ON THE COMPLEXITY OF EVALUATING MULTIVARIATE POLYNOMIALS

Mark Jerrum

Ph.D.

University of Edinburgh

1981

DECLARATION

With the exception of chapter 5, which represents the result of a collaboration with Marc Snir, the work described in this thesis is my own. Chapter 3 contains material previously presented in a paper "Applications of Algebraic Completeness to Problems of Network Reliability and Monomer-Dimer Systems", which has been published as a Departmental Report (CSR-45-79) and is to appear in SIAM Journal on Computing. Chapter 5 is based on a paper, co-authored with Marc Snir, entitled "Some Exact Complexity Results for Straight-line Computation over Semirings". This paper has appeared as a Departmental Report (CSR-58-80), and has been accepted for publication in Journal of the ACM.

The thesis has been composed by me.

ABSTRACT

A wide range of multivariate polynomials is considered and an attempt made to explain the varying amounts of computational effort required in their evaluation. Two approaches to this task are documented. In the first, a completeness class of polynomial families is introduced, the members of which are interrelated by means of elegant algebraic reductions known as projections. It is likely that no member of this completeness class may be evaluated using a number of arithmetic operations (+,-,x,/) which is bounded by a polynomial function of the number of indeterminates; the members of the class may reasonably be termed intractable.

Two polynomials arising in the study of the physical properties of crystal lattices, namely the generating functions for monomer-dimer arrangements on a 2-dimensional lattice and for the 3-dimensional Ising problem, are shown to be intractable in this sense. Further multivariate polynomials concerned with network reliability are shown to be similarly intractable, a particular example being the probability that all stations of an unreliable network can communicate with each other. For all the specific examples mentioned above there was previously no explanation of their apparent computational difficulty.

In the second of the two approaches, attention is restricted to monotone computations, that is computations involving only the arithmetic operations {+, ×} and non-negative real constants. The reward for restricting the domain of computation is that it becomes possible to obtain exact bounds on the number of multiplications required to compute various polynomials. In this way, provably optimal monotone algorithms can be exibited for computing the permanent of a n×n matrix (which is shown to require $n2^{n-1}-n$

multiplications) and the generating function for Hamiltonian circuits in the complete graph of order n $((n-1)(n-2)2^{n-3}+(n-1)$ multiplications). As a bonus, the results hold good for other computational domains, including the minimax algebra (real numbers with the operations of + and min). In particular, it will be shown that finding minimum weight matchings using a straight-line program in this algebra takes exponential time, whereas known algorithms using branching solve the problem in polynomial time.

CONTENTS

1. INTRODUCTION

This thesis is concerned with the computational properties of multivariate polynomials. In the field of computer science such polynomials arise not only explicitly in numerical computations, for example matrix multiplication, but also in implicit forms. Problems involving the counting of certain structures in graphs for example, perhaps more often regarded as combinatorial enumeration, can be viewed as the evaluation of appropriate polynomial generating functions. It is natural to ask how difficult are these polynomials to evaluate.

The usual model of computation employed in such a study is the straight line algorithm, which informally consists of a linear sequence of instructions each calling for the addition, multiplication etc. of two previously computed values or inputs. The inputs to the computation are indeterminates and elements of the field underlying the computation. The complexity of the computation is measured, perhaps in terms of the total number of arithmetic operations used, perhaps by the number of multiplications required. There are two essential features of the model. Firstly, the arithmetic operations, and the elements of the underlying field, are atomic units of computation and are considered indivisible. We are concerned neither with the representation of field elements nor with the implementation of the operations of a more fundamental level; furthermore, it is not considered important that the time taken to execute each step of the algorithm on a "reasonable" machine would be a function of the length of the representations of the values being manipulated. (It may be remarked in passing that this mirrors the situation in real computers, where approximations to real numbers are held in small, fixed multiples of words and are operated upon as single entities.) The second feature of the model is that it lacks uniformity of the kind possessed by Turing machines. A separate straight line algorithm is provided for each input size - in this context, input size is the number of indeterminates in the polynomial being computed. A Turing machine, on the other hand, is expected to act uniformly over all input sizes.

Ideally, we would like to know good bounds on the number of operations required to compute specific polynomials in such a model. Unfortunately, such bounds seem, in general, very difficult or impossible to obtain. We must content ourselves with results which are less cut and dried but still hopefully informative.

In this thesis two distinct approaches are described to the problem of determining the inherent complexity of evaluating multivariate polynomials. In chapter 2, a classification is described which enables, subject to a reasonable assumption, a broad division between tractable and intractable polynomials to be made. A class of polynomial families is described whose members may be informally characterised as having easily computable specifications. A notion of reducibility between polynomial families is presented which enables a "completeness class" of families to be identified. The members of the completeness class are characterised as being at least as difficult to compute as any member of the original class. (This situation mirrors the more familiar, machine based completeness class in NP.) The completeness class has the property that either all its members can be computed using a number of arithmetic operations which is polynomial in the number of indeterminates, or none can be. The heavy weight of circumstantial evidence points to the latter.

The aforementioned completeness class was introduced by Valiant [43], who showed that several naturally occurring polynomial families, including the permanent function and the generating function for Hamiltonian circuits in a complete graph, are contained within it. Perhaps the most interesting family to be classified in this way was the generating function for dimer arrangements on the 3-dimensional cubic lattice graph - a problem which has its origin in the study of the physical properties of crystal lattices. An efficient algorithm for computing the generating function on 2-dimensional lattices had been known for some time, and attempts had been made to extend the method to 3-dimensional lattices. The significance of the result was that it explained the failure of these attempts; since the generating function for a particular 3-dimensional lattice is a member of the above completeness class, it is unlikely that any efficient algorithm exists which solves the problem in general.

A closely related problem is that of computing the generating function of monomer-dimer arrangements on a crystal lattice. If the lattice is taken to be the 3-dimensional cubic lattice, the intractability of the task follows easily from the completeness result for dimer arrangements. It is tempting to suppose, however, that some technique akin to that employed in the enumeration of dimer arrangements on 2-dimensional lattices might be applicable to the related monomer-dimer problem. In chapter 3, this is shown to be very unlikely, as the generating function for monomer-dimer arrangements on the 2-dimensional rectangular lattice is also a member of the completeness class. In the same chapter, another crystal lattice enumeration problem, the so-called Ising problem, is shown to be intractable for 3-dimensional lattices; again this contrasts with the 2-dimensional case for which an efficient solution is known.

In chapter 4, the same classification is applied to the study of reliability measures for communication networks. Such networks have been intensively studied for some time and many, but not all, of the natural reliability measures have been shown to be computationally intractable. One measure, for which there was previously no evidence of intractability, is the probability that all stations of a network can communicate with each other. This measure is treated in chapter 4, and is shown to be a member of the algebraic completeness class.

The main observation to be made about chapters 3 and 4 is that the framework described above, and which will be established in chapter 2, enables problems to be treated which have escaped classification in other formulations. This is suggestive of the power and utility of the algebraic reductions employed.

In the final chapter, a number of precise bounds on the number of arithmetic operations required to compute certain polynomials are obtained, at the expense of limiting the model of computation somewhat. Computations are considered which involve only the arithmetic operations $\{+,\times\}$ and non-negative real constants, and lower bounds obtained using a combinatorial argument. As a bonus, the results hold good for computations over a number of other domains - a notable example being the minimax algebra (the real numbers with the operations of min and +) which has been used, on a number of occasions, for specifying and solving combinatorial optimisation problems.

This restricted form of computation, often termed monotone arithmetic, has been studied by other authors. Schnorr [32] has presented an argument for bounding the number of additions required in the computation of various multivariate polynomials.

Shamir and Snir [34,35], in addition to obtaining lower bounds on the formula depth of several polynomials, have determined bounds on the number of multiplications required to compute specific polynomials. Among the polynomials to which their technique has been applied are the permanent function on n^2 variables, for which a bound of $O(1.88^n)$ exists, and the generating function for Hamiltonian circuits in the complete graph on n vertices, for which a similar bound applies. Although exponential, these bounds are not tight, and chapter 5 will present a technique for obtaining exact, that is to say attainable, bounds for these and other multivariate polynomials. In particular, we shall show that $n2^{n-1}-n$ multiplications are both necessary and sufficient for the monotone computation of the permanent of an $n \times n$ matrix, and that $(n-1)(n-2)2^{n-3}+(n-1)$ multiplications is optimal for the generating function for Hamiltonian circuits on the complete graph of order n. In addition, provably optimal monotone algorithms are presented for iterated matrix multiplication and iterated convolution.

The results obtained in chapter 5 are interesting, even if the computational model is rather weak, as exact bounds are a rarity in the field of computational complexity. More importantly, however, the results act as pointers to where the seat of power lies in less restricted models. It is shown, for example, that an exponential number of arithmetic operations are necessary in any monotone computation of the generating function for spanning trees in a complete graph, whereas the same polynomial can be efficiently computed if negation is allowed. (The existence of such an exponential gap is not a new discovery - it had been demonstrated by Valiant [42] in connection with the dimer generating function on a particular 2-dimensional lattice.)

In the same vein, it is shown that for computations in the minimax algebra an exponential speed-up can be obtained by introducing branching. Computing a maximum matching in a bipartite graph, for example, requires an exponential number of {max, +} operations, whereas known algorithms (see Lawler [16] p. 205) accomplish this in a polynomial number of steps, if branching is allowed.

2. AN ALGEBRAIC COMPLETENESS CLASS

2.1 Informal Description

In classical, machine based, complexity theory, the objects of interest are computationally defined classes of languages, for example the class NP of languages accepted by non-deterministic Turing machines which halt within a number of steps bounded by a polynomial function of input size. It is a well-documented phenomenon that, within such classes of languages, certain naturally defined languages exist which are, in some sense, as hard to compute as any in the class. These naturally defined languages belong to a completeness class which has the following characterisation: for any language L in the class, and any L_c in the completeness class, there is an efficiently computable function f with the property that

$$w \varepsilon L \Longleftrightarrow f(w) \varepsilon L_c.$$

By taking the class NP and drawing f from P, the class of functions computable by polynomial time deterministic Turing machines, the well known NP-completeness class is obtained (see, for example, Karp [13]). Note that the class and the reductions which establish the completeness class are both computationally defined.

Following Valiant [43], we shall take an approach which mirrors this, but does away with all computational references (although the results obtained certainly retain computational interest and implications). Rather than dealing with classes of languages, we shall be considering classes whose elements are polynomial families. These families consist of polynomials in several indeterminates, the polynomials being indexed by the natural numbers. Our convention will be that the index of a polynomial shall be equal to the number of indeterminates on which it depends; in this sense the index corresponds to the notion of input size. The classes of polynomial families which we shall consider are defined in a way which is partly combinatorial, partly algebraic, but makes no reference whatsoever to machine models.

The reductions employed between polynomial families are straightforward substitutions of indeterminates by other indeterminates or constants. Considering the very simple nature of the reductions it is rather surprising that many naturally defined polynomial families (for example those described in chapters 3 and 4 of this thesis, or by Valiant in [43]) turn out to be elements of a completeness class defined in this algebraic way. It should be noted that if a polynomial family is a projection of another, then polynomials in the first family can be computed by evaluating elements of the second with suitable assignments to the indeterminates. This situation parallels the programming concept of package, since inputs are "plugged-in" without pre-computation.

The particular algebraically defined class we shall be dealing with is the class of "p-definable polynomials". Combinatorial enumeration problems are usually viewed in terms of evaluating a certain polynomial called the generating function; for many natural enumeration problems this polynomial is p-definable. Moreover, for many "hard" enumeration problems the associated generating function is complete in the algebraic sense. Note that if a polynomial family is shown to be complete, then an algorithm for evaluating members of the family may be used as a "package" for evaluating members of any p-definable family of polynomials. Since the class of p-definable polynomial familes contains many families which, for empirical reasons, are thought to be hard to compute, showing a polynomial family to be complete is considered to be good evidence that it is difficult to compute.

2.2 Notation and Definitions

Let F be a field and let $F[x_1,\ldots,x_n]$ be the ring of polynomials over the indeterminates $x_1,\ldots,x_n$ with coefficients drawn from F (see, for example, Godement [10]). We shall be dealing extensively with families of polynomials, conventionally represented by P and Q, of the following form

$$P = \{P_i \mid P_i \in F[x_1,\ldots,x_i],\ i=1,2,..\}.$$

If it is necessary explicitly to exhibit the arguments of P_i, we do so by writing $P_i(x_1,\ldots,x_i)$.

The set of all formulae over F is defined recursively as follows:

"c", where $c \in F$, is a formula.

"x_i" where x_i is an indeterminate, is a formula.

$$\left.\begin{array}{l}\text{"}(f \oplus g)\text{"}\\ \text{"}(f \otimes g)\text{"}\end{array}\right\}$$ where f and g are formulae, is a formula.

The symbols $\oplus$ and $\otimes$ are intended, for the moment, to be uninterpreted syntactic objects. The size, $|f|$, of a formula f is an integer which is also recursively defined:

$$|c| = 0$$

$$|x_i| = 0$$

$$|(f \oplus g)| = |f| + |g| + 1.$$

$$|(f \otimes g)| = |f| + |g| + 1.$$

Each formula represents a polynomial; the polynomial p_f represented by a formula f is obtained as follows:

$$p_c = c.$$

$$p_{x_i} = x_i.$$

$$p_{(f \oplus g)} = p_f + p_g.$$

$$p_{(f \otimes g)} = p_f p_g.$$

(On the right-hand side of the defining equations, we are employing ordinary addition and multiplication of polynomials.)

Clearly a polynomial can be represented by several formulae, for example the polynomial

$$x_1 x_2 + x_1 x_3$$

is represented both by the formula

$$((x_1 \otimes x_2) \oplus (x_1 \otimes x_3))$$

of size 3, and by the formula

$$(x_1 \otimes (x_2 \oplus x_3))$$

of size 2.

The <u>formula</u> <u>size</u> $|P_i|$ of the polynomial P_i is the minimum of $|f|$ over all formulae f which represent P_i.

We now introduce an important class, that of p-definable polynomial families. A polynomial family P over F is <u>p-definable</u> if there is a family Q and a one argument polynomial t(.) such that for each i there exists a j with the property that

$$P_i(x_1,\dots,x_i) = \sum_{(x_{i+1},\dots,x_j) \in \{0,1\}^{j-i}} Q_j(x_1,\dots,x_j)$$

<u>and</u> $|Q_j| \leqq t(i)$.

In the next chapter a sufficient condition for a polynomial family to be p-definable will be presented; for the present, let it just be said that the class of p-definable families is very rich and includes most of the generating functions associated with classical enumeration problems.

Having introduced the class of polynomial families we shall be working with, let us now consider a notion of reducibility between members of the class. We shall say that $P_i \varepsilon F[x_1,\ldots,x_i]$ is a projection of $Q_j \varepsilon F[x_1,\ldots,x_j]$ iff there is a mapping

$$\sigma\colon \{x_1,\ldots,x_j\} \rightarrow \{x_1,\ldots,x_i\} \cup F$$

such that

$$P_i(x_1,\ldots,x_i) = Q_j(\sigma(x_1),\ldots,\sigma(x_j)).$$

The family P is a p-projection of Q iff there is a one argument polynomial t(.) such that for all i, P_i is a projection of Q_j for some $j \leq t(i)$. It should be noted that if two families P and Q are p-projections of each other then they are of similar computational difficulty, for P can be used as a "package" for computing Q and vice versa; we need only make the correct assignments to the indeterminates.

We can now establish the completeness class containing those p-definable families which are "hardest to compute". A polynomial family P over F is complete over F if:

(i) P is p-definable

(ii) Every p-definable family Q is a p-projection of P.

Although it is not immediately apparent that complete families exist, we shall see in the next section that the completeness class is non-empty.

2.3 Some Key Results

Firstly, we introduce a useful, sufficient condition for a polynomial family to be p-definable.

Theorem 2.1 Suppose $P = \{P_1, P_2, \ldots\}$ is a family of polynomials over an arbitrary field F, in which every monomial of every member polynomial has coefficient 0 or 1. Suppose also that there is a polynomial time bounded Turing machine which can determine for any vector $\underline{v} \varepsilon \{0,1\}^i$ whether the coefficient of

$$\prod_{v_j=1} x_j$$

in P_i is 1. Then P is p-definable over F.

Proof Due to Valiant (see proposition 4 of [43].) □

This simple, yet powerful result is sufficient to demonstrate the p-definability of all the generating functions introduced in chapter 3. Each of the polynomial families considered there has easily computable 0-1 coefficients and the theorem may be directly applied. Theorem 2.1 cannot, however, be directly applied to the reliability polynomials considered in chapter 4. These polynomials are of the form

$$P_i(p_1,p_2,\ldots,p_i) = \sum_{\underline{v}\,\varepsilon\{0,1\}^i} \left[c(\underline{v}) \prod_{v_j=1} p_j \prod_{v_j=0} (1-p_j) \right]$$

where $c(\underline{v})$ an easily computable function mapping $\{0,1\}^i$ to $\{0,1\}$.

By theorem 2.1 the polynomial family

$$P'_{2i}(p_1,\ldots,p_i,q_1,\ldots,q_i) = \sum_{\underline{v}\,\varepsilon\{0,1\}^i} \left[c(\underline{v}) \prod_{v_j=1} p_j \prod_{v_j=0} q_j \right]$$

is p-definable, and hence, by the definition of p-definability, we are assured of the existence of a polynomial $Q'_{2i+k}(p_1,\ldots,p_i,q_1,\ldots,q_i,x_1,\ldots,x_k)$ with the properties

$$P'_{2i}(\underline{p},\underline{q})) = \sum_{\underline{x}\,\varepsilon\{0,1\}^k} Q'_{2i+k}(\underline{p},\underline{q},\underline{x})$$

and $|Q'_{2i+k}| \leqq t(i)$ for some polynomial $t(.)$.

Now

$$P_i(\underline{p}) = \sum_{\underline{x}\,\varepsilon\{0,1\}^k} Q'_{2i+k}(p_1,\ldots,p_i,(1-p_1),\ldots,(1-p_i),\underline{x})$$

$$= \sum_{\underline{x}\,\varepsilon\{0,1\}^k} Q_{i+k}(p_1,\ldots,p_i,\underline{x})$$

where $|Q_{i+k}| \leqq t(i)+i$. Hence, by definition, P is p-definable. When the polynomial families considered in chapters 3 and 4 are introduced, it should be immediately apparent, from the above discussion, that they are p-definable; no further comment will be made about this point.

Let us now turn our attention to the completeness class itself, i.e. the class of polynomial families which are complete over F. If $X_{n\times n} = \{x_{ij} \mid 1 \leqq i,j \leqq n\}$ is a matrix of indeterminates, then define the permanent function $\mathrm{per}_{n\times n}(X_{n\times n})$ (or just $\mathrm{per}(X_{n\times n})$ where no confusion arises) by

$$\mathrm{per}(X_{n\times n}) = \sum_{\pi\varepsilon S(n)} x_{1,\pi(1)}x_{2,\pi(2)}\cdots x_{n,\pi(n)}$$

where S(n) is the set of all permutations of the first n natural numbers. It will be seen that the permanent is similar to the more familiar determinant function, and differs from it formally only in the respect of the sign assigned to the coefficients of the monomials. The permanent is of great significance in combinatorial mathematics, and a comprehensive account of it is given by Minc [19]. The following theorem shows two things, firstly that the completeness class is non-empty, and secondly that the determinant and the permanent, despite definitional similarities, are apparently of widely separated computational complexities.

Theorem 2.2 The family $\{\mathrm{per}(X_{i\times i}) \mid i=1,2,\ldots\}$ is complete over any field not of characteristic 2.

Proof Due to Valiant (see theorem 2 of [43]). □

The practical importance of theorem 2.2 springs from the following considerations. Suppose P is a polynomial family which is known to be complete over F. It is immediate from the definition of p-projection that the relation "is a p-projection of" is a transitive one. It follows that, in order to demonstrate that another family Q is complete over F, it is sufficient to show that

(i) Q is p-definable

and (ii) P is a p-projection of Q.

Theorem 2.2, by explicitly exhibiting a complete polynomial, gives us a starting point from which we can prove other families complete.

We close this section by showing the completeness of the partial permanent function. The method serves as an illustration of the above technique, while the completeness result itself serves as a useful base for the reductions of chapters 3 and 4.

If $X_{n\times n}$ is as before, the partial permanent $\mathrm{per}^*_{n\times n}(X_{n\times n})$ is defined by

$$\mathrm{per}^*(X_{n\times n}) = \sum_{\pi\in S^*(n)} \prod_{i\in\mathrm{dom}(\pi)} x_{i,\pi(i)}$$

where $S^*(n)$ is the set of all injective (but not necessarily total) functions $\{1,2,\ldots,n\}\to\{1,2,\ldots,n\}$, and $\mathrm{dom}(\pi)$ is the domain of π. (The null product is taken to be 1; as a consequence, $\mathrm{per}^*(X_{n\times n})$ has a term 1 of degree 0.) Note that monomials of $\mathrm{per}^*(X_{n\times n})$ correspond to sets of indeterminates which are row-wise and column-wise disjoint in $X_{n\times n}$.

<u>Lemma 2.3</u> The family $\{\mathrm{per}^*(X_{i\times i}) \mid i=1,2,\ldots\}$ is complete over any field not of characteristic 2.

<u>Proof</u> Representing the n×n identity matrix by $I_{n\times n}$ and the n×n zero matrix by $O_{n\times n}$, the following is an explicit expression of the permanent function as a projection of the partial permanent:

$$\mathrm{per}(X_{n\times n}) = \mathrm{per}^*\left(\begin{array}{c|c} X_{n\times n} & -I_{n\times n} \\ \hline -I_{n\times n} & O_{n\times n}\end{array}\right).$$

By way of verification, we note that the right hand side of the identity is a linear combination of monomials of $\mathrm{per}^*(X_{n\times n})$ and, moreover, that each such monomial of degree d occurs with coefficient

$$\mathrm{per}^*\left(\begin{array}{c|c} -I_{(2n-2d)\times(2n-2d)} & O_{(2n-2d)\times d} \\ \hline O_{d\times(2n-2d)} & O_{d\times d} \end{array}\right)$$

This coefficient is clearly 1 if d=n, but otherwise is equal to

$$\binom{2n-2d}{0} + \binom{2n-2d}{1}(-1) + \binom{2n-2d}{2}(-1)^2 + \ldots + \binom{2n-2d}{2n-2d}(-1)^{2n-2d}$$

$$= \quad (1-1)^{2n-2d}$$

$$= \quad 0.$$

The result follows from the completeness of the permanent function over such fields (theorem 2.2). □

3. FIRST APPLICATION: TWO PROBLEMS FROM CRYSTAL PHYSICS

3.1 Enumeration Problems in Crystal Physics

The domain of crystal physics is one rich in combinatorial enumeration problems. A crystal lattice, consisting of a regular array of atoms and bonds joining them, is given, and we are asked to find the number of distinct figures which can be inscribed on the lattice and which satisfy a certain given condition. Such questions have been treated by many authors including Heilmann and Lieb [12], Kasteleyn [14], Montroll [21] and Percus [26]. Two problems of the above type are presented here and analysed using the methods introduced in chapter 2.

Our first example is motivated independently by two distinct physical models. A two-dimensional version of the problem arises in the mathematical treatment of the properties of a system of diatomic molecules, or dimers, which are adsorbed on the surface of a crystal. The dimers are attracted preferentially to pairs of adjacent lattice sites which they then occupy. The thermodynamic properties of the system are to some extent determined by the number of ways in which the dimers can be arranged on the crystal without overlap. The dimer problem is the enumeration problem which arises if we insist that all lattice sites be occupied, while the monomer-dimer problem is concerned with counting the arrangements which may occur if we allow vacant sites or monomers. An analogous three-dimensional version of the problem arises in the theories of binary mixtures and cell-clusters.

The second example is concerned with the "Ising model" of a crystalline system. In this model, each atom of a crystal can be in one of two states ; adjacent atoms which are in different states contribute a fixed amount of energy to the system whereas those in similar states contribute an amount which is equal but opposite in sign. It can be shown, see Kasteleyn [14], that computing the thermal properties of such a system is equivalent to an enumeration problem of the type which we are considering.

3.2 Graphs and Lattices

In this and the next chapter we shall be drawing on several concepts from graph theory. Here, for completeness, we include some basic graphical definitions; others will be introduced as and when required. It is intended that the terminology used should, for the most part, be consistent with that of Berge [3].

A graph G is specified by a pair (V,E), where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of vertices and E a set of edges. For a directed graph the set E is composed of ordered pairs of vertices, i.e. $E \subseteq \{(u,v) \mid u,v \in V\}$, and for an undirected graph E is a set of unordered pairs, i.e. $E \subseteq \{\{u,v\} \mid u,v \in V\}$. An edge (u,v) of a directed graph has endpoints u and v, and is said to be incident out of u and incident into v. The number of edges incident out of u is the outdegree of u; the number of edges incident into v is the indegree of v. An edge {u,v} of an undirected graph has similar endpoints and is said to be incident at u and v. The degree of u is the number of edges incident at u. The degree of an undirected graph is equal to that of a vertex of maximal degree in the graph. The order of a graph G is the cardinality of its vertex set V; the size of G, denoted by $|G|$, is the cardinality of the edge set E. The term node is used as a synonym for vertex.

It will prove convenient to supply additional structure for our graphs. A labelled graph is a graph G together with a mapping $\lambda: E \to \Lambda$ which takes edges of G into some label set Λ. The label sets we shall use will be of the form $\Lambda = F \cup X$ where F is a field and X is a set of indeterminates over F. We shall henceforth assume that all graphs are labelled graphs.

There are two graphs which we shall have cause to refer to frequently. The <u>complete graph</u>, K_n, <u>on n nodes</u> is defined in the undirected case by the triple

$$V = \{v_1, \ldots v_n\}$$

$$E = \{\{v_i, v_j\} \mid 1 \leqq i < j \leqq n\}$$

$$\lambda\colon E \to X, \quad \{v_i, v_j\} \mapsto x_{ij} \qquad (i \leqq j)$$

and in the directed case by

$$V = \{v_1, \ldots, v_n\}$$

$$E = \{(v_i, v_j) \mid 1 \leqq i, j \leqq n\}.$$

$$\lambda\colon E \to X, \quad (v_i, v_j) \mapsto x_{ij}$$

Here $X = \{x_{ij} \mid 1 \leqq i, j \leqq n\}$ is a set of indeterminates. The (undirected) <u>complete bipartite graph</u>, $K_{n,n}$, <u>on 2n nodes</u> is defined by the triple

$$V = \{u_1, \ldots, u_n, v_1, \ldots, v_n\}$$

$$E = \{\{u_i, v_j\} \mid 1 \leqq i, j \leqq n\}$$

$$\lambda\colon E \to X, \quad \{u_i, v_j\} \mapsto x_{ij}.$$

We shall, for the moment, restrict our attention to undirected graphs. Two methods of deriving smaller graphs from larger will interest us. If $G = (V,E)$ is a graph then $G' = (V',E')$ is said to be a <u>subgraph</u> of G iff $V' \subseteq V$ and $E' = \{\{u,v\} \in E \mid u,v \in V'\}$. (The subgraph G' is said to be <u>induced</u> by the set of vertices V'.) In addition, if G is as before, and E' is <u>any</u> subset of E, then $G' = (V,E')$ is said to be a <u>partial graph</u> of G.

For the purposes of the current chapter, in which we are concerned with crystal lattices, we shall need to consider certain graphs with a regular structure. Two such will be introduced here. R_n will denote the rectangular lattice graph whose n^2 vertices are arranged in two-dimensional Euclidean space according to the Cartesian coordinates $\{(i,j) \mid 0 \leqq i,j \leqq n-1\}$, and whose edges consist of pairs of nodes which are separated by unit distance. (It will be assumed that the edges are labelled with distinct indeterminates over some field.) A three-dimensional variant of the above, the cubic lattice graph C_n, has $2n^2$ vertices placed according to the coordinates $\{(i,j,k) \mid 0 \leqq i,j \leqq n-1, k=0,1\}$. Again, edges consist of pairs of nodes which are separated by unit distance.

3.3 Generating Functions and Polynomial Families

Suppose that S is a function which maps an arbitrary graph $G = (V,E)$ onto a subset of 2^E, for example $S(G)$ might be the set of all perfect matchings of G. Recall that the graphs we are considering are labelled and suppose that the label set is $\Lambda = F \cup X$ where F is a field and $X = \{x_1,\ldots,x_n\}$ is a set of indeterminates over F. Denote by M the set of all monomials in the indeterminates X, i.e. $M = \{x_1^{r_1} \ldots x_n^{r_n} \mid r_1,\ldots,r_n \in \mathbb{N}\}$. Then the labelling λ on E extends to a function on the subsets $A \subseteqq E$ in the following way:

$$\lambda : 2^E \to F \times M, \qquad A \mapsto \prod_{e \in A} \lambda(e).$$

(The product used here is the multiplication in the polynomial ring.) The pair (S,G) specifies an enumeration problem, namely evaluating the cardinality of S(G), and defines a corresponding <u>generating function</u> (g.f.):

$$GF(G,S) = \sum_{A \in S(G)} \lambda(A) \qquad (3.1)$$

(The sum used is the addition of the polynomial ring.) Note that if λ maps each edge of G onto a distinct indeterminate, then monomials of (3.1) correspond in a natural way to the objects we wish to enumerate.

In order to construct, from the defining function S of the enumeration problem, a polynomial family of the type considered in chapter 2, we need only specify a family of graphs; the generating functions for these graphs will comprise the polynomial family. As an example, the family of graphs $\{K_n \mid n = 1,2,\ldots\}$ generates the polynomial family $\{GF(K_n,S) \mid n = 1,2,\ldots\}$. This family is "universal" for the enumeration problem in the sense that the g.f. for an arbitrary graph of order n can be obtained from the n^{th} element of the family by a projection which maps certain of the indeterminates x_{ij} onto 0 while leaving the others fixed. Two polynomial families which we will be considering are the following:

$$\{GF(R_n,S) \mid n = 1,2,\ldots\}$$

$$\{GF(C_n,S) \mid n = 1,2,\ldots\}.$$

It will be recalled that R_n and C_n are the rectangular and cubic lattice graphs defined previously.

The generating function described above may be evaluated for 0,1 assignments to its indeterminates in order to yield solutions to the corresponding enumeration problem. In addition, a component of certain degree can be extracted from the polynomial in order to yield the number of structures of given size which exist in the lattice. The generating function also allows us to assign a weighting to each lattice edge corresponding to some physical quantity which is not uniform over all bonds. Kasteleyn [14] lists several reasons why enumeration problems on lattice graphs should be attacked via the corresponding generating function, and the empirical evidence to support this view is very strong:

generating functions consistute the only known method for solving those non-trivial lattice problems which are known to be tractable. We therefore argue that showing a g.f. to be complete for some family of lattices is good evidence that the corresponding enumeration problem on such lattices is intractable.

3.4 The Monomer-Dimer Problem

In order to discuss dimer arrangement problems we need to introduce the graph theoretic notion of matching. A partial matching of a graph $G = (V,E)$ is a subset M of E with the property that no pair of elements of M are incident at a common node. A vertex $v \in V$ is said to be saturated by M if there exists an edge in M which is incident at v. A perfect matching of G is a partial matching of G which saturates all the vertices in V. The monomer-dimer problem is that of enumerating partial matchings in a lattice graph, while the dimer problem is concerned with the enumeration of perfect matchings.

Let S_{MD} be the function which maps an arbitrary finite graph onto the set of all partial matchings of the graph. We may write down the generating function for monomer-dimer arrangements on a graph G as

$$MD(G) = GF(G,S_{MD}) \qquad (3.2)$$

where GF is defined as in (3.1). Likewise, if S_{DI} is the function which maps a graph onto the set of all its perfect matchings, then the g.f. for dimer arrangements may be written

$$DI(G) = GF(G,S_{DI}) \qquad (3.3)$$

In the case of the rectangular and cubic lattice graphs introduced earlier, in section 3.2, these g.f. yield the following interesting polynomial families:

$$\{DI(R_n) \mid n = 1,2,..\} \tag{3.4}$$

$$\{DI(C_n) \mid n = 1,2,..\} \tag{3.5}$$

$$\{MD(R_n) \mid n = 1,2,..\} \tag{3.6}$$

$$\{MD(C_n) \mid n = 1,2,..\} \tag{3.7}$$

Kasteleyn [14] shows how the g.f. for dimer coverings of a planar graph (for a definition of planar see Berge [3]) can be expressed as the square root of a determinant of size equal to the order, k, of the graph; such a determinant can be evaluated in $O(k^{2.53})$ operations using the matrix multiplication method of Schonhage [33] coupled with the LUP matrix decomposition algorithm described in Aho et al. ([1] p. 235). Each member of the family (3.4) can therefore be evaluated using $O(n^{5.05})$ arithmetic operations.

It is interesting to observe that when we pass from 2-dimensional to 3-dimensional lattices the g.f. becomes apparently much more difficult to evaluate. The essence of this phenomenon is captured by a result of Valiant [43] which asserts that the family (3.5) is complete over any field not of characteristic 2. It should be noted that using even the merest degree of freedom which 3-space allows (the cubic lattice C_n has only unit thickness) enables us to convert a computationally tractable problem into an intractable one. A second observation is that testing for the existence of a perfect matching in an arbitrary, possibly non-planar, graph of order k, can be achieved in time $O(k^3)$ using a method of Edmonds, which is described in Lawler [16] p. 233. This is a typical example of the now well-documented gap which can exist between the complexity of an existence problem and its corresponding enumeration problem. (See Valiant [41].)

It is the aim of the following section to show that the family

(3.6), and hence (3.7), is complete in the sense of chapter 2. The consequence of this result is that it is unlikely that enumerating monomer-dimer arrangements on a rectangular lattice is computationally feasible. It will afterwards be suggested that the exact nature of the lattice is immaterial to the result and that the generating function remains complete for a variety of other lattice graphs. This result stands in stark contrast to the tractability of the planar case of the dimer enumeration problem.

3.5 The Completeness of the Family $\{MD(R_n)\}$

Our starting point in this section is the family $\{MD(K_{n,n})\}$ of g.f.'s for monomer-dimer arrangements on complete bipartite graphs, the completeness of which is almost immediate. The bipartite graph will, for $n \geqq 3$, be non-planar, but we will provide a sequence of transformations of the bipartite graph which result in a planar graph of order $O(n^4)$. These transformations have the property that they preserve the monomer-dimer g.f. of the graph. In this way, we will have constructed a family of planar graphs, the monomer-dimer g.f.'s of which form a complete family. Finally a space-efficient technique is used to embed the members of this family of planar graphs, in instances R_k of the rectangular lattice graph.

Lemma 3.1 The family $\{MD(K_{n,n})\}$ is complete over any field not of characteristic 2.

Proof The monomials of $MD(K_{n,n})$ (see section 3.2) may be characterised as products of elements of the matrix $X = \{x_{ij} \mid 1 \leqq i,j \leqq n\}$ in which pairs of elements in the product are row-wise and column-wise disjoint. There is thus a 1-1 correspondence between monomials of the above polynomial and those of $\mathrm{per}^*(X)$. We deduce that $MD(K_{n,n}) = \mathrm{per}^*(X)$, and the result follows from Lemma 2.3. □

The transformations we shall apply to the complete bipartite graph to yield a planar graph are of the following form. Choose a subset of the vertices of the graph, excise the subgraph induced by those vertices, and insert some prescribed replacement subgraph. In order to make precise the notion that a transformation of a graph preserves the monomer-dimer g.f. of that graph, we require some new definitions.

We introduce a restricted form of the monomer-dimer g.f.. Suppose $H = (V,E)$ is a graph and $U \subseteq U_o \subseteq V$. Define

$$MD^*(H,U_o,U) = \sum_{A \varepsilon S(H,U_o,U)} \lambda(A) \qquad (3.8)$$

where $S(H,U_o,U)$ is the set of all partial matchings on H which saturate all the vertices in U but none of those in (U_o-U). We remark in passing that

$$MD(H) = \sum_{U \subseteq U_o} MD^*(H,U_o,U), \quad \textit{for any fixed } U_o.$$

Suppose that H is as before ; a <u>simulation</u> S of H consists of a graph $H_S = (V_S, E_S)$ together with an injective map $i_S: V \rightarrow V_S$. (Note that we allow the order of H_S to be strictly greater than that of H.) We say that a simulation S of H is <u>faithful</u> iff, for all $U \subseteq V$,

$$MD^*(H_S, i_S(V), i_S(U)) = MD^*(H,V,U).$$

It should be clear that the definition of faithful simulation captures the notion of generating function preserving transformation which we require. For if G is a graph which contains H as a subgraph, we may excise H from G and replace it by H_S (identifying the nodes according to the injective map i_S) without affecting the monomer-dimer g.f. on G. Three simulations, which are claimed to be faithful, are now presented. The first

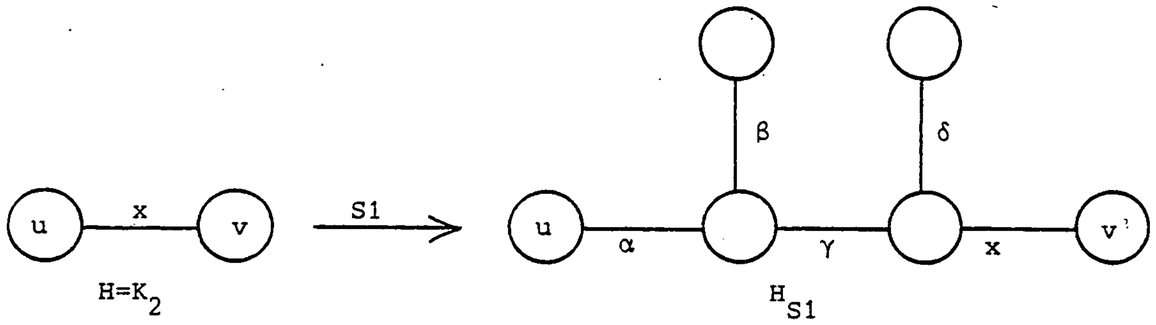

Figure 3.1

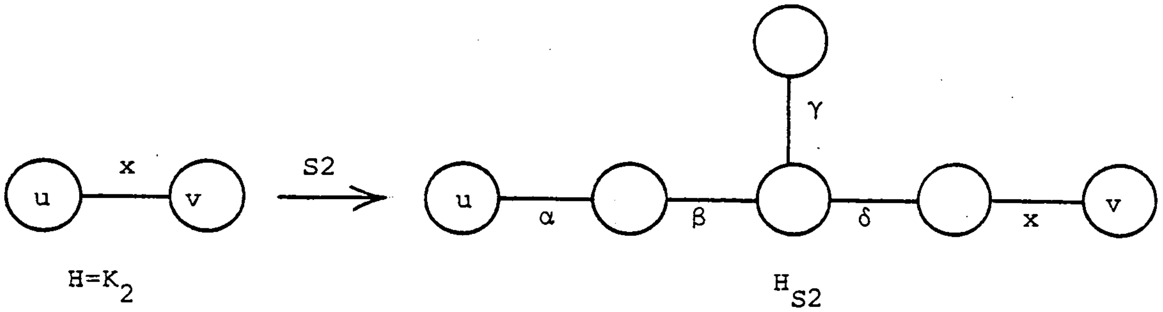

Figure 3.2

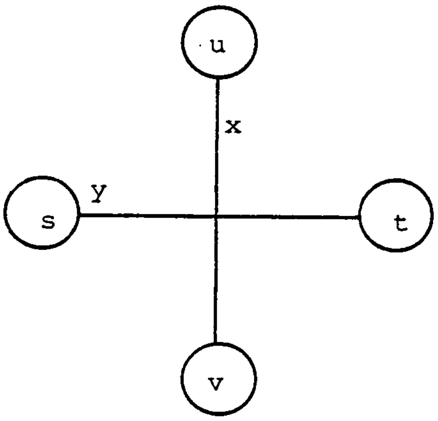

Figure 3.3

two show between them that a single edge may be simulated by a "chain" of edges of arbitrary length.

(S1) <u>Simulation of K_2</u>: The simulation is described in fig. 3.1. Here $\alpha = \gamma = 1$, $\beta = \delta = -1$ and x is an indeterminate. Note that the nodes labelled u and v in H_{S1} are really $i_{S1}(u)$ and $i_{S1}(v)$; however, no confusion should arise from this abuse of notation. The following four observations establish that the simulation is faithful. (For brevity, we let $f(U) = MD^*(H_{S1}, \{u,v\}, U)$ and $g(U) = MD^*(H, \{u,v\}, U)$)

$$\begin{aligned}
f(\emptyset) &= 1 + \beta + \gamma + \delta + \beta\delta = 1 = g(\emptyset)\\
f(\{u\}) &= \alpha + \alpha\delta \quad = 0 = g(\{u\})\\
f(\{v\}) &= x + \beta x \quad = 0 = g(\{v\})\\
f(\{u,v\}) &= \alpha x \quad = x = g(\{u,v\}).
\end{aligned}$$

(S2) <u>Simulation of K_2</u>: The simulation is described in fig. 3.2; $\alpha = -1$, $\beta = \delta = 1$ and $\gamma = -2$. The four cases are listed as before. (For brevity, we let $f(U) = MD^*(H_{S2}, \{u,v\}, U)$ and $g(U) = MD^*(H, \{u,v\}, U)$)

$$\begin{aligned}
f(\emptyset) &= 1 + \beta + \gamma + \delta \quad = 1 = g(\emptyset)\\
f(\{u\}) &= \alpha + \alpha\delta + \alpha\gamma \quad = 0 = g(\{u\})\\
f(\{v\}) &= x + \beta x + \gamma x \quad = 0 = g(\{v\})\\
f(\{u,v\}) &= \alpha x + \alpha\gamma x \quad = x = g(\{u,v\}).
\end{aligned}$$

(S3) <u>Simulation of a crossover</u>: Our aim is to construct a planar graph which simulates the crossover of fig. 3.3. We first remark that it is sufficient to treat the special case when $x,y = 1$. To see this we note that the edges $\{s,t\}$ and $\{u,v\}$ may be expanded using simulation S1 (see fig. 3.1), and the crossover arranged to take place on the $\gamma(=1)$ weighted edges.

We construct the simulating graph stepwise from simpler components. First consider the graph F1, with distinguished

F1:

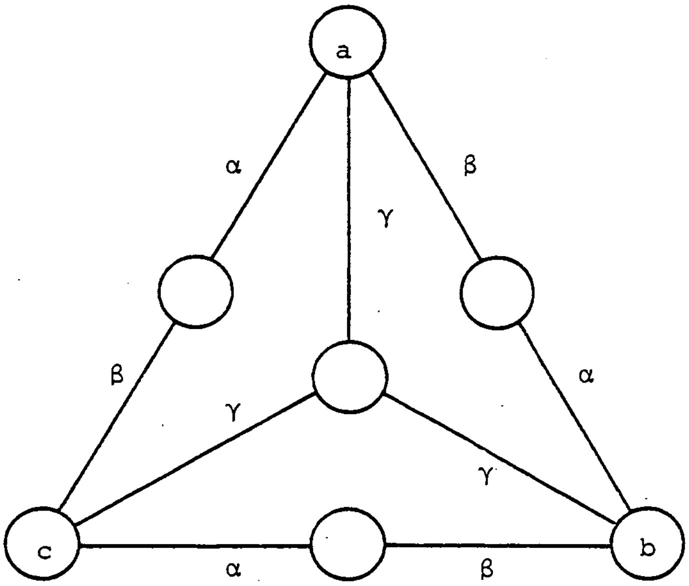

Figure 3.4

F2:

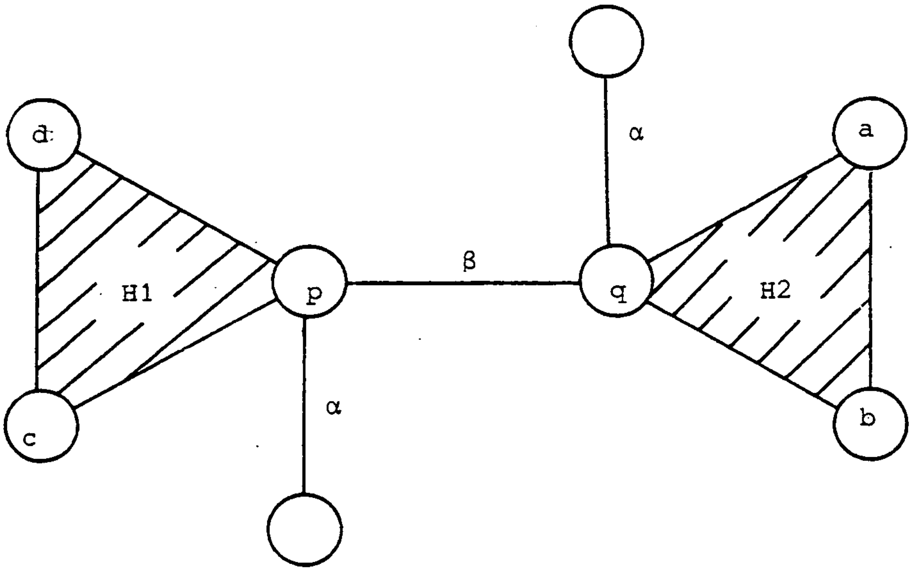

Figure 3.5

nodes a,b,c, which is described by fig. 3.4. Here α and β are the distinct roots of the equation $x^2 + \gamma x - \gamma^2 = 0$, and $\gamma = 2^{-1/6}$. (Note that α and β are real, and have the properties $\alpha + \beta = -\gamma$, $\alpha\beta = -\gamma^2$.) The properties of F1 may be summarised as follows; as before we employ the succinct notation

$f_1(A) = MD^*(F1, \{a,b,c\}, A)$:

$$
\begin{aligned}
f_1(\emptyset) &= 1 \\
f_1(\{a\}) &= f_1(\{b\}) = f_1(\{c\}) \\
&= \alpha + \beta + \gamma \\
&= -\gamma + \gamma \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
f_1(\{a,b\}) &= f_1(\{b,c\}) = f_1(\{a,c\}) \\
&= \alpha^2 + \beta^2 + \alpha\beta + 2\alpha\gamma + 2\beta\gamma \\
&= (\alpha + \beta)^2 - \alpha\beta + 2\gamma\ (\alpha + \beta) \\
&= \gamma^2 + \gamma^2 - 2\gamma^2 \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
f_1(\{a,b,c\}) &= \alpha^3 + \beta^3 + 3\alpha^2\gamma + 3\beta^2\gamma + 3\alpha\beta\gamma \\
&= (\alpha + \beta)(\alpha^2 - \alpha\beta + \beta^2) + 3\gamma(\alpha^2 + \beta^2 + \alpha\beta) \\
&= (\alpha + \beta)[(\alpha + \beta)^2 - 3\alpha\beta] + 3\gamma[(\alpha + \beta)^2 - \alpha\beta] \\
&= -\gamma(\gamma^2 + 3\gamma^2) + 3\gamma(\gamma^2 + \gamma^2) \\
&= 2\gamma^3 \\
&\doteq \sqrt{2}
\end{aligned}
$$

The second stage of the construction combines two copies of F1 into a single graph F2 as described in fig. 3.5. The subgraph H1 is a copy of F1 with vertices a,b relabelled d,p, and H2 is another copy with vertex c relabelled q. The scalars α and β are set to -1 and 2 respectively. We use the following abbreviated forms:

$$f_2(A) = MD^*(F2, \{a,b,c,d\}, A) \quad (A \subseteqq \{a,b,c,d\})$$

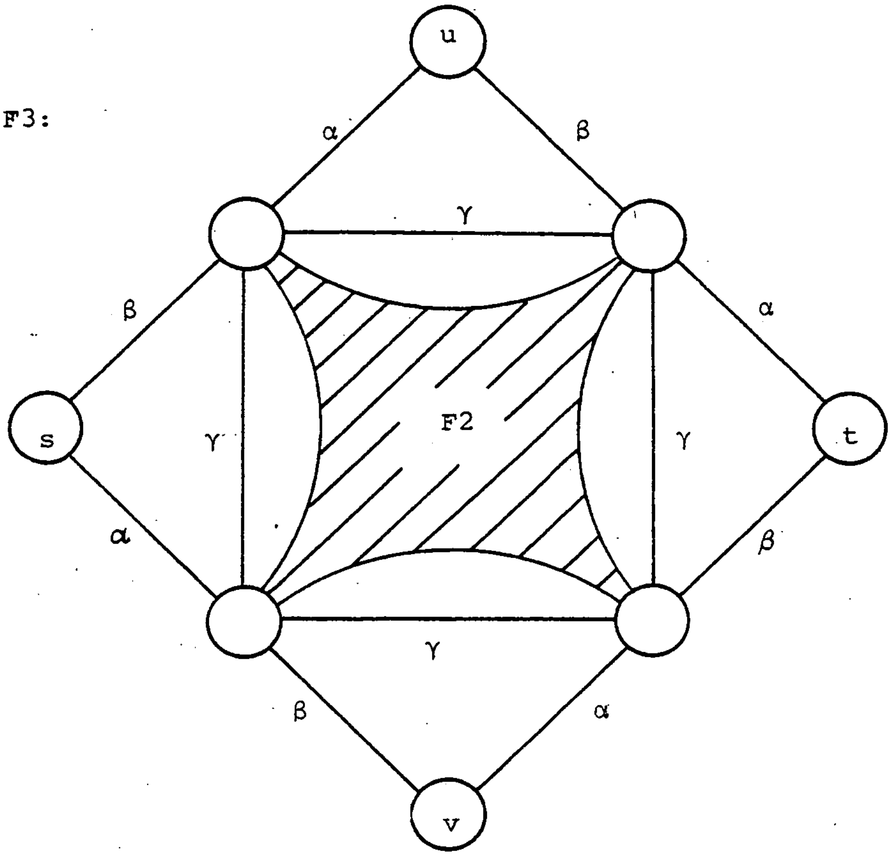

Figure 3.6

$$h_1(A) = MD^*(H1, \{c,d,p\}, A) \qquad (A \subsetneq \{c,d,p\})$$

$$h_2(A) = MD^*(H2, \{a,b,q\}, A) \qquad (A \subsetneq \{a,b,q\}).$$

Only for two values of its argument does f assume a non-zero value, these being

$$\begin{aligned} f_2(\emptyset) &= h_1(\emptyset)h_2(\emptyset)\ [1 + \beta + 2\alpha + \alpha^2] \\ &= 1.1.2 \\ &= 2, \end{aligned}$$

and

$$\begin{aligned} f_2(\{a,b,c,d\}) &= h_1(\{c,d,p\})h_2(\{a,b,q\})\ .1 \\ &= \sqrt{2}.\sqrt{2}.1 \\ &= 2 \end{aligned}$$

For all other arguments, f_2 is zero, a representative example being

$$\begin{aligned} f_2(\{a,b\}) &= h_1(\emptyset)h_2(\{a,b,q\})\ [1 + \alpha] \\ &= 1.\sqrt{2}.0 \\ &= 0 \end{aligned}$$

The function f_2 encapsulates the properties of the graph F2 which are relevant for the verification of the final stage of the construction. In this, F2 is augmented to a graph F3, described by fig. 3.6, which has the properties required of a faithful simulation of the crossover of fig. 3.3. The assignments to the scalars are $\alpha = 1/\sqrt{2}$, $\beta = -1/\sqrt{2}$, $\gamma = -1/2$. and we make the usual style of abbreviation:

$$f_3(U) = MD^*(F3, \{s,t,u,v\}, U) \qquad (U \subsetneq \{s,t,u,v\}).$$

The verification of F3 as a faithful simulation is purely mechanical:

$$\begin{aligned} f_3(\emptyset) &= f_2(\emptyset)\ [1 + 4\gamma + 2\gamma^2] + f_2(\{a,b,c,d\}).1 \\ &= 2(-1/2) + 2.1 \\ &= 1 \end{aligned}$$

$$\begin{aligned} f_3(\{s\}) &= \ldots = f_3(\{v\}) \\ &= f_2(\emptyset)\ [\alpha + \beta]\ [1 + 2\gamma] \\ &= 2.0.0 \\ &= 0. \end{aligned}$$

$$
\begin{aligned}
f_3(\{u,t\}) &= \ldots = f_3(\{s,u\}) \\
&= f_2(\emptyset)\ [\alpha^2 + \alpha^2\gamma + \beta^2 + \beta^2\gamma + \alpha\beta] \\
&= 2\ [1/2 - 1/4 + 1/2 - 1/4 - 1/2] \\
&= 0 \\
f_3(\{s,t\}) &= f_3(\{u,v\}) \\
&= f_2(\emptyset)\ [\alpha^2 + \beta^2 + 2\alpha\beta + 2\alpha\beta\gamma] \\
&= 2\ [1/2 + 1/2 - 1 + 1/2] \\
&= 1 \\
f_3(\{s,u,t\}) &= \ldots = f_3(\{v,s,u\}) \\
&= f_2(\emptyset)\ [\alpha^3 + \beta^3 + \alpha^2\beta + \alpha\beta^2] \\
&= 2(\alpha + \beta)(\alpha^2 + \beta^2) \\
&= 0 \\
f_3(\{s,t,u,v\}) &= f_2(\emptyset)\ [\alpha^4 + \beta^4] \\
&= 2\ [1/4 + 1/4] \\
&= 1.
\end{aligned}
$$

We thus see that F3 has exactly the properties required for a faithful simulation. This completes the construction.

We are now in a position to prove a preliminary theorem which will lead to the main result of the section.

<u>Theorem 3.2</u> There exists a family of graphs $\{G_n\}$, with the following properties:

(i) Each G_n is planar,

(ii) $|G_n| = o(n^4)$,

(iii) $\{MD(G_n)\}$ is a complete family over the real field $\mathbb{R}$.

<u>Proof</u> Our starting point is the complete bipartite graph $K_{n,n}$; we know from Lemma 3.1 that $MD(K_{n,n})$ is complete over $\mathbb{R}$. It will be shown that, using the simulations which we have presented, $K_{n,n}$ may be transformed into a planar graph G_n, whose size is $O(n^4)$. Since the simulations have the property that they

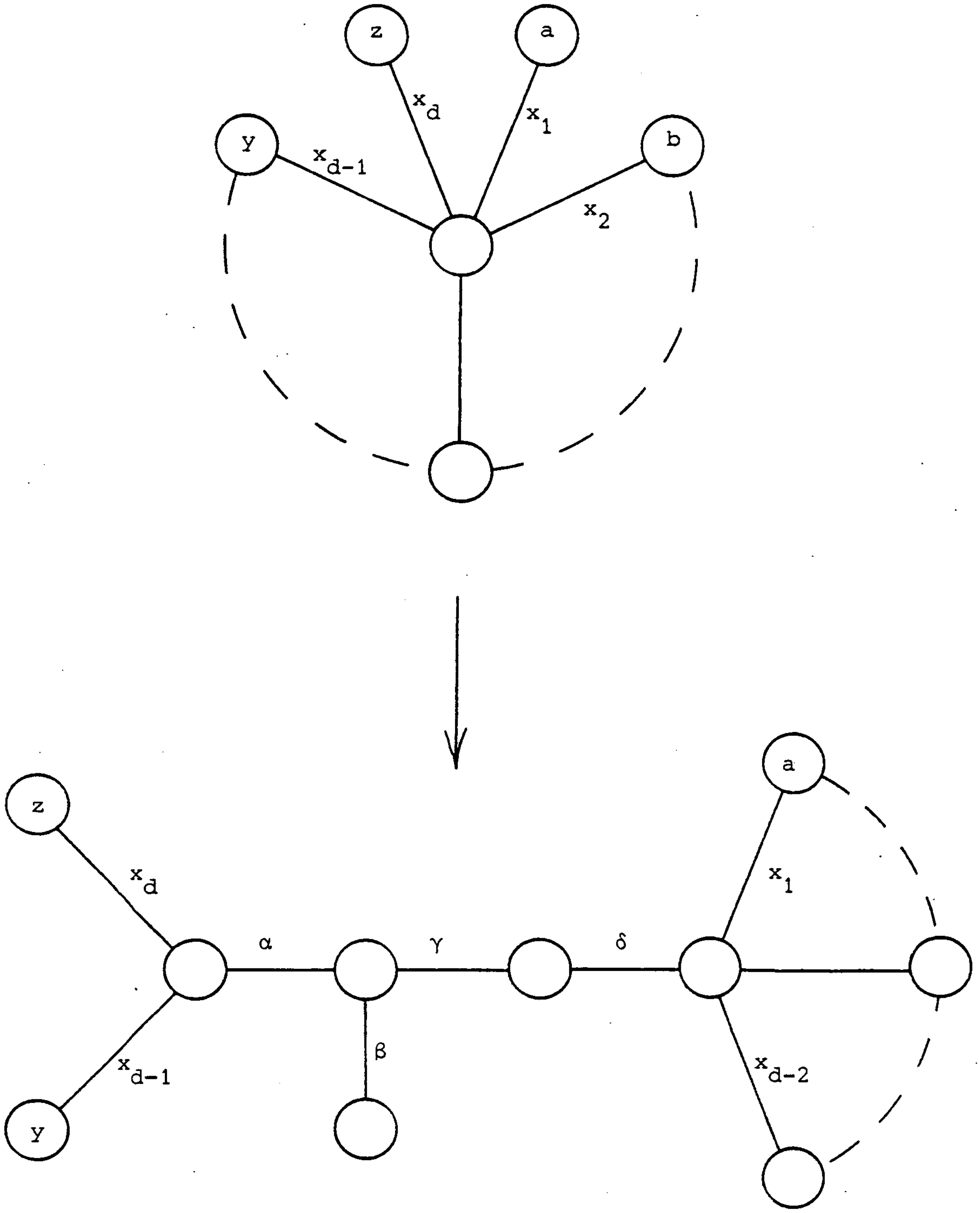

Figure 3.7

are faithful, $MD(G_n) = MD(K_{n,n})$ and the result follows.

The construction of G_n proceeds as follows. Each edge of $K_{n,n}$ is expanded into a chain of n^2 edges by recursively applying the simulation S1. As there are only n^2 edges in $K_{n,n}$ we may arrange that each crossover in the transformed graph takes place on a separate pair of edges. Each crossover is then transformed using simulation S3 into a planar subgraph; the resulting graph is G_n. □

We now arrive at the main theorem.

Theorem 3.3 The family $\{MD(R_n)\}$ is complete over $\mathbb{R}$.

Proof The proof proceeds in two steps. The planar graph G_n of the previous theorem is transformed by the "unfolding" of its vertices into a graph of degree 3. The resulting degree three graph is then efficiently embedded in R_n using a known result on planar embeddings. The result will follow from the observation that both of these steps preserve the monomer-dimer g.f..

The first step, of unfolding vertices of degree greater than 3, is illustrated in fig. 3.7. Here we consider the unfolding of a vertex v, of degree d, with incident edges labelled by indeterminates $x_1,\ldots,x_d$, into 4 vertices of degree less than or equal to 3 and one of degree d-1. Clearly the unfolding may be repeated until all vertices have degree less than of equal to 3. The values of the scalars used in the construction are $\alpha = 1$, $\beta = -2$, $\gamma = 1$ and $\delta = -1$.

Suppose that the graph of which v is a vertex is H and that the graph resulting from a single unfolding of the type shown in fig. 3.7 is H'. Denote by E_0 the original edge set of H, and by E_1 the set of added edges, i.e. those labelled α, β, γ, δ. We wish to show that $MD(H') = MD(H)$. Now by definition,

$$MD(H') = \sum_{A_0 \subseteq E_0} \sum_{A_1 \subseteq E_1} \lambda(A_0)\ \lambda(A_1)\chi_{H'}(A_0 \cup A_1)$$

where $\chi_H(A)$ is 1 if A is a partial matching of H, and 0 otherwise. Separating the two sums we obtain:

$$MD(H') = \sum_{A_0 \subseteq E_0} \lambda(A_0)f(A_0)$$

where $$f(A_0) = \sum_{A_1 \subseteq E_1} \lambda(A_0)\chi_{H'}(A_0 \cup A_1)$$

Our claim is that $f(A_0) = \chi_H(A_0)$, from which we may deduce the required identity $MD(H') = MD(H)$.

The claim may be justified by a direct calculation consisting of two parts.

(a) Suppose that $\chi_H(A_0) = 0$. Then there exists a pair of elements of A_0 which are incident at a common vertex of H. If that common vertex is other than v, then $\chi_{H'}(A_0 \cup A_1) = 0$ for any choice of A_1 and hence $f(A_0) = 0$. We may suppose, therefore, that the common vertex is v and, moreover, that the two edges incident at v in H are incident at distinct vertices of H'; suppose w.l.o.g. that the pair of edges in question are the x_1 and x_d labelled edges. Then

$$f(A_0) = \chi_{H'}(A_0)[1 + \beta + \gamma] = 0.$$

Thus we deduce that $(\chi_H(A_0) = 0 \Rightarrow f(A_0) = 0)$.

(b) Now assume, to the contrary, that $\chi_H(A_0) = 1$. Then each pair of edges in A_0 is vertex disjoint and thus at most one of the $x_1,\ldots,x_d$ labelled edges is a member of A_0. There are three cases to consider:

(i) No edge in A_0 is incident at v, in which case

$$f(A_0) = [1 + \alpha + \beta + \gamma + \delta + \alpha\delta + \beta\delta] = 1$$

(ii) Either the x_{d-1} or the x_d labelled edge is incident at v, in which case

$$f(A_0) = [1 + \beta + \gamma + \delta + \beta\delta] = 1$$

(iii) One of the $x_1, \ldots x_{d-2}$ labelled edges is incident at v, in which case

$f(A_0) = [1 + \alpha + \beta + \gamma] = 1$

Taken together, these three cases yield that ($\chi_H(A_0) = 1 \Rightarrow f(A_0) = 1$). From (a) and (b) we deduce that $\chi_H(A_0) = f(A_0)$, which was our claim.

By unfolding all the vertices of G_n, in the manner described above, we obtain an equivalent degree 3 graph which we shall denote by $G_n^{(3)}$. We remark that $|G_n^{(3)}| < 9 \cdot |G_n|$. For the second step, that of embedding the resulting degree three graph in the rectangular lattice graph, we employ a variant of the method of Valiant [44].

As $G_n^{(3)}$ is planar, it has a planar realisation. From such a realisation we may construct a sequence of vertices by repeating the following procedure until all the vertices of $G_n^{(3)}$ are included in the sequence: Choose a vertex on the outer boundary of the planar realisation, add it to the sequence, and delete all edges which are incident at that vertex. Clearly, the planar realisation of $G_n^{(3)}$ may be reconstructed starting with a single vertex, the last in the above sequence, and adding vertices, one by one, according to the reverse of the sequence. Together with each new vertex are added all the edges incident at that vertex and with some other vertex previously placed. The construction of the sequence allows us to arrange that each edge added lies within the exterior of the perimeter constructed so far.

The planar graph $G_n^{(3)}$ may be embedded in a rectangular lattice by a recursive method using this strategy. Suppose that the first i vertices of $G_n^{(3)}$ have been embedded in the rectangular grid R_{6i}. The method by which we embed the vertex

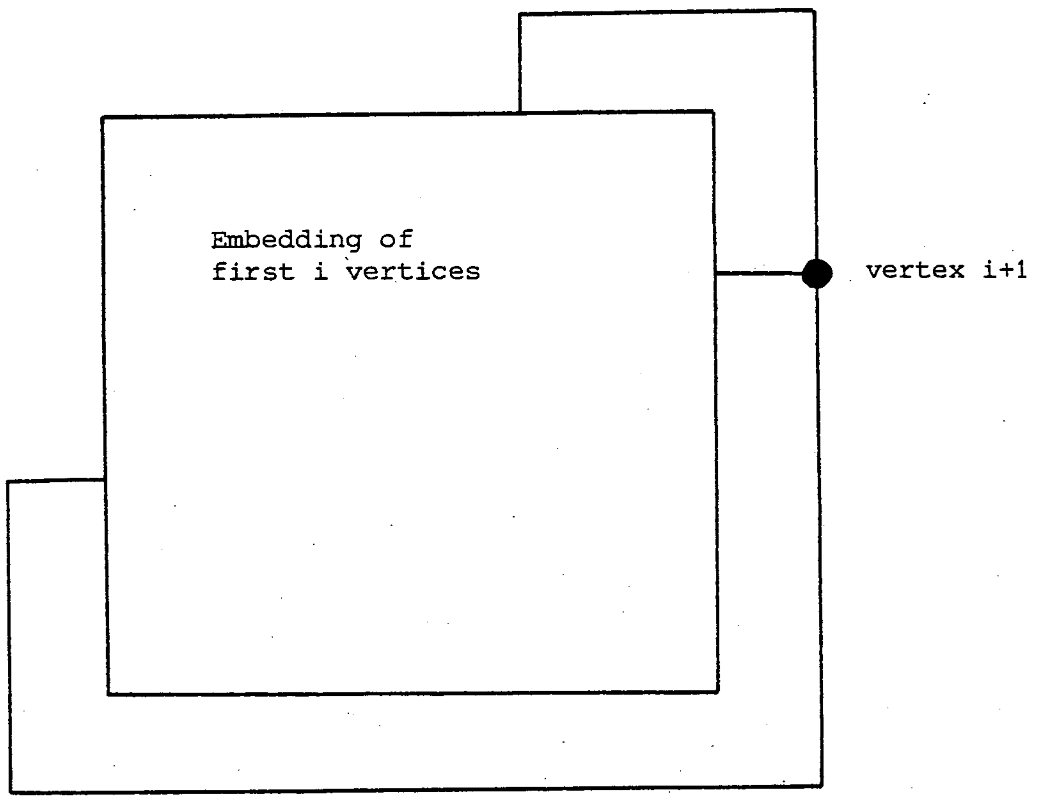

Figure 3.8

i+1 and its incident edges is described in fig. 3.8. The lines emanating from the newly added vertex in this figure correspond to edges incident at vertex i+1 in $G_n^{(3)}$ and are obtained by recursively applying the simulations S1 and S2 to the corresponding single edge in $G_n^{(3)}$. We remark that the reason why two different simulations are required is one of parity. Using only S1 we may construct only chains of odd length, while S1 and S2 together may be used to construct a chain of any length greater or equal to 3. From this consideration we see that the chains of edges in fig. 3.8 need never be distant more than 3 from the perimeter of the embedding of the first i vertices. Hence the first i+1 vertices may be embedded in $R_{6(i+1)}$. The fact that the embedding preserves the monomer-dimer g.f. is immediate from the observation that S1 and S2 are faithful. □

3.6 Some Extensions and Observations

It will perhaps have been noticed that the construction of the planar embedding of the last section does not rely heavily on the particular structure of the rectangular lattice and that the result should hold if R_n is replaced by a number of other lattice graphs.

Two possible families of lattice graphs we might consider are the hexagonal, whose n^{th} member has a vertex set defined by the Cartesian coordinates

$$\{(\sqrt{3},0)v_1+(\sqrt{3}/2,3/2)v_2+(\sqrt{3}/2,1/2)v_3 \mid 0 \leqq v_1,v_2 \leqq n;\ v_3=0,1\}$$

and the triangular with vertex set

$$\{(1,0)v_1+(1/2,\sqrt{3}/2)v_2 \mid 0 \leqq v_1,v_2 \leqq n\}.$$

In each case edges are considered to exist between pairs of vertices which are distant 1 apart. Without going into detail, it should be clear that the construction given in fig. 3.8 may be modified to yield an embedding in either of these lattices. We therefore deduce analogues of theorem 3.3 which declare that the monomer-dimer g.f.'s for these two lattice graphs are complete.

Another immediate corollary is the following: rather than enumerating all monomer-dimer arrangements on R_n (say) we may wish to enumerate those arrangements which have a certain number μ of monomers (i.e. we insist on a fixed monomer density). The g.f. for such an arrangement is simply a component of $MD(R_n)$ of specified degree. In the case $\mu = 0$, the g.f. is exactly $DI(R_n)$ which is efficiently computable as we have already remarked. However an efficient computational procedure for evaluating the g.f. for arbitrary values of μ would imply the existence of an efficient procedure for computing $MD(R_n)$. (We would merely sum over all values of μ.) It is therefore unlikely that counting monomer-dimer arrangements with specific monomer density is computationally feasible.

3.7 The Ising Problem

The second of two examples drawn from crystal physics is the so called Ising problem. Suppose that we have a crystal lattice in which each atom can be in one of two states. The state of an atom at vertex v_i of the lattice is described by a variable σ_i which can assume values from $\{-1,1\}$. Two adjacent atoms v_i, v_j contribute an "interaction energy" $- J_{ij}\sigma_i\sigma_j$ to the system, where J_{ij} is a constant; the total energy of the system is given by

$$- \sum_{i,j}^{adj} \sigma_i\sigma_j J_{ij}.$$

The summation is over i and j with v_i and v_j adjacent. The thermodynamic properties of the system are described by the "partition function"

$$\sum_{(\sigma_1,\ldots,\sigma_N)\,\varepsilon\{-1,1\}^N} \exp\left(\sum_{i,j}^{adj} K_{ij}\sigma_i\sigma_j\right) \qquad (3.11)$$

where N is the number of vertices in the lattice and $K_{ij} = J_{ij}/kT$.The symbols k and T represent physical constants. The evaluation of the above partition function for particular values of J_{ij} constitutes the Ising problem.

It is shown by Kasteleyn ([14] p. 100) that there is a close relationship between (3.11) and the g.f. for closed partial graphs of the lattice graph. A graph is said to be closed if each of its vertices has even (possibly zero) degree. If G = (V,E) is a graph then let $S_{IS}(G)$ denote the set

$$\{A \subseteq E \mid (V,A) \text{ is a closed graph}\}.$$

The g.f. for the Ising problem is simply:

$$IS(G) = GF(G,S_{IS}) \qquad (3.12)$$

The relationship, which is derived in the above reference, between the Ising problem and the g.f. IS is the following. If G is a lattice graph with vertices $v_1,\ldots,v_N$ and edges $\{v_i,v_j\}$ labelled x_{ij} then the expression (3.11) is equal to

$$2^N(\prod_{i,j}^{adj} \cosh(K_{ij}))IS(G)$$

evaluated at the point $x_{ij} = \tanh(J_{ij}/kT)$.

The generating function (3.12) applied to the rectangular and cubic lattice graphs, introduced in section 3.2, yields two polynomial families, viz.

$$\{IS(R_n) \mid n = 1,2,\ldots\} \qquad (3.13)$$

$$\{IS(C_n) \mid n = 1,2,\ldots\} \qquad (3.14)$$

The method used for computing the g.f. of dimer arrangements on planar lattice graphs can be modified (see Kasteleyn [14] p. 101) to yield an algorithm for evaluating members of the family (3.13) which uses a number of arithmetic operations which is polynomial $(O(n^{5.05}))$ in the index n. On the other hand, the evaluation of members of (3.14) appears much more difficult, and no efficient procedure is known. We throw some light on this phenomenon by showing that the family (3.14) is complete in the sense of chapter 2. We remark that this is another example where moving from 2 to 3 dimensions introduces apparent intractability.

3.8 The Completeness of the Family $\{IS(C_n)\}$

We approach the main result via a series of lemmata. The first of these parallels lemma 3.1. Recall that DI is the dimer generating function defined in equation (3.3).

Lemma 3.4 The family $\{DI(K_{n,n})\}$ is complete over any field not of characteristic 2.

Proof A typical monomial of $DI(K_{n,n})$, say $x_{i_1j_1}..x_{i_nj_n}$, is characterised by

$i_1,..,i_n$ distinct and in the range [1,n]

$j_1,..,j_n$ distinct and in the range [1,n]

i.e. is of the form $x_{1\pi(1)}...x_{n\pi(n)}$ for some permutation π. Moreover to each such permutation corresponds a monomial of $DI(K_{n,n})$. Hence $DI(K_{n,n}) = per(x_{ij})$. The result follows from theorem 2.2. □

Lemma 3.5 For any graph G, there exists a graph H with the following properties:

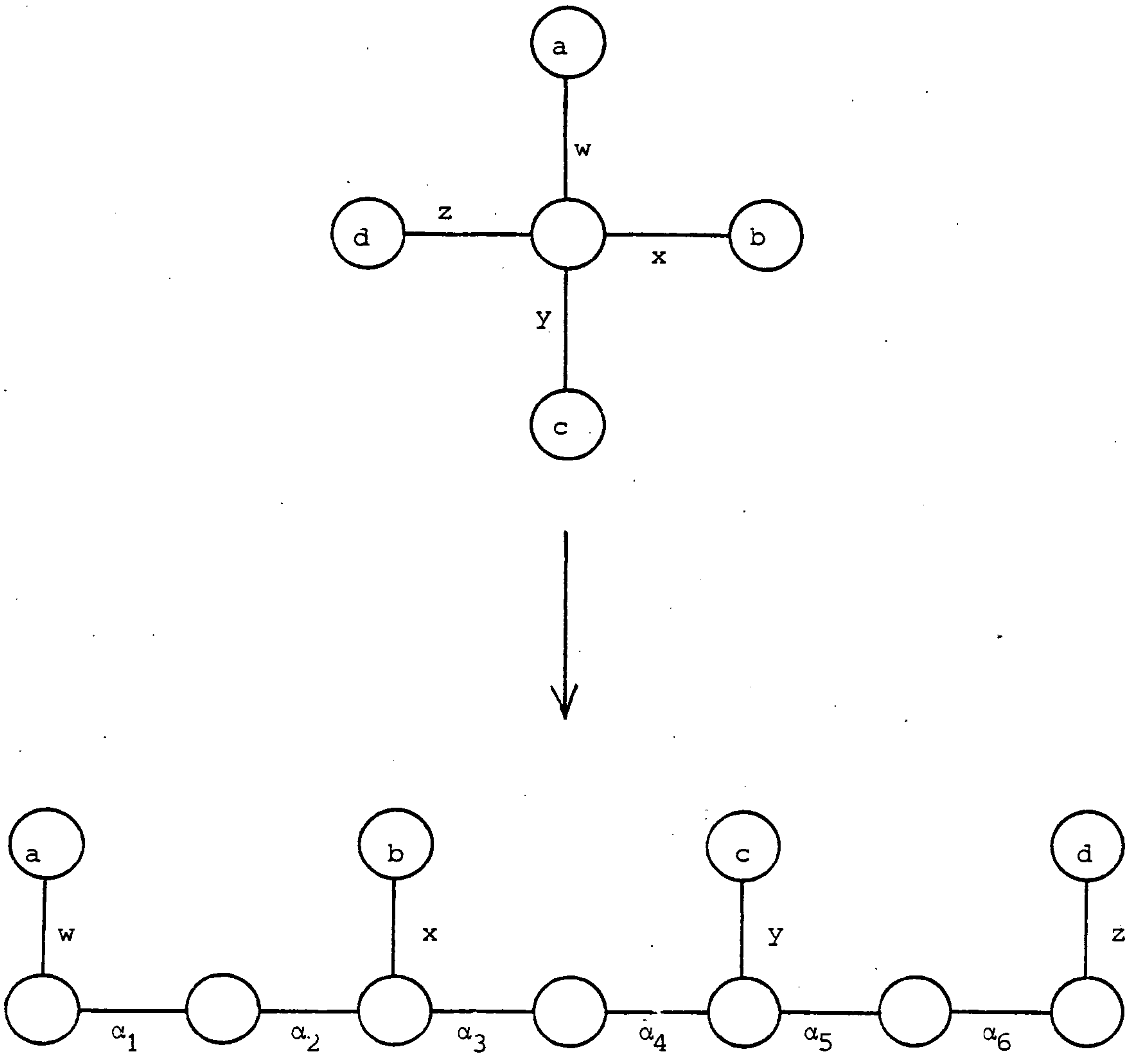
a
w
z
d
b
x
y
c
a
b
c
d
w
x
y
z
α_1
α_2
α_3
α_4
α_5
α_6

Figure 3.9

(i) $DI(H) = DI(G)$

(ii) H is of degree 3.

(iii) $|H| < 5 \cdot |G|$

Proof The graph G is transformed into H by "unfolding" vertices of degree greater than 3, a process analogous to that employed in theorem 3.3. The unfolding is illustrated by fig. 3.9, where we consider the case of a vertex of degree 4 with incident edges labelled by indeterminates w,x,y,z; the extension to vertices of higher degree will be immediately apparent. The scalars $\alpha_1,\ldots,\alpha_6$ are all set to 1.

Suppose that H is obtained from G by applying this unfolding to each of its vertices. Any perfect matching of G may be extended to a perfect matching of H in a unique way. For example, a perfect matching of G which includes the x labelled edge will extend to one of H which includes the x labelled edge together with the edges labelled $\alpha_1,\alpha_4,\alpha_6$. Conversely a perfect matching on H must include exactly one of the edges w,x,y,z; hence, such a matching is the extension of some matching on G. From this 1-1 correspondence between perfect matchings of G and H we deduce that $DI(H) = DI(G)$. This shows that H meets condition (i): conditions (ii) and (iii) are easily verified. □

It proves convenient to introduce a g.f. complementary to IS, which we denote by $\overline{IS}$, defined as the g.f. of partial graphs in which each vertex has odd degree.

Lemma 3.6 For any graph G of degree 3, there exists a graph H with the following properties:

(i) $\overline{IS}(H) = DI(G)$

(ii) $|H| \leq 15|G|$.

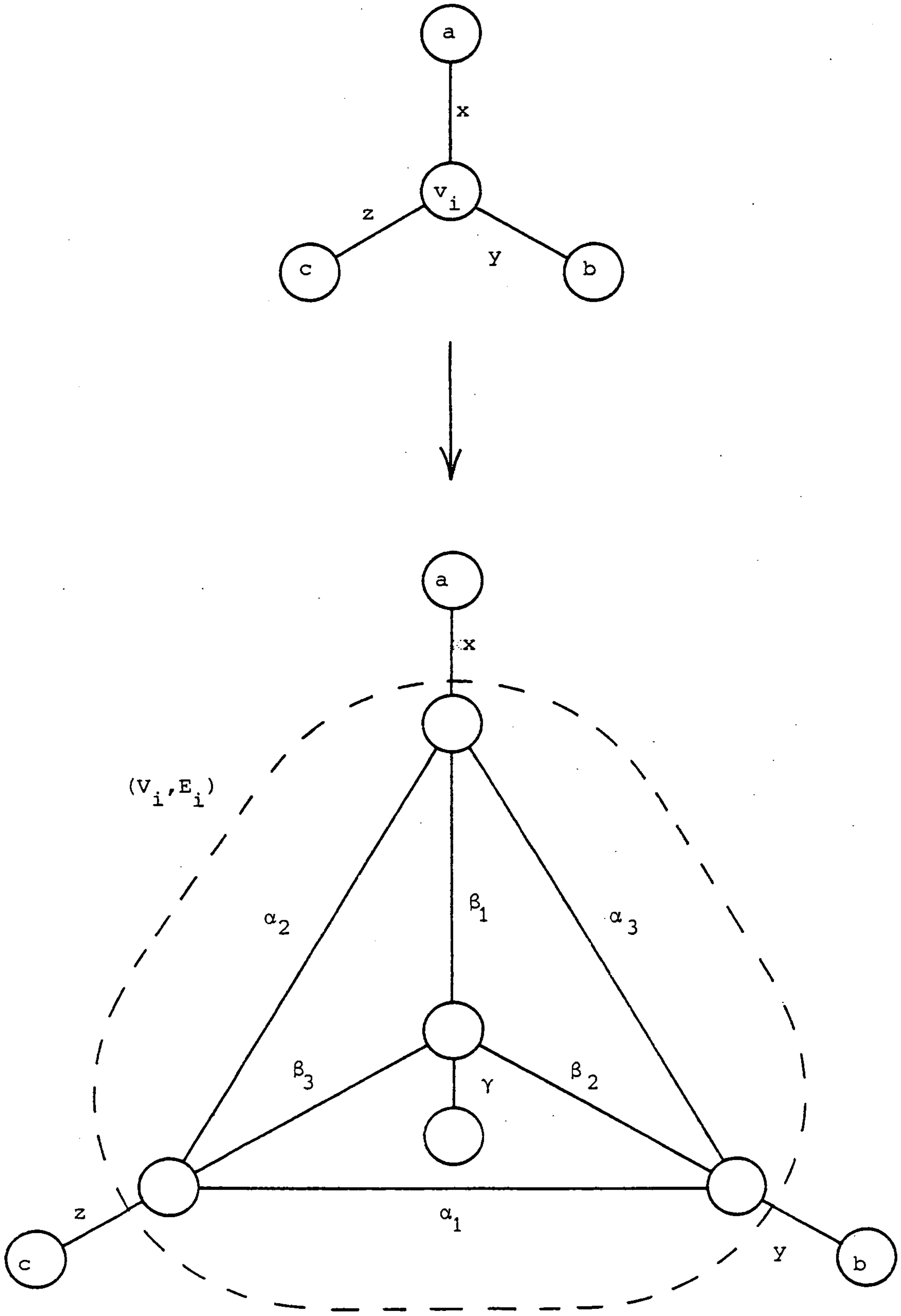
a
x
v_i
z
c
y
b
a
x
(V_i, E_i)
α_2
β_1
α_3
β_3
γ
β_2
z
α_1
c
y
b

Figure 3.10

Proof $H = (V,E)$ is constructed from G as follows. For each vertex v_i in G (we will suppose that G has n vertices $v_1,\dots,v_n$) there corresponds a set V_i of 5 vertices in H ; V is then taken to be the union $V_1 \cup \dots \cup V_n$. The vertices within each V_i are interconnected by a set E_i of 7 edges as described in fig. 3.10. In addition, for each edge $\{v_i, v_j\}$ in G there is an edge in H connecting a vertex of V_i to one of V_j. Distinct edges of G, incident at a common vertex v_i in G, correspond to edges of H which are incident at distinct vertices of V_i, as suggested by fig. 3.10. The set of edges of H which correspond 1-1 with the edges of G is denoted by E_0 ; E is then the union $E_0 \cup E_1 \cup \dots \cup E_n$. Note that $|H| \leq 15 \cdot |G|$.

Now define $\chi: 2^V \times 2^E \to \{0,1\}$ by $\chi(U,A) = 1$ iff for all $u \in U$ the degree of the vertex u in the partial graph (V,A) is odd. Then by the definition of $\overline{IS}$:

$$\overline{IS}(H) \equiv \sum_{A_i \subseteq E_i (0 \leq i \leq n)} \lambda(A_0)\dots\lambda(A_n)\chi(V, A_0 \cup \dots \cup A_n)$$

$$\equiv \sum_{A_0 \subseteq E_0} \lambda(A_0) \sum_{A_i \subseteq E_i (1 \leq i \leq n)} \lambda(A_1)\dots\lambda(A_n)\ \chi(V, A_0 \cup \dots \cup A_n) \qquad (3.15)$$

As the edges of H which are in distinct E_i are vertex disjoint, we may expand $\chi(V, A_0 \cup \dots \cup A_n)$ as

$$\prod_{1 \leq i \leq n} \chi(V_i, A_0 \cup A_i).$$

Hence, substituting in (3.15), we obtain

$$\overline{IS}(H) \equiv \sum_{A_0 \subseteq E_0} \lambda(A_0) \prod_{1 \leq i \leq n} \left[\sum_{A_i \subseteq E_i} \lambda(A_i)\ \chi(V_i, A_0 \cup A_i) \right] \qquad (3.16)$$

We claim that the sum

$$\sum_{A_i \subseteq E_i} \lambda(A_i)\ \chi(V_i, A_0 \cup A_i) \qquad (3.17)$$

is 1 if exactly 1 edge in A_0 is incident at V_i (we shall say that an edge is incident at a set of vertices iff it is incident at some vertex in the set), and 0 otherwise. Hence the identity (3.16) simplifies to

$$\overline{\mathrm{IS}}(H) = \mathrm{DI}(G).$$

The claim may be demonstrated by direct calculation. Firstly, we note that if exactly 0 or 2 edges of A_0 are incident at V_i, then necessarily $\chi(V_i, A_0 \cup A_i) = 0$ for any choice of A_i, and hence (3.17) is zero. (The number of odd degree vertices in any graph is even, see for example Even [8] p. 1.) If 3 edges of A_0 are incident at V_i then the sum (3.17) is

$$\gamma[1 + \alpha_1\alpha_2\alpha_3 + \beta_1\beta_2\alpha_3 + \beta_1\beta_2\alpha_1\alpha_2 + \beta_2\beta_3\alpha_1 + \beta_2\beta_3\alpha_2\alpha_3$$
$$+\beta_1\beta_3\alpha_2 + \beta_1\beta_3\alpha_1\alpha_3]$$

which evaluates to zero if we make the following assignment to the scalars in the construction:

$\alpha_1 = \alpha_2 = \alpha_3 = -1/3$, $\beta_1 = \beta_2 = \beta_3 = \sqrt{13}/3$ and $\gamma = 243/128$.

(The terms in the expression correspond naturally to partial graphs of (V_i, E_i).) If exactly one edge of A_0 (say the x labelled edge in fig. 3.11) is incident at V_i then (3.17) is

$$\gamma[\alpha_1 + \alpha_2\alpha_3 + \beta_1\beta_2\alpha_2 + \beta_1\beta_2\alpha_1\alpha_3 + \beta_2\beta_3 + \beta_2\beta_3\alpha_1\alpha_2\alpha_3 + \beta_1\beta_3\alpha_3 + \beta_1\beta_3\alpha_1\alpha_2]$$

which evaluates to one under the same assignment.

<u>Lemma 3.7</u> For any graph G there exists a graph H such that:

(i) $\mathrm{IS}(H) = \overline{\mathrm{IS}}(G)$

(ii) $|H| \leq 5|G|$.

<u>Proof</u> Firstly, we note that if G has an odd order then $\overline{\mathrm{IS}} = 0$. (The number of vertices of odd degree in a graph is even.) It is trivial, in this case, to construct a suitable H; we might, for example, take the complete graph on three vertices with edges labelled 1,1 and -1. We may therefore suppose that the order of G is even. Partition the 2n modes of G into two sets,

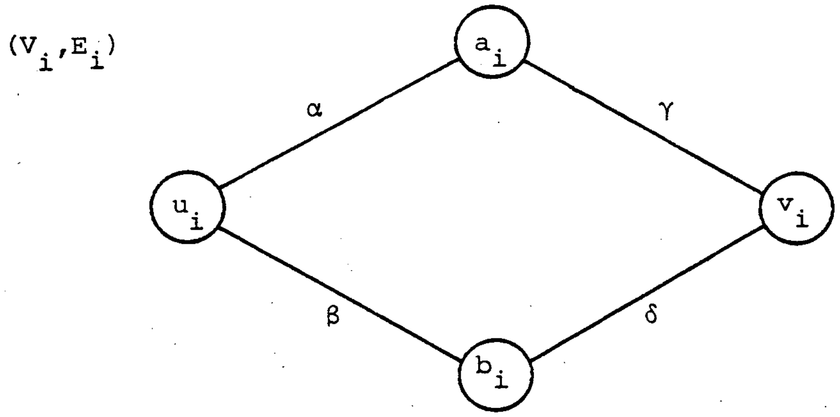

Figure 3.11

$u_1,\ldots,u_n$ and $v_1,\ldots,v_n$, in an arbitrary fashion. Construct $H = (V,E)$ by composing G with the n graphs described by fig. 3.11, identifying the similarly labelled vertices. Denote the original set of edges of G by E_0, the set of vertices $\{u_i,v_i,a_i,b_i\}$ by V_i and the set of edges $\{\{u_i,a_i\},\{u_i,b_i\},\{v_i,a_i\},\{v_i,b_i\}\}$ by E_i. Note that $V = V_1 \cup \ldots \cup V_n$ and $E = E_0 \cup E_1 \cup \ldots \cup E_n$. Let $\chi: 2^V \times 2^E \to \{0,1\}$ be defined by $\chi(U,A) = 1$ iff, in the partial graph (V,A), there are an <u>even</u> number of edges incident at u for each vertex $u \in U$. Then by definition:

$$IS(H) = \sum_{A_i \subseteq E_i (0 \leq i \leq n)} \lambda(A_0)\ldots\lambda(A_n)\ \chi(V, A_0 \cup \ldots \cup A_n)$$

$$= \sum_{A_0 \subseteq E_0} \lambda(A_0) \sum_{A_i \subseteq E_i (1 \leq i \leq n)} \lambda(A_1)\ldots\lambda(A_n)\chi(V, A_0 \cup \ldots \cup A_n) \qquad \ldots \qquad (3.18)$$

In a manner similar to that used in the proof of lemma 3.6, we may express $\chi(V, A_0 \cup \ldots \cup A_n)$ as the product

$$\prod_{1 \leq i \leq n} \chi(V_i, A_0 \cup A_i).$$

Substituting in (3.18) we obtain:

$$IS(H) = \sum_{A_0 \subseteq E_0} \lambda(A_0) \prod_{1 \leq i \leq n} \left[\sum_{A_i \subseteq E_i} \lambda(A_i)\ \chi(V_i, A_0 \cup A_i) \right] \qquad \ldots \qquad (3.19)$$

We now claim that the sum

$$\sum_{A_i \subseteq E_i} \lambda(A_i)\chi(V_i, A_0 \cup A_i) \qquad (3.20)$$

is 1 if there are an odd number of edges incident at both u_i and v_i, and 0 otherwise. Hence the product in (3.19) is 1 if the degree of each node of the partial subgraph $(\{u_1,\ldots,u_n,v_1,\ldots v_n\},A_0)$ is odd, and 0 otherwise. Hence (3.19) simplifies to $IS(H) = \overline{IS}(G)$.

We justify the claim by direct calculation. Firstly, we observe that if an odd number of edges in A_0 are incident at v_i

and an even number at u_i (or vice versa) then (3.20) is necessarily 0. (This arises from the usual parity consideration.) If the number of edges in A_0 incident at u_i is even and at v_i is even then (3.20) is

$$1 + \alpha\beta\gamma\delta$$

which is zero under the assignment $\alpha=1$, $\beta=1$, $\gamma= (1+\sqrt{5})/2$, $\delta = (1-\sqrt{5})/2$, On the other hand, if the number of edges in A_0 incident at u_i is odd and at v_i is odd then (3.20) is

$$\alpha\gamma + \beta\delta$$

which is 1 under the same assignment. This substantiates the claim. □

We can now state and prove a theorem analogous to theorem 3.2.

Theorem 3.8 There exists a family $\{G_n\}$ of graphs with the following properties:

(i) $\{IS(G_n) \mid n = 1,2,..\}$ is a complete family over the real field $\mathbb{R}$.

(ii) $|G_n| = O(n^2)$

Proof Combining lemmata 3.5, 3,6, 3.7 we deduce the existence of a graph G_n with the property that $IS(G_n) = DI(K_{n,n})$. It is clear from the statements of the lemmata that $|G_n| \leqq 375.|K_{n,n}|$. The result follows from the completeness of $\{DI(K_{n,n})\}$ over $\mathbb{R}$ (lemma 3.4). □

The analogue of theorem 3.3 is

Theorem 3.9 The family $\{IS(C_n) \mid n = 1,2,..\}$ is complete over the real field $\mathbb{R}$.

Proof The construction is in two steps:

(i) transformation of the graph G_n of the previous theorem into a degree-3 graph $G_n^{(3)}$ which satisfies $IS(G_n^{(3)}) = IS(G_n)$

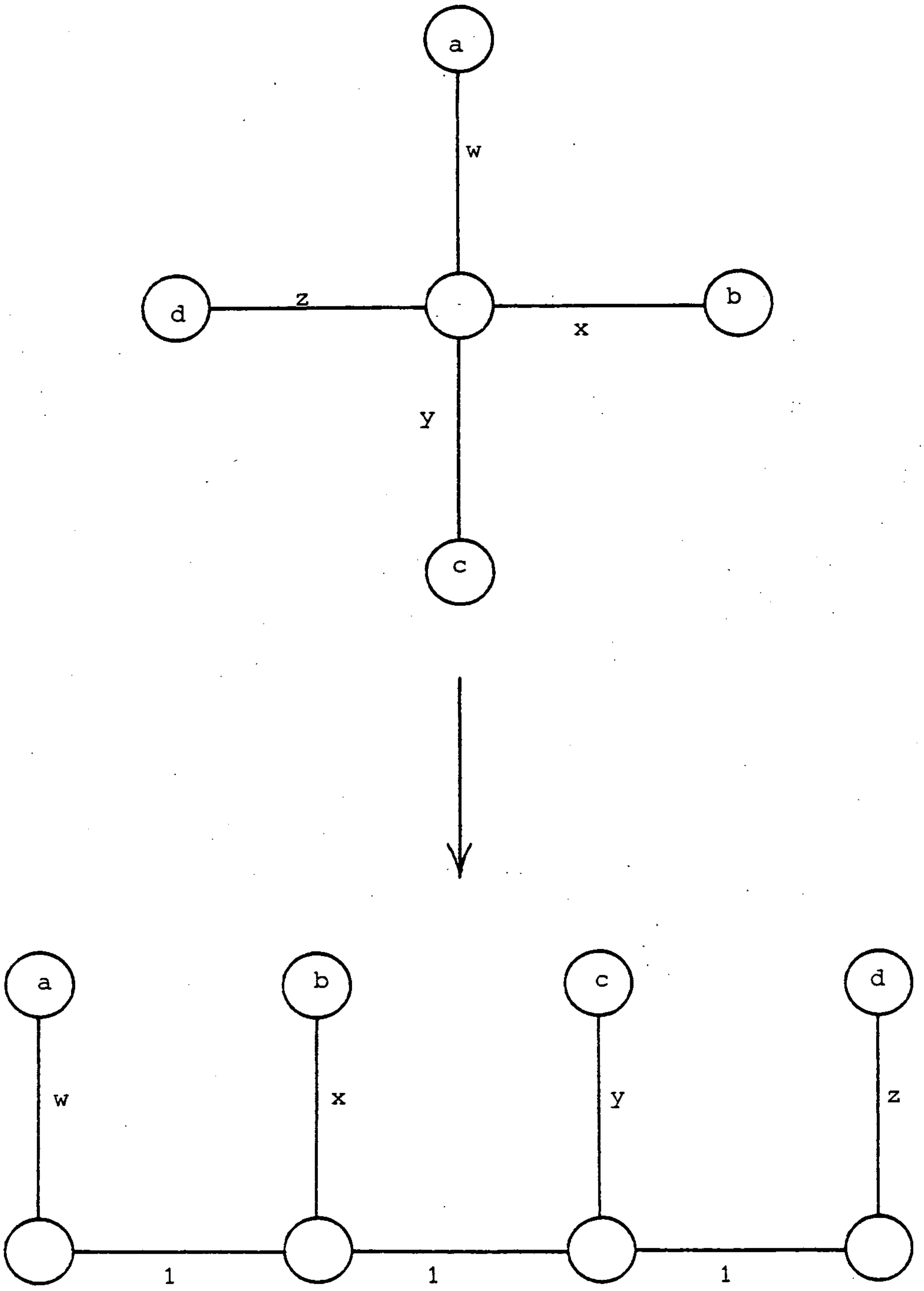

Figure 3.12

(ii) embedding of the degree-3 graph $G_n^{(3)}$ homeomorphically in the cubic lattice $C_{p(n)}$ for some polynomial p(.).

The construction of $G_n^{(3)}$ from G_n is effected by unfolding the vertices of G_n in a fashion described by fig. 3.12, which illustrates the degree-4 case. The extension to vertices of higher degree and the fact that the g.f. is preserved should be immediately apparent.

Suppose $G_n^{(3)}$ has k vertices $v_1, \dots v_k$ and m edges $e_1, \dots e_m$. $G_n^{(3)}$ may be embedded in C_{3k} as follows. The vertex v_i is mapped onto that vertex of C_{3k} which has Cartesian coordinates (3i-2,0,0). For each edge e_j connecting vertices v_i and $v_{i'}$ in $G_n^{(3)}$ there is a chain of edges in C_{3k} passing through the following points and running in straight lines between them: (3i-2+d,0,0), (3i-2+d,j,0), (3i-2+d,j,1), (3i'-2+d',j,1), (3i'-2+d',j,0), (3i'-2+d',0,0); d and d' take values in {-1,0,1} and are chosen so that each of the three edges incident at a vertex of the original graph $G_n^{(3)}$ is mapped onto a distinct chain of edges in C_{3k}. Note that where chains cross in the x,y-plane, they always do so at different "levels" i.e. their z-coordinates differ.

Each edge of such a chain is labelled 1, except an arbitrary distinguished one which is given the same label as the corresponding edge in $G_n^{(3)}$. All the other edges of C_{3k} are assigned weight 0.

It will be apparent that the Ising g.f. of the embedded graph is the same as that of $G_n^{(3)}$. The result follows from theorem 3.8.

□

4. SECOND APPLICATION : NETWORK RELIABILITY

4.1 An Introduction to Network Reliability

As a second example of the use of the ideas developed in chapter 2, we examine the problem of determining the reliability of communication networks. Informally a communication network is composed of a number of transmission stations which communicate via links. The elements of the network (stations and links) are unreliable and fail with known probabilities; the failures of distinct elements are assumed to be probabilistically independent events. Two stations are said to be communicating if they are connected by a chain of links and stations, none of which has failed. It is of fundamental importance to designers of communication networks that they be able to ascertain the robustness of a proposed network to element failures. They are thus led to consider various reliability measures for networks, such as the probability that all stations can communicate with each other, or the probability that two given stations can communicate.

Such networks have been studied extensively by many authors, for example Ball [2], Misra [20] and Rosenthal [30]; despite this effort, no procedure is known which computes a non-trivial reliability measure, which can be applied to arbitrary networks, and which runs in time polynomial in the size of the network. Polynomial time algorithms are known only for some very special cases, for example the series-parallel networks which are considered by Misra [20].

It would be pleasing to be able to make some statement about this apparent intractability. One way of doing this is to show that some problem related to network reliability is NP-hard in the sense of Karp [13]. This is not a very natural or satisfying method. It is not natural because an evaluation or enumeration problem is being translated into a decision problem: it is not satisfying because we do not obtain a precise characterisation of the complexity of the problem, only that it is harder than some complexity class. However, this technique has been used by Rosenthal [30] to show that some reliability measures are hard to compute, assuming that both station and link failures are allowed. Another approach is to view the computation of a reliability measure as the problem of enumerating all partial graphs of a graph which possess a given property; in this way, Valiant [41] has demonstrated the intractability of evaluating the probability that two given stations communicate, even if we assume that all stations are perfectly reliable. In this chapter we use the algebraic formulation developed in chapter 2, which forms a natural framework in which to consider reliability problems.

The intractability of most reliability measures in the case when stations as well as links are allowed to fail has already been established. It might be supposed that if we restrict our attention to networks with perfectly reliable stations then computation becomes much easier - we shall show that this is not the case. The aim of this chapter is to present proofs of the intractability of several reliability measures. For one of these measures, namely the probability that in an undirected network all stations can communicate with each other, there was previously no evidence of intractability. The question of the computational difficulty of this measure was raised by Rosenthal [30] and a solution has been particularly elusive.

4.2 Network Definitions

Although we may consider networks with links which are either bidirectional or unidirectional (i.e. in which transmission can take place in both directions or only in one), we will concentrate our attention first on the case of bidirectional links. The completeness results for reliability measures on unidirectional networks are usually easy corollories of the results for bidirectional ones as will be shown in section 4 of this chapter. Accordingly, we model a communication network as an undirected graph $G = (V,E)$; the vertex set V is taken to represent a number of stations, E is the set of links joining them, and the edge labellings represent the probabilities that the corresponding edges are functioning. There are two points to be stressed: firstly that the stations are thought of as perfectly reliable, and secondly that the link failure probabilities are assumed to be probabilistically independent.

The notion of two stations of a network being able to communicate carries over to the graph theoretic coneept of <u>connected components</u> of a graph. Suppose that $G = (V,E)$ is a graph and $A \subseteq E$. The set of edges A defines a relation on V whereby $u,v \in V$ are related iff $\{u,v\} \in A$; define the equivalence relation $\longleftrightarrow$ on V to be the reflexive, transitive closure of this. The equivalence relation, $\longleftrightarrow$, partitions V into equivalence classes; the subgraphs induced by the equivalence classes are called the (<u>connected</u>) <u>components</u> of G. Two stations of a network which can communicate correspond to vertices of G which are in the same connected component.

In the usual manner of probability theory (see Rao [28] p. 80) we associate with G a set of <u>elementary events</u> Ω_G - in this case we simply take $\Omega_G = 2^E$. We assume, as in chapter 3, that λ is a labelling of G taking values in $\mathbb{F} \cup X$, where $\mathbb{F}$ is a field and $X = \{x_1, \ldots x_n\}$ is a finite set of indeterminates over that field. The labelling λ induces a probability distribution $p : \Omega_G \to \mathbb{F}[x_1, \ldots, x_n]$ specified by

$$p(A) = \prod_{e \varepsilon A} \lambda(e) \prod_{e \varepsilon (E-A)} (1-\lambda(e)).$$

Here addition and multiplication are those of the polynomial ring $\mathbb{F}[X]$.

Suppose now that S is an <u>event</u>, that is $S \subseteqq \Omega_G$. The event S has an associated <u>probability</u> <u>of</u> <u>occurrence</u>, a polynomial specified by

$$Pr(S) = \sum_{A \varepsilon S} p(A). \tag{4.1}$$

We remark that the probability of an event and its complement are related by

$$Pr(S) + Pr(\Omega_G - S) = 1 \tag{4.2}$$

Using this notation, a few reliability measures for undirected graphs are listed:

(i) CONNECTED(G)=$Pr\{A \varepsilon \Omega_G | (V,A)$ has exactly one connected component$\}$. (The probability that, in the network modelled by G, all stations can communicate.)

(ii) s-t-CONNECTED(G)=$Pr\{A \varepsilon \Omega_G |$ s and t are in the same connected component of $(V,A)\}$. (Measures the probability that s and t can communicate.)

Both the reliability polynomials so far defined have a natural interpretation; the next is artificial, but serves as a useful stepping-stone in our reductions:

(iii) s-t-PARTITION(G)=$Pr\{A \varepsilon \Omega_G | (V,A)$ has exactly two connected components, one containing s and the other t$\}$.

The final polynomial is redundant in that it is the complement of one previously defined, however, it too serves as a conceptual aid:

(iv) s-t-SEPARATED(G)=Pr$\{A \varepsilon \Omega_G \mid$ s and t are in distinct components of $(V,A)\}$.

Note that, by identity (4.2),

$$\text{s-t-SEPARATED(G)=1-(s-t-CONNECTED(G)).} \qquad (4.3).$$

By considering these reliability measures applied to the complete graph K_n for increasing n, we generate the associated polynomial families, for example:

$\{\text{s-t-CONNECTED}(K_n) \mid n=1,2,..\}$

and $\{\text{CONNECTED}(K_n) \mid n=1,2,..\}$

The main results of section 4 of this chapter will be to show that these two polynomial families are complete in the sense of chapter 2. In section 4.5 the completeness of several reliability measures for directed graphs will be deduced as corollories.

4.3 Computing Reliability Measures of Synthesised Networks

In the proofs of the main theorems of the next section we will need to compute reliability polynomials for large graphs which are composed from small component graphs. In preparation for these tasks, we introduce a method due to Rosenthal [31] for simplifying such calculations.

Suppose G = (V,E) is a graph constructed from a set of component subgraphs, $H_i = (V_i,E_i)$ $(1 \leqq i \leqq m)$, by identifying certain of the vertices in distinct H_i. The identified vertices will be termed external, and the remainder internal, vertices. Suppose also that we wish to compute some reliability polynomial on G. For the sake of definiteness we shall work with the s-t-SEPARATED polynomial, but the method can be applied to any natural measure of reliability.

Consider one of the component subgraphs H_i with external vertices $u_1, \ldots, u_k$. Any subset A of the edges E_i of H_i induces a partition π of the external vertices of H_i, that is to say divides $\{u_1, \ldots, u_k\}$ into a set of <u>blocks</u> such that each u_j is in exactly one block of π; two vertices are in the same block of π iff they are in the same connected component of (V_i, A). In this way, to every partition π of the external vertices of H_i corresponds a <u>class</u> of elementary events on H_i:

$$C_\pi = \{A \subseteq E_i \mid A \text{ induces the partition } \pi \text{ of the external vertices of } H_i\}.$$

Each class has a <u>class probability</u>, namely

$$\Pr(C_\pi) = \sum_{A \in C_\pi} \Pr(A).$$

We introduce a succinct notation for classes. A class is specified by listing the external vertices of a subgraph, say H_i, enclosing in square brackets those vertices which belong to the same connected component. For example, when k=3, $[u_1, u_3][u_2]$ is the class $\{A \subseteq E_i \mid u_1$ and u_3 are contained in a single connected component of (V_i, A), which is distinct from that which contains $u_2\}$.

If H_i and H_j are distinct component subgraphs with classes and class probabilities defined as above, we can combine them to form a single, larger component, H say, by identifying certain of the external vertices. The set of external vertices of H contains those vertices of H_i and H_j which are also shared by component subgraphs other than these two.

The classes of H are in 1-1 correspondence with partitions of the external vertices of H. The important observation is that the class probabilities of H_i and H_j encode exactly enough information to enable the computation of the class probabilities of H to proceed. In particular it should be noted that each class of H is a union of sets of the form

$$\{A_i \cup A_j \mid A_i \varepsilon C_i,\ A_j \varepsilon C_j\} \qquad (4.4)$$

where C_i and C_j are classes of H_i and H_j respectively. Using this observation it can be seen that a product of classes $C = C_i \times C_j$ can be defined, where C is that class of H which contains the set (4.4). We can express the class probability of C as

$$\Pr(C) = \sum_{C_i \times C_j = C} \Pr(C_i)\Pr(C_j) \qquad (4.5)$$

By combining component subgraphs, using a repeated application of this procedure, we may evaluate the class probabilities of the synthesised graph G. The polynomial s-t-SEPARATED(G) is then simply that class probability of G which corresponds to a partition of s and t into separate blocks.

As a concrete example, consider the component subgraph of fig. 4.1. It has 5 classes, corresponding to the 5 partitions of the external vertices s,t,u_{ij}. The class probabilities are listed assuming the assignments $p_1=1/2$, $p_2=1/2$ and $p_3=3/2$. The convention $q_i=1-p_i$ is used

(i) $\Pr([s,t,u_{ij}]) = p_1p_2p_3 = 3/8$

(ii) $\Pr([s,t][u_{ij}]) = p_1p_2q_3 = -1/8$

(iii) $\Pr([s,u_{ij}][t]) = p_1q_2p_3 = 3/8$

(iv) $\Pr([t,u_{ij}][s]) = q_1p_2p_3 = 3/8$

(v) $\Pr([s][t][u_{ij}]) = q_1q_2q_3+p_1q_2q_3+q_1p_2q_3+q_1q_2p_3 = 0$

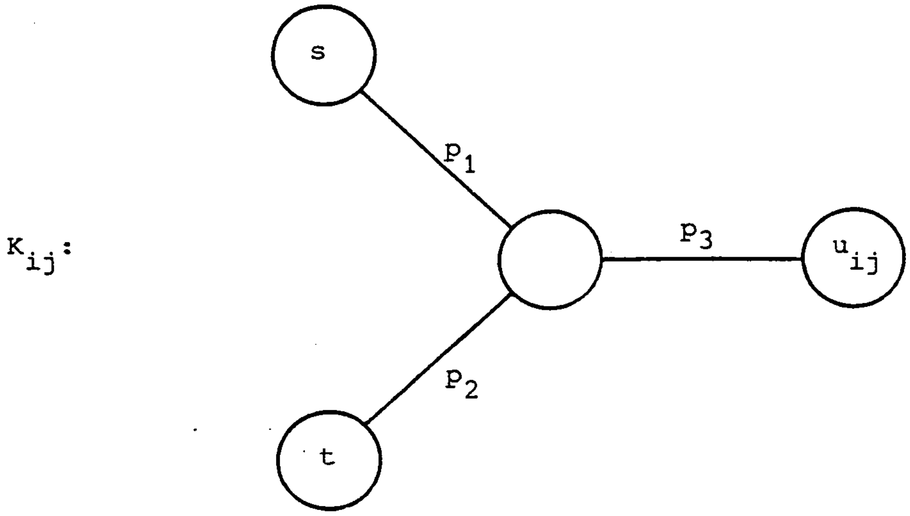

Figure 4.1

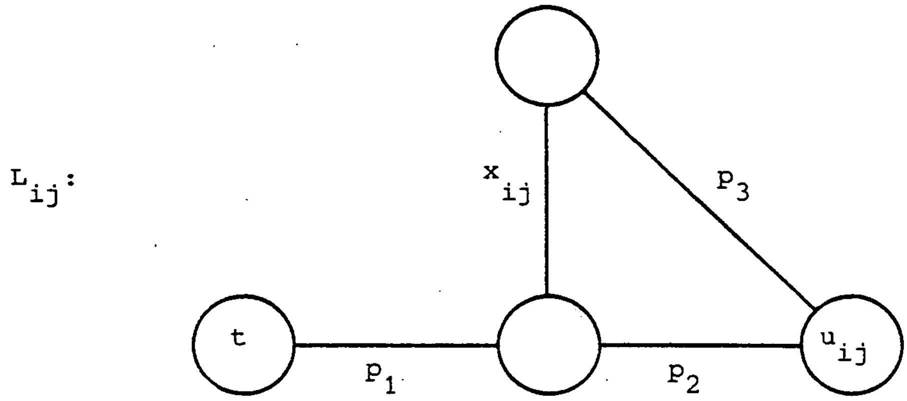

Figure 4.2

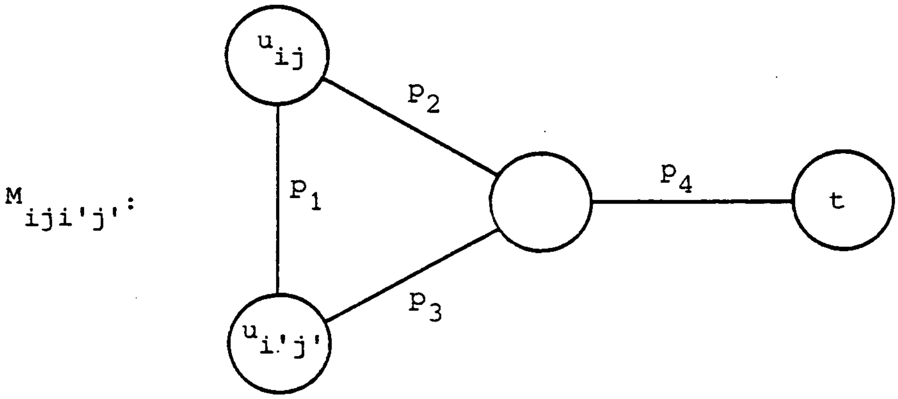

Figure 4.3

In this example, it will be noted that the classes $[s,t,u_{ij}]$ and $[s,t][u_{ij}]$ may, in the present context, be neglected as they already place s and t in the same connected component - we shall call such classes <u>inconsistent</u>. In addition the class $[s][t][u_{ij}]$ can be neglected as it has associated probability 0 - we shall call such classes <u>null</u>. This leaves only two classes which are neither null nor inconsistent - such classes will be termed <u>contributory</u>. By working with the 2 contributory classes instead of the 8 elementary events, the computational effort is substantially reduced.

We are now prepared for the main results of the chapter.

4.4 Completeness Results

<u>Theorem 4.1</u> There exists a family of graphs $\{G_n\}$, with distinguished vertices s and t, which possesses the following properties:

(i) $|G_n| = O(n^3)$.

(ii) $s\text{-}t\text{-SEPARATED}(G_n)=(3/8)^{n^2} \text{per}^*(X)$

<u>Proof</u> The graph G_n is constructed by composing the component subgraphs K_{ij}, L_{ij} $(1 \leqq i,j \leqq n)$ and $M_{iji'j'}$ $(1 \leqq i,j,i',j' \leqq n,$ $i=i' \oplus j=j')$ of figs. 4.1, 4.2, 4.3 respectively. The symbol $\oplus$ is here used to denote "exclusive or". Similarly labelled vertices are identified in the composition, while separate occurrences exist of the unlabelled vertices. We consider the classes of each of the component subgraphs and compute their respective class probabilities, in preparation for evaluating the required reliability measure on G_n

(i) The components K_{ij} have 3 external vertices, s,t,u_{ij}, and hence 5 classes. The scalars are assigned values

$p_1=1/2$, $p_2=1/2$, $p_3=3/2$. The class probabilities were computed in the last section but are repeated here for convenience:

$\Pr([s,t,u_{ij}]) = 3/8$ (inconsistent).

$\Pr([s,t][u_{ij}]) = -1/8$ (inconsistent).

$\Pr([s,u_{ij}][t]) = 3/8$.

$\Pr([t,u_{ij}][s]) = 3/8$.

$\Pr([s][t][u_{ij}]) = 0$ (null).

(ii) The components L_{ij} have 2 external vertices t and u_{ij} and hence only 2 classes. The scalars are assigned values $p_1=1/2$, $p_2=2$, $p_3=2$.

$$\begin{aligned}\Pr([t,u_{ij}]) &= x_{ij}(p_1p_2p_3 + p_1p_2q_3 + p_1q_2p_3) \\ &\quad + (1-x_{ij})(p_1p_2p_3 + p_1p_2q_3) \\ &= x_{ij}(2-1-1) + (1-x_{ij})(2-1) \\ &= (1-x_{ij}).\end{aligned}$$

$\Pr([t][u_{ij}]) = x_{ij}$ (by identity 4.2)

(iii) The components $M_{iji'j'}$ have 3 external vertices $u_{ij}, u_{i'j'}$ and t and 5 classes. The scalars are assigned values $p_1=-7/5$, $p_2=1/2$, $p_3=1/2$, $p_4=4/3$.

$$\begin{aligned}\Pr([u_{ij},u_{i'j'},t]) &= p_1p_2p_3p_4 + q_1p_2p_3p_4 + \\ &\quad p_1q_2p_3p_4 + p_1p_2q_3p_4 \\ &= -3/5.\end{aligned}$$

$$\begin{aligned}\Pr([u_{ij},u_{i'j'}][t]) &= q_1p_2p_3q_4 + p_1q_2q_3q_4 + \\ &\quad p_1p_2q_3q_4 + p_1q_2p_3q_4 + \\ &\quad p_1q_2q_3p_4 + p_1p_2p_3q_4 \\ &= -1/5.\end{aligned}$$

$$\begin{aligned}\Pr([u_{ij},t][u_{i'j'}]) &= q_1p_2q_3p_4 \\ &= 4/5\end{aligned}$$

$$
\begin{aligned}
\Pr([u_{i'j'},t][u_{ij}]) &= q_1q_1p_3p_4 \\
&= 4/5 \\
\Pr([u_{ij}][u_{i'j'}][t]) &= q_1q_2q_3q_4 + q_1p_2q_3q_4 + \\
&\quad q_1q_2p_3q_4 + q_1q_2q_3p_4 \\
&= 1/5
\end{aligned}
$$

Having analysed each of the component subgraphs, let us now proceed to apply the method of the previous section. Firstly each pair of component subgraphs K_{ij}, L_{ij} is combined to form a single component H_{ij} with external vertices $\{s,t,u_{ij}\}$. As has been noted, K_{ij} has only two contributory classes, namely $[s,u_{ij}][t]$ and $[t,u_{ij}][s]$, both of which have associated probability 3/8. H_{ij} thus has two contributory classes:

(i) $[s,u_{ij}][t]$, formed by producting the class $[s,u_{ij}][t]$ of K_{ij} with the class $[t][u_{ij}]$ of L_{ij}. By equation 4.5 the associated class probability is $(3/8)x_{ij}$.

(ii) $[t,u_{ij}][s]$, formed by producting the class $[t,u_{ij}][s]$ of K_{ij} with either class of L_{ij}. By equation 4.5 the associated class probability is $(3/8)[x_{ij}+(1-x_{ij})] = 3/8$.

Next, the component subgraphs H_{ij} thus produced are combined into one component H with external vertices $\{s,t,u_{11},u_{12},\dots,u_{nn}\}$ Each H_{ij} has exactly two contributory classes - one which places u_{ij} in the connected component containing s and another which places it in that containing t. Each contributory class of H corresponds to a partition of the external nodes into two blocks, one containing s and the other t, i.e. each such class of H is of the form

$$C_S = [s,u_{ij}((i,j)\varepsilon S)][t,u_{ij}((i,j)\not\varepsilon S)]$$

for some $S\subseteq\{(i,j)|1\leqq i,\leqq n\}$. The class C_S is formed by producting together the classes $[s,u_{ij}][t]$ of H_{ij} for $(i,j)\varepsilon S$ and the classes $[t,u_{ij}][s]$ of H_{ij} for $(i,j)\not\varepsilon S$. Thus:

$$\begin{aligned}Pr(C_S) &= \prod_{(i,j)\varepsilon S} (3/8)x_{ij} \prod_{(i,j)\not\varepsilon S} (3/8)\\ &= (3/8)^{n^2} \prod_{(i,j)\varepsilon S} x_{ij}.\end{aligned}$$

Finally, we combine the above class probabilities with those of the component subgraphs $M_{iji'j'}$ in order to compute s-t-SEPARATED(G_n). The $2^{n\times n}$ class probabilities of H correspond precisely to the $2^{n\times n}$ possible linear monomials in the indeterminates $\{x_{11},\ldots,x_{nn}\}$; the function of the subgraphs $M_{iji'j'}$ is to pick out those monomials which occur in the expansion of per*(X), while annihilating others.

We investigate which classes of $M_{iji'j'}$, when producted with the class C_S of H, produce the class [s][t] of G_n. There are four cases:

(i) If $(i,j)\not\varepsilon S$ and $(i',j')\not\varepsilon S$ then any of the classes of $M_{iji'j'}$ will combine. The factor contributed by such subgraphs is thus 1.

(ii) If $(i,j)\varepsilon S$ and $(i',j')\not\varepsilon S$ then the classes which combine are $[t][u_{ij}][u_{i'j'}]$ and $[u_{ij}][u_{i'j'},t]$. The factor contributed is $P_r([t][u_{ij}][u_{i'j'}]) + Pr([u_{ij}][u_{i'j'},t]) = 1/5 + 4/5 = 1$.

(iii) The case when $(i,j)\not\varepsilon S$ and $(i',j')\varepsilon S$ is similar to (ii) by symmetry.

(iv) If $(i,j)\varepsilon S$ and $(i',j')\varepsilon S$ then the classes which combine are $[t][u_{ij}][u_{i'j'}]$ and $[t][u_{ij},u_{i'j'}]$. The factor

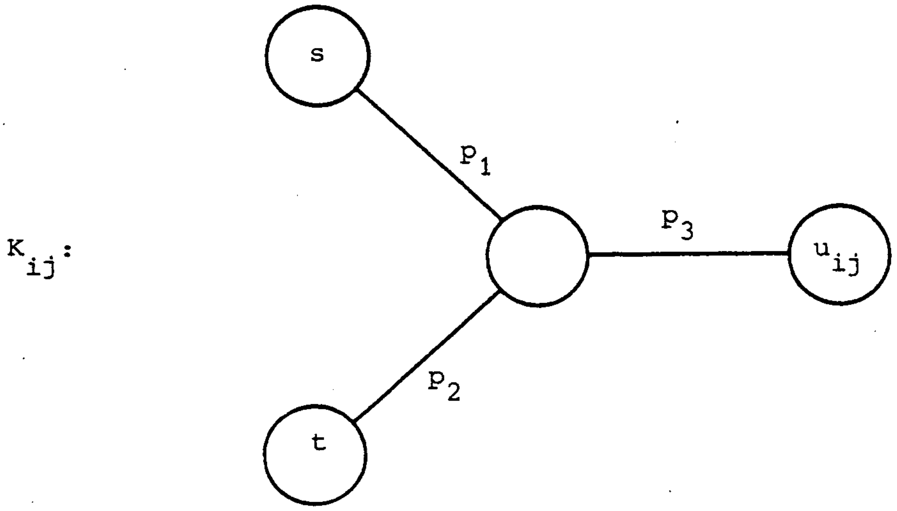

Figure 4.4

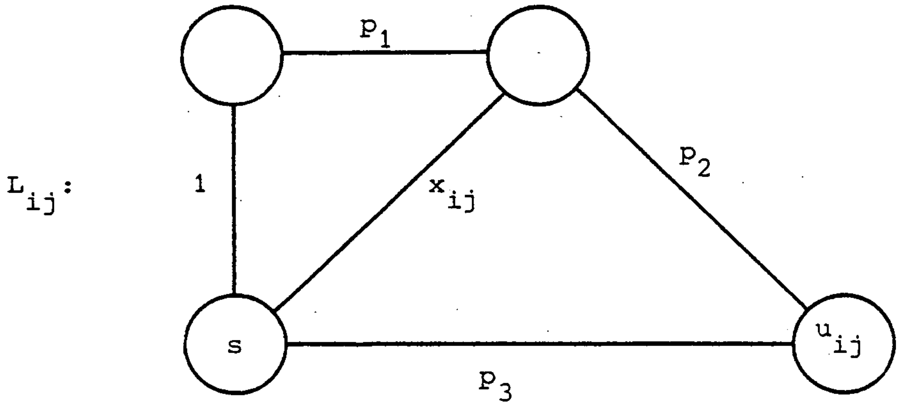

Figure 4.5

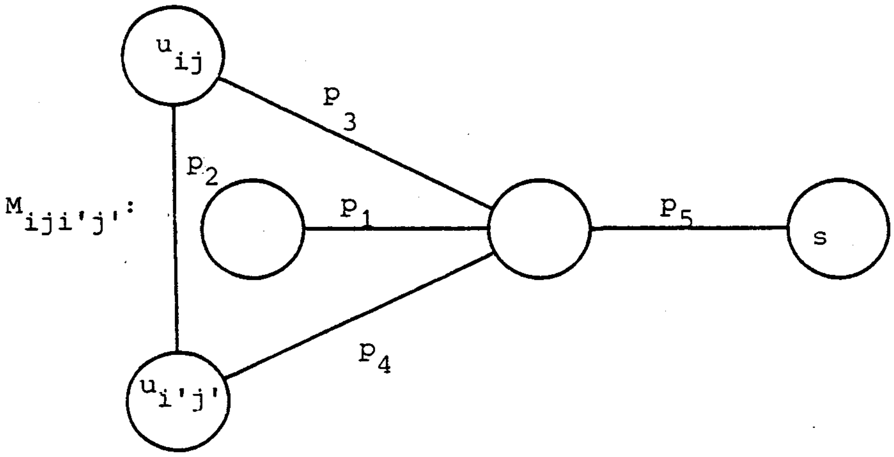

Figure 4.6

contributed is $\Pr([t][u_{ij}][u_{i'j'}]) + \Pr([t][u_{ij}, u_{i'j'}])$
$= 1/5 - 1/5 = 0.$

A component subgraph $M_{iji'j'}$ exists for all $1 \leq i,j,i',j' \leq n$ with $(i=i') \oplus (j=j')$. Hence

$$\text{s-t-SEPARATED}(G_n) = \sum_{S \subseteq \{(i,j) \mid 1 \leq i,j \leq n\}} \chi(S) \Pr(C_S)$$

where

$\chi(S)=1$ if every pair of elements in S differs in both components, and

$=0$ otherwise.

i.e.
$$\text{s-t-SEPARATED}(G_n) = \sum_{S \subseteq \{(i,j) \mid 1 \leq i,j \leq n\}} \left[(3/8)^{n^2} \chi(S) \prod_{(i,j) \in S} x_{ij} \right]$$
$$= (3/8)^{n^2} \text{per}^*(X) \qquad \square$$

Theorem 4.2 There exists a family of graphs $\{G_n\}$, with distinguished vertices s and t, which possesses the following properties:

(i) $|G_n| = O(n^3)$

(ii) $\text{s-t-PARTITIONED}(G_n) = (1/2)^{n^2} \text{per}^*(X)$.

Proof The proof of this theorem is analogous to that of theorem 4.1. The graph G_n is constructed by composing the component subgraphs K_{ij}, L_{ij} $(1 \leq i,j \leq n)$ and $M_{iji'j'}$ $(1 \leq i,j,i',j' \leq n$, $i=i' \oplus j=j')$, described by figs. 4.4, 4.5 and 4.6, which are modified versions of those employed in the last construction. To accommodate the s-t-PARTITION reliability measure we must slightly redefine the notion of class. Suppose $H_i = (V_i, E_i)$ is a component subgraph with external vertices $u_1, \ldots, u_k$. As before, any subset of the edges E_i of H_i induces a partition π

of the vertices $u_1,\dots,u_k$. To each such partition corresponds a class, C_π, of elementary events on H_i:

$$C_\pi = \{A \subseteq E_i \mid (A \text{ induces the partition } \pi \text{ of the external vertices of } H_i) \wedge (\text{every connected component of } (V_i,A) \text{ contains an external vertex of } H_i)\}$$

The classes are in 1-1 correspondence with those defined for the s-t-SEPARATED measure; the only difference is that we insist that no internal vertex is isolated from all the external vertices. With the redefined classes it is clear that s-t-PARTITIONED(G_n) may be expressed as the class probability Pr([s][t]) of G_n.

Although the classes have been redefined, the rules for computing class probabilities when component subgraphs are combined remain unchanged. The altered classes require that the component subnetworks be modified from those used in theorem 4.1, however each performs essentially the same function in both constructions. We shall therefore content ourselves with computing the class probabilities of the component subgraphs and appending a sketch of how they combine.

The class probabilities of the component subgraphs are now listed:

(i) The components K_{ij} have 3 external vertices, s,t,u_{ij} and 5 classes. The scalars are assigned values $p_1=1/2$, $p_2=1/2$, $p_3=2$. The class probabilities are

$$\begin{aligned}
&\Pr([s,t,u_{ij}]) &&= p_1p_2p_3 = 1/2 && \text{(inconsistent)}\\
&\Pr([s,t][u_{ij}]) &&= p_1p_2q_3 = -1/4 && \text{(inconsistent)}\\
&\Pr([s,u_{ij}][t]) &&= p_1q_2p_3 = 1/2 &&\\
&\Pr([t,u_{ij}][s]) &&= q_1p_2p_3 = 1/2 &&\\
&\Pr([s][t][u_{ij}]) &&= p_1q_2q_3 + q_1p_2q_3 + q_1q_2p_3 &&\\
& &&= -1/4-1/4+1/2 &&\\
& &&= 0 && \text{(null)}
\end{aligned}$$

(ii) The components L_{ij} have 2 external vertices s and u_{ij}, and 2 classes. The scalars are assigned values $p_1=-1$, $p_2=1/2$, $p_3=-1$.

$$\begin{aligned}
\Pr([s,u_{ij}]) &= x_{ij}(p_1p_2p_3 + p_1q_2p_3 + p_1p_2q_3 + q_1p_2p_3 + \\
&\qquad q_1q_2p_3 + q_1p_2q_3) + (1-x_{ij})(p_1p_2p_3 + \\
&\qquad p_1q_2p_3 + p_1p_2q_3 + q_1p_2p_3) \\
&= x_{ij}(1/2+1/2-1-1-1+2) + (1-x_{ij})(1/2+1/2-1-1) \\
&= x_{ij}-1.
\end{aligned}$$

$$\begin{aligned}
\Pr([s][u_{ij}]) &= x_{ij}(p_1q_2q_3 + q_1q_2q_3) + (1-x_{ij})(p_1q_2q_3 + q_1p_2q_3) \\
&= x_{ij}(-1+2) + (1-x_{ij})(-1+2) \\
&= 1.
\end{aligned}$$

(iii) The components $M_{iji'j'}$ have 3 external vertices u_{ij}, $u_{i'j'}$, s and 5 classes. The scalars are assigned values $p_1=4/9$, $p_2=-7/2$, $p_3=1/2$, $p_4=1/2$, $p_5=-3$.

$$\begin{aligned}
\Pr([s,u_{ij},u_{i'j'}]) &= p_1(p_2p_3p_4p_5 + p_2q_3p_4p_5 + p_2p_3q_4p_5 + q_2p_3p_4p_5) \\
&= 4/9(21/8+21/8+21/8-27/8) \\
&= 2.
\end{aligned}$$

$$\begin{aligned}
\Pr([u_{ij},u_{i'j'}][s]) &= p_1(p_2p_3p_4q_5 + p_2p_3q_4q_5 + p_2q_3p_4q_5 + \\
&\qquad p_2q_3q_4p_5 + q_2p_3p_4q_5) \\
&= 4/9(-7/2-7/2-7/2+21/8+9/2) \\
&= -3/2.
\end{aligned}$$

$$\begin{aligned}
\Pr([s,u_{ij}][u_{i'j'}]) &= p_1q_2p_3q_4p_5 \\
&= -3/2.
\end{aligned}$$

$$\begin{aligned}
\Pr([s,u_{i'j'}][u_{ij}]) &= p_1q_2q_3p_4p_5 \\
&= -3/2.
\end{aligned}$$

$$\begin{aligned}
\Pr([s][u_{ij}][u_{i'j'}]) &= p_1(q_2p_3q_4q_5 + q_2q_3p_4q_5 + q_2q_3q_4p_5) \\
&= 4/9(9/2+9/2+27/8) \\
&= 5/2.
\end{aligned}$$

Let us now combine the component subgraphs, as in the proof of the previous theorem, computing the class probabilities as we proceed. Firstly each pair of component subgraphs K_{ij}, L_{ij} is combined to form a single component H_{ij} with external vertices $\{s,t,u_{ij}\}$. The class probabilities of H_{ij} are

$$\Pr([s,u_{ij}][t] = 1/2[(x_{ij}-1)+1]$$
$$= (1/2)x_{ij}$$

and $$\Pr([t,u_{ij}][s]) = 1/2\cdot 1$$
$$= 1/2.$$

Next, the H_{ij} are combined into a single component H with external vertices $\{s,t,u_{11},\ldots,u_{nn}\}$ and classes $\{C_S \mid S\subseteq\{(i,j)\mid 1\leq i,j\leq n\}\}$ where $C_S = [s,u_{ij}((i,j)\in S)][t,u_{ij}((i,j)\notin S)]$. The class probabilities of H are given by

$$\Pr(C_S) = (1/2)^{n^2}\prod_{(i,j)\in S} x_{ij}.$$

Finally combining H with the component subgraphs $M_{iji'j'}$, using an argument analogous to that used in the previous proof, yields the required result.

The completeness results for the reliability polynomials introduced in section 4.2 follow easily from the above two theorems. □

Corollary 4.3 The polynomial families (i) and (ii) are both complete over the field of rationals, $\mathbb{Q}$:

(i) $\{\text{s-t-CONNECTED}(K_n) \mid n=1,2,..\}$

(ii) $\{\text{CONNECTED}(K_n) \mid n=1,2,..\}$.

Proof

(i) Consider the graph G_n, with s-t-SEPARATED$(G_n)=(3/8)^{n^2}\text{per*}(X)$, whose existence is assured by theorem 4.1. Relabel vertex t as t' and augment the resulting graph to produce G'_n as in fig. 4.7.

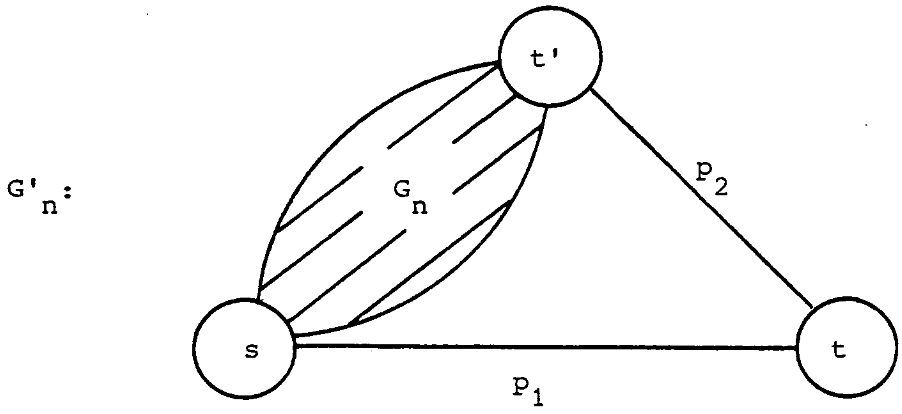

Figure 4.7

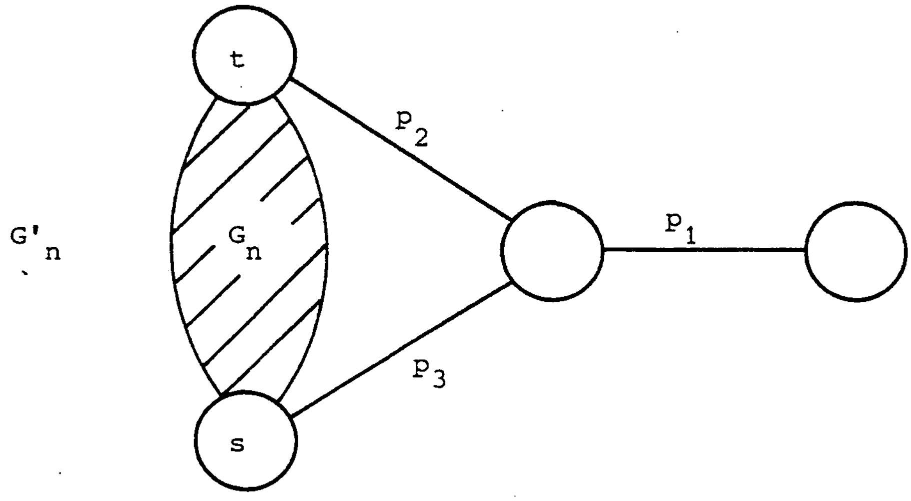

Figure 4.8

The scalars p_1 and p_2 assume the values $(8/3)^{n^2}$ and $[1-(3/8)^{n^2}]^{-1}$ respectively. Then:

$$\begin{aligned}
&\text{s-t-CONNECTED}(G'_n) \\
&= p_1 + (1-p_1)p_2[\text{s-t-CONNECTED}(G_n)] \\
&= p_1 + (1-p_1)p_2[1- \text{s-t-SEPARATED}(G_n)] \\
&= (p_1+p_2-p_1p_2) + (p_1p_2-p_2)[\text{s-t-SEPARATED}(G_n)] \\
&= \text{per}^*(X).
\end{aligned}$$

The result follows from lemma 2.3.

(ii) By theorem 4.2, there exists a graph G_n, of small size, with s-t-PARTITIONED$(G_n)=(1/2)^{n^2}$ per*(X). Augment G_n to G'_n as indicated in fig. 4.8. The scalars are assigned values $p_1=-2^{(n^2+1)}$, $p_2=1/2$, $p_3=-1$. Then:

$$\begin{aligned}
&\text{CONNECTED}(G'_n) \\
&= (p_1p_2p_3 + p_1q_2p_3 + p_1p_2q_3)[\text{CONNECTED}(G_n)] \\
&\quad + p_1p_2p_3[\text{s-t-PARTITIONED}(G_n)] \\
&= \text{per}^*(X).
\end{aligned}$$

A second application of lemma 2.3. yields the result. □

4.5 Networks with Unidirectional Links

A network in which transmission along links takes place in one direction only is modelled by an directed graph. In this section only, we break from our convention, and all graphs mentioned will be directed unless otherwise stated. The notion of two stations of a network communicating is captured as follows. Suppose $G=(V,E)$ is a directed graph, representing a network, and A is a subset of E, representing the functioning links of the network. The edge set A defines a relation on V whereby u,v are related iff $(u,v)\varepsilon A$; define the relation $\overset{A}{\rightarrow}$ on V to be the reflexive, transitive closure of this.

Using the relation $\overset{A}{\rightarrow}$, a number of reliability polynomials can

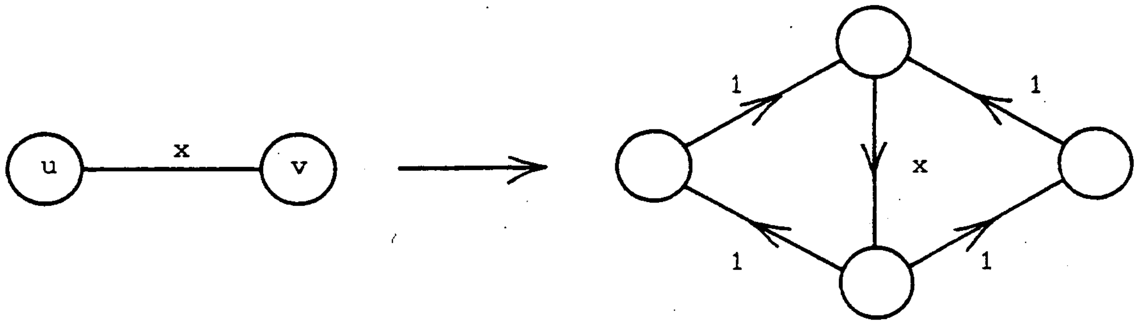

Figure 4.9

be defined for directed graphs. Again we represent by Ω_G the set of elementary events; $\Omega_G = 2^E$.

$$\text{s-t-CONNECTED}(G)=\Pr\{A\varepsilon\Omega_G | s \overset{A}{\rightarrow} t\} \tag{4.6}$$

$$\text{STRONGLY-s-t-CONNECTED}(G)=\Pr\{A\varepsilon\Omega_G | (s \overset{A}{\rightarrow} t)\wedge(t \overset{A}{\rightarrow} s)\} \tag{4.7}$$

$$\text{s-V-CONNECTED}(G)=\Pr\{A\varepsilon\Omega_G | \forall v\varepsilon V, s \overset{A}{\rightarrow} v\} \tag{4.8}$$

$$\text{CONNECTED}(G)=\Pr\{A\varepsilon\Omega_G | \forall u,v\varepsilon V, (u \overset{A}{\rightarrow} v)\wedge(v \overset{A}{\rightarrow} u)\} \tag{4.9}$$

The polynomial (4.6) represents the probability that s can communicate to t, (4.7) represents the probability that s and t can communicate with each other, (4.8) the probability that s can communicate to all other stations, and (4.9) that all pairs of stations can communicate.

As in the undirected case, we generate polynomial families by considering the reliability polynomials, for the complete graph K_n on n vertices, for increasing values of n. The completeness of these polynomial families is a direct corollory of the results obtained in the previous section for undirected graphs.

Corollory 4.4 The following polynomial families are complete over the field of rationals $\mathbb{Q}$.

(i) $\{\text{s-t-CONNECTED}(K_n) | n=1,2,..\}$.

(ii) $\{\text{STRONGLY-s-t-CONNECTED}(K_n) | n=1,2,..\}$.

(iii) $\{\text{s-V-CONNECTED}(K_n) | n=1,2,..\}$.

(iv) $\{\text{CONNECTED}(K_n) | n=1,2,..\}$.

Proof An arbitrary undirected graph G=(V,E) may be transformed into a directed graph G' by replacing each edge $\{u,v\}\varepsilon E$ by the subgraph of fig. 4.9, consisting of 4 vertices and 5 edges. The communication probabilities are unaltered by this transformation and the following identities hold:

$$\text{s-t-CONNECTED}(G') = \text{s-t-CONNECTED}(G)$$

$$\text{STRONGLY-s-t-CONNECTED}(G') = \text{s-t-CONNECTED}(G)$$

$$\text{s-V-CONNECTED}(G') = \text{CONNECTED}(G)$$

and $\text{CONNECTED}(G') = \text{CONNECTED}(G)$.

Note that the reliability polynomials on the left hand sides are for directed graphs, while those on the right hand sides are for undirected graphs. By taking G to be the complete undirected graph on n vertices, we see that we have exhibited projections from the polynomial families (i)-(iv) onto other polynomial families which are known, by corollory 4.3 to be complete. □

4.6 Discussion

A number of reliability measures for networks have been proposed, and all have been shown to be complete in the sense of chapter 2. This observation suggests that computing network reliability is an inherently difficult task. Intuitively, what makes computation of such measures difficult is the subtlety of the probabilistic dependencies; it is impossible to decompose a reliability measure so that its dependence on individual failure probabilities becomes explicit. In fact reliability measures seemingly much easier than the ones studied in sections 4.4 and 4.5 appear to be intractable. As an example consider the following reliability polynomial for an undirected graph $G=(V,E)$

$$\text{NO-ISOLATED-VERTEX}(G)=\Pr\{A\varepsilon\Omega_G \mid \text{every component of } (V,A) \text{ has order at least } 2\}.$$

The intuitive reading of this is the probability that every station of a network can communicate with some other station. One might expect this to be easy to compute, as it represents the probability of an event which is determined by purely local considerations at each vertex; the previously referred to probabilistic dependencies are reduced to a minimum in this case.

The intractability of this measure is, however, easily demonstrated. If $p(x_1,\ldots,x_n)$ is a polynomial of degree d in n indeterminates, let le(p) denote the lower envelope of p, that is to say the sum of all the monomials of lowest degree in p. We remark that the lower envelope of a polynomial is not substantially easier to compute than the polynomial itself. For suppose we wish to evaluate le(p) at the point $(\alpha_1,\alpha_2,\ldots,\alpha_n)$. Consider the polynomial $p(\lambda\alpha_1,\lambda\alpha_2,\ldots,\lambda\alpha_n)$ of degree d in the single indeterminate λ; the value we wish to compute is precisely the coefficient of the term of lowest degree in λ. The coefficients of $p(\lambda\alpha_1,\ldots,\lambda\alpha_n)$ can be determined by evaluating the polynomial at d+1 distinct values of λ. In this way the evaluation of le(p) has been reduced to d+1 evaluations of p. Now, if $K_{n,n}$ is the complete bipartite graph on 2n vertices with the usual labelling, then the monomials of le(NO-ISOLATED-VERTEX($K_{n,n}$)) correspond to partial graphs of $K_{n,n}$ in which every component has order exactly 2, i.e. to perfect matchings in $K_{n,n}$. Hence

$$\text{le(NO-ISOLATED-VERTEX}(K_{n,n}))=\text{per}(X).$$

The lower envelope is thus complete, and, by the above discussion, the reliability polynomial itself difficult to compute.

The objection might be raised that our reductions employ constants outside the range [0,1] of realisable probabilities. However an appeal to intuition suggests that it is no easier to compute a multivariate polynomial when we restrict all values of its indeterminates to a certain range (say[0,1]) than it is to compute it for arbitrary values. By way of justification, we might remark that an arithmetic circuit which correctly computes the polynomial within the restricted range will work (except at a few singularities where division by zero occurs) over the whole range.

We finish with a caveat. The results obtained here are for arbitrary networks; it is conceivable that computing the reliability of some useful subclass of networks, for example planar networks, is radically easier than the general case. Although in the field of network reliability this seems unlikely, there is a precedent for this effect in Kasteleyn's method for enumerating perfect matchings in a planar graph, which was cited in chapter 3.

4.7 New Completeness Results from Old

As has been remarked in chapter 1, our approach differs from that of machine-based complexity theory in two respects. Firstly, our notions are non-uniform, for example the p-projections used to reduce one polynomial family to another which specify a separate translation for each member of the family (i.e. for each input size). In practice this is of no consequence; the polynomial families which we have considered are certainly uniform (i.e. can be described by an effective procedure), while the reductions of this and the last chapter are not only Turing computable but efficiently so. The second difference is potentially more important - we have viewed problems through the medium of generating functions rather than as pure combinatorial enumeration. There is a doubt that additional complexity may be introduced when we move from the discrete combinatorial world to the continuous algebraic one.

Although it should be stressed that the algebraic completeness results we have obtained are strong statements of intractability in their own right, it might be illuminating to give an example of how such a result can be used, with not too much effort, to prove a statement about the complexity of an associated combinatorial enumeration problem.

We assume familiarity with the class #P and its associated completeness class #P-complete introduced by Valiant in [40,41]. In brief, #P is the class of integer functions computed by polynomial time bounded counting Turing machines. A counting T.M. is a standard non-deterministic T.M. with the additional facility of outputting the number of accepting computations. A problem is #P-complete if it is complete in #P with respect to polynomial time Turing reduction. The class #P-complete includes many classical 'hard' enumeration problems. In particular, the following problem is known to be #P-complete (for a proof see [40]):

0-1 PERMANENT

Input: 0-1 matrix U.

Output: per(U).

We introduce an enumeration problem, which is closely associated with one of the reliability measures defined in this chapter, and show it to be #P-complete using theorem 4.2. The problem is:

#CONNECTED P-GRAPHS

Input: graph G

Output: the number of connected partial graphs of G.

Two points should be emphasised. Firstly, although the proof of #P-completeness is ad hoc, the techniques employed, namely polynomial interpolation and the 'encoding' of field elements, are probably of wider application in this area. Secondly, no direct proof is known of #P-completeness of the above problem. The proof of theorem 4.7 may be interpreted as further evidence of the utility of the algebraic approach.

Two preparatory lemmata are required. For computational purposes, we shall assume that graphs are represented in some standard form, for example as adjacency matrices. Rational numbers will be held as pairs of binary integers, representing the numerator and denominator of a fraction in reduced form. The length of such a representation of a rational q will be denoted by $|q|$.

Lemma 4.5 Suppose $p(x_1,\dots,x_k)$ is a polynomial in k indeterminates, of degree d, and with rational coefficients. Let $A=\{q_1,q_2,\dots,q_{d+1}\}$ be a set of distinct rationals. Then the coefficients of p can be computed, in deterministic polynomial time, from the set of $(d+1)+(d+1)^k$ values

$$A\cup\{p(a_1,\dots,a_k)\,|\,\underline{a}\in A^k\}$$

Proof Firstly we claim that if two polynomials $f(x_1,\dots,x_k)$ and $g(x_1,\dots,x_k)$, of degree d, agree at all points in the set

$$\{q_1,\dots,q_{d+1}\}^k$$

then they are identically equal. Setting

$$h(x_1,\dots,x_k) = f(x_1,\dots,x_k) - g(x_1,\dots,x_k),$$

the claim is equivalent to showing that h is identically 0. This is a slight extension of the fundamental theorem of algebra (see Godement [10]) which can be proved by straightforward induction on k. Now consider the polynomial

$$f(\underline{x}) = \sum_{\underline{a}\varepsilon A^k}\left[p(\underline{a})\prod_{1\leq i\leq k}\ \prod_{q_j\neq a_i}\frac{(x_i-q_j)}{(a_i-q_j)}\right]. \qquad (4.10)$$

If $\underline{x}\varepsilon A^k$, all but one of the terms in the sum are zero; the remaining term is equal to $p(\underline{x})$. Hence f agrees with p on all points of A^k, and, by our claim, f is identically equal to p. The formula 4.10 is thus an explicit expression of p in terms of $\{q_1,\ldots,q_{d+1}\}$ and $\{p(\underline{a})\,|\,\underline{a}\in A^k\}$. It only remains to show that the coefficients of p may be computed, using expression 4.10, in time polynomial in the input length, l. This should be apparent from the following observations.

(i) If the input length is l, then $|q_i|\leq l \;\; \forall i$, $|p(\underline{a})|\leq l \;\; \forall \underline{a}\in A^k$, and $(d+1)^k \leq l$.

(ii) The polynomials manipulated in the computation may be represented by vectors of coefficients having $(d+1)^k$ $(\leq l)$ components.

(iii) The sum in (4.10) consists of $(d+1)^k$ $(\leq l)$ terms, each having kd $(\leq l)$ factors.

(iv) At no point in the computation do we need to handle rationals whose representation requires more than $O(l^3)$ space.

Lemma 4.6 Suppose that G is a graph with label set $\Lambda=\{1/2,1/3,\ldots,1/k\}$. Let the number of edges with label $1/j$ be n_j. Then there exists an efficiently computable unlabelled graph H with the property

$$\text{CONNECTED}(G)=\prod_{2\leq j\leq k} j^{-n_j}\times(\text{number of connected partial graphs of } H.)$$

Proof Suppose that $G=(V,E_G)$. The graph $H=(V,E_H)$ is constructed from G by simply replacing each $1/j$ labelled edge, e, of G by a chain, C_e, of $(j-1)$ unlabelled edges, the endpoints of C_e being the same as those of the original edge e. Suppose $A_H\subseteq E_H$ is such that (V,A_H) is connected. For each chain of edges C_e in H, A_H either contains all the edges of C_e, or contains all but one of the edges. (Otherwise, vertices of C_e would exist which are isolated from the endpoints.) Define a mapping ν, from connected partial graphs of H to connected partial graphs of G, as follows:

$$\nu(A_H) = \{e \in E_G | A_H \text{ contains all the edges in } C_e\}.$$

The mapping ν is surjective, but not injective. However a simple expression exists for the number of partial graphs of H which map to a fixed one (V,A_G) of G:

$$|\{A_H \subseteqq E_H | \ \nu(A_H)=A_G\}| = \prod_{e\in E_G - A_G} (\lambda(e)^{-1}-1).$$

(The factors in the product correspond to the number of ways of choosing a single edge from the chain C_e.) Hence:

$$|\{A_H \subseteqq E_H | \nu(A_H)=A_G\}|$$
$$= \left(\prod_{e\in E_G} \lambda(e)^{-1}\right)\left(\prod_{e\in A_G} \lambda(e)\right)\left(\prod_{e\in E_G - A_G} (1-\lambda(e))\right)$$
$$= \left(\prod_{2\leqq j\leqq k} j^{n_j}\right) p(A_G)$$

The result follows by summation over all connected partial graphs (V,A_G) of G. □

Theorem 4.7 #CONNECTED P-GRAPHS is #P-complete.

Proof That the problem is in #P is immediate - given a graph G we simply test in parallel each partial graph of G and accept iff it is connected. The testing can be done in time $O|G|)$ by using, for example, depth first search [8]. Therefore it is sufficient to show that 0-1 PERMANENT is polynomial time Turing reducible to #CONNECTED P-GRAPHS.

Suppose U is an n×n 0-1 matrix. Combining the projections explicitly presented in lemma 2.3, theorem 4.2 and corollory 4.3 we see that a graph $G_{U,\underline{\alpha}}$ exists with the following properties.

(i) $\quad \mathrm{per}(U) = \mathrm{CONNECTED}(G_{U,\underline{\alpha}})$.

(ii) $G_{U,\underline{\alpha}}$ is efficiently constructable from U, i.e. the mapping $U \mapsto G_{U,\underline{\alpha}}$ is computable in deterministic polynomial time.

(iii) The label set Λ of $G_{U,\underline{\alpha}}$ is $\{\alpha_1,\ldots,\alpha_k\}$ where k is a fixed integer (independent of n), and $\alpha_i \varepsilon Q$. (Explicitly k=9 and $\{\alpha_1,\ldots,\alpha_k\} = \{-2^{n^2-1},-7/2,-3,-1,0,4/9,1/2,1,2\}$.)

Let $G_{U,\underline{x}}$ be the graph formed from $G_{U,\underline{\alpha}}$ by replacing each rational label α_i by an indeterminate x_i, let $p_U(x_1,\ldots,x_k)$=CONNECTED($G_{U,\underline{x}}$), and let d be the degree of p_U. We remark that the coefficients of p_U are integers in the range $[0,2^d]$ and that $d(=|G_{U,\underline{x}}|)$ is bounded by a polynomial function of n. By lemma 4.5, the coefficients of p_U and hence the value of $p_U(\alpha_1,\ldots,\alpha_k)$ may be efficiently computed if we know the set of values

$$\{p_U(b_1,\ldots,b_k)|\underline{b}\varepsilon\beta^k\}$$

where $\beta=\{1/2,1/3,\ldots,1/(d+2)\}$. However we note that $p_U(b_1,\ldots,b_k)$ is just CONNECTED($G_{U,\underline{b}}$), where $G_{U,\underline{b}}$ is the graph formed from $G_{U,\underline{x}}$ by replacing each label x_i by the rational b_i. By lemma 4.6 there is a graph $H_{U,\underline{b}}$ and a rational $q_{U,\underline{b}}$, both efficiently computable from $G_{U,\underline{b}}$ such that:

CONNECTED($G_{U,\underline{b}}$)

= $q_{U,\underline{b}}\times$(number of connected partial graphs of $H_{U,\underline{b}}$).

The whole reduction is now summarised:

(i) Compute $G_{U,\underline{x}}$.

(ii) Compute $\{G_{U,\underline{b}}|\underline{b}\varepsilon\beta^k\}$

(iii) Compute $\{H_{U,\underline{b}}|\underline{b}\varepsilon\beta^k\}$ using lemma 4.6.

(iv) Use a subroutine for #CONNECTED P-GRAPHS to enumerate, for each of the graphs found at stage (iii), the number of connected partial graphs.

(v) Scale the results from (iv) in order to obtain

$\{\mathrm{CONNECTED}(G_{U,\underline{b}}) | \underline{b} \varepsilon \beta^k\}$.

(vi) Apply the algorithm of lemma 4.5 to compute

$\mathrm{per}(U) = \mathrm{CONNECTED}(G_{U,\underline{\alpha}})$. □

Note: Recently, a direct proof of the result of Theorem 4.7 has been provided by J.S. Provan and M.O. Ball. (See "The Complexity of Counting Cuts and of Computing the Probability that a Graph is Connected", University of Maryland working paper MS/S 81-002, (1981).)

5. EXACT LOWER BOUNDS FOR RESTRICTED ALGEBRAIC MODELS

5.1 Introduction

As remarked in chapter 1, the topic of this final chapter is in some sense distinct from that of the previous three. The common thread which binds the two parts is the idea of a model of computation which is non-uniform and in which arithmetic operations are elementary. In chapters two to four the underlying model is implicit and the results are of a relative nature; in this chapter the computational model is precisely defined and the complexity results obtained are in many cases exact.

We shall be considering the question of finding the number of arithmetic operations required to compute various polynomial functions, by now a classic goal of complexity theory. If we allow operations to be drawn from the set {+,×,-}, or possibly {+,×,-,/}, then it is an unresolved question how many of these operations are required to compute seemingly such a simple function as matrix multiplication. Profound algebraic methods are required to obtain all but the most trivial results in such a system and, indeed, fast algorithms can be built using non-trivial algebraic properties of the domain of computation. Examples of techniques used to provide lower bounds in arithmetic complexity are to be found in Borodin and Munro [5], while Strassen's celebrated fast matrix multiplication algorithm [37] is an illustration of the possibilities which exist for subtle exploitation of the properties of the domain of computation. Such exploitation reaches cunning heights in the work of Pan [24] and Bini et al. [4].

An obvious and cowardly escape from the convolutions of the general problem is provided by restricting the computational model in some way, and this is the path we shall be following in the present chapter. In the field of Boolean complexity, for example, much work has been done on monotone Boolean computations, analysis of which has proved more tractable than computations using negations (see [15,17,25,27,46]). Similar work has been undertaken, by Schnorr, Shamir and Snir, on monotone arithmetic computations, that is computations using only positive constants, additions and multiplications [32,34,35,36]. In both models it is possible to prove that multiplication of n×n matrices requires n^3 scalar multiplications. Of the same flavour are results concerning regular expressions not using complementation or intersection [7,11].

In order to justify considering restricted computational models a number of desirable features of such models may be listed. Miller [18], for example, shows that monotone arithmetic computations have absolute numerical stability. Such computations also possess a kind of universality, stemming from the property that their correctness may be deduced merely from the associativity, commutativity and distributivity of addition and multiplication. By redefining the operations of addition and multiplication suitably, therefore, we may reinterpret the computation in a number of different domains; this is a feature of monotone arithmetic which we shall be returning to later. Perhaps the main argument in favour, however, is that considering restricted models gives us insight into where the power of more general models lies; we shall show, for example, that introducing negative constants into the domain of computation enables a startling gain in efficiency to be made in the computation of certain polynomials.

The material described in this chapter is motivated by computation in the semiring of non-negative real numbers with the usual addition and multiplication (monotone arithmetic). The results obtained are, however, valid for a number of other (easily characterisable) semirings, and the treatment will therefore be given in a general setting. The results apply, for example, to monotone arithmetic and to computation in the semiring of real numbers with the operations of minimum and addition. This latter structure has frequently been used (for example by Aho et al. [1] and Cuninghame-Green [6]) to formulate and solve optimisation problems.

Later in this chapter we show that the problem of computing a polynomial function in these semirings is reducible to the problem of computing a formal polynomial over the semiring. This in turn is as hard as computing a formal polynomial over the Boolean semiring B ({0,1} with the two operations or, and). Formal polynomials over B are essentially finite sets of integer valued vectors with addition being union and multiplication being componentwise addition. Computations in this semiring are combinatorial in character, and in section 5.4 a combinatorial method is developed which yields lower bounds on the number of multiplications needed to compute certain polynomials. This is achieved essentially by abstracting from the computational task considered a suitable combinatorial optimisation problem. In section 5.5, the technique is applied to several specific polynomials and precise lower bounds obtained on the number of multiplications required to compute them. A discussion of the results follows in section 5.6.

5.2 Semirings, Polynomials and Computations

Although the algebraic terminology we shall be using is fairly standard, we begin this section with a brief review.

A semiring is a system $(S, \oplus, \otimes, 0, 1)$, where S is a set, $\oplus$ (addition) and $\otimes$ (multiplication) are binary operations on S, and 0 and 1 are elements of S having the following properties:

(i) $(S, \oplus, 0)$ is a commutative monoid, that is $\oplus$ is associative and commutative and 0 is an identity.

(ii) $(S, \otimes, 1)$ is a commutative monoid.

(iii) $\otimes$ distributes over $\oplus$, that is $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$.

(iv) $a \otimes 0 = 0$.

The semirings we shall be using are the following:

(i) The Boolean semiring $B=(\{0,1\}, \vee, \wedge, 0,1)$ ($\vee$ being Boolean disjunction, $\wedge$ being conjunction).

(ii) The semiring $R=(\mathbb{R}^{+},+,\cdot,0,1)$ of non-negative real numbers with the usual addition and multiplication.

(iii) The semiring $M=(\mathbb{R}^{*},\min,+,+\infty,0)$, where $\mathbb{R}^{*}=\mathbb{R}\cup\{+\infty\}$, min is the binary minimum operator and + is the usual addition.

(iv) The semiring $M^{+}=(\mathbb{R}^{+*},\min,+,+\infty,0)$, which is the subsemiring of M obtained by restricting the domain to non-negative real numbers.

Let S be a semiring and $X = \{x_1,\dots,x_n\}$ be a finite set of indeterminates. Denote by $S[X]$ the semiring of (formal) polynomials obtained from S by adjunction of the indeterminates $x_1,\dots,x_n$. Each monomial $m=x_1^{i_1}\dots x_n^{i_n}$ is uniquely determined by the vector of exponents $(i_1,\dots,i_n)$, so that we can identify monomials with elements of $\mathbb{N}^n$. Each polynomial $p \varepsilon S[X]$ may uniquely be written in the form

$$p = \bigoplus_{(i_1,\dots,i_n)\varepsilon \mathbb{N}^n} a_{i_1\dots i_n} x_1^{i_1} \dots x_n^{i_n} \qquad (5.1)$$

where only finitely many coefficients $a_{i_1\dots i_n}$ are different from zero, so we may identify formal polynomials with functions from $\mathbb{N}^n$ to S with finite support. Thus if $p \varepsilon S[X]$, $m \varepsilon \mathbb{N}^n$, then p_m will denote the value of the coefficient of p with index vector m and 2.1 may be rewritten as

$$p = \bigoplus_{m \varepsilon \mathbb{N}^n} p_m m. \qquad (5.2)$$

S is imbedded in S[X] by identifying each element $s \varepsilon S$ with the constant polynomial $sx_1^0 \ldots x_n^0$. (For a more elaborate treatment see for example [29], §67.)

Some terminology concerning the polynomial semiring is now introduced. We assume henceforth that p is a polynomial given by equation 5.2 and that $m=(i_1, \ldots, i_n)$ is a monomial. Define the monomial set of p by

$$\text{mon}(p)=\{m \varepsilon \mathbb{N}^n | p_m \neq 0\},$$

the degree of m by

$$\deg(m) = \sum_{j=1}^{n} i_j$$

and the degree of p by

$$\deg(p)=\max\{\deg(m) | m \in \text{mon}(p)\}.$$

The polynomial p is said to be homogeneous if all its monomials have the same degree; m is linear if $m \varepsilon \{0,1\}^n$ and p is linear if all its monomials are linear.

Note that the formal polynomials so far introduced are purely syntactic objects. We can however define a natural mapping ν which assigns to each formal polynomial a functional interpretation. If $p \varepsilon S[X]$ is a formal polynomial then the associated polynomial function $\nu p: S^n \rightarrow S$ is the function whose value at $(a_1, \ldots, a_n)$ is obtained by substituting a_i for x_i in p. The map ν is a homomorphism from S[X] to the semiring of functions $[S^n \rightarrow S]$ with pointwise addition and multiplication. We denote by $P_n(S)$ the image of $S[X]$ under ν, that is the subsemiring of polynomial functions. The map ν need not be injective as two different polynomials, e.g. x and x^2 in $B[X]$, can represent the same function.

The model of computation and its associated complexity measures will now be introduced. (Certain terminology from graph theory will be employed, which can be referenced in chapter 3.) Let S be a semiring. A *computation* Γ in S with *input set* $I \subset S$ is a labelled, directed acyclic graph (d.a.g.) with the following properties:

(i) Vertices of Γ with indegree 0, termed *input vertices*, are labelled by elements of I.

(ii) The vertices of Γ which are not input vertices all have indegree 2 and are labelled either by $\oplus$ or $\otimes$.

(iii) There is a unique vertex, ρ, of Γ, of outdegree 0, termed the *result vertex*.

Let V be the vertex set of Γ and let $V_\oplus$, $V_\otimes$ respectively be the $\oplus$ and $\otimes$ labelled elements of V. If $\alpha,\beta \varepsilon V$ and there is an edge in Γ directed from α to β, then α is a *predecessor* of β, and β a *successor* of α. The *ancestor* relation is the transitive closure of the predecessor relation; the *descendant* relation is the transitive closure of the successor relation.

A result function, res: $V \to S$, is defined recursively on the vertices of Γ in the following manner:

(i) If α is an input vertex labelled by $i \varepsilon I$ then $\text{res}(\alpha)=i$.

(ii) If $\alpha \varepsilon V_\oplus$ with predecessors β,γ then $\text{res}(\alpha)=\text{res}(\beta) \oplus \text{res}(\gamma)$.

(iii) If $\alpha \varepsilon V_\otimes$ with predecessors β,γ then $\text{res}(\alpha)=\text{res}(\beta) \otimes \text{res}(\gamma)$.

Note that the condition that Γ is acyclic ensures that res is well-defined. We say that *Γ computes s* if $\text{res}(\rho)=s$, where ρ is the result vertex of Γ.

The $\otimes(\oplus)$-complexity of Γ is simply the cardinality of $V_{\otimes}(V_{\oplus})$. The $\otimes(\oplus)$-complexity of $s\varepsilon S$ with respect to $I\subset S$ is the minimum $\otimes(\oplus)$-complexity of a computation with input set I computing s. Of particular interest to us will be computations of polynomials in $S[X]$, and polynomial functions in $P_n(S)$. For computations in $S[X]$ the input set will always be assumed to be $S\cup X$ and for computations in $P_n(S)$ it will accordingly consist of the constant functions and projection functions. Thus the $\otimes(\oplus)$-complexity of a formal polynomial or polynomial function will be understood to mean the $\otimes(\oplus)$-complexity with respect to these input sets.

Whenever an algebraic structure is **a homomorphic image of another,** computations in the first structure are related to computations in the second, and so complexity results for the second structure translate into results for the first. Indeed we have:

Lemma 5.1 Let S,S' be semirings, $\tau:S\rightarrow S'$ be a homomorphism. Let Γ compute $s\varepsilon S$ with input set $I\subset S$. Let Γ' be obtained from Γ by relabelling each input vertex with label $i\varepsilon I$ by $\tau(i)$. Then Γ' is a computation in S' with input set $\tau(I)$; for each vertex α of Γ, if $r=\mathrm{res}(\alpha)$, then $\tau(r)$ is the result of α in Γ'. In particular Γ' computes $\tau(s)$.

Proof Easy induction on V □

and in consequence:

Corollary 5.2: Let S,S' semirings, $\tau:S\rightarrow S'$ be a homomorphism.

(i) The $\otimes(\oplus)$-complexity of $s\varepsilon S$ with respect to $I\subset S$ is no smaller than the $\otimes(\oplus)$-complexity of $\tau(s)$ with respect to $\tau(I)$.

(ii) If τ is surjective, then the $\otimes(\oplus)$-complexity of $s'\varepsilon S'$ with respect to $I\subset S'$ is equal to the minimum $\otimes(\oplus)$-complexity of an element $s\varepsilon\tau^{-1}(s')$ with respect to $\tau^{-1}(I)$.

Proof Immediate from lemma 5.1 □

As an important application of cor. 5.2 we have:

Corollory 5.3: The $\otimes(\oplus)$-complexity of a polynomial function is equal to the minimum $\otimes(\oplus)$-complexity of a polynomial representing it.

Proof: Take τ in cor. 5.2 to be the canonical homomorphism ν from polynomials to polynomial functions. □

The foregoing observation is especially useful in semirings where each polynomial function is represented by a unique polynomial; the semiring R is such a case. For such semirings the $\otimes(\oplus)$-complexity of a polynomial and of the function it represents are equal. This is not true in general for the semirings M, M^+ where there is no unique representation of polynomial functions. The next section of the chapter will deal with this problem.

Our complexity results will be derived in the first instance for polynomials in $B[X]$. These results can be extended using cor. 5.2 to any other polynomial semiring $S[X]$, provided that we can exhibit a homomorphism from $S[X]$ to $B[X]$ mapping $S \cup X$ into $B \cup X$. But any homomorphism $\tau: S \to B$ extends naturally to a homomorphism which maps S into B and x_i onto itself. For all three semirings R, M, M^+ such a homomorphism exists, and is given by

$$\tau(a) = \begin{cases} 0_B \text{ iff } a=0_S \\ \\ 1_B \text{ iff } a \neq 0_S \end{cases} \qquad (5.3)$$

(0_S is 0 in R and $+\infty$ in M, M^+).

Two points are perhaps worth making at this juncture. Firstly, τ maps polynomials with 0-1 coefficients into formally identical polynomials, and thus, any lower bound obtained for the $\otimes(\oplus)$-complexity of a polynomial $p \in B[X]$ yields immediate lower bounds on the $\otimes(\oplus)$-complexity of the formally identical polynomials in $R[X]$, $M[X]$ and $M^+[X]$. Secondly, it is at this point that the method presented here for obtaining lower bounds would formally break down were we to attempt to apply it to general arithmetic computations (with negative constants). For in the general case, taking S to be the semiring $(\mathbb{R},+,\cdot,0,1)$, the map τ defined by equation 5.3 would no longer be a homomorphism $(\tau(1_S)+\tau(-1_S)=1_B$, $\tau(0_S)=0_B)$. That 5.3 defines a homomorphism is a characterisation of the semirings to which the lower bounds obtained in section 5.5 apply; we might term such semirings monotone.

As has been remarked, in the case of $M[X]$ and $M^+[X]$ the canonical homomorphism, ν, from formal polynomials to polynomial functions, is not an isomorphism. The next section - which is self contained and can be omitted - establishes the machinery required to deal with this problem.

5.3 Envelopes and Computations in min,+.

As will be seen, the methods presented in the following section for obtaining lower bounds are applicable only to homogeneous polynomials. It is possible, however, to extract, from any polynomial, homogeneous components which are simpler to compute than the polynomial itself. By arguing about these components it is therefore possible to obtain lower bounds on the complexity of non-homogeneous polynomials.

Let $p\varepsilon S[X]$ be given by equation 5.2, and let $k=\min\{\deg(m)\,|\,m\varepsilon\mathrm{mon}(p)\}$. The lower envelope of p is given by

$$\mathrm{le}(p) = \bigoplus_{\deg(m)=k} p_m\, m.$$

Similarly, if $K=\max\{\deg(m)\,|\,m\varepsilon\mathrm{mon}(p)\}$, then the higher envelope of p is given by

$$\mathrm{he}(p) = \bigoplus_{\deg(m)=K} p_m\, m.$$

Informally, le(p) (he(p)) is obtained from p by preserving only the terms of minimal (maximal) degree. Assuming we restrict our attention to semirings for which the map τ, defined by equation 5.3, is indeed a homomorphism, the following properties of lower envelopes can be deduced ($p_1, p_2 \varepsilon S[X]$):

(i) If $\deg(le(p_1))=\deg(le(p_2))$ then

$le(p_1 \oplus p_2)=le(p_1) \oplus le(p_2)$.

(ii) If $\deg(le(p_1))<\deg(le(p_2))$ then $le(p_1 \oplus p_2)=le(p_1)$.

(iii) $le(p_1 \otimes p_2)=le(p_1) \otimes le(p_2)$.

Similar relations hold for the higher envelope. The complexity of a polynomial and that of its higher and lower envelopes are related as follows:

Lemma 5.4 The $\otimes(\oplus)$-complexity of p is no smaller than the $\otimes(\oplus)$-complexity of $le(p)$ $(he(p))$.

Proof From the properties of lower and higher envelopes listed above, it is clear that any computation for $p\varepsilon S[X]$ may be restructured, by appropriately discarding some of its additions, into a computation of $le(p)$ $(he(p))$. The additions to be discarded are those whose summands have lower (higher) envelopes of unequal degrees. □

Let us now turn to the semirings M and M^+. We shall investigate how the structure of a polynomial is determined by the function it represents. We assume that $p\varepsilon M[X]$ $(p\varepsilon M^+[X])$ is given by

$$p=\bigoplus_{i=1}^{k} c_i m_i$$

where $c_i \neq +\infty$, $m_i \varepsilon N^n$. The function f represented by p is

$$f(u)=f(u_1,\ldots,u_n)=\min_{1\leq i\leq k}(<m_i \cdot u>+c_i),$$

where $<u\cdot v>$ denotes the scalar product of u and v. We shall obtain a characterisation of the class of polynomials which represent a given function f; this characterisation rests on the basic separation theorem in convexity theory, due to Farkas, whose statement follows.

Theorem 5.5 Let $a, a_i \varepsilon \mathbb{R}^n$, $b, b_i \varepsilon \mathbb{R}$ for $i=1,\ldots,k$. The following two assertions are equivalent:

(i) The system of inequalities

$\langle a_i \cdot u \rangle \geqq b_i \qquad i=1,\ldots,k$

implies the inequality

$\langle a \cdot u \rangle \geqq b$.

(ii) $\exists \lambda_1,\ldots,\lambda_k$ such that

$\lambda_i \geqq 0$

$a = \sum \lambda_i a_i$

$b \leqq \sum \lambda_i b_i$

Proof See [9], theorem 4. □

The following theorem, informally stated, tells us that any polynomial representing a given function is composed of a fixed set of "essential terms" together with a (possibly empty) set of "redundant terms"; the set of possible redundant terms has an elegant characterisation.

Theorem 5.6 Let $f \varepsilon P_n(M)$ be a polynomial function over M. There exists a unique set of terms $T=\{c_i m_i \mid 1 \leqq i \leqq t\}$. such that if p represents f in M[X] then

(i) Each term of T occurs in p;

(ii) If cm is a term of p then there exist $\lambda_1,\ldots,\lambda_t$ such that:

$\lambda_i \geqq 0$, $i=1,\ldots,t$;

$\sum \lambda_i = 1$;

$m = \sum \lambda_i m_i$;

$c \geqq \sum \lambda_i c_i$.

Proof   Associate with f the set $\underline{Gr}(f) \subset \mathbb{R}^{n+1}$ which is bounded above by the graph of f.

$$\underline{Gr}(f) = \{(u_1,\dots,u_n,v) \mid v \leqq f(u_1,\dots,u_n)\}$$
$$= \{(u,v) \mid v \leqq (\langle m_i \cdot u\rangle + c_i) \text{ for } i=1,\dots,k\}.$$

$\underline{Gr}(f)$ is the intersection of k closed halfspaces corresponding to the k terms of p, and has non-empty interior (unless $p=-\infty$). There is a unique minimal family of halfspaces whose intersection yields $\underline{Gr}(f)$, each halfspace being bounded by a hyperplane which contains one of the n-dimensional faces of the n+1 dimensional polyhedron $\underline{Gr}(f)$. It follows that there is a unique set T of terms of p which appear in any polynomial representing f. This deals with part (i) of the theorem - the characterisation of the remaining terms of p follows almost immediately from theorem 5.5. For if cm is a redundant term of p then:

$$\forall u \in \mathbb{R}^n, \ \langle m \cdot u\rangle + c \geqq \min_{1 \leqq i \leqq t} (\langle m_i \cdot u\rangle + c_i)$$

which is equivalent to the assertion that in $\mathbb{R}^{n+1}$ the system of inequalities

$$\langle m_i \cdot u\rangle + c_i \geqq u_{n+1}, \quad i=1,\dots,t$$

implies the inequality

$$\langle m \cdot u\rangle + c \geqq u_{n+1}$$

where $u_{n+1} \in \mathbb{R}$ is an independent variable. Denoting the vector $(u_1,\dots,u_n,u_{n+1})$ by u^*, the assertion may be rewritten as

$$\langle (m_i,-1) \cdot u^*\rangle \geqq -c_i, \ i=1,\dots,t$$

implies

$$\langle (m,-1) \cdot u^*\rangle \geqq -c,$$

which by theorem 5.5 is equivalent to the existence of $\lambda_1,\dots,\lambda_t$ with the properties

$\lambda_i \geqq 0, \quad i=1,\dots,t$

$(m,-1)= \sum \lambda_i (m_i,-1)$

$-c \leqq \sum \lambda_i (-c_i)$.

The result follows immediately. □

The characterisation of redundant terms supplied by theorem 5.6 yields a unique representation theorem for certain classes of functions.

Theorem 5.7 Let $p,q \in M[x]$ represent the same function. Then

(i) If p is linear then p=q.

(ii) If le(p) (he(p)) is linear then le(p)=le(q) (he(p)=he(q)).

Proof

(i) Let $T=\{c_i m_i\}$ be the set of essential terms occurring both in p and q. We claim that no other term occurs in p or q. Indeed, let cm be a term of p (or q). Then, by theorem 5.6, $m= \sum \lambda_i m_i$ with $\lambda_i \geqq 0$, $\sum \lambda_i =1$. However, the m_i are 0-1 valued vectors and no non-trivial convex combination of them can yield an integer valued vector (the interior of the unit cube does not contain lattice points). Thus the monomial m occurs in T and so $cm \varepsilon T$.

(ii) Let $k=\min_i \deg(m_i)$. We claim that the terms of le(p) (le(q)) are precisely the minimal degree terms of T. If $\deg(m_i)=k$ then $c_i m_i$ occurs in le(p) and le(q). On the other hand, let cm be a term of le(p) (or le(q)). Then deg(m)=k and, by th. 5.6, $m= \sum \lambda_i m_i$ with $\lambda_i \geqq 0$, $\sum \lambda_i =1$. But $\deg(m)= \sum \lambda_i \deg(m_i) \geqq \min \deg(m_i)=k$, and equality can occur only if $\lambda_i=0$ whenever $\deg(m_i)>k$. Thus m is a convex combination of the minimum degree monomials in T, and, by the same argument used in (i), it follows that $cm \varepsilon T$. The proof for higher envelopes is similar. □

The relevance of the unique representation theorem is that it allows us to relate the complexities of polynomial functions and formal polynomials representing them.

Corollary 5.8 Let $p\varepsilon M[X]$ represent the function $f\varepsilon P_n(m)$. Then:

(i) If p is linear, then the $\otimes(\oplus)$-complexity of f is equal to the $\otimes(\oplus)$-complexity of p.

(ii) If le(p) (he(p)) is linear, then the $\otimes(\oplus)$-complexity of f is no smaller than the $\otimes(\oplus)$-complexity of le(p) (he(p)).

Proof Use cor. 5.3, th. 5.4 and th. 5.7. □

When the domain of computation is restricted to non-negative numbers, there is greater freedom in choosing representations for functions, as the following analogue of th. 5.6 suggests.

Theorem 5.9 Let $f\varepsilon P_n(M^+)$ be a polynomial function over M^+. There exists a unique set of terms $T=\{c_i m_i \mid 1\leqq i\leqq t\}$ such that if p represents f in $M^+[X]$ then:

(i) Each term of T occurs in p.

(ii) If cm is a term of p then there exist

$\lambda_1,\ldots,\lambda_t$ such that

$\lambda_i \geqq 0 \quad i=1,\ldots,t;$

$\sum \lambda_i = 1$

$m \geqq \sum \lambda_i m_i$

$c \geqq \sum \lambda_i c_i$.

Proof The construction of the set of redundant terms T is identical to that of theorem 5.6. For part (ii) of the theorem, suppose that cm is a redundant term of p. Then

$$\forall u\varepsilon \mathbb{R}^n,\ u\geqq 0 \Rightarrow\ <m.u>+c \geqq \min_{1\leqq i\leqq t}(<m_i.u>+c_i)$$

which is equivalent to the assertion that in $\mathbb{R}^{n+1}$ the set of inequalities

$u_j \geqq 0,\ j=1,\ldots,n$

$<m_i\cdot u>+c_i \geqq u_{n+1}, \quad i=1,\ldots,t$

implies the inequality

$$\langle m\cdot u\rangle + c \geqq u_{n+1}$$

where $u_{n+1} \varepsilon \mathbb{R}$ is an independent variable. Denoting the vector $(u_1,\ldots,u_n,u_{n+1})$ by u^* and the j^{th} unit vector by e_j, the assertion may be rewritten as

$$\langle e_j\cdot u^*\rangle \geqq 0,\ j=1,\ldots,n$$

$$\langle (m_i,-1)\cdot u^*\rangle \geqq -c_i,\ i=1,\ldots,t$$

implies

$$\langle (m,-1)\cdot u^*\rangle \geqq -c$$

which by theorem 5.5 is equivalent to the existence of $\lambda_1,\ldots,\lambda_{n+t}$ with the properties

$$\lambda_i \geqq 0,\ i=1,\ldots,n+t$$

$$(m,-1) = \sum_{j=1}^{n} \lambda_j e_j + \sum_{i=1}^{t} \lambda_{n+i}\ (m_i,-1)$$

$$-c \leqq \sum_{i=1}^{t} \lambda_{n+1}\ (-c_i).$$

The result follows immediately. □

The unique representation theorem for M^+ is rather weaker than the corresponding one (th. 5.7) for M.

Theorem 5.10 Let $p,q \varepsilon M^+[X]$ represent the same function. Then

(i) If le(p) is linear, then le(p)=le(q)

(ii) If p is linear and homogeneous, then p=le(q).

Proof

(i) The argument in proving th. 5.7 carries over if we replace the appeal in the proof to th. 5.6 by one to th. 5.9. (There is no analogous argument for higher envelopes.)

(ii) If p is homogeneous, then p=le(p) and (ii) follows from (i). □

Corollary 5.11 Let $p \varepsilon M^{+}[X]$ represent the function $f \varepsilon P_{n}(M^{+})$. Then

(i) If le(p) is linear, then the $\otimes(\oplus)$-complexity of f is no smaller than the $\otimes(\oplus)$-complexity of le(p).

(ii) If p is linear and homogeneous, then the $\otimes(\oplus)$-complexity of f is equal to the $\otimes(\oplus)$-complexity of p.

Proof Use cor. 5.3, th. 5.4 and th. 5.10. □

5.4 A Combinatorial Lower Bound Argument

In this section, we restrict our attention to computations in B[X], the semiring of formal polynomials over the Boolean semiring B. The results obtained here extend to other semirings by the considerations introduced in sections 5.2, 5.3. Throughout the following, Γ will denote an arbitrary computation in B[X] with result vertex ρ and $res(\rho)=p \varepsilon B[X]$. $V_{\oplus}$, $V_{\otimes}$, E will respectively denote the set of $\oplus$-labelled vertices, $\otimes$-labelled vertices and edges, of Γ.

Let us now extend some of our earlier notation. If α is in the vertex set V of Γ then mon(α) is the monomial set of res(α) and deg(α) is the degree of res(α). pred(α) will denote the set of predecessors of α. Γ is said to be linear (homogeneous) if res(α) is a linear (homogeneous) polynomial for all $\alpha \varepsilon V$. We now show that when obtaining lower bounds on the $\otimes$-complexity of computing a certain linear (homogeneous) polynomial, we can as well restrict ourselves to computations which are themselves linear (homogeneous).

Lemma 5.12 Suppose that $p \varepsilon B[X]$, Γ computes p and that Γ is optimal in the sense that no Γ' computing p has fewer $\otimes$-vertices. Then

(i) Γ is linear if and only if p is.

(ii) Γ is homogeneous if and only if p is.

(iii) If α,β are in the vertex set V of Γ, β is a descendant of α and $m\varepsilon\text{mon}(\alpha)$ then $\text{mon}(\beta)$ contains a monomial of the form mm'.

Proof

(i) The only if part is immediate from the definition of linearity. For the other implication, suppose to the contrary that p is linear and that $\alpha\varepsilon V$ with $\text{res}(\alpha)$ not linear. The conditions that Γ is acyclic and that ρ is the unique vertex of outdegree 0 in Γ imply that there is a directed path $\alpha=\alpha_0,\alpha_1,\ldots,\alpha_j=\rho$ from α to ρ in Γ. Consider two adjacent vertices α_i,α_{i+1} in the path with the property that α_{i+1} is linear but α_i not. It must be the case that α_{i+1} is a $\otimes$-vertex, and that its predecessor β distinct from α_i has $\text{res}(\beta)=0$. But then $\text{res}(\alpha_{i+1})=0$ and α_{i+1} could be replaced by an input vertex labelled by 0. The amended computation would have one fewer $\otimes$-vertex, contradicting the optimality of Γ..

(ii), (iii) Analogous to (i). □

As the polynomials for which we will be obtaining lower bounds are both linear and homogeneous,parts (i) and (ii) of the previous lemma assure us that we may safely confine attention to computations which are both linear and homogeneous, and we assume henceforth that Γ has these properties. Part (iii) of the lemma captures the property of computation in B[X] which makes it amenable to treatment in the style of [34] or of the present chapter. Stated informally, once a monomial has been created, it must find its way into the final result; this "conservation of monomials" ensures that no "invalid" monomials are formed, and severely limits the rate at which monomials may be accumulated in the computation. Let us now introduce some definitions which help make precise this idea.

If $\alpha \varepsilon V$, then the complement of α is the set

$$\text{complement}(\alpha)=\{m=x_1^{i_1}x_2^{i_2}\ldots x_n^{i_n} \mid \forall m' \varepsilon \text{mon}(\alpha), mm' \varepsilon \text{mon}(\rho)\}$$

and the content of α is the set

$$\text{content}(\alpha)=\{mm' \mid m \varepsilon \text{complement}(\alpha), m' \varepsilon \text{mon}(\alpha)\}.$$

We remark that $\text{content}(\alpha) \subseteqq \text{mon}(\rho)$.

The fundamental construction on which our argument rests is that of the parse tree of a monomial, which is now described. If $\alpha \varepsilon V$ and $m \varepsilon \text{mon}(\alpha)$, $m \neq 1$ (the unit monomial), then the parse tree of m rooted at α is denoted by $PT(\alpha,m)$ and is a recursively defined subtree of Γ. We will, in fact, define $PT(\alpha,m)$ by specifying its edge set $E(\alpha,m)$; the vertex set of $PT(\alpha,m)$ then contains those vertices in V which are endpoints of edges in $E(\alpha,m)$. The recursive definition of $E(\alpha,m)$ is as follows:

(i) **α is an input vertex:** **Define $E(\alpha,m)$ to be $\emptyset$.**

(ii) **$\alpha \varepsilon V_{\oplus}$:** Let $\text{pred}(\alpha)=\{\beta,\gamma\}$.
Since $m \varepsilon \text{mon}(\alpha)$ we may deduce that either $m \varepsilon \text{mon}(\beta)$ or $m \varepsilon \text{mon}(\gamma)$ (or both). Without loss of generality we may suppose the former. Define $E(\alpha,m)$ in that case to be $E(\beta,m) \cup \{(\beta,\alpha)\}$. Note that although some freedom may exist in choosing between β and γ the definition may be made good by providing an ordering on the predecessors of α.

(iii) **$\alpha \varepsilon V_{\otimes}$:** Again let $\text{pred}(\alpha)=\{\beta,\gamma\}$. Since $m \varepsilon \text{mon}(\alpha)$ there must exist $m_1 \varepsilon \text{mon}(\beta)$, $m_2 \varepsilon \text{mon}(\gamma)$ such that $m=m_1m_2$. We may suppose that m_1 is not equal to 1, the unit monomial, for if it were the homogeneity of Γ would imply $\text{res}(\beta)=1$ and hence $\text{res}(\alpha)=\text{res}(\gamma)$. A smaller computation for p could then be obtained by removing the vertex α from Γ and restructuring. By a similar argument we may suppose $m_2 \neq 1$. Define $E(\alpha,m)$ to be

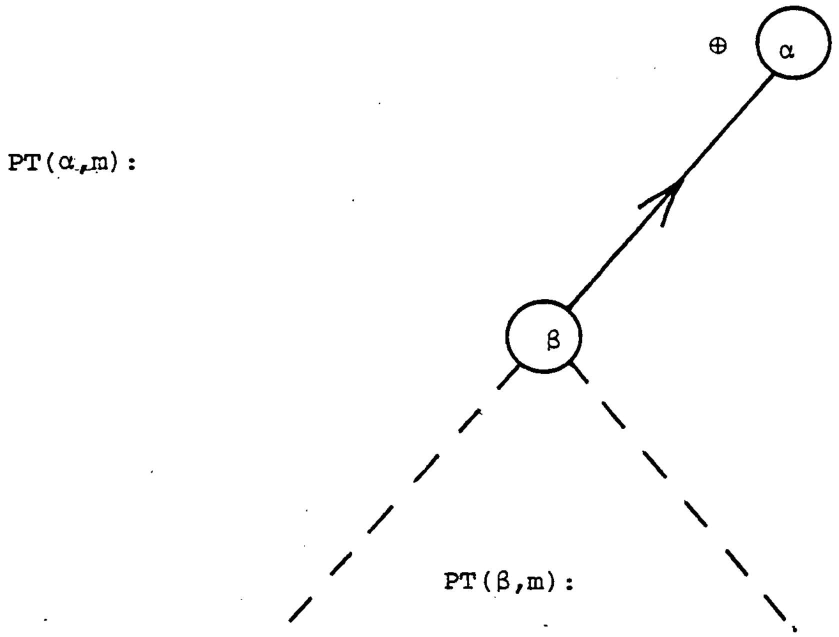

Figure 5.1

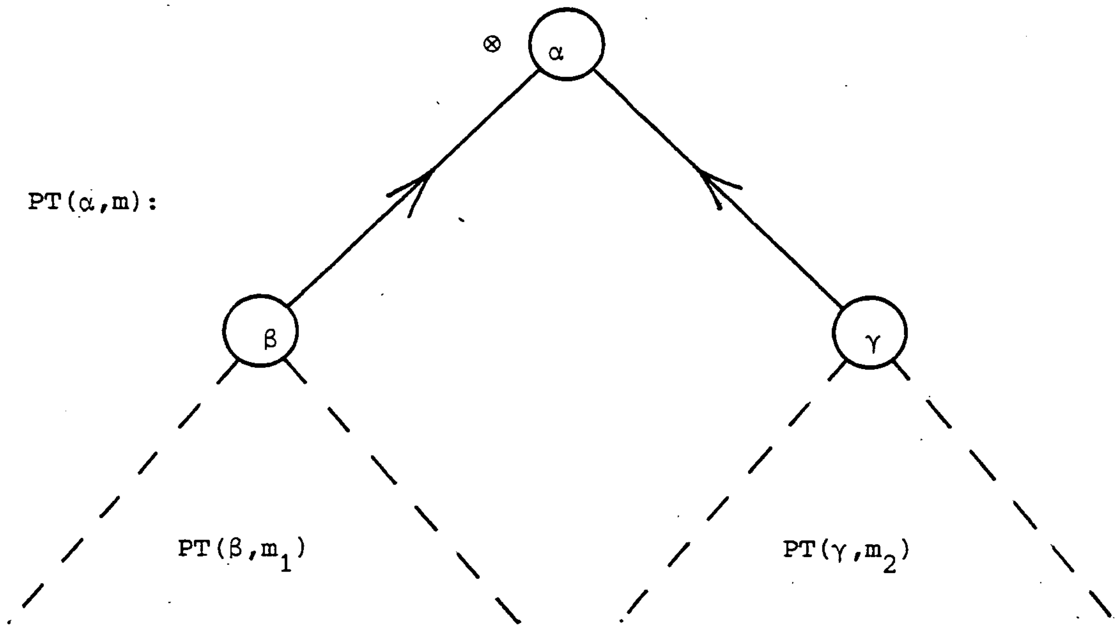

Figure 5.2

$E(\beta,m_1) \cup E(\gamma,m_2) \cup \{(\beta,\alpha),(\gamma,\alpha)\}$. (Again m_1 and m_2 may not be uniquely defined, but we may provide a rule for choosing such a pair.) The diagrams 5.1 and 5.2 are added as an aid to visualising the construct.

That $PT(\alpha,m)$ is well defined is a consequence of Γ being acyclic, it only remains to check that it is indeed a tree. But if $PT(\alpha,m)$ is not a tree then it must contain two distinct directed paths from some vertex β to α. Now if $m \varepsilon mon(\beta)$, three applications of lemma 5.12 (iii) yield that $mon(\alpha)$ contains a monomial of the form m^2m', which violates the linearity of Γ.

The parse tree of a monomial is intended to be an intuitively appealing construct; essentially it is a family tree which charts the generation of that monomial within the computation. Those familiar with the work of Ehrenfeucht and Zeiger [7] will note the similarity with the "parse function" which is defined there on elements of regular sets. An important property of parse trees is the following:

<u>Theorem 5.13</u> Let m be an element of $mon(\rho)$. If α is in the vertex set of $PT(\rho,m)$ then $m \varepsilon content(\alpha)$.

<u>Proof</u> Following the recursive construction of $PT(\rho,m)$, let $m_\alpha \varepsilon mon(\alpha)$ be the monomial whose parse tree $PT(\alpha,m_\alpha)$ is precisely the subtree of $PT(\rho,m)$ rooted at α. We are done if we can show that for each α in the vertex set of $PT(\rho,m)$ there exists a monomial m'_α such that

$$m_\alpha m'_\alpha = m \qquad (5.4)$$

$$m'_\alpha n \; \varepsilon \; mon(\rho) \qquad \forall n \varepsilon mon(\alpha). \qquad (5.5)$$

For if 5.4, 5.5 are satisfied, $m'_\alpha \varepsilon\, complement(\alpha)$, $m_\alpha \varepsilon mon(\alpha)$ and hence $m = m_\alpha m'_\alpha \varepsilon\, content(\alpha)$. The existence of m'_α satisfying equations 5.4, 5.5 is established by induction on the vertices of $PT(\rho,m)$.

(i) The hypothesis is true for the root, ρ. Take $m'_\rho=1$, then 5.4, 5.5 are trivally satisfied.

(ii) Assume true for $\oplus$-vertex α. Let β be the predecessor of α in PT(ρ,m) and let m'_α satisfy 5.4, 5.5. To see that the hypothesis holds for β, note that, by construction, $m_\beta= m_\alpha$ and that 5.4 may be satisfied by taking $m'_\beta=m'_\alpha$. Also, since

$$\begin{aligned}\{m'_\beta n \mid n\varepsilon\text{mon}(\beta)\} &\subseteqq \{m'_\beta n \mid n\varepsilon\text{mon}(\alpha)\}\\ &=\{m'_\alpha n \mid n\varepsilon\text{mon}(\alpha)\}\\ &\subseteqq \text{mon}(\rho),\end{aligned}$$

5.5 is satisfied.

(iii) Assume true for $\otimes$-vertex α. Let pred(α)=$\{\beta,\gamma\}$, and let m'_α satisfy 5.4, 5.5. We show that the hypothesis holds for β (and by symmetry for γ). Set $m'_\beta=m'_\alpha m_\gamma$ and observe that, since $m_\beta m'_\beta=m_\beta m'_\alpha m_\gamma=m_\alpha m'_\alpha=m$, equation 5.4 is satisfied. Additionally:

$$\begin{aligned}\{m'_\beta n \mid n\varepsilon\text{mon}(\beta)\} &= \{m'_\alpha m_\gamma n \mid n\varepsilon\text{mon}(\beta)\}\\ &\subseteqq \{m'_\alpha n \mid n\varepsilon\text{mon}(\alpha)\}\\ &\subseteqq \text{mon}(\rho)\end{aligned}$$

which verifies 5.5 for β. □

Theorem 5.13 suggests a method for obtaining lower bounds. Γ contains $|\text{mon}(\rho)|$ parse-trees corresponding to the distinct monomials of p. Distinct parse-trees may share vertices of Γ, but the amount of sharing that takes place is limited by the previous theorem. In order to make this qualitative argument precise we introduce a weight function for parse-trees.

Suppose T is a parse tree in Γ. Define the weight of T, w(T) by

$$w(T) = \sum_\alpha |\text{content}(\alpha)|^{-1}$$

with summation being over all $\otimes$-vertices in T.

Theorem 5.14 $\displaystyle\sum_{m\varepsilon\mathrm{mon}(\rho)} w(PT(\rho,m))\leq|V_\otimes|$

(= $\otimes$-complexity of Γ).

Proof Denote by U(m) the set of $\otimes$-vertices of PT(ρ,m). Then:

$$\begin{aligned}\sum_{m\varepsilon\mathrm{mon}(\rho)} w(PT(\rho,m)) &= \sum_{m\varepsilon\mathrm{mon}(\rho)}\ \sum_{\alpha\varepsilon U(m)} |\mathrm{content}(\alpha)|^{-1}\\ &= \sum_{\alpha\varepsilon V_\otimes} |\{m|\alpha\varepsilon U(m)\}|.|\mathrm{content}(\alpha)|^{-1}\\ &\leq \sum_{\alpha\varepsilon V_\otimes} |\{m|m\varepsilon\mathrm{content}(\alpha)\}|.|\mathrm{content}(\alpha)|^{-1}\\ &\qquad\qquad\text{(by theorem 5.13)}\\ &= |V_\otimes| \qquad \square\end{aligned}$$

Now suppose that for a specified linear, homogeneous polynomial p we have some bound on the content of vertices in the computation. Specifically, we assume the existence of a function c($\cdot$,$\cdot$) of two integer variables which satisfies

$$c(j,k)\geqq\max\{|\mathrm{content}(\alpha)|\,|\alpha\varepsilon V_\otimes,\deg(\mathrm{pred}(\alpha))=\{j,k\}\}.$$

We use c to construct a lower bound on w(PT(α,m)) which depends only on the degree of α.

Theorem 5.15 If the function w*($\cdot$) of one integer variable is defined by

$$w^*(1)=0 \tag{5.6}$$

$$w^*(i)=\min_{\substack{1\leqq j\leqq k\\ j+k=i}}\{w^*(j)+w^*(k)+1/c(j,k)\}\quad(i\geqq 2) \tag{5.7}$$

then $w(PT(\alpha,m))\geqq w^*(\deg(\alpha))$ for all $\alpha\varepsilon V$, $m\varepsilon\mathrm{mon}(\alpha)$. In particular $w(PT(\rho,m))\geqq w^*(\deg(p))$.

Proof Since Γ is acyclic we may perform a topological sort ([1]p.70) of the vertices V of Γ, that is to say order the vertices in such a way that each edge of Γ is directed from a vertex lower in the order to one higher. We proceed by induction on this order. The hypothesis is clearly true when deg(α)=0 and the induction step trivial if $\alpha\varepsilon V_\oplus$. Assume, therefore, that $\alpha \varepsilon V_{\otimes}$, $\text{pred}(\alpha)=\{\beta,\gamma\}$ and let $i=\deg(\alpha)$. Then

$$\begin{aligned} w(PT(\alpha,m)) &= w(PT(\beta,m_{\beta}))+w(PT(\gamma,m_{\gamma}))+|\text{content}(\alpha)|^{-1} \\ &\geqq w^*(\deg(\beta))+w^*(\deg(\gamma))+1/c(\deg(\beta),\deg(\gamma)) \\ &\qquad \text{(by inductive hypothesis)} \\ &\geqq \min_{j+k=i} \{w^*(j)+w^*(k)+1/c(j,k)\} \\ &= w^*(i) \end{aligned}$$ □

It may be remarked that the theorem remains true if the equalities of 5.6, 5.7 are replaced by inequalities ($\leqq$). This observation can be useful if an exact solution to the original equations is hard to obtain.

Corollary 5.16 For linear, homogeneous $p \varepsilon B[X]$

$$|\text{mon}(\rho)| \cdot w^*(\deg(p)) \leqq \otimes\text{-complexity of } p$$

Proof Take Γ in theorem 5.14 to be a computation for p which minimises $|V_{\otimes}|$, obtaining

$$\sum_{m \varepsilon \text{mon}(\rho)} w(PT(\rho,m)) \leqq \otimes\text{-complexity of } p.$$

Applying theorem 5.15 we obtain

$$\sum_{m \varepsilon \text{mon}(\rho)} w^*(\deg(p)) \leqq \otimes\text{-complexity of } p$$ □

In the next section we compute content bounds for specific polynomials and derive the corresponding weight bounds. We show that for several polynomials the lower bound implied by corollary 5.16 is tight. In order to help solve the recurrences 5.6, 5.7 for practical examples we introduce a final lemma.

Lemma 5.17 If for all integers j,k satisfying $1 \leqq j \leqq k-2$, $4 \leqq j+k \leqq n$, the inequality

$$1/c(j+1,k-1)+1/c(1,j)-1/c(j,k)-1/c(1,k-1) \geqq 0$$

holds, the solution to the recurrences 5.6, 5.7 is

$$w^*(i) = \sum_{i'=1}^{i-1} 1/c(1,i') \qquad (2 \leqq i \leqq n)$$

Proof by induction on i. Trivially true for i=2,3, otherwise

$$w^*(i)=\min_{\substack{1\leq j\leq k\\ j+k=i}}\left\{\sum_{j'=1}^{j-1} 1/c(1,j')+\sum_{k'=1}^{k-1} 1/c(1,k') +1/c(j,k)\right\}$$

$$=\min_{1\leq j\leq i/2} g(j)$$

regarding j as an independent variable and k as dependent. We observe that g is a monotonically increasing function in its range $1\leq j\leq\lfloor i/2\rfloor$ since

$$g(j+1)-g(j)=1/c(1,j)-1/c(1,k-1)+1/c(j+1,k-1)-1/c(j,k)$$

$$\geq 0 \text{ by stated condition.}$$

Thus $w^*(i)=g(1)$

$$=\sum_{i'=1}^{i-1} 1/c(1,i') \qquad \square$$

5.5 The Complexity of Specific Polynomials

(i) Iterated matrix multiplication

Suppose $X^{(1)},X^{(2)},\ldots,X^{(n)}$ are d×d matrices; $X^{(k)}=x_{ij}^{(k)}$ $(1\leq i,j\leq n)$. We are interested in the number of multiplications required to compute the product

$$(X^{(1)}\ldots X^{(n)})_{ij}=\bigoplus_{1\leq i_k\leq d} x_{ii_2}^{(1)}x_{i_2i_3}^{(2)}x_{i_3i_4}^{(3)}\ldots x_{i_nj}^{(n)}.$$

We note that any computation for the above can be transformed into a computation for the related polynomial

$$p=\bigoplus_{1\leq i_k\leq d} x_{i_1i_2}^{(1)}x_{i_2i_3}^{(2)}\ldots x_{i_ni_{n+1}}^{(n)}x_{i_{n+1}i_1}^{(n+1)}$$

by the addition of at most d^2 ⊗-vertices. The number of multiplications necessary for matrix multiplication is thus no smaller than (⊗-complexity of p)$-d^2$.

The first step in establishing a bound on the complexity of p is to compute a suitable content bound $c(\cdot,\cdot)$. Suppose q is a polynomial with indeterminates of the form $x_{ij}^{(k)}$. Define the index set I_q to be the set of superscripts of indeterminates occurring in q. Now consider polynomials, a,b,c, of degrees $r(\geqq 1), s(\geqq 1), n-r-s+1$, with the property that $\mathrm{mon}(abc) \subseteqq \mathrm{mon}(p)$. By considering the form of monomials of p we see immediately that I_a, I_b, I_c are disjoint and, moreover, $|I_a| \geqq r, |I_b| \geqq s, |I_c| \geqq n-r-s+1$. Hence $\{I_a, I_b, I_c\}$ is a partition of $\{1,2,\ldots,n+1\}$. Define the set A of <u>articulations</u> to be

$$A=\{k \mid (2 \leqq k \leqq n+1, \text{ k and k-1 are in distinct index sets})$$
$$\vee (k=1, \text{ 1 and n+1 are in distinct index sets})\}.$$

Next consider a general element of mon(abc)

$$x^{(1)}_{i_1 i_2} \, x^{(2)}_{i_2 i_3} \cdots x^{(n+1)}_{i_{n+1} i_1} \, .$$

Observe that if k is an articulation ($k \varepsilon A$) then the subscript i_k is necessarily fixed by the condition $\mathrm{mon}(abc) \subseteqq \mathrm{mon}(p)$; otherwise i_k is free to assume any of the d possible values 1,...,d. Hence

$$|\mathrm{mon}(abc)| \leqq d^{n-|A|+1}$$

If $r+s<n+1$ then $I_a, I_b, I_c \neq \emptyset$, which implies $|A| \geqq 3$; if $r+s=n+1$ then $I_a, I_b \neq \emptyset, I_c = \emptyset$ and $|A| \geqq 2$. Consequently we take as our content bound

$$c(r,s)=\begin{cases} d^{n-2} & (r+s<n+1) \\ d^{n-1} & (r+s=n+1). \end{cases}$$

The recurrence relations 5,6, 5,7 are easily solved in this case, where $c(\circ,\bullet)$ is essentially a constant. The condition of lemma 5.17 is trivially satisfied, from which we obtain

$$w^*(n+1)= \sum_{i=1}^{n} 1/c(1,i)$$
$$=(n-1)d^{(2-n)}+d^{(1-n)} .$$

Hence by cor. 5.16:

$\otimes$-complexity of $p \geqq [(n-1)d^{(2-n)}+d^{(1-n)}] \cdot |\mathrm{mon}(p)| = (n-1)d^3+d^2$

and, by our initial observation, the number of multiplications required for matrix multiplication is $(n-1)d^3$. (For the case

$n=2$, this result is implied by a stronger one, obtained by Paterson and others [17,25,27], for the monotone Boolean matrix product.) The obvious algorithm, derived from the definition of matrix multiplication yields an upper bound of $(n-1)d^3$ and illustrates that our bound is tight. Note that since p is linear and homogeneous the conditions of cor. 5.8, 5.11 are satisfied and the lower bound is valid for matrix multiplication over R,M,M^+.

(ii) Iterated wrapped convolution

Suppose $\bar{x}^{(1)}, \bar{x}^{(2)}, \dots, \bar{x}^{(n)}$ are d-vectors; $\bar{x}^{(k)} = x_i^{(k)}$ $(0 \leq i \leq d-1)$. The wrapped convolution of these vectors is the k-vector $\bar{y}$ whose components are given by

$$y_j = \bigoplus_{i_1+i_2+\dots+i_n \equiv j \pmod d} x_{i_1}^{(1)} x_{i_2}^{(2)} \dots x_{i_n}^{(n)}$$

As before we define the related polynomial

$$p = \bigoplus_{i_1+\dots+i_{n+1} \equiv 0 \pmod d} x_{i_1}^{(1)} x_{i_2}^{(2)} \dots x_{i_n}^{(n)} x_{i_{n+1}}^{(n+1)}$$

where $\bar{x}^{(n+1)}$ is a d-vector, and remark that the number of multiplications required to compute $\bar{y}$ is at least ($\otimes$-complexity of p)$-d$.

Consider polynomials a,b,c of degrees $r,s,n-r-s+1$ with the property that $\text{mon}(abc) \subseteq \text{mon}(p)$. As before define the index set I_q of a polynomial q to be the set of all superscripts occurring in the indeterminates which form q. Again, I_a, I_b, I_c form a partition of $\{1,2,\dots,n+1\}$. If we now consider a general monomial

$$m_a m_b m_c = x_{i_1}^{(1)} x_{i_2}^{(2)} \dots x_{i_{n+1}}^{(n+1)}$$

of mon(abc), we see from the definition of p that

$$\sum_{k\varepsilon I_a} i_k + \sum_{k\varepsilon I_b} i_k + \sum_{k\varepsilon I_c} i_k \equiv 0 \pmod d$$

and, letting m_a range over mon(a) while holding m_b, m_c fixed, we deduce that $\sum_{k\varepsilon I_a} i_k$ is congruent to a constant, modulo d. Similar arguments apply to I_b, I_c and hence $|\text{mon}(abc)|$ is bounded by the number of assignments which can be made to $i_1, i_2, \ldots, i_{n+1}$ which fix the three above sums. If $r+s<n+1$, then I_a, I_b, I_c are all non-empty and the number of assignments which can be made is d^{n-2}; if $r+s=n+1$ then $I_c=\emptyset$ and there are d^{n-1} possible assignments. Our content bound is thus

$$c(r,s)=\begin{cases} d^{n-2} & (r+s<n+1) \\ d^{n-1} & (r+s=n+1) \end{cases}$$

Observing that this bound is identical to that derived in the previous example we can immediately write down

$$w^*(n+1)=(n-1)d^{(2-n)}+d^{(1-n)}$$

and so by cor. 5.16

$$\begin{aligned}\otimes\text{-complexity of } p &\geqq [(n-1)d^{(2-n)}+d^{(1-n)}]\cdot|\text{mon}(p)| \\ &=(n-1)d^2+d.\end{aligned}$$

The number of multiplications required to compute the wrapped convolution is thus at least $(n-1)d^2$. That this bound is tight may be seen by considering the algorithm derived from the definition. Again the bound is valid for R,M and M^+.

(iii) Permanent

Suppose X is an n×n matrix of indeterminates x_{ij} $(1\leqq i,j\leqq n)$. The permanent function on X is defined to be

$$\text{per}(X)=p=\bigoplus_{\pi\varepsilon S(n)} x_{1,\pi(1)}\, x_{2,\pi(2)} \circ\cdots x_{n,\pi(n)}$$

where S(n) is the set of all permutations of the first n natural numbers. In its arithmetic interpretation, the permanent was introduced in chapter 2 and its importance as the generating function for perfect matchings in bipartite graphs discussed. It will be recalled that the permanent is algebraically complete; we should not expect therefore to be able to compute the permanent using a number of arithmetic operations bounded by a polynomial in n, even if arbitrary constants are allowed. We shall in fact show that in any monotone computation of the permanent (i.e. computation in R[X]) there are an exponential number of multiplications.

Re-interpreting the semiring operations, $\oplus$, $\otimes$ as min,+, the significance of the permanent is that it computes the minimum weight of a perfect matching in a bipartite graph (the so-called "assignment problem"). In contrast to the arithmetic interpretation, the problem of finding a minimum weight matching in a bipartite graph is tractable and an $O(n^3)$ algorithm can be found in Lawler [16].

To study the complexity of monotone computation of the permanent we first determine a content bound, $c(\cdot,\cdot)$. Suppose a,b and c are polynomials of degrees r,s and n-r-s respectively, with $\text{mon}(abc) \subseteq \text{mon}(p)$. If q is a polynomial with indeterminates x_{ij}, we denote by I_q and J_q the sets

$$I_q=\{i|x_{ij} \text{ occurs in } q\}$$

$$J_q=\{j|x_{ij} \text{ occurs in } q\}.$$

If we consider a general element of mon(abc)

$$m_a m_b m_c = x_{1,\pi(1)} x_{2,\pi(2)} \cdots x_{n,\pi(n)}$$

we can see that the sets I_a, I_b, I_c are disjoint and

$$|I_a|=r, \ |I_b|=s, |I_c|=n-r-s$$

so that $\{I_a,I_b,I_c\}$ is a partition of $\{1,2,\ldots,n\}$. Since π is a permutation, the same argument yields that $\{J_a, J_b, J_c\}$ is also a partition. Elements of mon(abc) correspond to permutations π which observe the restrictions

$$\pi(I_a)=J_a,\ \pi(I_b)=J_b,\ \pi(I_c)=J_c.$$

The total number of such permutations is clearly $r!s!(n-r-s)!$ and so we may take as our content bound

$$c(r,s)=r!s!(n-r-s)!$$

We claim that this bound satisfies the condition of lemma 5.17. In order to show this we need the following easily verified lemma:

<u>Lemma 5.18</u> If $r \geq 0$, $s \geq 2$ then $\binom{r+s}{r} \geq r(s+1)$ □

For the particular content bound we have just computed the condition of lemma 5.17 becomes

$\forall r,s$ satisfying $1 \leq r \leq s-2$, $4 \leq r+s \leq n$:

$$1/(r+1)!(s-1)!(n-r-s)!+1/r!(n-r-1)!-1/r!s!(n-r-s)!-1/(s-1)!(n-s)! \geq 0$$

By multiplying throughout by $r!(s-1)!(n-r-s)!$ the following equivalent condition is obtained

$\forall r,s$ satisfying $1 \leq r \leq s-2$, $4 \leq r+s \leq n$:

$$(r+1)^{-1}+\binom{n-r-1}{n-r-s}^{-1}-s^{-1}-\binom{n-s}{n-r-s}^{-1} \geq 0$$

Finally, making the substitution $t=r+s$, we arrive at a final equivalent condition

$\forall t,r$ satisfying $4 \leq t \leq n$, $2 \leq 2r \leq t-2$:

$$f(t,r) \equiv (r+1)^{-1}+\binom{n-r-1}{n-t}^{-1}-(t-r)^{-1}-\binom{n-t+r}{n-t}^{-1} \geq 0 \qquad (5.8)$$

In fact, it proves easier to show that $f(t,r) \geq 0$ in the slightly extended range $4 \leq t \leq n$, $2 \leq 2r \leq t-1$. Our approach will be to show that $f(t,r)$, with t fixed, is a monotonically decreasing function of r in the range $2 \leq 2r \leq t-1$. The problem is thus reduced to showing $f(t,r)$ to be non-negative when r assumes its maximum value i.e. $\lfloor (t-1)/2 \rfloor$.

For the monotonicity of f, consider the difference $f(t,r-1)-f(t,r)$. From the definition of f we have

$$f(t,r-1)=r^{-1}+\binom{n-r}{n-t}^{-1}-(t-r+1)^{-1}-\binom{n-t+r}{n-t}^{-1} \geqq 0 \qquad (5.9)$$

Forming differences of corresponding terms in equations 5.8, 5,9 we obtain

$$\begin{aligned}&f(t,r-1)-f(t,r)\\ &= 1/r(r+1)+\binom{n-r}{n-t}^{-1}\cdot[1-(n-r)/(t-r)]+1/(t-r)(t-r+1)\\ &\qquad +\binom{n-t+r}{n-t}^{-1}\cdot[1-(n-t+r)/r]\\ &=1/r(r+1)-\binom{n-r}{n-t}^{-1}\cdot(n-t)(t-r)^{-1}+1/(t-r)(t-r+1)-\binom{n-t+r}{n-t}^{-1}\cdot(n-t)r^{-1}\end{aligned}$$

By lemma 5.18, the binomial coefficient $\binom{n-r}{n-t}$ is bounded below by $(n-t)(t-r+1)$ while the binomial coefficient $\binom{n-t+r}{n-t}$ is bounded by $(n-t)(r+1)$. Replacing each coefficient by its bound the terms of right hand side cancel in pairs, yielding

$$f(t,r-1)-f(t,r)\geqq 0.$$

i.e. that f is a monotonically decreasing function of its second argument.

It only remains to show that f is non-negative when its second argument assumes its maximum value, i.e. that

$$f(t,\lfloor(t-1)/2\rfloor)\geqq 0 \quad \forall t, 4\leqq t\leqq n.$$

Considering the cases when t is respectively odd and even:

$$\begin{aligned}&f(t,(t-1)/2)\\ &=(\tfrac12 t+\tfrac12)^{-1}+\binom{n-\frac12 t-\frac12}{n-t}^{-1}-(\tfrac12 t+\tfrac12)^{-1}-\binom{n-\frac12 t-\frac12}{n-t}^{-1}\\ &=0\end{aligned}$$

and $f(t,\frac12 t-1)$

$$\begin{aligned}&=(\tfrac12 t)^{-1}+\binom{n-\frac12 t}{n-t}^{-1}-(\tfrac12 t+1)^{-1}-\binom{n-\frac12 t-1}{n-t}^{-1}\\ &=4/t(t+2)+\binom{n-\frac12 t}{n-t}^{-1}[1-(n-\tfrac12 t)/\tfrac12 t]\\ &=4/t(t+2)-\binom{n-\frac12 t}{n-t}^{-1}\cdot 2(n-t)t^{-1}.\end{aligned}$$

By lemma 5.18, $\binom{n-\frac{1}{2}t}{n-t}$ is bounded below by $(n-t)(\frac{1}{2}t+1)$, and hence $f(t,\frac{1}{2}t-1)\geqq 0$.

Invoking lemma 5.17, whose condition we now see to be satisfied , we deduce

$$w^*(n) = \sum_{i=1}^{n-1} 1/(i-1)!(n-i)!$$

$$= \sum_{i=1}^{n-1} \binom{n-1}{i} / (n-1)!$$

$$= (2^{n-1}-1)/(n-1)!$$

By cor. 5.16

$$\otimes\text{-complexity of } p \geqq n!(2^{n-1}-1)/(n-1)!$$
$$= n(2^{n-1}-1)$$

This lower bound is in fact achievable using a permanental equivalent of Laplace's expansion rule for determinants. This is essentially a dynamic programming method: the permanents of all i×i submatrices contained in the first i rows of X (the "subpermanents" of the first i rows) are computed using the values obtained for the subpermanents of the first i-1 rows. Clearly we can obtain variants of this algorithm by permuting rows of X and transposing X; what is more interesting is that the optimal algorithm lacks uniqueness in a non-trivial way, this stemming from the observation that several "shapes" of parse tree all have optimum weight. More specifically, the value of

$$w^*(i)+w^*(j)+1/c(i,j)$$

is a constant for all non-negative integers i,j summing to n-1, which leads to the following family of optimal algorithms for $1\leqq t\leqq n-2$:

(1) Evaluate all t×t subpermanents of the first t rows using Laplace's expansion.

(2) Evaluate the $(n-t-1)\times(n-t-1)$ subpermanents of the rows $t+1, t+2, \ldots, n-1$ in the same way.

(3) Use the results of (1) and (2) to compute all $(n-1)\times(n-1)$ subpermanents of the first $n-1$ rows.

(4) From (3) compute per(X) by Laplace's expansion.

We remark that, once more, the lower bound is valid for R,M and M^+.

(iv) Hamiltonian circuit polynomial

Suppose again that X is an $n\times n$ matrix of indeterminates x_{ij} $(1\leq i,j\leq n)$. The Hamiltonian circuit polynomial is

$$HC_{n\times n}=p=\bigoplus_{\pi\varepsilon C(n)} x_{1,\pi(1)}x_{2,\pi(2)}\ldots x_{n,\pi(n)}$$

where C(n) is the set of all cyclic permutations of the first n natural numbers. Identifying each indeterminate x_{ij} with the (i,j) edge of the complete graph K_n on the n vertices $\{1,2,\ldots,n\}$, it will be seen that monomials of p correspond to Hamiltonian circuits in K_n. Over R the polynomial can be viewed as the generating function for Hamiltonian circuits in the complete graph, while the corresponding interpretation for M^+ is that of finding the shortest circuit which visits all the vertices of a graph - the so-called "Travelling Salesman Problem".

In the usual way, we let a,b,c be polynomials of degrees r,s,n-r-s respectively with $\text{mon}(abc)\subseteq\text{mon}(p)$. Using the same reasoning as for the permanent, a,b and c define two partitions of $\{1,2,\ldots,n\}$ namely

$$\{I_a,I_b,I_c\} \text{ and } \{J_a,J_b,J_c\}.$$

If we consider a general monomial of abc

$$m_am_bm_c=x_{1,\pi(1)}x_{2,\pi(2)}\ldots x_{n,\pi(n)}$$

we have

$$\pi(I_a)=J_a,\ \pi(I_b)=J_b,\ \pi(I_c)=J_c$$

and so $|\mathrm{mon}(abc)|$ is bounded by the number of cyclic permutations π which satisfy these constraints. Suppose we fix m_b, m_c i.e. fix π on $I_b \cup I_c$; we wish to know the number of possible choices of m_a i.e. the number of ways of extending π to I_a. Define

$$\pi^*: I_a \to I_a$$

$$\pi^*(i) = \pi^\alpha(i)$$

where α is the smallest positive number such that $\pi^\alpha(i) \varepsilon I_a$ (such an α exists since π is cyclic). Note that π is completely determined by π^* and the restriction of π to $I_b \cup I_c$. We observe that π^* is a cyclic permutation, and hence the number of distinct permutations π which agree on $I_b \cup I_c$ is bounded by the number of cyclic permutations on r objects

i.e. $|\mathrm{mon}(a)| \leqq (r-1)!$

similarly $|\mathrm{mon}(b)| \leqq (s-1)!$

and $|\mathrm{mon}(c)| \leqq (n-r-s-1)!$ $\quad (r+s<n)$

$= 1$ $\quad (r+s=n)$

the second case case being the degenerate one where $I_c = \emptyset$.

Consequently we take as content bound

$$c(r,s) = \begin{cases} (r-1)!(s-1)!(n-r-s-1)! & (r+s<n) \\ (r-1)!(s-1)! & (r+s=n) \end{cases}$$

By an argument completely analogous to the case of the permanent, we can show that this bound satisfies the condition of lemma 5.17 and hence

$$w^*(n) = 1/(n-2)! + \sum_{i=1}^{n-2} 1/(i-1)!(n-i-2)!$$

$$= 1/(n-2)! + \sum_{i=1}^{n-2} \binom{n-3}{i-1} / (n-3)!$$

$$= [(n-2)2^{(n-3)} + 1]/(n-2)!$$

By cor. 5.16 :

$$\otimes\text{-complexity of } p \geq (n-1)!\,[(n-2)2^{(n-3)}+1]/(n-2)!$$
$$=(n-1)\,[(n-2)2^{(n-3)}+1].$$

Again this bound is valid for the semirings R,M and M^+, and is attainable.

Denote by N the set containing the first n natural numbers i.e. $N=\{1,2,\ldots,n\}$. Let $p_{i,S,j}$ for $i,j \in N$, $S \subsetneq N-\{i,j\}$ be the polynomial whose monomials correspond 1-1 with the simple paths in K_n which start at vertex i, terminate at vertex j and visit exactly those vertices in S. A dynamic programming approach may be used to compute $p_{1,S,j}$ for all permissible j,S; the relevant relations are

$$p_{1,\emptyset,j}=x_{1j} \qquad (j\in N\backslash 1)$$

$$p_{1,S,j}=\bigoplus_{i\in S} p_{1,S\backslash i,i}\, x_{ij} \quad (j\in N\backslash 1, S\neq\emptyset).$$

Generating the set $\{p_{1,S,j} \mid |S|=s\}$ of polynomials from the set $\{p_{1,S,j} \mid |S|=s-1\}$ can be achieved using $s(n-1)\binom{n-2}{s}$ multiplications; by iterating this process we can compute $p_{1,N-\{1,j\},j}$ in

$$\sum_{s=1}^{n-2} s(n-1)\binom{n-2}{s}$$
$$=(n-1)(n-2)\sum_{s=1}^{n-2}\binom{n-3}{s-1}$$
$$=(n-1)(n-2)2^{(n-3)} \quad \text{multiplications.}$$

Now $HC_{n\times n}=\bigoplus_{j=2}^{n} p_{1,N-\{1,j\},j}\,x_{j1}$

which can be computed in

$$(n-1)(n-2)2^{(n-3)}+(n-1) \text{ multiplications.}$$

A polynomial closely allied to the Hamiltonian circuit polynomial is the generating function for simple paths between two distinguished vertices of a complete graph. Define the <u>simple paths polynomial</u> to be

$$SP_{n\times n}=\bigoplus_{S\subsetneq N-\{1,n\}} p_{1,S,n}$$

so that monomials correspond to simple paths from vertex 1 to vertex n in the complete graph K_n. We remark that ⊗-complexity of $HC_{(n-1)\times(n-1)} \leq$ ⊗-complexity of $p_{1,N-\{1,n\},n}$ since a computation for the latter may be transformed into one for the former by changing the inputs x_{in} to x_{i1}. Hence the lower bound for $HC_{n\times n}$ implies a lower bound of $(n-2)[(n-3)2^{(n-4)}+1]$ multiplications for $p_{1,N-\{1,n\},n}$. However $p_{1,N-\{1,n\},n}=he(SP_{n\times n})$ and so, by theorem 5.4 and corollary 5.8, we obtain a lower bound of $(n-2)[(n-3)2^{(n-4)}+1]$ multiplications for $SP_{n\times n}$ when working with the semirings R and M (but not M^+, where a minimum length path between two vertices is necessarily simple - the significance of this observation will be discussed in the next section.)

(v) Spanning tree polynomial

Suppose X is an n×n matrix of indeterminates x_{ij} $(1\leq i,j\leq n)$. Define the spanning tree polynomial to be

$$ST_{n\times n}=p=\bigoplus_{t\in T(n)} x_{2,t(2)}x_{3,t(3)}\cdots x_{n,t(n)}$$

where $T(n)=\{t:\{2,3,\ldots,n\}\rightarrow\{1,2,\ldots,n\}\,|\,\forall i\,\exists k,\ t^k(i)=1\}$.

The polynomial is the generating function of directed trees spanning K_n and rooted at vertex 1. The lower bound obtained for this polynomial is not claimed to be attainable; it is in any case difficult to envisage the form that an optimal monotone computation would take in this case. We therefore content ourselves with a crude bound on the content of a vertex, which is, however, good enough to yield an exponential lower bound on the ⊗-complexity of $ST_{n\times n}$.

Let a,b,c be polynomials of degrees r,s,n-r-s-1 satisfying mon(abc) $\subsetneq$ mon(p). In the usual way we define the index set I_q of a polynomial q to be

$$I_q=\{i|x_{ij} \text{ is an indeterminate of } q\}$$

and note that $\{I_a,I_b,I_c\}$ is a partition of $\{2,3,\ldots,n\}$.

Let X_i $(2\leqq i\leqq n)$ be defined by

$$X_i=\{x_{ij}|x_{ij} \text{ is an indeterminate of } a,b \text{ or } c\}$$

Obviously, $\sum_{i=2}^{n}|X_i|$ is bounded by the number of distinct indeterminates which appear in $ST_{n\times n}$, i.e. $(n-1)^2$, but this trivial bound may be improved through the following observation. Suppose $i_a\varepsilon I_a$ and $i_b\varepsilon I_b$; then the indeterminates $x_{i_a i_b}, x_{i_b i_a}$ cannot both appear in $\bigsqcup_{i=2}^{n} X_i$, for if they did $x_{i_a i_b}$ would appear in a, $x_{i_b i_a}$ would appear in b and the invalid monomial $x_{i_a i_b} x_{i_b i_a} m$ would appear in mon(abc). Thus a better restriction is

$$\begin{aligned}\sum_{i=2}^{n}|X_i| &\leqq (n-1)^2-|I_a|\cdot|I_b|-|I_b|\cdot|I_c|-|I_c|\cdot|I_a|\\ &=(n-1)^2-rs-s(n-r-s-1)-(n-r-s-1)r\\ &=(n-1)^2-rs-(r+s)(n-r-s-1)\\ &\leqq(n-1)^2-(r+s)(n-r-s-1).\end{aligned}$$

The number of monomials in mon(abc) is clearly bounded by the number of functions t,

$$t:\{2,3,\ldots,n\}\to\{1,2,\ldots,n\}$$

which respect $x_{i,t(i)}\varepsilon X_i$ for all i, $2\leqq i\leqq n$; this number is just $\prod_{i=2}^{n}|X_i|$. This product is maximised, subject to the constraint on the sum $\sum_{i=2}^{n}|X_i|$, when $|X_i|$ is independent of i, thus:

$$\begin{aligned}|\mathrm{mon}(abc)| &\leqq \prod_{i=2}^{n}|X_i|\\ &\leqq[(n-1)-(r+s)(n-r-s-1)/(n-1)]^{(n-1)}\end{aligned}$$

and a (crude) content bound is

$$c(r,s)=[(n-1)-(r+s)(n-r-s-1)/(n-1)]^{(n-1)}$$

It is an elementary observation that any parse tree rooted at ρ must contain at least one $\otimes$-vertex, α, whose degree lies in the range $[n/3, 2n/3]$. The content of this vertex α is bounded by

$$|\text{content}(\alpha)| \leq \max_{n/3 \leq r+s \leq 2n/3} c(r,s)$$

Now $c(r,s)$ is dependent only on the sum $(r+s)$ and achieves its maximum in the stated range at $r+s=2n/3$. Hence

$$\begin{aligned}|\text{content}(\alpha)| &\leq [(n-1)-2n(n-3)/9(n-1)]^{(n-1)} \\ &= [(7n^2-12n+9)/9(n-1)]^{(n-1)} \\ &\leq (7n/9)^{(n-1)} \qquad (n \geq 2)\end{aligned}$$

The weight of any parse tree rooted at ρ is certainly bounded below by $|\text{content}(\alpha)|^{-1}$ and hence, by theorem 5.14

$$\otimes\text{-complexity of } p \geq |\text{mon}(p)|\,(9/7n)^{(n-1)}.$$

The cardinality of mon(p) is precisely the number of directed spanning trees, rooted at 1, of the complete graph K_n. The number of such trees is $n^{(n-2)}$ (see Moon [22]), from which

$$\begin{aligned}\otimes\text{-complexity of } p &\geq n^{(n-2)}(9/7n)^{(n-1)} \\ &= n^{-1}(9/7)^{(n-1)}\end{aligned}$$

Thus we obtain an exponential bound valid for R,M, and M^+, that is, for the problems of counting the number of directed spanning trees of a graph, or of finding in a graph such a tree of minimum weight.

5.6 Discussion of Results

In the previous section, lower bounds were obtained for the ⊗-complexity of a wide range of functions in different semirings. Some of the results, such as the exponential lower bound for the minimum spanning tree computation, stand in stark contrast to the known tractability of the problem, and raise questions as to the relevancy of the results to actual computations. The lower bounds can therefore be interpreted in two complementary ways: on the one hand they deny the existence, for many problems, of fast "combinatorial" algorithms which work independently of the domain of computation, while on the other hand they affirm the power of algorithms which exploit the algebraic idiosyncracies of a specific problem. Let us use the results of the previous section to explore the efficiency which can be gained by using less restrictive models of computation.

Our model of computation suffers from two weaknesses; the more obvious is the restriction on the allowed operations. In the arithmetic case, only computations not involving subtraction were considered. That such a restriction could entail an exponential penalty was already known; Valiant [42] treats the example of the generating function of perfect matchings in a planar graph. In the same vein, the results presented here indicate an exponential gap for the spanning tree polynomial. From example (v) of the previous section we learn that any monotone arithmetic computation for the spanning tree polynomial requires at least $n^{-1}(9/7)^{(n-1)}$ multiplications, while in contrast, if negative constants are allowed, the same polynomial can be expressed as an n×n determinant whose elements are linear combinations of the indeterminates (see for example Moon [22]). The determinant may be evaluated via the method of Aho et al. [1], coupled with the matrix multiplication technique of Schonhage, using $O(n^{2.52})$ multiplications/divisions; an observation of Strassen [38] allows the divisions to be eliminated at the expense of increasing the number of multiplications to $O(n^{3.52})$.

Even for functions which have polynomial monotone complexity, subtraction is still helpful. From example (i) we have that, in the monotone case, multiplication of two n×n matrices requires at least n^3 multiplications, whereas, allowing negative constants, Schonhage's method [33] computes the product in $O(n^{2.52})$ multiplications. Similarly a gain can be achieved for the convolution of example (ii) using the fast Fourier transform method ([1] p.257). A very modest gain can be demonstrated for the permanent function of example (iii): any monotone computation requires at least $n(2^{n-1}-1)$ multiplications, however, using a modification of the inclusion-exclusion technique of Ryser ([23], p.158), the same computation can be effected using subtraction in only $(n-1)2^{n-1}+3$ multiplications. The interest in this case is that, although small, the complexity gap is the only one known for a 0-1 polynomial which is algebraically complete in the sense of chapter 2.

All this evidence points to the value of complex algorithms which exploit the particular characteristics of the domain of computation, in this case the ability to form monomials which cancel out in subtle ways in the result. Of particular interest is the power of linear algebra to make tractable polynomials whose monotone complexity is exponential. In contrast, it is noteworthy that augmenting the allowed set of operations with division and performing computation over the rational functions is of limited value, as division can be simulated by truncated power series (Strassen [38]).

The second weakness of the model is less obvious, since it is not usually encountered in algebraic complexity. What is essentially a straight-line algorithm (s.l.a) model is used to measure the complexity of computation, neglecting the additional computational power that branching (test and branch instructions) can provide. It is well known (see for example Strassen [39]) that branching cannot help in the computation of polynomials over an infinite field, so that the model is adequate for R in this respect. The situation is however completely different in M or M^+ where branching can yield dramatically shorter computations. To return to the example of the spanning tree polynomial ((v) of the previous section), we learn that $n^{-1}(9/7)^{(n-1)}$ additions are necessary to compute the polynomial using a straight-line algorithm, whereas the same polynomial can be computed in $O(n^2 \log n)$ min, + operations if branching is allowed ([16],p.348). As another demonstration of an exponential gain, we might consider the permanent, which over M^+ is connected with the minimal assignment problem. In the min, + algebra, the computation of the permanent requires $n(2^{(n-1)}-1)$ additions, but with branching the same computation can be performed using only $O(n^3)$ operations ([16], p.205). Indeed, we can paraphrase Valiant [42] and assert that "branching can be exponentially powerful".

A final lesson we may drawn is that the algebraic idiosyncrasies of different semirings can cause functions described by the same formal polynomial to have radically different complexities. In fact, only one consistent relation emerged from a study of the semirings considered here: it is always easier to compute a 0-1 polynomial over B than M, M^+ or R. (Loosely speaking, checking the existence of a solution to a problem is always easier than finding an optimum solution or counting their number.) This gap can be exponential: the spanning tree polynomial ST has exponential complexity over M, M^+ and R, but polynomial over B. Over B, $ST_{n\times n}(X)=1$ iff the graph whose adjacency matrix is X has a directed spanning tree rooted at 1, that is to say if a directed path exists from each vertex to 1. However the latter condition may be checked by computing the transitive closure X^* of X and-ing the elements in the first column of X^*, a procedure which can be accomplished in $O(n^3)$ operations ([1], p.199). Another interesting case is provided by the simple paths polynomial SP of example (iv). If x_{ij} is the length of edge (i,j) in K_n, then $SP_{n\times n}(X)$ represents, over M,M^+, the length of a minimum simple path from vertex 1 to vertex n. In M^+ this is equal to the minimum length of a path from 1 to n and can be computed in $O(n^3)$ operations ([1], p.202). Over M, however, the polynomial has exponential complexity. The same exponential bound is valid, over R, for the problem of enumerating simple paths (where X is the adjacency matrix of the graph).

ACKNOWLEDGEMENT

It is a pleasure to thank those who assisted in the production of this thesis. Threefold thanks are due to Leslie Valiant: for initially suggesting fruitful lines of enquiry, for helpful discussions and encouragement while the work was in progress, and finally for reading the manuscript. I would like to record my gratitude to Marc Snir; through my collaboration with him I learnt something of the nature and method of research. In addition I would like to thank Heather Carlin for seeing through a difficult typing job.

REFERENCES

1. A.V. Aho, J.E. Hopcroft and J.D. Ullman. The Design and Analysis of Computer Algorithms. Addison Wesley, 1974.

2. M.O. Ball. Computing Network Reliability. Operations Research 27 (1979), pp. 823-838.

3. C. Berge. Graphs and Hypergraphs. North-Holland, Amsterdam, 1973.

4. D. Bini, M. Capovani, G. Lotti and F. Romani. $O(n^{2.7799})$ Complexity for Matrix Multiplication. Information Processing Letters 8 (1979), pp. 234-235.

5. A. Borodin and I. Munro. The Complexity of Algebraic and and Numerical Problems. American Elsevier, New York, 1974.

6. R. Cuninghame-Green. Minimax Algebra. Springer Verlag, 1979.

7. A. Ehrenfeucht and P. Zeiger. Complexity Measures for Regular Expressions. Proc. 6th ACM Symposium on Theory of Computing (1974), pp. 75-79.

8. S. Even. Graph Algorithms. Computer Science Press, Potomac, Maryland, 1979.

9. K. Fan. On Systems of Linear Inequalities. In Linear Inequalities and Related Systems, H.W. Kuhn and A.W. Tucker (eds.). Princeton Univ. Press, Princeton N.J., 1956, pp. 99-156.

10. R. Godement. Algebra, Hermann, 1968.

11. G.B. Goodrich, R.E. Ladner and M.J. Fisher. Straight-line Programs to Compute Finite Languages. Proc. of a Conference on Theoretical Computer Science. Univ. of Waterloo, 1977, pp. 221-229.

12. O.J. Heilmann and E.H. Lieb. Theory of Monomer-Dimer Systems. *Commun. Math. Phys. 25* (1972), pp. 190-232.

13. R.M. Karp. Reducibility Among Combinatorial Problems. In *Complexity of Computer Computations*, R.E. Miller and J.W. Thatcher (eds). Plenum Press, New York, 1972.

14. P.W. Kasteleyn. Graph Theory and Crystal Physics. In *Graph Theory and Theoretical Physics*, F. Harary (ed.). Academic Press 1967, pp. 43-110.

15. E.A. Lamanga and J.E. Savage. Combinatorial Complexity of some Monotone Functions. *Proc. 15th IEEE Symposium on Switching and Automata Theory* (1974), pp. 140-144.

16. E.L. Lawler. *Combinatorial Optimisation*: *Networks and Matroids*. Holt Rinehart and Winston, New York, 1976.

17. K. Melhorn and Z. Galil. Monotone Switching Circuits and Boolean Matrix Product. *Computing* *16* (1976), pp. 99-111.

18. W. Miller. Computational Complexity and Numerical Stability. *JACM* *22* (1975), pp. 512-521.

19. H. Minc. *Permanents*. Encyclopedia of Mathematics and its Applications, 6. Addison-Wesley, 1978.

20. K.B. Misra. An Algorithm for the Reliability Evaluation of Redundant Networks. *IEEE Trans. Rel. 19* (1970) pp. 146-151.

21. E.W. Montroll. Lattice Statistics. In *Applied Combinatorial Mathematics*, E.F. Beckenbach (ed.). Wiley 1964, pp. 96-143.

22. J.W. Moon. *Counting Labelled Trees*. Canadian Mathematical Congress, Montreal, 1970.

23. A. Nijenhuis and H.W. Wilf. *Combinatorial Algorithms*. Academic Press, 1975.

24. V. Ya. Pan. New Fast Algorithms for Matrix Operations. *SIAM J. on Computing 9* (1980), pp. 321-342.

25. M.S. Paterson. Complexity of Monotone Networks for Boolean Matrix Product. *Theoretical Computer Science 1* (1975), pp. 13-20.

26. J.K. Percus. *Combinatorial Methods*. Applied Mathematical Sciences 4. Springer Verlag, 1971.

27. V.R. Pratt. The Power of Negative Thinking in Multiplying Boolean Matrices. *SIAM J. on Computing 4* (1975) pp. 326-330.

28. C.R. Rao. *Linear Statistical Inference and its Applications*. Wiley, 1973.

29. L. Redei. *Algebra Vol. 1.* Pergamon Press, Oxford, 1967.

30. A. Rosenthal. A Computer Scientist looks at Reliability Computations. In *Reliability and Fault Tree Analysis*, R.E. Barlow et al. (eds.). SIAM, Philadelphia, 1975, pp. 133-152.

31. A. Rosenthal. Computing the Reliability of Complex Networks. *SIAM J. Appl. Math. 32*, (1977), pp. 384-393.

32. C.P. Schnorr. A Lower Bound on the Number of Additions in Monotone Computations. *Theoretical Computer Science 2* (1976), pp. 305-315.

33. A. Schonhage. Partial and Total Matrix Multiplication. Manuscript, Mathematisches Institut, Universitat Tubingen, Tubingen, Germany.

34. E. Shamir and M. Snir. Lower Bounds on the Number of Multiplications and the Number of Additions in Monotone Computations. IBM, T.J. Watson Research Center Report, RC 6757, 1977.

35. E. Shamir and M. Snir. On the Depth Complexity of Formulas. Mathematical Systems Theory (to appear).

36. M. Snir. On the Size Complexity of Monotone Formulas. Proc. 7th International Colloquium on Automata, Languages and Programming (1980). Lecture Notes in Computer Science 85, Springer Verlag, pp. 621-631.

37. V. Strassen. Gaussian Elimination is not Optimal. Numerische Mathematik 13 (1969), pp. 354-356.

38. V. Strassen. Vermeidung von Divisionen. J. Reine und Angewandte Mathematik 264 (1973), pp. 182-202.

39. V. Strassen. Berechnung un Programm II. Acta Informatica 2 (1973), pp. 64-79.

40. L.G. Valiant. The Complexity of Computing the Permanent. Theoretical Computer Science 8 (1979), pp. 189-201.

41. L.G. Valiant. The Complexity of Enumeration and Reliability Problems. SIAM J. on Computing 8 (1979), pp. 410-421.

42. L.G. Valiant. Negation can be Exponentially Powerful. Proc. 11th ACM Symposium on Theory of Computing (1979), 189-196.

43. L.G. Valiant. Completeness Classes in Algebra. Proc. 11th ACM Symposium on Theory of Computing (1979), pp. 249-261.

44. L.G. Valiant. Universality Considerations in VLSI Circuits. IEEE Trans. on Computers 30 (1981), pp. 135-140.

45. L.G. Valiant. Reducibility by Algebraic Projections. University of Edinburgh Report CSR-64-80 (1980).

46. I. Wegener. Switching Functions whose Monotone Complexity is Nearly Quadratic. Theoretical Computer Science 9 (1979), pp. 83-87.