COMPLETENESS CLASSES IN ALGEBRA

L.G. Valiant
Computer Science Department
Edinburgh University
Edinburgh, Scotland.

0. Introduction

In the theory of recursive functions and computational complexity it has been demonstrated repeatedly that the natural problems tend to cluster together in "completeness classes". These are families of problems that (A) are computationally interreducible and (B) are the hardest members of some computationally defined class. The aim of this paper is to demonstrate that for both algebraic and combinatorial problems this phenomenon exists in a form that is purely algebraic in both of the respects (A) and (B). Such computational consequences as NP-completeness are particular manifestations of something more fundamental.

The core of the paper is self-contained, consisting as it does essentially of the two notions of "p-definability" and the five algebraic relations that are proved as theorems. In the remainder our aim is to elucidate the computational consequences of these basic results. Hence in the auxiliary propositions and discussion for convenience we do assume familiarity with algebraic and Boolean complexity theory [3,20].

Our basic technique is that of reducing polynomials to each other, or Boolean functions to each other, by projections (i.e. substitutions for indeterminates of constants or other indeterminates.) Our main conclusions can be summarized roughly as follows:

(a) Linear algebra offers essentially the only fast technique for computing multivariate polynomials of moderate degree.

(b) Numerous well-known but apparently intractable polynomials are interreducible via projections. Because of the closeness of this relationship they are not only computationally but also mathematically equally intractable. Furthermore we can identify them as belonging to the completeness class for p-definable polynomials.

(c) The question as to whether these complete polynomials can be computed fast is equivalent to the purely algebraic question as to whether they are projections of a moderate size determinant.

(d) Many NP-complete problems when regarded as finite Boolean functions are interreducible by projection. Furthermore they are complete for a class of functions that can be defined in terms of finite Boolean properties. The completeness class is meaningful even if $P = NP$.

(e) These NP-complete problems can be computed by polynomial size formulae if and only if they are projections of the transitive closure function of moderate size. Some purely algebraic or combinatorial approaches to proving $P \neq NP$ also suggest themselves.

1. Algebraic Definitions

Let F be a field and $F[x_1,\ldots,x_n]$ the ring of polynomials over indeterminates $x_1,\ldots,x_n$ with coefficients from F. P,Q and R will denote infinite families of polynomials where typically

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

and similarly for Q and R.

A formula f over F is an expression that is of one of the following forms:

See page ii

(i) "c" where $c \in F$, or (ii) "x_j" where x_j is an indeterminate, or (iii) "$(f_1 \circ f_2)$" where f_1 and f_2 are themselves formulae over F and $\circ$ is one of the two ring operators $\{+,\times\}$. The size of a formula f is the number of operations of type (iii) needed in its construction, and is denoted by $|f|$. For example the formula "$(((x_1 + 1) \times (x_2 + 2)) + 2)$" has size four. Any formula specifies a polynomial in the obvious way. The formula size $|P_i|$ of polynomial P_i is the size of the minimal size formula that specifies it.

If X is a set of indeterminates and A a set of polynomials then any mapping $\sigma: X \to A$ can be regarded as a substitution. If $P_i \in F[x_1,\ldots,x_i]$ and $X \subseteq \{x_1,\ldots,x_i\}$ then the substitution σ can be made in P_i and the resulting polynomial is denoted by P_i^σ. If $A = Y \cup F$ where Y is a set of indeterminates, then a mapping $\sigma: X \to A$ is a simple substitution.

Definition $Q_i \in F[y_1,\ldots,y_i]$ is a projection of $P_j \in F[x_1,\ldots,x_j]$ iff there is a simple substitution σ over F such that $Q_i = P_j^\sigma$.

Functions from positive integers to positive integers we shall denote typically by t. Such a t is p-bounded if for some constants K,k, for all n $t(n) \le K + n^k$.

Definition If P,Q are families of polynomials then Q is a t-projection of P if for all i there exist $j \le t(i)$ and σ such that $Q_i = P_j^\sigma$. It is a p-projection if it is a t-projection for some p-bounded t.

An example of a pair of polynomial families that are not projections of each other is the following:

$$P = \{P_i = \sum_1^i x_j\} \quad \text{and} \quad Q = \{Q_i = \prod_1^i x_j\}.$$

Note that our notion of substitution is very restricted as compared with some reasonable alternatives. (If in A we had allowed arbitrary linear combinations of indeterminates then P would have been a 1-projection of Q, and if arbitrary monomials had been allowed then the converse would have held.)

2. Universality of the Determinant

We show that every polynomial of formula size u is the projection of the $(u+2) \times (u+2)$ determinant. To interpret this note that the determinant itself has formula size $2^{0(\log^2 n)}$ [6,14] which is less than strictly exponential (i.e. 2^{n^ε} for $\varepsilon > 0$.) We conclude that for the problem of finding a subexponential formula for a polynomial when one exists, linear algebra is essentially the only technique in the sense that it is always applicable.

A more significant interpretation as far as computation follows from Hyafil's result [14]. He showed that for some constant α any polynomial of degree d that can be computed by a straight-line program in C steps has formula size $C^{\alpha \log d}$. Consider polynomial families in which the degree is p-bounded in terms of the number of indeterminates. Define a function $t(n)$ to be qp-bounded (quasi-polynomial) if it is bounded above by $2^{\log^k(n)}$ for some constant k. Then the class of polynomials with qp-bounded formula size is the same as the class of polynomials that can be computed by programs of qp-bounded length. Hence a polynomial with p-bounded degree can be computed in qp-time if and only if it is the projection of a determinant of qp-bounded dimensions.

Let Y be an $n \times n$ matrix of indeterminates $\{y_{ij} \mid 1 \le i,j \le n\}$. Let G be a directed graph on n nodes $\{1,2,\ldots,n\}$ in which edge (i,j) is given weight y_{ij}. A cycle cover of G is a set of n edges that together form a set of disjoint directed cycles in G (i.e. every node must be visited exactly once.) Now

$$\mathrm{Det}(Y) = \sum_\pi (-1)^{\mathrm{sgn}(\pi)} \prod_{i=1}^n y_{i,\pi(i)}$$

where summation is over all $n!$ permutations on $\{1,\ldots,n\}$. Clearly there is a one-to-one correspondence between cycle covers in G and permutations. Also, each π is the product of cyclic permutations, and the latter correspond one-to-one with the cycles in the cycle cover. Since $\mathrm{sgn}(c)$ is $+1$ if and only if c is a cycle of odd length, if the cycle covers in G consist entirely of odd length cycles then

$$\sum_{cc} (\text{product of weights on } cc)$$

summed over all the cycle covers of G will equal $\det(Y')$ where (a) $(i,j) \in G \Rightarrow y'_{ij} = y_{ij}$ and (b) $(i,j) \notin G \Rightarrow y'_{ij} = 0$. Denoting the $n \times n$ determinant by $\mathrm{Det}_{n \times n}$ we prove the following:

Theorem 1 If $P_i \in F[x_1,\ldots,x_i]$ then P_i is the projection of $\mathrm{Det}_{s \times s}$ where $s = |P_i| + 2$.

Proof We first define a mapping

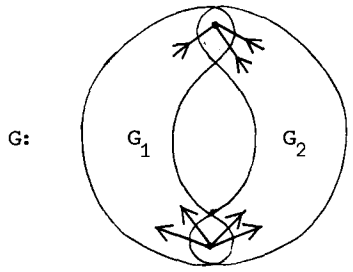
$$H : \{\text{formulae}\} \to \{\text{graphs}\} \times \{0,1\}$$

recursively in the construction of the formula. Note that for any formula f if $H(f) = (G,r)$ then (i) G will be acyclic with distinguished source and sink nodes s and t respectively, and (ii) either every path from s to t is of odd length, in which case $r = 1$, or every path is of even length in which case $r = 0$.

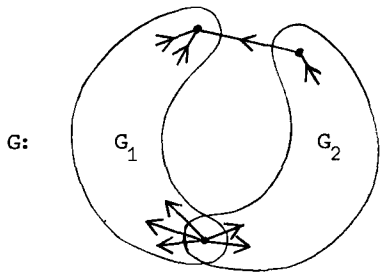
(a) If $f =$ "c" or "y_j" then $H(f) = (G,1)$ where G has node set $\{s,t\}$ and just the one edge (s,t) which is given weight c or y_j as appropriate.

(b) If $f = (f_1 + f_2)$ where $H(f_1) = (G_1,r_1)$ and $H(f_2) = (G_2,r_2)$ then

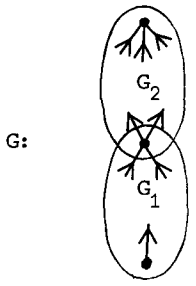
(i) if $r_1 = r_2$ then $H(f) = (G,r_1)$ where G is the disjoint union of G_1 and G_2 but with the two source nodes and the two sink nodes identified:



(ii) if $r_1 \neq r_2$ then $H(f) = (G,r_1)$ where G is the disjoint union of G_1 and G_2 with the two sources identified, and with an additional edge, weighted one, from the sink of G_2 to the sink of G_1 :



(c) If $f = (f_1 \times f_2)$ where $H(f_1) = (G_1,r_1)$ and $H(f_2) = (G_2,r_2)$ then $H(f) = (G,r_1 + r_2 \bmod 2)$ where G is the disjoint union of G_1 and G_2 but with the sink of G_1 identified with the source of G_2



We claim that for any f if $H(f) = (G,r)$ then

$$\sum_{stp} \text{product of weights on } stp,$$

summed over all directed paths from s to t in G, is equal to the polynomial represented by f. The reader may verify this easily by induction on the construction of G. Also by induction it follows that G has at most $s = |f| + 2$ nodes.

To establish the theorem consider $H(f) = (G,r)$ where f is a minimal formula for P_i. Modify G to G' as follows: First add a self-loop (i.e. edge (k,k)) weighted one to every node k of G that is not s or t. If $r = 0$ add a new edge weighted one from the sink to the source. If $r=1$ identify the sink and the source. In either case every cycle cover of G' will consist of one non-trivial cycle of odd length and a number of self-loops (i.e. of length one). Now the polynomial represented by f will equal

$$\sum_{cc} (\text{product of weights on } cc)$$

with summation over all cycle covers of G. By the preliminary observations this will equal the appropriate projection of the $s \times s$ determinant. □

3. p-Definability

For numerous widely studied polynomials only exponential size formulae are known. In the majority of cases it turns out that they can still be described succinctly because their coefficients can be specified by a small formula. For reasons which should become clear later, we capture this notion as follows:

Definition A family P of polynomials over F is p-definable iff

either (a) there is a family Q over F and a p-bounded t such that for all i $|Q_i| \leq t(i)$ and

$$P_i = \sum_{\sigma} Q_i^{\sigma} \prod_{\sigma(x_k)=1} x_k ,$$

summation being over the 2^j substitutions $\{x_1,\ldots,x_j\} \to \{0,1\}$ for some $j,(0 \leq j \leq i)$, if $Q_i \in F[x_1,\ldots,x_i]$,

or (b) P is the p-projection of some p-definable family.

[N.B. Allowing $j < i$ is useful but technically redundant. As we shall see every p-definable family is the p-projection of some family that can be defined with $i = j$.]

If Q is a family that satisfies condition (a) of this definition for P then we say Q defines P. Clearly every family of p-bounded formula size defines itself since we can take $j = 0$.

In the next two sections we shall be concerned with p-definable polynomials that in all probability do not have p-bounded formulae and are intractable. In the remainder of this section we shall give techniques for identifying polynomials as p-definable. Direct verification is often very cumbersome.

Given an $n \times n$ matrix Y of indeterminates $\{y_{ij} \mid 1 \leq i,j \leq n\}$ we define the permanent of Y as

$$\mathrm{Perm}_{n\times n} = \sum_{\pi} \prod_{i=1}^{n} y_{i,\pi(i)}$$

with summation over all $n!$ permutations of $\{1,\ldots,n\}$.

Proposition 1 Over any F the permanent is defined by the family Q where

$$Q_{n\times n} = \left(\prod_{i=1}^{n} \sum_{j=1}^{n} y_{ij} \right) \left(\prod_{\substack{i=k \\ \text{or } j=m}} (1 - y_{ij}y_{km}) \right)$$

Proof By expanding the shorthand notation it is clear that $Q_{n\times n}$ has formula size $O(n^3)$. Also, for input values from the set $\{0,1\}^{n^2}$, $Q_{n\times n}$ is zero if some row is all zero, or if any two ones are in the same row or column. □

Since p-definability is concerned with the evaluation of algebraic formulae at $\{0,1\}$ the consideration of Boolean formulae is often useful. (See §7 and [20]).

Proposition 2 For any F there is a mapping from Boolean functions to polynomials over F that keeps formula size p-bounded and has the following property: any $g \in B[x_1,\ldots,x_n]$ maps to $f \in F[x_1,\ldots,x_n]$ such that for all vectors $\underline{v} \in \{0,1\}^n$, $g(\underline{v}) = 1 \Rightarrow f(\underline{v}) = 1$ and $g(\underline{v}) = 0 \Rightarrow f(\underline{v}) = 0$.

Proof We use the well-known result that there is a translation that takes any Boolean formula of size s to an equivalent one of depth $O(\log s)$ and size p-bounded by s ([20],p26)., Now each Boolean operation can be simulated in any F on the domain $\{0,1\}$: $x \wedge y$ by xy , $\bar{x}$ by $1-x$ and $x \vee y$ by $x+y-xy$. If a Boolean formula of depth d translates into an algebraic one of size $S(d)$ then $S(d) \leq cS(d-1)$ where c is the maximal number of algebraic operations needed to realise any of the three Boolean operators. It follows that a formula of size S will be translated to one of size $S(d) \leq c^d \leq c^{O(\log s)} \leq s^k$ for some constant k. □

It is sometimes useful to employ other Boolean operators also. The following illustrates the use of "exclusive-or". Clearly this causes no extra problems since $x \oplus y$ can be simulated by $x+y-2xy$ just as well.

Proposition 3 The determinant is defined by the family

$$\bar{Q}_{n\times n} = Q_{n\times n}(1 - 2\tilde{Q}_{n\times n})$$

where $Q_{n\times n}$ is as in Proposition 1, and $\tilde{Q}_{n\times n}$ is the formula that on domain $\{0,1\}$ equals the following Boolean formula

$$\bigoplus_{i,m} \bigoplus_{\substack{j>m \\ \text{and} \\ k<i}} x_{im}x_{kj}$$

Proof By inspection. □

Usually it is sufficient to establish the existence of a small defining formula, and we do not need to find an elegant one. In these

circumstances the following is very often sufficient.

Proposition 4 Suppose $P = \{P_1, P_2, \ldots\}$ is a family of polynomials over F where every monomial has coefficient one (or zero). Suppose that there is a p-time algorithm that for any vector $\underline{v} \in \{0,1\}^n$ can determine whether the coefficient of

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

is one. Then P is p-definable over F.

Proof Consider a deterministic $t(n)$-time bounded one-tape Turing acceptor M for the hypothesized problem. Then computation sequences of M can be described by sequences of $O(t^2)$ binary symbols. Furthermore, there is a Boolean formula g of p-bounded size that determines for such a sequence of symbols whether it represents an accepting computation for a specified input. Now translate g to a formula f over F that is "equivalent" to it in the sense of Proposition 2. Suppose the indeterminates of g are $x_1, x_2 \ldots, x_n, x_{n+1}, \ldots, x_r$ where $x_1, \ldots, x_n$ correspond one-to-one to $v_1, \ldots, v_n$, and call the polynomial that g represents Q_r. Then clearly

$$P'_r = \sum Q_r^{\sigma} \prod_{\substack{\sigma(k)=1 \\ 1 \le k \le r}} x_k$$

is p-definable if summation is over the 2^r substitutions $\sigma: \{x_1, \ldots, x_r\} \to \{0,1\}$. But for each $\underline{v} \in \{0,1\}^n$ that corresponds to an accepting computation of M there is exactly one substitution σ that agrees with it for $1 \le i \le n$ and gives $Q_r^{\sigma} = 1$, namely the one describing the correct computation. Hence P_n is the p-projection of P'_r under the substitution that sets to one each x_k with $k \ge n$, and leaves the others unchanged. □

Remark 1 By the same argument it follows that for each predicate computable in nondeterministic p-time there is an associated p-definable polynomial, but now the coefficient of each monomial is the number of accepting computations rather than unity.

Using this last result it is easy to verify that most of the frequently occurring generating polynomials for combinatorial structures are p-definable. The examples below are specified as follows: Let G be the complete directed graph on the n nodes $\{1, \ldots, n\}$ with edge (i,j) labelled by indeterminate y_{ij}. Let S be a set of subsets $\{E_1, \ldots, E_m\}$ of the edges of G. Then the polynomial for S over F is defined as

$$\sum_{k=1}^{m} \prod_{(i,j) \in E_k} y_{ij} .$$

From Proposition 4 it is clear that each of the following polynomial families is p-definable.

I Permanent: S = {cycle covers.}

II Self Avoiding Walks: S = {paths from node 1 to node 2 that do not go through any node more than once.}

III Hamiltonian Paths: S = {self-avoiding walks from node 1 to node 2 of length exactly $n-1$.}

IV Hamiltonian Circuits: S = {self-avoiding cycles of length n.}

V Spanning Trees: S = {spanning trees in which each edge is directed away from node 1.}

VI Reliability: $S = \{E_i \mid$ there is a path from node 1 to 2 in $E_i\}$.

For each of the above we get the corresponding undirected case if we identify each y_{ij} with y_{ji}. In that case the following are further natural p-definable problems.

VII Matchings: $S = \{E_i \mid$ no two edges in E_i are incident with the same node.}

VIII Perfect matchings: S = {matchings consisting of exactly $n/2$ edges.} (n assumed even.)

IX Connected components: $S = \{E_i \mid E_i$ forms a single connected component.}

4. Complete Problems

We now show that some of the above defined polynomials are of maximal intractability in the following strong sense:

Definition A p-definable family P over F is complete over F if every p-definable family Q over F is a p-projection of P.

Note that the problems that are complete over any one field F are all p-projections of each other. Hence they share all mathematical and computational properties that are preserved under

simple substitutions. Computational complexity is one example of such a property.

To identify a p-definable family as being complete we need to show that some known complete problem is a p-projection of it. In practice the following two theorems appear to suffice as starting points.

Theorem 2 If F is any field with characteristic not equal to two, then the Permanent is complete over F.

Proof We consider an arbitrary p-definable family P and show that it is the p-projection of the permanent. Therefore suppose that P is the p-projection of some $\bar{P}$ which in turn is defined by the family Q. Consider a particular member P_m which is therefore the projection of some $\bar{P}_i$ such that

$$\bar{P}_i(x_1,\ldots,x_j) = \sum_{\sigma\in\{0,1\}^j} Q_i^{\sigma} \prod_{\sigma(x_k)=1} x_k$$

where summation is over the 2^j assignments to $\{x_1,\ldots,x_j\}$. Now consider a minimal size formula f for Q_i, and construct from it a graph G' exactly as in the proof of Theorem 1. [N.B. Keeping track of the parity r is actually superfluous in the current proof.] By the argument given there the projection of $Perm_{n\times n}$ that is specified by the edge weights of G' will equal Q_i. What we need to do is to modify G' to G'' so that G'' specifies a projection of $Perm_{n\times n}$ that equals not Q_i but the polynomial $\bar{P}_i$ that it defines.

To do this we first add an isolated cycle labelled x_k for each k $(1\le k\le j)$. Then we superimpose a global structure that ensures that in any cycle cover that contains the x_k cycle all x_k weighted edges in G' have effective weight one, while in any cover not containing the x_k-cycle all x_k edges have effective weight zero. If this can be achieved then the permanent of G'' will equal $\bar{P}_i$ since the coefficient of each Πx_k product will be just the value of Q_i evaluated at the appropriate input vector from $\{0,1\}^j$.

The global structure connects each x_k edge in G' with the corresponding x_k cycle via a separate co-ordinator. The introduction of each co-ordinator involves eight new nodes, as shown in Figure 1.

A co-ordinator consists of two identical 4-node junctions. Denoting the nodes by $\{1,2,3,4\}$ a junction has the property that in any cycle cover that enters it at node 1 and leaves it at 4, or vice versa, its contribution is a multiplicative factor of 4. In any other kind of cycle cover its contribution is a factor of zero, and hence all such cycle covers are effectively cancelled out. The construction of the co-ordinator ensures that in any non-vanishing cycle cover either both junctions are traversed $4\to1$, or both $1\to4$. In other words either both the x_k cycle and the x_k weighted edge in G' are effectively included, or neither one is.

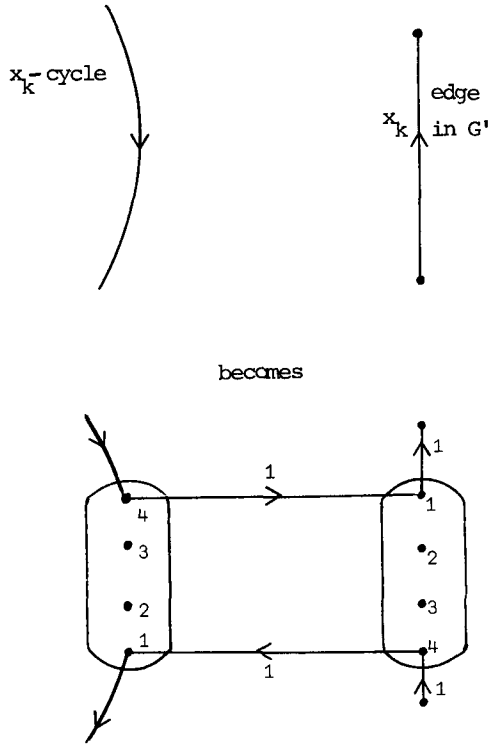


Figure 1: A co-ordinator of G"

Each function is implemented by a weighted directed graph proposed in [21], whose adjacency matrix is

$$X = \begin{pmatrix} 0 & 1 & -1 & -1 \\ 1 & -1 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 3 & 0 \end{pmatrix}$$

If $X[\gamma;\delta]$ denotes X but with rows γ and columns δ removed then

$\text{Perm } X(1;4) = \text{Perm } X(4;1) = 4$, but

$\text{Perm } X = \text{Perm } X(1;1) = \text{Perm } X(4;4) = \text{Perm } X(1,4;1,4) = 0.$

It can be verified that these properties ensure that the functions behave as claimed.

In G'' each x_k cycle will contain a number of junctions joined in a ring by a number of edges. Amongst the latter edges we label just one by x_k and the rest by unity. All the aims of the construction are now achieved except that each junction contributes a factor of four rather than one. To compensate for this we insist on an edge from the sink of G' to the source and give it weight $(2^{-1})^{2J}$ where J is the total number of junctions in G''. The permanent of the adjacency matrix of G'' is then $\bar{P}_j$ as required. □

Remark 2 If char $F = 2$ the proof fails because 2^{-1} does not exist. Furthermore the technique itself fails since the permanent and determinant are then identical, and no matrix with the determinental properties required of X exists.

Remark 3 The question as to whether there is a matrix transformation that translates a permanent into a determinant, or vice versa, was asked for the first time apparently by Polya [18]. Except for the trivial case of $n = 2$ no positive result was previously known. The strongest negative result was that of Marcus and Minc [17] who showed that even if substitutions of linear forms are allowed but the matrix size is preserved, neither function is the image of the other.

Proposition 5 For some constant c for all n $\text{Det}_{n\times n}$ is the projection of $\text{Perm}_{m\times m}$ for $m = cn^4$.

Proof If char $F = 2$ then the two polynomials are identical. Otherwise apply Theorem 2 to Proposition 3. □

Proposition 6 For all n $\text{Perm}_{n\times n}$ is the projection of $\text{Det}_{t(n)\times t(n)}$ where $t(n) < n^2 2^n$.

Proof Ryser [19, p26] gives a formula for the permanent of size $(n^2 2^n)$. The result follows from Theorem 1 for any F. □

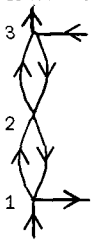
Proposition 7 For any given F with char $F \neq 2$ Proposition 6 holds for a qp-bounded t if and only if every qp-definable polynomial family has qp-bounded formula size over F. □

Our second completeness result is for the Hamiltonian Circuit polynomial defined in §3. It is different in that it holds for any field F.

Theorem 3 The Hamiltonian circuit polynomial is complete over any field F.

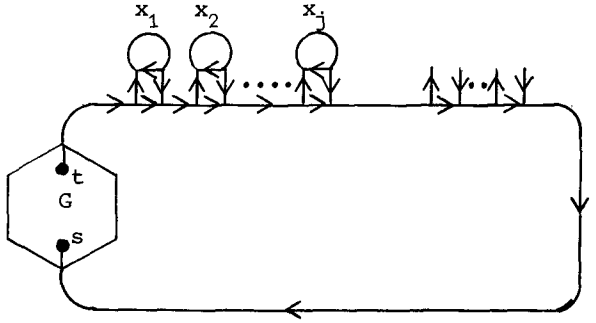
Proof We denote the directed Hamiltonian circuit polynomial over n nodes by $HC_{n\times n}$. As observed before, it is p-definable. To show that any p-definable family P is the p-projection of it we consider P_m, $\bar{P}_i$ and Q_i as in Theorem 2, and construct G' from the minimal formula for Q_i exactly as there.

To obtain the necessary G'' from G' we first note that junctions can be much simplified to

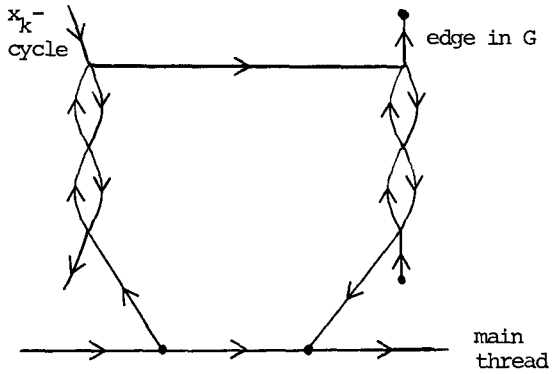


Since every Hamiltonian circuit must enter at node 1 and leave at 3, or vice versa, each edge internal to the junction can be weighted one.

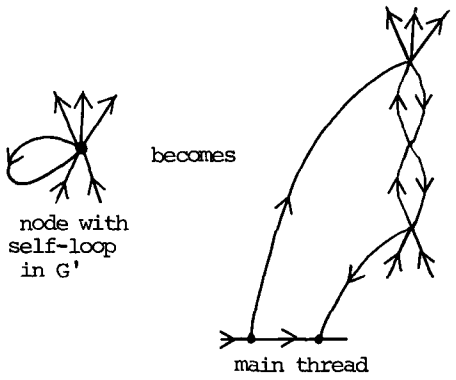
We introduce an x_k- cycle for each $k(1 \le k \le j)$ but break each one and "thread" them together with the main cycle of G' as follows:



The unspecified "beads" at the right of the diagram thread all the remaining potential cycles in G' , namely the self-loops and the main cycle of each co-ordinator. Thus each co-ordinator is now of the form:



To cope with self loops we modify every node in G other than s and t so that it becomes a junction.



It can be verified that the projection of HC specified by G" gives $\bar{P}_j$. □

5. More Complete Problems

Multivariate polynomials of the kind defined in §3 occur in diverse contexts. For example, the reliability and connected components problems have obvious applications to unreliable networks [8] and are also related to percolation problems [9,23]. The polynomials for self-avoiding walks, matchings and perfect matchings appear as generating functions for the corresponding counting problems in several branches of the physical sciences [2,12,16].

Despite exhaustive research nearly all of them have defied detailed mathematical or computational analysis. Even mere approximations for special cases appear difficult to obtain [13].

There are essentially only two interesting cases that are known to be tractable. The undirected spanning tree polynomial was solved by Kirchhoff, and the result later extended to the directed case (see [4]). Perfect matchings for planar graphs were solved by Kasteleyn and Fisher (see [16]). Both of these results express the required polynomial in terms of a determinant, a fact which suggests that the interpretation we claimed for our Theorem 1 has some validity.

In this section we shall illustrate how the intractability of many of the remaining problems can be explained away in terms of our notion of completeness.

Our first example shows that the considerable efforts made to extend the Kasteleyn-Fisher technique just to regular rectangular lattices in three dimensions was doomed to fail. The reader should note that here there was no previous concrete indication that counting solutions was difficult. This contrasts with Hamiltonian circuits where solutions are hard even to detect [15] and general perfect matchings where solutions are hard to count [21]. The example highlights the fact that for counting combinatorial structures in apparently harmless special cases the algebraic approach can easily introduce evident intractability.

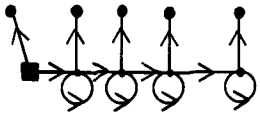
Proposition 8 Denote by G_n the graph on $2n^2$ nodes arranged as a 3-dimensional slab with integer co-ordinates

$$\{(i,j,k) \mid 1 \le i,j \le n \ ; \ k = 0,1 \},$$

with every pair of nodes separated by unit distance connected by an edge. Then the perfect matching polynomial for this restricted family of graphs is complete for F if $\mathrm{char}\, F \neq 2$.

Proof First note that given any weighted directed graph G (e.g. the complete graph) we can construct a G' , with maximal indegree and outdegree of 3, that has the same permanent. For each node we transform the edge set incident from it as follows:

becomes



The edge sets incident into the nodes are treated similarly.

In turn we can translate G' to an undirected bipartite graph G'' of maximal degree three, of which the perfect matching polynomial is the permanent of G. We conclude that there is a family $\{H_1, H_2, \ldots\}$ of undirected bipartite graphs of maximal degree three such that the associated family of perfect matching polynomials is complete over F.

We now show that the perfect matching polynomial for $\{G_n\}$ is also complete, by suitably embedding each H_n into G_m where $m = r^2 + 1$ and $r = 6n$.

The idea of the embedding is to map each edge in H_n to a chain of edges of odd length in G_m such that these chains are node-disjoint except for the ends. An edge that is matched in H_n will correspond to a chain in which the two endmost edges are matched in G_m.

In particular if H_n has nodes $\{1,2,\ldots,n,1',2',\ldots,n'\}$ then node i will map to $(ir,0,0)$ and node i' to $(ir,m,0)$. An edge connecting i to j' in H_n is mapped ideally to the three straight sequences of edges successively joining $(ir,0,0)$, $(ir,ir+j,0)$, $(jr,ir+j,0)$ and $(jr,m,0)$. Since up to six chains may compete for the same path in the first or third of these sequences, they will be displaced by the appropriate number of units in the horizontal dimension. When a horizontal chain crosses over a vertical one the collision is avoided by rerouting it to the $k = 1$ plane. Note that irrespective of the implementation details each chain will be of odd length since the total horizontal displacement is even, and the vertical one odd. □

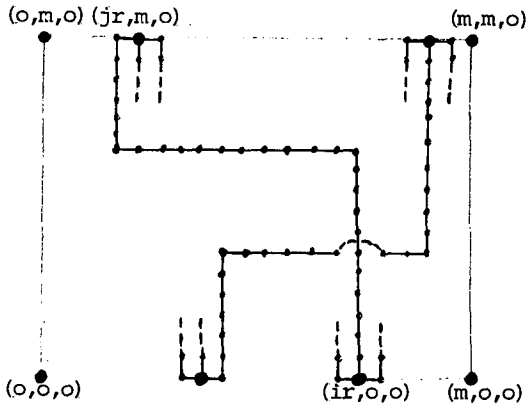


Fig.2 Embedding of two intersecting edges.

In the remainder of this section we shall observe that the intractability of several polynomials for regular lattices is already implicit in published proofs of NP-completeness. For example, since Hamiltonian paths in planar graphs are NP-complete [10] we would expect that by embedding in the 2-dimensional rectangular lattice the special planar graphs used in the proof we can obtain our algebraic reduction. In this manner one can verify algebraic intractability for many NP- and #P-complete problems when restricted to such regular graphs.

For suitable definitions of the appropriate polynomials and using constructions from [10,11,15] one can establish such reduction sequences as the following: Hamiltonian circuits → Satisfiability → Chromatic number → Exact 3-cover → Planar directed Hamiltonian paths → Directed self-avoiding walks in 2-D rectangular lattice, and Exact 3-cover → Connected components in 2-D rectangular lattice.

The reader can verify that these reductions establish for each problem either (i) that it is complete or (ii) that it is a homogeneous component of a complete problem, or (iii) that it is a certain coefficient in a complete multilinear family. The latter two properties are certainly equivalent to the first as far as computational complexity. Any program can be modified to produce just one homogeneous component of it with only quadratic increase in size [14]. If it is multilinear then any coefficient can be abstracted by similar techniques.

Among the above defined problems some are already polynomial families in the strict sense

that there is one member for each cardinality of indeterminates. In others, such as satisfiability, we have an exponential number, while for Hamiltonian paths we have a polynomial number by the freedom to choose nodes 1 and 2. Problems of the latter sort are characterized by a family $\{\mathfrak{I}_1, \mathfrak{I}_2, \ldots\}$ where $\mathfrak{I}_i$ is a class of polynomials with i indeterminates. In this case we define a problem to be complete if for some choice $\{P_i \in \mathfrak{I}_i\}$ we get a complete family in the normal sense.

Finally we note that the Hamiltonian circuit polynomial HC should not be confused with the Hamiltonian Graph polynomial

$$HG_{n\times n} = \sum_{S} \prod_{(i,j)\in E_k} y_{ij}$$

where S characterizes the graphs that contain Hamiltonian circuits. Clearly if $P = NP$ then by Proposition 4 and Theorem 3 HG would be a p-projection of HC. By proving the nonexistence of such a relation in a particular field one could in principle prove $P \neq NP$ by an algebraic (or combinatorial if $F = GF(2)$) argument.

6. Operations on Polynomials

Consider the problem of finding some specified coefficient of a multivariate polynomial P_i. If P_i is multilinear then no coefficient can be much more difficult to compute than P_i itself. That the problem is difficult in the general case, however, follows from the fact that the coefficient of $y_1 \cdots y_n$ in the trivial polynomial

$$\prod_{k=1}^{n} \sum_{i=1}^{n} x_{ki} y_i \qquad (*)$$

is the permanent of the $\{x_{ki}\}$ matrix. What we shall observe in this section is that the maximal difficulty of deriving coefficients is well characterized by this example.

Definition If $P_n \in F[x_1, \ldots, x_n]$ and m is a monomial $x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ then the coefficient of m in P_n is the unique polynomial Q_n where (i) $P_n = mQ_n + R_n$, (ii) Q_n and m have no indeterminate in common, and (iii) each monomial in R_n differs from m in the exponent of at least one indeterminate.

Proposition 9 If P has p-bounded formula size and Q a family such that Q_i is a coefficient in $P_{t(i)}$ then Q is p-definable with respect to the parameter $t(i)$.

Proof By Theorem 1 P is the p-projection of the determinant. For a particular $P_n \in F[x_1, \ldots, x_n]$ and monomial m let the $r \times r$ matrix Y be a matrix projection of minimal dimensions such that $\det Y = P_n$. Introduce r^2 new indeterminates $Z = \{z_{ij} \mid 1 \le i,j \le r\}$ and let W be the matrix such that $W_{ij} = Y_{ij} z_{ij}$ for each i,j pair. We now claim that there is a polynomial $R \in F[Z \cup \{x_1, \ldots, x_n\}]$ of p-bounded formula size such that for $\sigma : Z \to \{0,1\}$

$$R^{\sigma}_{r^2+n} = \text{monomial } m' \text{ if } \prod_{\sigma(z_{ij})=1} Y_{ij}$$

is of the form mm' where m and m' have no indeterminate in common, and equals zero otherwise.

If $\bar{Q}_{r\times r} \in F(Z)$ is the polynomial of Proposition 3 then

$$\sum_{\sigma} \bar{Q}^{\sigma}_{r\times r} R^{\sigma}_{r^2+n} \prod z_{ij}$$

is clearly p-definable. But setting each z_{ij} to one then gives the coefficient of m in $\det Y = P_n$. □

Remark 4. Deducing a coefficient from a small formula is difficult even in the univariate case. If in (*) each y_i is replaced by z^{2^i}, and the x's replaced by integers then the problem of computing the coefficient of $z^{2^{n+1}-1}$ is equivalent to evaluating an integer permanent. Evidence of the difficulty of this is given in [21].

7. Boolean Definitions

Boolean analogues to the algebraic results of the previous sections can be developed in several ways. We shall restrict ourselves here to Boolean functions of arguments that range over $\{0,1\}$. Other possibilities include Boolean polynomials (i.e. formal polynomials where only the constant coefficients obey the Boolean Laws) or Boolean

polynomials with some additional identities (e.g. $x_i^2 = x_i$ or $x_i x_j + x_i = x_i$.)

Let $B[x_1,\ldots,x_n]$ be the class of 2^{2^n} Boolean functions of the arguments $\{x_1,\ldots,x_n\}$. P,Q,R will denote infinite families of such functions where typically $P = \{P_i \mid P_i \in B[x_1,\ldots,x_i]\}$. A Boolean formula is an expression that is either of the form (i) "c" for $c \in \{0,1\}$, or (ii) an argument "x_j" or a negated argument "$\bar{x}_j$", or (iii) an expression "$(f_1 \circ f_2)$" where f_1, f_2 are formulae and $\circ$ is one of the two operations "and" or "or" (denoted by $\times$ and $+$ respectively.)

The size $|f|$ of f is the number of operations of type (iii) needed in its construction. A formula represents a Boolean function in the obvious way. The formula size of $P_i \in B[x_1,\ldots,x_i]$ is the size of the minimal size formula for it, and is denoted by $|P_i|$. It is well known that our measure is p-bounded in terms of corresponding measures for all other choices for the $\circ$ operation [20].

If X is a set of arguments and A a set of Boolean functions then any mapping $\sigma : X \to A$ can be regarded as a substitution. If $P_i \in B[x_1,\ldots,x_i]$ and $X \subseteq \{x_1,\ldots,x_i\}$ then the substitution σ can be made in P_i and the resulting function is denoted by P_i^σ. A substitution is simple if $A = \{0,1\} \cup \{y_1,\ldots,y_r\} \cup \{\bar{y}_1,\ldots,\bar{y}_r\}$ where each y_j is an argument.

Definition $Q_i \in B[y_1,\ldots,y_i]$ is a projection of $P_j \in B[x_1,\ldots,x_j]$ iff there is a simple substitution σ such that $Q_i = P_j^\sigma$. The family Q is a p-projection of the family P if for some p-bounded t for all i there is a $j \le t(i)$ such that Q_i is the projection of P_j.

Note that such pairs of trivial families as $P = \{\sum_1^i x_j\}$ and $Q = \{\prod_1^i x_j\}$ are not projections of each other.

8. Universality of Transitive Closure

Suppose that Y is a matrix of n^2 argument symbols $\{y_{ij} \mid 1 \le i,j \le n\}$. Define the transitive closure function $\mathrm{Trans}_{n\times n} \in B[Y]$ by

$$\sum_{i=0}^{\infty} Y^i$$

It is well known that $\mathrm{Trans}_{n\times n}$ is defined and equal to $(I+Y)^n$ where I is the $n\times n$ identity matrix. Hence Trans is a p-projection of Y^n. This latter function will also therefore be proved to be universal.

Theorem 4 If $P_i \in B[x_1,\ldots,x_i]$ then P_i is the projection of the $(1,s)$ entry of $\mathrm{Trans}_{s\times s}$ where $s = |P_i| + 2$.

Proof From P_i we construct a graph G exactly as in Theorem 1. Now, with summation over all paths from s to t,

$$\sum_{stp} \text{products of weights on } stp$$

will equal the function P_i. But if in Y we set $y_{ij} = 0$ if edge (i,j) is absent from G, and y_{ij} to the weight of edge (i,j) otherwise, then the $(1,s)$ entry of Y^j will equal the contribution to P_i given by the s-t paths of length exactly j. The result follows. □

We conclude that every Boolean function of small formula size is the projection of the transitive closure function of a correspondingly small matrix. Since $\mathrm{Trans}_{n\times n}$ has formula size at most $n^{0(\log n)}$, the question as to whether some given family P has qp-bounded formula size is equivalent to this explicit combinatorial property in Boolean algebra. We note that in the Boolean case, although no analogue of Hyafil's result is known, the logarithm of formula size is intimately related to the space required to compute the function.

9. p-Definability for Boolean Functions

Next we observe that the formal Boolean analogue of the previously defined algebraic notion of p-definability is closely related to the concept of nondeterminism as traditionally applied to discrete computations.

Definition A family P of Boolean functions is p-definable iff either (a) there is a family Q and a p-bounded t such that for all i $|Q_i| \le t(i)$ and

$$P_i = \sum_{\sigma} Q_i^\sigma \prod_{\sigma(x_k)=1} x_k,$$

summation being over the 2^j substitutions

$\{x_1,\ldots,x_j\} \to \{0,1\}$ for some j $(0 \le j \le i)$, if $Q_i \in B[x_1,\ldots,x_i]$,
or (b) P is the p-projection of some p-definable family.

[N.B. Again, allowing $j < i$ is redundant since every p-definable family is a p-projection of some complete family that can be defined with $i = j$. In some cases, however, it allows for a more tractable defining family.]

Proposition 10 Suppose that S is a family $S_1,S_2,\ldots$ where S_n is a set of subsets of $\{1,2,\ldots,n\}$. Suppose that there is a polynomial time nondeterministic Turing Machine that, given any n and $s' \subseteq \{1,2,\ldots,n\}$, determines whether $s' \in S_n$. Then the function

$$P_n = \sum_{s \in S} \prod_{j \in s} x_j$$

specifies a p-definable family.

Proof Exactly as in the proof of Proposition 4, given n we can find Q_r and P'_r such that

$$P'_r = \sum Q_r^{\sigma} \prod_{\substack{\sigma(k)=1 \\ 1 \le k \le r}} x_k$$

In the Boolean case Q_r^{σ} will equal 1 for inputs that are accepted even by nondeterministic computations. Hence P_n is the p-projection of P'_r under the substitution that sets to one each x_k with $k \ge n$ and leaves the others unchanged. □

For such monotone functions as Hamiltonian circuits, p-definability can be verified trivially even without using the nondeterminsitic facility of Proposition 10. Using the terminology of §§3,5 but with a Boolean interpretation, this function is simply

$$HC_{n \times n} = HG_{n \times n} = \sum_{k=1}^{m} \prod_{(i,j) \in E_k} y_{ij}$$

where E_k is the k^{th} Hamiltonian circuit. Checking whether some set of edges is a Hamiltonian circuit can be done fast deterministicly. Note, however, that the function $HG = HC$ checks for an arbitrary graph whether it contains some Hamiltonian circuit rather than just whether it is one.

For some other monotone functions we do need nondeterminism as in the following example of the Satisfiability problem. We define it as a function of $2n^2$ arguments $X = \{x_{ij},y_{ij} \mid 1 \le i,j \le n\}$. A truth assignment to X will denote a conjunctive normal form formula f with n clauses and arguments $\{z_1,\ldots,z_n\}$ such that z_j appears in clause i iff $x_{ij} = 1$, and $\bar{z}_j$ appears in clause i iff $y_{ij} = 1$. Then the desired satisfiability function is

$$\sum_{X' \subseteq X} a(X') \prod_{x_{ij} \in X'} x_{ij} \prod_{y_{ij} \in X'} y_{ij}$$

where $a(X')$ is 1 or 0 according to whether the formula f corresponding to just the X' arguments being set to one is satisfiable or not.

[N.B. In §5 Satisfiability was a family of polynomials each of which was defined for a specific Boolean function. Thus the polynomial in $F[x_1,\ldots,x_i]$ corresponding to $Q_i \in B[y_1,\ldots,y_i]$ is

$$\sum_{Y' \subseteq Y} b(Y') \prod_{y_i \in Y'} x_i$$

where $b(Y')$ is 1 or 0 according to whether Q_i is 1 or 0 for the input values defined by: $y_i = 1 \Leftrightarrow y_i \in Y'$.]

Problems in NP that are not monotone (e.g. exact cover [15]) are also easily seen to be p-definable. By taking some natural representation of them as Boolean functions and renaming each pair $\{x_j,\bar{x}_j\}$ by new distinct arguments $\{y_j,z_j\}$ we obtain monotone p-definable functions. The original function is still p-definable since it can be recaptured by taking the projection $y_j \leftarrow x_j$ and $z_j \leftarrow \bar{x}_j$ for each j.

The converse implication that p-definability implies containment in NP is, of course, false since no uniformity is assumed within a family of functions.

10. Complete Boolean Functions

The p-definable Boolean functions have a completeness class that appears to contain the majority of those NP-complete problems that can be expressed as monotone Boolean functions. (e.g. satisfiability, cliques, colourability, Hamiltonian circuits.) Our purpose here is to point out

(a) that the class in which they are complete can be specified in terms of finite Boolean functions, in contrast with NP which is an infinite concept, and (b) that the complete problems are even more closely related to each other than previously realised - they can be obtained from each other by simple substitutions.

Definition A p-definable family P of Boolean functions is complete if every p-definable family Q is a p-projection of it.

Theorem 5 The Hamiltonian circuit function HC is complete.

Proof The construction is identical to that of Theorem 3 except for the following modifications. For each argument x_k we have both an x_k-cycle (weighted x_k) and an $\bar{x}_k$-cycle (weighted one.) Each x_k edge is linked to the x_k-cycle by a co-ordinator, and each $\bar{x}_k$ edge to the $\bar{x}_k$-cycle similarly. Furthermore each x_k cycle intersects with the $\bar{x}_k$ cycle at a junction to ensure that in each Hamiltonian circuit exactly one of them is traversed. The result follows. □

The reader can verify that such monotone functions as HC, satisfiability and cliques are all p-projections of each other. Anti-monotone problems such as node cover and colourability are also complete if represented in a nonstandard way in terms of absent edges. For non-monotone NP-complete functions such as exact cover one can usually find a subset of special cases that have a monotone non-standard representation and correspond to a monotone complete problem. Such a subset is often already implicit in known reductions.

11. References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman. The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass.(1974).

[2] M.N. Barber and B.W. Ninham. Random and Restricted Walks. Gordon and Breach, New York (1970).

[3] A. Borodin and I. Munro. The Computational Complexity of Algebraic and Numeric Problems. American Elsevier, New York (1975).

[4] S. Chaiken and D.J. Kleitman. Matrix tree theorems. J. Combinatorial Theory, Series A 24 (1978) 377-381.

[5] S.A. Cook. The complexity of theorem proving procedures. Proc. 3rd ACM Symp. on Theory of Computing (1971) 151-158.

[6] L. Csanky. Fast parallel inversion algorithms. SIAM J. on Computing, 5:4 (1976) 618-623.

[7] M.E. Fisher. Statistical mechanics of dimers on a plane lattice. Phys. Rev. 124 (1961) 1664-1672.

[8] H. Frank and I.T. Frisch. Communication, Transmission and Transportation Networks. Addison-Wesley (1971).

[9] H.N. Frisch and J.M. Hammersley. Percolation processes and related topics. J. Siam Appl. Math. 11 (1963) 894-918.

[10] M.R. Garey, D.S. Johnson and L.J. Stockmeyer. Some simplified NP-complete problems. Proc. 6th ACM Symp. on Theory of Computing. (1974) 47-63.

[11] M.R. Garey, R.L. Graham and D.S. Johnson. Some NP-complete geometric problems. Proc. 8th ACM Symp. on Theory of Computing (1976) 10-22.

[12] H.S. Green and G.A. Hurst. Order-Disorder Phenomena. Interscience. London (1964).

[13] J.M. Hammersley. Existence theorems and Monte Carlo methods for the monomer-dimer problem. Research Papers in Statistics 125-146.

[14] L. Hyafil. On the parallel evaluation of multivariate polynomials. Proc. Tenth ACM Symp. on Theory of Computing (1978) 193-195.

[15] 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).

[16] P.W. Kasteleyn. Graph theory and crystal physics. In Graph Theory and Theoretical Physics, (F. Harary, ed.), Academic Press (1967).

[17] M. Marcus and H. Minc. On the relation between the determinant and the permanent. Illinois J. Math 5 (1961) 376-381.

[18] G. Polya. Aufgabe 424. Arch. Math. Phys (3) 20 (1913) 27.

[19] H.J. Ryser. Combinatorial Mathematics. Carus Math. Monograph no. 14 (1963).

[20] J.E. Savage. The Complexity of Computing, Wiley, New York (1976).

[21] L.G. Valiant. The complexity of computing the permanent. Theoretical Computer Science. (to appear).

[22] L.G. Valiant. The complexity of enumeration and reliability problems. SIAM J. on Computing, (to appear).

[23] D.J.A. Welsh. Percolation and related topics. Science Progress, Oxford 64 (1977) 65-83.