Theoretical Computer Science 12 (1980) 303–314

NEGATION CAN BE EXPONENTIALLY POWERFUL

L.G. VALIANT

Computer Science Department, Edinburgh University, Edinburgh, United Kingdom

Communicated by M.S. Paterson
Received February 1979
Revised October 1979

1. Introduction

Among the most remarkable algorithms in algebra are Strassen's algorithm for the multiplication of matrices and the Fast Fourier Transform method for the convolution of vectors. For both of these problems the definition suggests an obvious algorithm that uses just the monotone operations + and ×. Schnorr [18] has shown that these algorithms, which use $\theta(n^3)$ and $\theta(n^2)$ operations respectively, are essentially optimal among algorithms that use only these monotone operations.

By using subtraction as an additional operation and exploiting cancellations of computed terms in a very intricate way Strassen showed that a faster algorithm requiring only $O(n^{2.81})$ operations is possible. The *FFT* method for convolution achieves $O(n \log n)$ complexity in a similar fashion.

The question arises as to whether we can expect even greater gains in computational efficiency by such judicious use of cancellations. In this paper we give a positive answer to this, by exhibiting a problem for which an exponential speedup can be attained using $\{+, -, \times\}$ rather than just $\{+, \times\}$ as operations. The problem in question is the multivariate polynomial associated with perfect matchings in planar graphs. For this a fast algorithm is implicit in the Pfaffian technique of Fisher and Kasteleyn [6, 8]. The main result we provide here is the exponential lower bound in the monotone case.

For discrete computations the question of the power of negation can be phrased in terms of the size of combinational circuits for computing Boolean functions. The problem is to determine whether there are functions that require a large circuit when only 'and' and 'or' gates are allowed, but can be computed efficiently when 'not' operators are available in addition. For computing sets of functions simultaneously, as in Boolean sorting, convolution and matrix multiplication, small polynomial speedups have been found [10, 12, 13, 15, 16, 17, 23]. Since no nonlinear lower bound is currently known for the monotone complexity of any single Boolean function an exponential 'negation versus complexity' tradeoff for circuits appears to be beyond current proof techniques. Among plausible vehicles for demonstrating

eventually such a tradeoff we would suggest the Boolean permanent (i.e. perfect matchings in bipartite graphs) as a likely candidate.

For our lower bound we use a global argument in the style of Shamir and Snir [19]. Previous such bounds were all for polynomials that were algebraicly complete, in the sense of [22]. Our task was to find an example that is easily computed with negations.

Although it is our purpose to demonstrate the restrictedness of the monotone model the reader should note the following relationships: for any unrestricted Boolean circuit over n arguments there exists a monotone circuit not much larger over $2n$ arguments that can compute the same function if the original arguments and their negations are provided as input [5]. In the algebraic case we shall observe that any polynomial computed easily over the reals with $\{+, -, \times\}$ is the difference of two polynomials computed easily with $\{+, \times\}$. Thus a single subtraction is enough to give an exponential gain.

From the last relation we can deduce that there are (families of) polynomials P, Q, R such that $Q = P + R$, where R and Q are polynomial time monotone computable, but P requires exponential monotone time. Thus a hard polynomial P can be annihilated by the addition of an easy one R. While this phenomenon is self-evident in Boolean algebra, where $1 = P + 1$ is axiomatic, it appears rather intriguing in the case of polynomials over the reals.

2. Preliminaries

Although monotone polynomial computations arise most naturally when coefficients are from the field of real or rational numbers, we shall give a more general treatment that applies to some other fields also. Adopting conventions from [2], we denote the ring of polynomials over indeterminates $\{x_1, \ldots, x_n\}$ with coefficients from the field F by $F[x_1, \ldots, x_n]$. A *program* over $F[x_1, \ldots, x_n]$ is a finite sequence of instructions of the form $a \leftarrow b \circ c$, where

(i) a is a variable,

(ii) $\circ$ is one of the two ring operations $+$ or $\times$, and

(iii) b and c can each be either a constant from F, an indeterminate, or a previously computed variable.

Each instruction computes a polynomial from $F[x_1, \ldots, x_n]$ in the natural way, and this we call its *value*. The value (or values) of a program is defined as that of a single (or set of) designated instruction(s). The *complexity* of a program is the number of instructions in it, and the complexity of a polynomial is the complexity of the shortest program for it.

A subset $H \subseteq F$ is *monotone* if the closure of H under $+$ and $\times$ does not contain 0. A program is *monotone* if the set of constants appearing in it together with unity form a monotone set. A polynomial is *monotone* if its nonzero coefficients form a monotone subset of F. For example, if F is the field of reals, then any polynomial with non-negative coefficients is monotone.

A *monomial* of polynomial P is a term of the form $\alpha x_1^{i_1} x_2^{i_2} x_3^{i_3} \cdots x_n^{i_n}$ where each i_j is a non-negative integer, and $\alpha \neq 0$ is the coefficient of $x_1^{i_1} \cdots x_n^{i_n}$ in P. The *degree* of the monomial is $\sum_{j=1}^{n} i_j$. A polynomial is *homogeneous* if all its monomials have the same degree. A program is homogeneous if all its instructions have homogeneous values.

Lemma 1. *If P is a homogeneous monotone polynomial over F, then any monotone program for it of minimal complexity is homogeneous.*

Proof. Suppose that the value of some instruction $a \leftarrow b \circ c$ in a monotone program contains X and Y as monomials of different degrees. Clearly if this instruction were replaced by $a \leftarrow y \times 1$, where y is a new indeterminate, then the polynomial computed would contain a monomial $y^i Z$ for some $i \geq 1$ and some Z independent of y, for otherwise the program would not be minimal. But then $\alpha X^i Z$ and $\beta Y^i Z$ must be monomials in P, which is a contradiction since they are of different degrees. The important point is that neither α nor β can be zero, for that would imply that zero is in the closure of the relevant monotone subset $H \subseteq F$. (To see this verify that the contrary would imply that there is non-null program that has value $0 \cdot X^i Z = 0$ and involves constants only from the subset H.)

The following is implicit in [21].

Lemma 2. *If P is homogeneous of degree d and there is a program of complexity C for it, then there is also a homogeneous program for it of complexity $C(d+1)^2$.*

Proof. Replace each variable a_i in the original program by $d+1$ variables $a_{i0}, \ldots, a_{id}$. The intention is that the value of any instruction $a_{ij} \leftarrow \cdots$ will be the sum of all the monomials of a_i that are of degree j. This can be achieved if each addition $a_i \leftarrow a_m + a_n$ is replaced by the $d+1$ instructions $\{a_{ij} \leftarrow a_{mj} + a_{nj} \mid 0 \leq j \leq d\}$ and each multiplication $a_i \leftarrow a_m \times a_n$ by the $(d+1)^2$ instructions entailed in computing

$$a_{ij} = \sum_{k=0}^{j} a_{mk} \times a_{n(j-k)} \quad \text{for } 0 \leq j \leq d.$$

The next fact occurs in [7] with a different proof, and indirectly in the monotone case in [19].

Lemma 3. *If P is a polynomial of degree d that is computed by a homogeneous program of complexity C, then*

$$P = \sum_{i=1}^{C} Q_i R_i$$

for some $Q_1, \ldots, Q_C$, $R_1, \ldots, R_C$ all of degree at most $\frac{2}{3}d$. Furthermore, if the program is monotone for $H \subseteq F$, then so are Q_i, R_i.

Proof. Some instruction $a \leftarrow b \times c$ in the program must have the property that its value Q_1 has degree in the range $[\frac{1}{3}d, \frac{2}{3}d]$. Let S_1 be the value computed by the program obtained by replacing the above instruction by $a \leftarrow 0+0$. Then $P = Q_1R_1 + S_1$ for some R_1. Furthermore, if the program was monotone for $H \subseteq F$, then Q_1, R_1 and S_1 must also be monotone for H. Since S_1 is computed by a program with at most $C-1$ nontrivial instructions, the desired result follows by induction on C.

Note. In Lemma 3 C can be replaced clearly by M the number of nonscalar multiplications. By a fanin argument it also follows that C can be replaced by $\frac{1}{2}A$, where A is the number of additions.

We next observe that Strassen's technique for doing reciprocals by ring operations alone also works for taking square roots.

Lemma 4. *Let F be a field of characteristic zero. Suppose $P, Q \in F[x_1, \ldots x_n]$ such that $P = Q^2$ and $\deg(Q) = d$. Then if P has complexity C, then Q has complexity $O(d^3 + d^2C)$.*

Proof. Suppose $P = k^2(1+R)$, where R has no constant term, and $k \in F$. Consider the binomial expansion

$$T = k(1 + \tfrac{1}{2}R - \tfrac{1}{8}R^2 + \tfrac{1}{16}R^3 - \tfrac{5}{128}R^4 + \cdots).$$

Now since F is a field $F[x_1, \ldots, x_n]$ is an integral domain and hence any quadratic equation such as $P = Q^2$ has at most two roots in $F[x_1, \ldots, x_n]$. Since T and $-T$ are both square roots of P, these are the only possible values of Q.

If Q has degree d, then all the contributions to monomials in Q are contained in the first $d+1$ terms of the power series for T. The sum of these terms can be computed in $C + 3d + 2$ steps. By Lemma 2 the first $d+1$ homogeneous components can therefore be computed in $(d+1)^2(C+3d+2)$ steps, and Q can be found in a further $d+1$ step.

The assumption $P = k^2(1+R)$ is certainly valid if P and Q have constant terms f, g respectively, for, then $f = g^2$. If they do not have constant terms, then we can find a translation of the variables to $y_i = x_i - \lambda_i$ such that $P(y_1, \ldots, y_n)$ does have a constant term. Q can then be computed fast in this ring and translated back.

Lemma 5. *If $P \in F[x_1, \ldots, x_n]$, where F is the field of reals or rationals and P has complexity C, then $P = Q - R$ for some monotone Q and R that can be computed simultaneously by a monotone program of complexity $6C$.*

Proof. For each variable a in the program for P there will correspond variables a^+ and a^- in the new program such that

$$\text{value}(a) = \text{value}(a^+) - \text{value}(a^-).$$

Each instruction $a \leftarrow b + c$ will be replaced by the pair

$$a^+ \leftarrow b^+ + c^+ \quad \text{and} \quad a^- \leftarrow b^- + c^-,$$

and each multiplication $a \leftarrow b \times c$ by the six instructions that assgn $b^+c^+ + b^-c^-$ to a^+ and $b^+c^- + b^-c^+$ to a^-. Indeterminates are treated as positive, and constants according to their sign.

3. Main results

Except where otherwise stated a *graph* will be undirected with no self-loops or multiple edges. Let G be a graph with a set V of i vertices and a set E of edges. A *matching* of G is a subset E' of E such that no two edges in E' are incident with the same vertex. The matching E' is *perfect* if its cardinality is $\frac{1}{2}i$, where i is even. Suppose now that the edge set E is $x_1, \ldots, x_r$ and that G has m perfect matchings $E_1, \ldots, E_m$. Then we define the *perfect matching polynomial* for G as

$$P_G = \sum_{i=1}^{m} \prod_{x_j \in E_i} x_j.$$

Let G_n be the *triangular grid* graph defined inductively as shown in Fig. 1. Then G_n has $N = \frac{1}{2}n(n+1)$ vertices and $\frac{3}{2}n(n-1)$ edges. Note that P_G is nonzero if and only if $n = 4j$ or $4j-1$ for some integer j. We shall from now on assume that n has one such non-trivial value, in which case P_G has degree $\frac{1}{4}n(n+1)$.

Our main result concerns this polynomial and is as follows:

Theorem 1. *There is a constant $c > 1$ such that for any field F any monotone program for computing P_G with $G = G_n$ has complexity at least c^n.*

To prove the result it is shown that each product P_iQ_i that generates monomials exclusively from P_G generates only an exponentially vanishing fraction of them

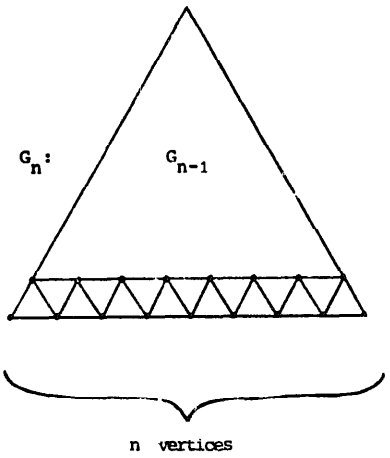


Fig. 1

provided the degrees of P_i and Q_i are at most two-thirds of that of P_G. An appeal to Lemma 3 then gives the answer. In order to establish this fact for P_iQ_i it is first shown that the variables corresponding to $\Omega(n)$ of the edges must be absent from P_iQ_i (Lemma 6). It is then proved that the matchings that consist only of the remaining edges are a vanishing fraction of the total. The special property of G_n required (Lemma 8) can be rephrased as follows: For some constant L if $n > L$, E' is a perfect matching of G_n and edge x is not in E', then there is an augmenting cycle with respect to E' that includes x and has at most L edges.

An analogue of the following lemma but for square grids is given in [4]:

Lemma 6. *Suppose that the N vertices of G_n are partitioned into two sets V_b and V_r where $|V_b| = kN$ and $0 < k < 1$. Then there is a function g depending only on k such that at least $g(k)n$ edges of G_n have one endpoint in V_b and the other in V_r.*

Proof. We fix the orientation of G_n as in Fig. 1, and call a set of vertices on the same horizontal level a *row*. Suppose V_b and V_r are coloured blue and red respectively. A row will be designated *full*, *empty* or *mixed* according to whether all, none, or some but not all of the vertices in it are coloured blue.

Let $g(k) = \frac{1}{2}\min\{1-\sqrt{1-k},\ 1-\sqrt{k}\}$. If there are at least $g(k)n$ mixed rows, then the result is established. What we now show is that if there are fewer mixed rows, then there must be at least $g(k)n$ empty rows and at least $g(k)n$ full rows. The result then follows since, if we turn the grid through 120°, then we would have at least $g(k)n$ mixed rows in this new orientation.

Therefore suppose that there were fewer than $g(k)n$ mixed rows and that the longest empty row is of length en and the longest full row of length fn. Then at least $n(1-g(k)-e)$ rows must be full and at least $n(1-g(k)-f)$ rows empty. Hence, there are at least $\frac{1}{2}n(n+1)(1-g(k)-e)^2$ blue vertices, and at least $\frac{1}{2}n(n+1)(1-g(k)-f)^2$ red ones. Since the numbers of blue and red vertices are in fact exactly $\frac{1}{2}kn(n+1)$ and $\frac{1}{2}(1-k)n(n+1)$ respectively, it follows that

$$e \geq 1-\sqrt{k}-g(k) \quad \text{and} \quad f \geq 1-\sqrt{1-k}-g(k)$$

and hence that $e, f \geq g(k)$, as desired.

The *distance* between two grid vertices is the number of edges in the shortest chain of edges between them. A *corner* of G_n is one of the three vertices of degree two. A *side* of G_n is one of the three chains of $n-2$ nodes of degree four linking a pair of corners.

Lemma 7. *For any integer t there is a constant $h > 0$ such that for all sufficiently large n the following holds: if $\{V_b, V_r\}$ is a partition of the vertices of G_n with $|V_b| \leq |V_r| \leq 2|V_b|$, then there is a set L of hn edges such that*

(i) *every edge in L has one endpoint in V_b and the other in V_r,*

(ii) *the distance between any two edges in L is at least $3t$, and*

(iii) *the distance from any edge in L to any vertex in a side of G_n is greater than t.*

Proof. Colour the vertices of G_n blue or red according to whether they belong to V_b or V_r. Remove all vertices within distance t of a side. In the remaining $G_{n'}$ where $n' = n - 3t - 3$ there will be exactly $\frac{1}{2}kn'(n'+1)$ blue vertices for some $k(\frac{1}{4} < k < \frac{3}{4})$, provided n is large enough. By Lemma 6 there will be $n'g(k)$ edges linking a blue vertex to a red one. Some fraction $f(t)$ of these can be selected so that no two of them are within distance $3t$ of each other. The result follows.

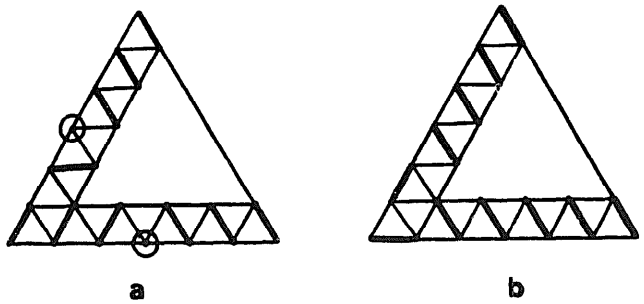
If $n = 3j+1$ for any integer j, then G_n has a vertex equidistant from the three corners, which we call the *centre*. Any edge incident to the centre is a *radius*. A *remnant of* G_n is any graph obtained by removing some subset V' of the vertices on the sides and corners of G_n and all edges incident to V'.

Lemma 8. *Suppose* $n = 12i+10$ *for some integer* $i \geq 1$, *and e is a radius of* G_n. *If a remnant G of* G_n *has some perfect matching, then it has a perfect matching containing the radius e.*

Proof. Define an s-remnant of G_n to be either G_n itself, or G_n with exactly two vertices removed, the two being on different sides of G_n and not within distance two of a corner. First we show that if $n = 12i+4$ and an s-remnant of G_n has a perfect matching then it has a perfect matching containing any desired radius. Fig. 2 illustrates how in the two cases the problem is reduced to that of finding a perfect matching in G_{n-4}. The omitted nodes are circled, and the matching shown in heavy lines.

In order that the centre of the grid be preserved, it is necessary to apply the reduction of Fig. 2(b) in the two other orientations also, thereby reducing the problem to G_{n-12}. In this manner the problem can be reduced to that of finding a perfect matching containing any specified radius in G_4. That G_4 does have this property can be verified easily by inspection.

It remains now to prove that the problem of finding a perfect matching in an arbitrary remnant of G_n, with $n = 12i+10$, can be reduced to that of finding a perfect matching in some s-remnant of G_{n-6}. To do this we show how, by starting at any two corners and proceeding towards each other along the common side and the row adjacent to it, we can cover by a matching all the vertices in these two rows of the remnant, and meet in the middle, with at most one vertex remaining uncovered, and this being in the inner of the two rows.



a b

Fig. 2

Suppose that we are constructing such a matching while proceeding from left to right. Fig. 3 shows the three possibilities that can occur at each step, where a circle indicates a vertex that is either absent from the remnant or already matched. In each case we add to the matching the edge shown by a heavy line, and proceed to the new case that arises on the right. (In the diagram the bottom line corresponds to the side of the grid.)

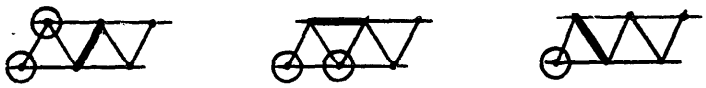


Fig. 3

When two such processes meet in the middle of a side we deal with each of the six possible cases as shown in Fig. 4.

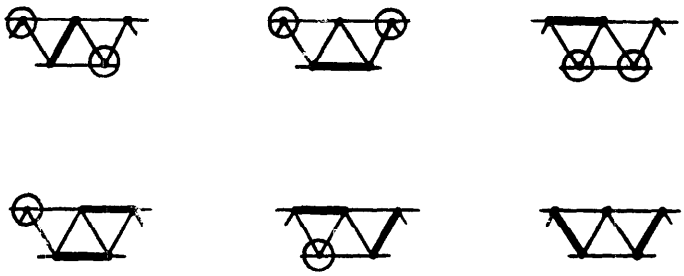


Fig. 4

It only remains to verify that we can start off such processes from a corner consistently along two sides. This can be done by inspecting the thirty distinct cases (modulo symmetry) that can arise at the corner of a remnant in which a perfect matching exists. Fig. 5 illustrates how each case is dealt with. In each diagram the corner is at the bottom left and circled vertices are those omitted from the remnants.

Corollary 9. *There is a constant $K > 1$ with the following property: if e is any radius of G_{22} and G'_{22} any remnant of it such that the number of perfect matchings in it is $x \geq 0$ altogether, and of these $y \geq 1$ do not contain e, then $x/y > K$.*

Proof. Immediate from Lemma 8.

Using those facts we can now deduce the main result:

Proof of Theorem 1. Suppose that the monotone complexity of P_G, where $G = G_n$ is C. Then by Lemma 3

$$P_G = \sum_{i=1}^{C} P_i Q_i,$$

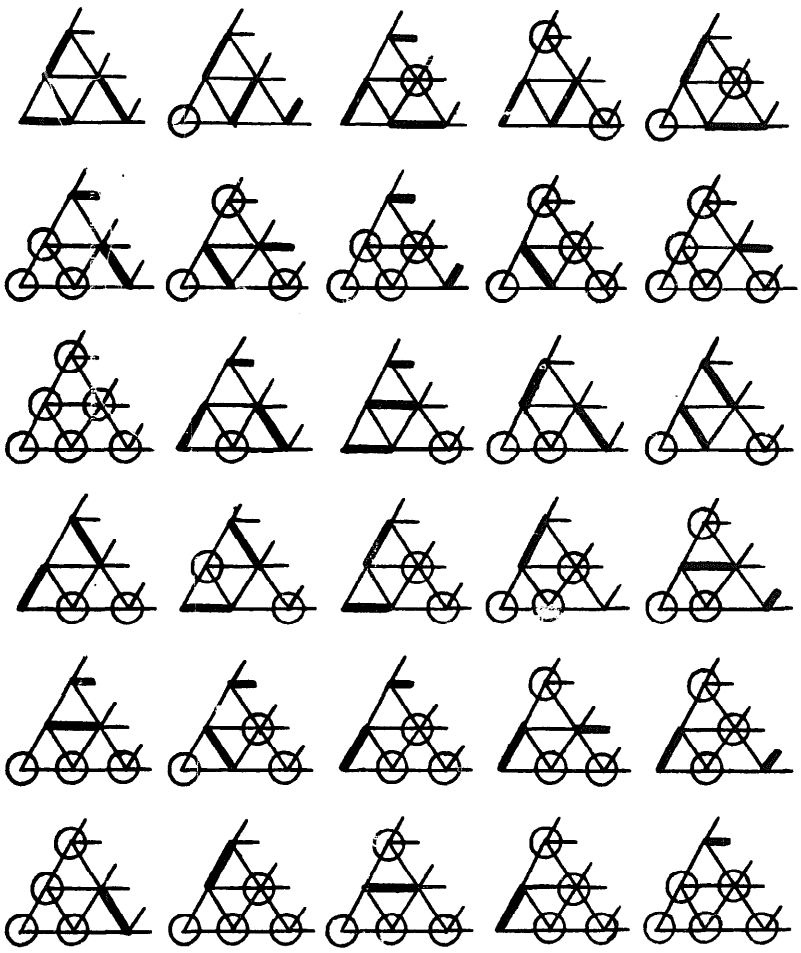


Fig. 5

where P_i, Q_i are monotone and of degree in the range $[\frac{1}{3}d, \frac{2}{3}d]$. Now for each i the variables in each monomial of P_iQ_i form a perfect matching in G_n. Monomials of other kinds cannot occur in P_iQ_i since, as in the proof of Lemma 1, there is no possibility of these being cancelled by the addition of other P_jQ_j terms.

Fix i and let V_b, V_r be sets of vertices that are covered by some matching in P_i and some matching in Q_i respectively. Then $\{V_b, V_r\}$ must be a partition of the vertices of G_n for otherwise P_iQ_i would contain undesirable monomials. For the same reason V_b, V_r must be the vertex sets of all the matchings in P_i, Q_i respectively. Applying Lemma 7 directly to this partition, and choosing $t = 22$ gives that there exists a set L of hn edges in G_n such that each one is at least distance 23 from a side and 66 from another, and each one has one endpoint restricted to P_i and the other restricted to Q_i. The last condition implies that no edge in L occurs in any matching in P_iQ_i.

Now for each $e_k \in L$ one can select from G_n a G_{22} subgraph that has e_k as a radius. By construction these subgraphs will be vertex-disjoint.

Let E_j be a matching in G_n that covers

(i) all the vertices of G_n outside the designated G_{22} subgraphs, and

(ii) possibly some of the vertices on the sides of some of these subgraphs, but
(iii) no vertex that is 'inside' one of them.

Suppose that $E_1, \ldots, E_u$ is the totality of all such matchings. Then the perfect matching in G_n corresponds one-to-one to expressions of the form

$$E_j \cup \bigcup_{k=1}^{hn} M_{kj}, \tag{1}$$

where $1 \leq j \leq u$, and each M_{kj} is a perfect matching in the remnant of the kth G_{22} subgraph induced by E_j, (i.e. the omitted nodes in the subgraph are just those contained in E_j.) On the other hand perfect matchings in P_iQ_i do not contain elements of L, and therefore can each be expressed by a union

$$E_j \cup \bigcup_{k=1}^{hn} M'_{kj}, \tag{2}$$

where $1 \leq j \leq n$ and M'_{kj} is some perfect matching not containing e_k of the remnant of the kth G_{22} subgraph induced by E_j.

Applying Corollary 9 to (1) and (2) we obtain that the ratio of the number of perfect matchings in G_n to those in P_iQ_i is at least K^{hn} for some $K > 1$. It follows that $C \geq K^{hn} = c^n$, where $c = K^h > 1$.

The contrasting positive result is the following:

Theorem 2. *There is a constant k such that for any field F of characteristic zero if $G = G_n$, then there is a program for P_G of complexity* $O(n^k)$.

Proof. Kasteleyn [9] showed that for any planar graph G with m nodes a skew-symmetric $m \times m$ matrix A_G exists (and can be found easily) such that

$$P_G = \text{Pfaffian}(A_G).$$

But for any skew-symmetric matrix, the Pfaffian is the positive square root of the determinant. Hence

$$P_G = \sqrt{\text{Det}(A_G)}.$$

Strassen has shown that a determinant over an infinite field can be computed in $O(m^{3.81})$ $\{+, \times\}$ operations [20, 21]. From Lemma 4 above it follows that P_G can be computed in $O(m^{5.81})$ $\{+, \times\}$ steps. But if $G = G_n$, then $m = O(n^2)$.

Corollary 10. *If F is the field of real or rational numbers, then for each positive integer i there exist monotone polynomials $Q_i, P_i, R_i \in F[x_1, \ldots, x_i]$ such that*

$$Q_i = P_i + R_i,$$

where $\{Q_i\}$ and $\{R_i\}$ are computable in monotone time polynomial in i, but $\{P_i\}$ requires time $\Omega(c^{\sqrt{i}})$ for some $c > 1$.

Proof. For values of i such that $i = j(4j+1)$ we let $P_i = P_G$, where $G = G_{4j}$. Q_i and R_i are derived in the manner of Lemma 5 from the fast algorithm for P_i implicit in Theorem 2. For other values of i we let $P_i = P_{i'}$ where $i' = \max\{i'' \mid i'' \leq i$ and $i'' = j(4j+1)$ for integral $j\}$.

In conclusion we note that our Lemma 8 is false for regular rectangular grids. The monotone complexity of these closely related polynomials is therefore unresolved.

4. Summary

For computing multivariate polynomials we have shown that cancellation of terms can be exploited in at least one instance to produce an exponential gain in efficiency as compared with monotone computations in which no cancellations are allowed. This result may be interpreted as confirming the potential power of sophisticated algebraic algorithms, and establishes the Fisher–Kasteleyn Pfaffian method for counting perfect matchings in planar graphs as a prime example. Whether negations help substantially in computing algebraicly complete problems such as the permanent remains a major open problem.

References

[1] C. Berge, *Graphs and Hypergraphs* (North-Holland, Amsterdam, 1973).
[2] A. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numeric Problems* (American Elsevier, New York, 1975).
[3] S. Chaiken and D.J. Kleitman, Matrix tree theorems, *J. Combinatorial Theory Ser. A* **24** (1978) 377–381.
[4] R.A. De Millo, S.C. Eisenstat and R.J. Lipton, The complexity of control structures and data structures, *Proc. 7th ACM Symposium on Theory of Computing* (1975) 186–193.
[5] M.J. Fischer, The complexity of negation-limited networks, MAC TM 65, MIT (1975).
[6] M.E. Fisher, Statistical mechanics of dimers on a plane lattice, *Phys. Rev.* **124** (1961) 1664–1672.
[7] L. Hyafil, On the parallel evaluation of multivariate polynomials, *Proc. 10th ACM Symposium on Theory of Computing* (1978) 193–195.
[8] P.W. Kasteleyn, The statistics of dimers on a lattice, *Physica* **27** (1961) 1209–1225.
[9] P.W. Kasteleyn, Graph theory and crystal physics, in: F. Harary, Ed., *Graph Theory and Theoretical Physics* (Academic Press, Reading, MA, 1967) 43–110.
[10] E.A. Lamagna and J.E. Savage, Combinational complexity of some monotone functions, *Proc. 15th IEEE Symposium on Switching and Automata Theory* (1974) 140–144.
[11] E.L. Lawler, *Combinatorial Optimization: Networks and Matroids.* (Holt, Rinehart and Winston, New York, 1976).
[12] K. Melhorn and Z. Galil, Monotone switching circuits and Boolean matrix product, *Computing* **16** (1976) 99–111.
[13] M.S. Paterson, Complexity of monotone networks for Boolean matrix product, *Theoret. Comput. Sci.* **1** (1975) 13–20.
[14] J.K. Percus, *Combinatorial Methods* (Springer, Berlin, 1971).
[15] N. Pippenger and L.G. Valiant, Shifting graphs and their applications, *J. ACM* (1976) 423–433.
[16] V.R. Pratt, The power of negative thinking in multiplying Boolean matrices, *SIAM J. Comput.* (1975) 326–330.

[17] J.E. Savage, *The Complexity of Computing* (Wiley, New York, 1976).
[18] C.P. Schnorr, A lower bound on the number of additions in monotone computations, *Theoret. Comput. Sci.* **2** (1976) 305–315.
[19] E. Shamir and M. Snir, Lower bounds on the number of multiplications and the number of additions in monotone computations, Research Report, IBM, Yorktown Heights, NY (1977).
[20] V. Strassen, Gaussian elimination is not optimal, *Numer. Math.* **13** (1969) 354–356.
[21] V. Strassen, Vermeidung von Divisionen, *J. Reine Angew. Math.* **264** (1973) 182–202.
[22] L.G. Valiant, Completeness classes in algebra, *Proc. 11th ACM Symposium on Theory of Computing*, Atlanta, GA (1979) 249–261.
[23] I. Wegener, Switching functions whose monotone complexity is nearly quadratic, *Proc. 10th ACM Symposium on Theory of Computing* (1978) 143–149.