-disjoint.

In the case of the composition $D_1[D_2]$ we have

$$A(D_1[D_2]) = A(D_1 \otimes K_{p_2}) + A(D_1 \times D_2)$$

$$= A(D_1) \otimes A(K_{p_2}) + A(D_1) \otimes I_{p_2} + I_{p_1} \otimes A(D_2)$$

$$= A(D_1) \otimes (A(K_{p_2}) + I_{p_2}) + I_{p_1} \otimes A(D_2)$$

$$A(D_1[D_2]) = A(D_1) \otimes J_{p_2} + I_{p_1} \otimes A(D_2) \qquad (4)$$

where $J_{p_2} = A(K_{p_2}) + I_{p_2}$. Notice that J_{p_2} is the edge-disjoint sum of graphs. The partitioned form of (4) (with $p_1 = 4$ and $A(D_1)$ as above) is

$$\begin{bmatrix} 0 & 0 & J_{P_2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & J_{P_2} & 0 & J_{P_2} \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} A(D_2) & 0 & 0 & 0 \\ 0 & A(D_2) & 0 & 0 \\ 0 & 0 & A(D_2) & 0 \\ 0 & 0 & 0 & A(D_2) \end{bmatrix}$$

Clearly, the block by block sums are edge-disjoint and the conclusion follows.

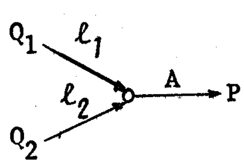
Q.E.D.

3. ALGORITHMS, CHARACTERISTIC POLYNOMIALS, AND EIGENVALUES.

An *algorithm* $Q \xrightarrow{A} P$ is an effectively computable process which maps object Q to object P. The *composition* and *cartesian product* of algorithms are analogous to those of functions; if

$Q \xrightarrow{A_1} P$ and $P \xrightarrow{A_2} R$ then $Q \xrightarrow{A_2 \circ A_1} R$ and

$Q \times P \xrightarrow{A_1 \times A_2} P \times R$. If $Q_1 \times Q_2 \xrightarrow{A} P$, then we use the notation



(5)

where o denotes that $Q_1 \times Q_2$ is the domain of A, and $\ell_i (i=1,2)$ denote the injections $Q_1 \xrightarrow{\ell_1} Q_1 \times Q_2 \xleftarrow{\ell_2} Q_2$.

A *tree algorithm* (T-algorithm) on n objects

$$Q_1 \times Q_2 \times \ldots\ldots\ldots \times Q_n \xrightarrow{T} P$$

is an algorithm which is the composition of a finite number of algorithms such that

i) T is (represented by) a rooted, directed, labelled, binary tree [3];

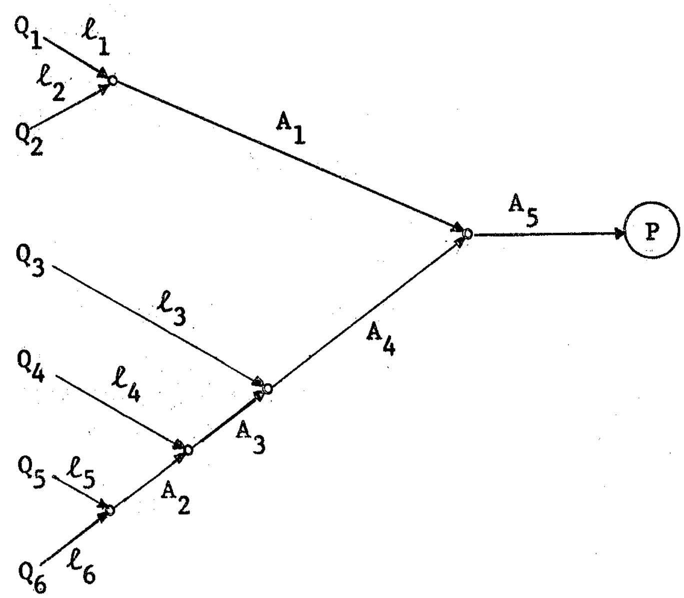
ii) P is the root of T;

iii) The arc labels of T are algorithm names;

iv) The "leaves" of T are the objects $Q_i, i \in \underline{n}$;

v) The internal vertices denote o as in (5), or algorithm objects.

The *T-diagram* for a typical T-algorithm is depicted in Figure 1.



A TYPICAL T-DIAGRAM

Figure 1

It is implicit in Figure 1 that the range object of A_2 serves as one of the domain objects of algorithm A_3, etc. In general, the intermediate objects will not be of interest, while the root of the T-diagram will be of special interest.

The *characteristic polynomial* $C(D)$ of a digraph D is $C(D) = \det(A(D) - Ix)$. The *eigenvalues* of D are the roots of $C(D)$. Let $\lambda(D)$ denote the set of eigenvalues of D.

Algorithm A: [6, p. 55]

If $C(D)$ is of the form

$$C(D) = x^n - c_1x^{n-1} + c_2x^{n-2} - \ldots\ldots + (-1)^n c_n$$

and

$$\lambda(D) = \{\lambda_1, \lambda_2, \ldots\ldots \lambda_n\}$$

then the r-th coefficient c_r of $C(D)$ is the sum of all products of the n eigenvalues taken r at a time.

Thus, algorithm A is denoted as $\lambda(D) \xrightarrow{A} C(D)$.

Fact Given $D = (V,E)$ a digraph and $\lambda(D)$, the coefficient c_1 of $C(D)$ is zero. Here we exclude identity graphs.

This result is immediate from the identity [6]

$$c_1 = \sum_{i=1}^{n} \lambda_i = \text{trace}(A(D)) = \sum_{j=1}^{n} a_{jj} = 0$$

It is worthwhile to note that if D is a digraph, A(D) is binary-valued, the coefficients of C(D) are integers, and the eigenvalues of D are either real or complex numbers. Although A(D) is binary-valued, at times we will treat it as complex-valued, as in the next few algorithms.

Algorithm A_1^* : [7]

Let G = (V,E) be a graph. Since A(G) is a real symmetric matrix whose eigenvalues are real, there exists an algorithm $A(G) \xrightarrow{A_1} \Lambda(G)$, where $\Lambda(G)$ is the diagonal matrix consisting of the eigenvalues of G. Let $A(G) \xrightarrow{A_1^*} \lambda(G)$.

Algorithm A_2^* : [6]

Let D = (V,E) be a digraph. There exists an algorithm $A(D) \xrightarrow{A_2} J(D)$, where J(D) is the block diagonal Jordan form matrix whose diagonal consists of the eigenvalues of D. Let $A(D) \xrightarrow{A_2^*} \lambda(D)$.

The third algorithm relies on a theorem whose proof can be found in Lancaster [6, p. 259]

Theorem 1

Let $\phi(x,y)$ be a polynomial in x and y with complex coefficients c_{ij} of the form

$$\phi(x,y) = \sum_{i,j=0}^{p} c_{ij} x^i y^j \tag{6}$$

Suppose A and B are complex-valued $n \times n$ and $m \times m$ matrices, respectively. Consider the $mn \times mn$ matrix of the form

$$\phi(A;B) = \sum_{i,j=0}^{p} c_{ij} A^i \otimes B^j$$

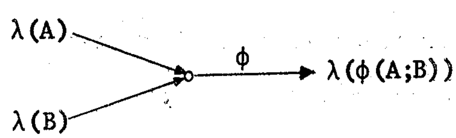
If $\lambda(A) = \{\lambda_1,\ldots,\lambda_n\}$ and $\lambda(B) = \{u_1,\ldots,u_m\}$

then

$$\lambda(\phi(A;B)) = \{\phi(\lambda_r,u_s) | \lambda_r \varepsilon\ \lambda(A),\ u_s \varepsilon\ \lambda(B),\ r\ \varepsilon\ \underline{n},\ s\ \varepsilon\ \underline{m}\}.$$

Algorithm ϕ :

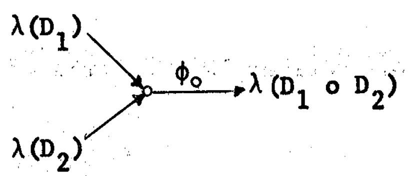
Given $\lambda(A)$, $\lambda(B)$ and $\phi(x,y)$, there exists a T-algorithm which computes $\lambda(\phi(A;B))$.



4. THE CHARACTERISTIC POLYNOMIAL OF GRAPH PRODUCTS

We are now prepared to study the characteristic polynomial of the product of graphs. Our goal is to apply the "algebraic machinery" developed for complex numbers to graphs. We begin by characterizing those boolean operations which have an associated ϕ-polynomial.

__Lemma 2__ Let D_1 and D_2 be digraphs with $\lambda(D_1)$ and $\lambda(D_2)$ given. Let $\Gamma = \{\otimes , x , \gamma\text{-product}\}$. For any boolean operation o in Γ, there exists a T-algorithm ϕ which computes $\lambda(D_1 \circ D_2)$, i.e., for $o \in \Gamma$



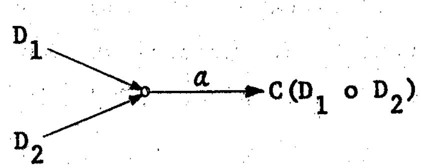
__Proof__ The respective ϕ-polynomials are $\phi_\otimes = xy$, $\phi_x = x + y$, $\phi_\gamma = x + y + xy$.

For those products whose ϕ-polynomials involve sums, we must insure that the sum of adjacency matrices modulo 2 is also valid over the complex numbers (as in Theorem 1). Clearly, this is the case since both x and γ-product are edge-disjoint by Lemma 1.

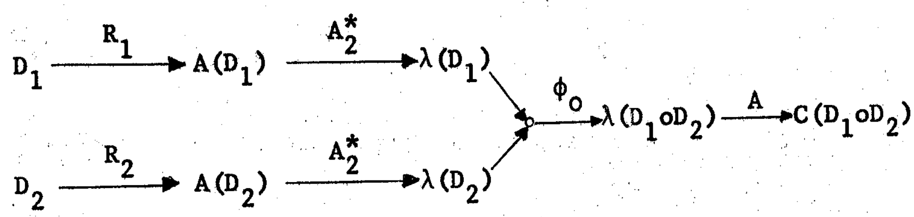
Q.E.D.

We conclude that those boolean operations which involve the sum of digraphs and also admit ϕ-polynomials preserve edge-disjointness. Note however that the composition operation $D_1[D_2]$ preserves edge-disjointness, but has no associated ϕ-polynomial.

<u>Theorem 3</u> Let D_1 and D_2 be digraphs. For any $o \in \Gamma$, there exists a T-algorithm a of the form



<u>Proof</u> The T-diagram below defines a in terms of previously defined algorithms:



$$a = A \circ \phi_o \circ (A_2^* \times A_2^*) \circ (R_1 \times R_2)$$

where R_i maps D_i to its adjacency matrix $A(D_i)$, $i = 1,2$.

Q.E.D.

An obvious extension of Theorem 3 is to include the composition of ϕ-algorithms. For this we introduce the notion of a boolean expression of digraphs. The definition is recursive and follows that of Even [2, p. 140].

Let $\mathcal{D} = \{\mathcal{D}_i \mid i \in \underline{k}\}$ denote a finite set of digraphs and Γ^* a set of boolean operations. Let $B(\mathcal{D}, \Gamma^*)$ denote the set of *boolean expressions*, each consisting of a sequence of boolean operations from Γ^* on digraphs in $\mathcal{D}$.

A *well-formed* boolean expression satisfies the following conditions:

1. The empty expression is well-formed.

2. If A and B are well-formed, then for any $o \in \Gamma^*$, A o B is well-formed.

3. If A is well-formed, then so is (A).

4. There are no other well-formed expressions.

Henceforth, we consider only well-formed boolean expressions. From Table 1 it is clear that every boolean expression can be expressed in its *normal form*, which consists of the sum of the Kronecker products of digraphs. The normal form of a boolean expression is *disjoint* if the sum of Kronecker products of digraphs is edge-disjoint.

Lastly, the characteristic polynomial of a boolean expression is *T-realizable* if there exists a T-algorithm consisting of the composition T-algorithms ϕ such that the root of T is the characteristic polynomial of the expression.

Example 1

$$\text{Let} \quad B = (\overline{(D_1 \times D_2) \vee \bar{D}_3}) \otimes D_4$$

The normal form for B is

$$NF(B) = ((D_1 \times D_2) \times D_3 + (D_1 \times D_2) \otimes D_3) \otimes D_4$$

$$= ((D_1 \otimes I_{p_2} + I_{p_1} \otimes D_2) \times D_3 + (D_1 \otimes I_{p_2} + I_{p_1} \otimes D_2) \otimes D_3) \otimes D_4$$

$$= ((D_1 \otimes I_{p_2} + I_{p_1} \otimes D_2) \otimes I_{p_3} + I_{p_1 p_2} \otimes D_3 + D_1 \otimes I_{p_2} \otimes D_3 +$$

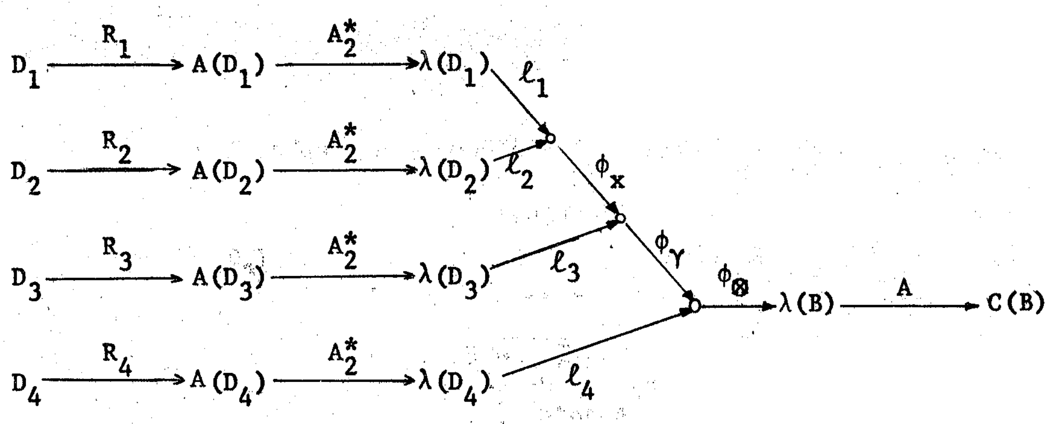
$$+ I_{p_1} \otimes D_2 \otimes D_3) \otimes D_4$$

$$= D_1 \otimes I_{p_2} \otimes I_{p_3} \otimes D_4 + I_{p_1} \otimes D_2 \otimes I_{p_3} \otimes D_4 + I_{p_1} \otimes I_{p_2} \otimes D_3 \otimes D_4 +$$

$$+ D_1 \otimes I_{p_2} \otimes D_3 \otimes D_4 + I_{p_1} \otimes D_2 \otimes D_3 \otimes D_4$$

It can be verified that $NF(B)$ is disjoint. Moreover, this can be ascertained directly from B by noting that $\times$ and the γ-product have disjoint normal forms.

Clearly C(B) is T-realizable and has the T-diagram.



<u>Theorem 4</u> Every T-realizable, well-formed boolean expression $B \in B(\mathcal{D}, \Gamma^*)$ has a disjoint normal form.

<u>Proof</u> Since B is T-realizable then all ϕ-algorithms involving sums of graphs preserve edge-disjointness, as do their compositions. Thus, the sequence of expansions which leads to the normal form for B preserves edge-disjointness, thereby insuring that B has a disjoint normal form.

Q.E.D.

Clearly, there are well-formed boolean expressions which have a disjoint normal form and yet are not T-realizable. This is the case for $B = D_1[D_2]$. If we restrict Γ^*, however, a disjoint normal form for B becomes a sufficient , as well as a necessary condition, for T-realizability.

<u>Theorem 5</u> Let $\Gamma^* = \{ \otimes, x, \gamma \}$. The characteristic polynomial of every well-formed boolean expression $B \in B(\mathcal{D}, \Gamma^*)$ is T-realizable.

Obviously, it is of interest to make Γ^* as large as possible. In the case of complete graphs, we obtain some interesting results.

<u>Lemma 3</u> Let K_{p_1} and K_{p_2} be complete graphs. Then

i) $K_{p_1} \times K_{p_2} = K_{p_1} \oplus K_{p_2}$

ii) $K_{p_1}[K_{p_2}] = K_{p_1} \vee K_{p_2} = \overline{(\bar{K}_{p_1} \vee \bar{K}_{p_2})}$

iii) $K_{p_1 p_2} = K_{p_1}[K_{p_2}] = K_{p_2}[K_{p_1}]$

iv) $K_{p_1} * K_{p_2} = 0_{p_1 p_2}$.

<u>Proof</u> The results can be verified by direct substitution in Table 1 and by comparing adjacency matrices.

<u>Theorem 6</u> Let $m \geq 2$ be a natural number. Then there exists a T-algorithm a^* of the form

$$K_{p_1} \times K_{p_2} \times \dots \times K_{p_k} \xrightarrow{a^*} C(K_m)$$

where p_i , $1 \leq i \leq k$ are prime numbers.

<u>Proof</u> Every natural number m has a decomposition as the product of powers of primes,

$$m = p_1^{e_1} \cdot p_2^{e_2} \cdots p_k^{e_k}$$

where $2 \leq p_1 < p_2 < \ldots < p_k$.

By repeated use of Lemma 3iii, we obtain

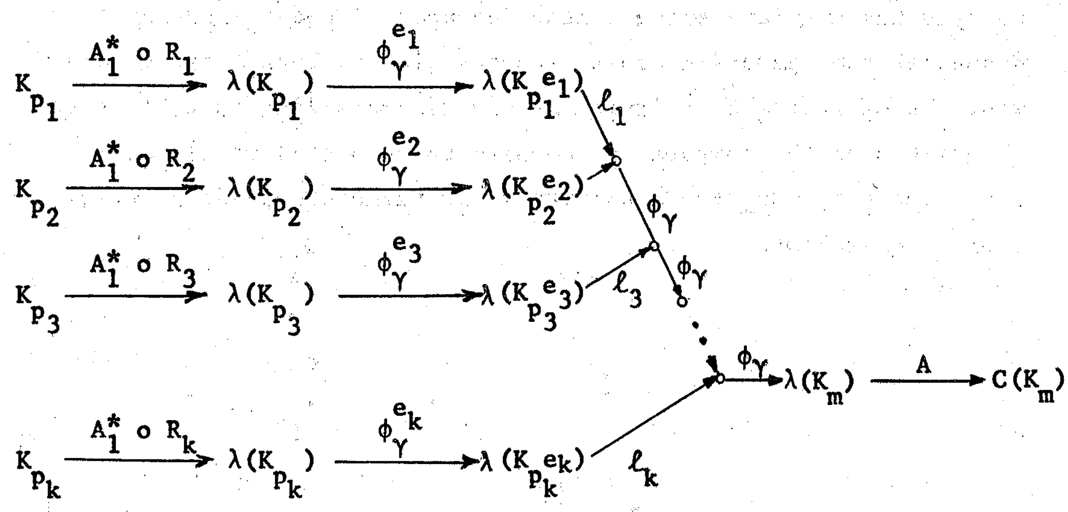
$$K_m = K_{p_1}^{e_1} [K_{p_2}^{e_2} [\ldots\ldots K_{p_{k-1}}^{e_{k-1}} [K_{p_k}^{e_k}]] \ldots\ldots]$$

where for each $i \in \underline{k}$

$$K_{p_i}^{e_i} = K_{p_i} [K_{p_i} [\ldots\ldots K_{p_i} [K_{p_i}]] \underbrace{\ldots\ldots\ldots]}_{e_i\text{-times}}$$

From Lemma 3ii we know that the composition of complete graphs preserves edge-disjointness, thereby admiting a T-realization for $C(K_m)$. Clearly, the tree will be composed of ϕ_γ-algorithms. For simplicity let $\phi_\gamma^{e_i}$ denote the algorithm

$\lambda(K_{p_i}) \xrightarrow{\phi_\gamma^{e_i}} \lambda(K_{p_i}^{e_i})$. The T-diagram for $C(K_m)$ is



Q.E.D.

Theorem 7 Let $K = \{K_{p_i} \mid p_i$ a prime and $i \in \underline{n}\}$ be a finite set of complete graphs. Let $\bar{\Gamma} = \{ \otimes, x, \gamma, [\,], \oplus, V \}$. The characteristic polynomial of every well-formed boolean expression $B \in B(K, \bar{\Gamma})$ is T - realizable.

Proof The conclusion is a direct consequence of Lemma 3 and Theorem 5.

5. CONCLUDING REMARKS

We have presented an algebraic method involving tree algorithms which computes the characteristic polynomial of graph products (well-formed boolean expressions) in terms of the eigenvalues (and hence the characteristic polynomials) of the factor graphs. Necessary and sufficient conditions have been presented for the T-realizability of the characteristic polynomial of a class of boolean expressions.

Sabidussi [9] has shown that every connected graph of finite type has a unique prime factor decomposition with respect to the Kronecker sum operation. This decomposition is T-realizable, and it would be of interest to implement his decomposition as well as the T - algorithms on the computer to compare the proposed method with a direct method, such as Mowshowitz's, in terms of speed, accuracy, and store requirements.

REFERENCES

1. F.H. Clarke, A graph polynomial and its applications, Discrete Mathematics 3 (1972) 305-313.

2. S. Even, Algorithmic combinatorics (Macmillan, New York, 1973).

3. F. Harary, Graph theory (Addison-Wesley, Reading, 1969).

4. F. Harary and G.W. Wilcox, Boolean operations on graphs, MATH.SCAND. 20 (1967), 41-51.

5. F. Harary, On the group of the composition of two graphs, Duke Math. J. 26 (1959), 29-34.

6. P. Lancaster, Theory of matrices (Academic Press, New York, 1969).

7. E. Isaacson and H.B. Keller, Analysis of numerical methods (John Wiley, New York, 1966).

8. A. Mowshowitz, The characteristic polynomial of a graph, Journal of Combinatorial Theory, 12 (1972), 177-193.

9. G. Sabidussi, Graph multiplication, Math. Z. 72 (1960), 446-457.

10. G. Sabidussi, The lexicographic product of graphs, Duke Math. J. 28 (1961), 573-578.