Quantum Computers that can be Simulated Classically in Polynomial Time*

Leslie G. Valiant
Division of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138

ABSTRACT

A model of quantum computation based on unitary matrix operations was introduced by Feynman and Deutsch. It has been asked whether the power of this model exceeds that of classical Turing machines. We show here that a significant class of these quantum computations can be simulated classically in polynomial time. In particular we show that two-bit operations characterized by 4×4 matrices in which the sixteen entries obey a set of five polynomial relations can be composed according to certain rules to yield a class of circuits that can be simulated classically in polynomial time. This contrasts with the known universality of two-bit operations, and demonstrates that efficient quantum computation of restricted classes is reconcilable with the Polynomial Time Turing Hypothesis. In other words it is possible that quantum phenomena can be used in a scalable fashion to make computers but that they do not have superpolynomial speedups compared to Turing machines for any problem. The techniques introduced bring the quantum computational model within the realm of algebraic complexity theory. In a manner consistent will one view of quantum physics, the wave function is simulated deterministically, and randomization arises only in the course of making measurements. The results generalize the quantum model in that they do not require the matrices to be unitary. In a different direction these techniques also yield deterministic polynomial time algorithms for the decision and parity problems for certain classes of read-twice Boolean formulae. All our results are based on the use of gates that are defined in terms of their graph matching properties.

1. BACKGROUND

The now classical theory of computational complexity is based on the computational model proposed by Turing[30] augmented in two ways: On the one hand random operations are often allowed in addition to the originally proposed deterministic ones[22]. On the other, the number of computational steps is restricted to be at most polynomial in the input length, which allows such striking phenomena as NP-completeness to be expressed[10]. Taken together the resulting theory has provided strong circumstantial evidence for the robustness of the resulting model[16]. The following statement is a variant of how this robustness is sometimes expressed, and might be called the Polynomial Time Turing Hypothesis:

> *Any physical computing device can be simulated by a randomizing Turing machine that, as the input instances vary, takes a number of steps that grows as at most some fixed polynomial in the quantity $T + S + E$ where T, S and E are the time, space and energy used by the computing device.*

While no generally accepted counterexample is known, this hypothesis has, and can be expected to continue to, come under repeated challenge. The fundamental sources of challenge are the accepted laws of physics that are expressed in terms of functions that are not known to be polynomial time computable classically. For example the wave function for indistinguishable bosons can be expressed as a matrix permanent i.e. the determinant with all positive signs, (e.g.[26,29]). This function is known to be $\#P$-complete[31], or, in other words, as hard as counting the number of solutions of NP-complete problems and therefore at least as hard as the decision question for NP-complete problems. Such a situation poses a challenge in that it implies that among the following three statements at least one must be true: (a) the Polynomial Time Turing Hypothesis is false, (b) some reason excludes the possibility of a physical device evaluating appropriate instances of the hard problem that appears in the statement of the given physical law, or (c) the supposed hard problem can be computed classically in polynomial time by a yet undiscovered algorithm. Identifying the statement or statements in such a list that are true would clearly add to our knowledge and offers a valid scientific goal. This paper is concerned with a particular instance of (c).

The law of physics that has received most attention from this viewpoint, and which is the subject of this paper, is the formulation of quantum mechanics as unitary operations. Benioff[2] observed that classical Boolean operations could be formulated in these terms. Subsequently Feynman17] and Deutsch[12] suggested that unitary transformations in general offer a model of computation that may violate the

*This research was supported in part by grants NSF-CCR-95-04436, NSF-CCR-98-77049 and ARO-DAAL-03092-G-0115

STOC'01, July 6-8, 2001, Hersonissos, Crete, Greece.

Polynomial Time Turing Hypothesis and asked whether it did. Bernstein and Vazirani[3] showed how uniformity over all input sizes could be incorporated in the quantum computation model and hence that quantum polynomial time class BQP could be defined. While it is not generally believed that this class contains all of #P or even NP, Shor[27] has shown that two particular problems, namely integer factorization and discrete logarithms, which are not known to be polynomial time computable, do lie in BQP.

In the model of quantum computation that we consider a gate operating on k Boolean bits is represented as a $2^k \times 2^k$ unitary matrix. A unitary matrix A is any matrix with complex number entries such that $AA^{*t} = I$, where I is the identity matrix, A^* is the complex conjugate of A, and A^{*t} is the transpose of A^*. We note, however, that our constructions are all based on general matrix properties in which unitarity plays *no* essential part, and, in that sense, our treatment generalizes the quantum computational model.

Some basic results on the quantum computational model can be found in [12-15]. Most notably, it is known that there exist 2-bit gates which in conjunction with arbitrary 1-bit gates can approximate arbitrary unitary matrices. On the other hand, it is easy to see that circuits composed of one-bit gates alone can be simulated deterministically in polynomial time since it is possible to follow the evolution of each bit independently. A central component of Grover's algorithm is in this class [19]. A more interesting class where only a discrete set of gates is allowed but following the evolution of a small set of basis operators is sufficient, is attributed to Knill[18].

Our approach can be viewed as a branch of algebraic complexity theory in which computation gates are simulated by graph matching properties [7, 8, 28, 32]. In [31,32] this is done in the context of the matrix permanent. In this paper we use encodings by matrix properties that are known to be polynomial time computable, namely the Pfaffian function, the decision problem for matchings, and the parity problem for matchings, respectively.

2. PFAFFIANS

We describe some standard graph-theoretic notions and their relation to the Pfaffian of a matrix.

A weighted undirected graph, or simply a *graph*, G is a triple (V, E, W) where V is a set of *vertices* each represented by a distinct positive integer, E is a set of *edges* or unordered pairs (i, j) of the vertices $i, j \in V$, and W is the set of *weights*, each weight $w(i, j)$ corresponding to the edge $(i, j) \in E$. For example, $V = \{1, 2, 3\}$, $E = \{(1, 2), (2, 3), (1, 3)\}$, $w(1, 2) = w(2, 3) = w(1, 3) = 2$, is the complete graph on three vertices in which every edge has weight 2.

An $n \times n$ matrix B is *skew-symmetric* if for all i, j $(1 \leq i, j \leq n)$ $B(i, j) = -B(j, i)$. The *matrix of the graph* $G = (V, E, W)$ where $V = \{1, 2, \ldots, n\}$ is the $n \times n$ matrix $M(G)$ where the (i, j) entry $M(G)(i, j)$ is defined to equal:

(i) $w(i, j)$ if $i < j$,

(ii) $-w(i, j)$ if $i > j$, and

(iii) 0 otherwise.

In the more general case that $V = \{k_1, k_2, \ldots, k_n\}$ where $k_1 < k_2 < \ldots < k_n$, weight $w(k_i, k_j)$ replaces $w(i, j)$ in (i) and (ii) in this definition. For brevity we shall abbreviate $M(G)$ by G whenever it is clear that a matrix is intended.

The Pfaffian of an $n \times n$ skew-symmetric matrix B is defined to be zero if n is odd, one if $n = 0$, and if n is even with $n = 2k$ and $k > 0$ then it is defined as:

$$\mathrm{Pf}(B) = \sum_{\pi} \epsilon_\pi w(i_1, i_2) w(i_3, i_4) \ldots w(i_{2k-1}, i_{2k})$$

where

(i) $\pi = [i_1, i_2, i_3, \ldots, i_{2k}]$ is a permutation on $[1, 2, \ldots, n]$,

(ii) summation is over all such permutations π where further
$i_1 < i_2, i_3 < i_4, \ldots, i_{2k-1} < i_{2k}$, and
$i_1 < i_3 < i_5 < \ldots < i_{2k-1}$, and

(iii) $\epsilon_\pi \in \{-1, 1\}$ is the sign of the permutation, i.e., it is -1 or $+1$ according to whether the number of transpositions or swaps of pairs of distinct elements i_j, i_k, needed to reorder π to the identity permutation is odd or even. (An equivalent definition in this context is that it is the sign or parity of the number of overlapping pairs, where a pair of edges (i_{2r-1}, i_{2r}), (i_{2s-1}, i_{2s}) is *overlapping* iff $i_{2r-1} < i_{2s-1} < i_{2r} < i_{2s}$ or $i_{2s-1} < i_{2r-1} < i_{2s} < i_{2r}$. Note that it is implicit here that $i_{2r-1} < i_{2r}$ and $i_{2s-1} < i_{2s}$.)

A *matching* $E^* \subseteq E$ of G is a set of edges such that if (i, j), (r, s) are distinct edges in E^* then i, j, r, s are all distinct vertices. In a graph with an even number $2k$ of nodes a matching E^* is *perfect* if it contains k edges. (In other words every $i \in V$ is an endpoint of, or is *saturated* by, some edge in E^*.)

If B is the matrix of the graph G then each monomial in the Pfaffian corresponds to a distinct perfect matching in G. The monomial $w(i_1, i_2)\ w(i_3, i_4) \ldots w(i_{2k-1}, i_{2k})$ in Pf (G) corresponds to the perfect matching $\{(i_1, i_2), (i_3, i_4), \ldots, (i_{2k-1}, i_{2k})\}$ in G. This term will have coefficient ϵ_π that depends on the sign of this permutation of the indices in the manner indicated in the definition of the Pfaffian.

The following fact, due to Cayley [9] (see also [5] Theorem 9.5.2) relates the Pfaffian to the determinant.

Theorem 1. *For any skew-symmetric matrix* B

$$\mathrm{Det}\,(B) = (\mathrm{Pf}\,(B))^2.$$

It is known that for matrices with elements from any field the complexity of computing the determinant is to within a constant multiplicative factor the same as that of matrix multiplication[28]. Further for $n \times n$ matrices this cost is known to be upper bounded by $O(n^\alpha)$, where $\alpha < 2.38$ [11]. It follows that the square of the Pfaffian is polynomial time computable, and for complex matrices, so is the square of the modulus $|\mathrm{Pf}(B)|^2 = |(\mathrm{Pf}(B))^2|$. Efficient parallel algorithms for these functions also follow from similar results for the determinant [4].

For any $n \times n$ matrix B we call a set $A = \{i_1, i_2, \ldots, i_r\} \subseteq \{1, 2, \ldots, n\}$ an *index set*. Further we denote by $B[i_1, i_2, \ldots, i_r]$ or $B[A]$ the $(n - r) \times (n - r)$ matrix obtained by deleting the i_j^{th} row and column from B for all the r values of j $(1 \leq j \leq r)$.

3. PFAFFIAN SUMS

We start by defining the notion of a Pfaffian Sum, which expresses the Pfaffian summed over exponentially many minors of a matrix. We then go on to show that the Pfaffian Sum can be expressed in terms of the standard Pfaffian, and its square is therefore computable in polynomial time.

Definition. The *Pfaffian Sum* of an $n \times n$ skew-symmetric matrix B is a polynomial over indeterminates $\lambda_1 \ldots \lambda_n$ such that

$$\mathrm{PfS}(B) = \sum_A \left(\prod_{i \in A} \lambda_i \right) \mathrm{Pf}(B[A]).$$

Summation here is over the various principal minors obtained from B by deleting some subset A of the indices. In this paper we shall only need the instances in which each λ_i is fixed to be 0 or 1. The i for which $\lambda_i = 0$ can be thought of as the *unomittable* indices, and those with $\lambda_i = 1$ as the *omittable* indices. Then for this *(0,1)-case* the Pfaffian Sum is simply the sum of the $\mathrm{Pf}(B[A])$ over those A that contain only omittable indices.

We note that one can similarly define a *Determinant Sum*

$$\mathrm{DetS}(B) = \sum_A \left(\prod_{i \in A} \lambda_i \right) \mathrm{Det}\,(B[A]).$$

It is easy to verify that for any matrix B

$$\mathrm{DetS}(B) = \mathrm{Det}\,(B + \Delta)$$

where $\Delta(i,j) = 0$ if $i \neq j$, and $\Delta(i,i) = \lambda_i$ for every i. Thus a summed determinant is as easy to compute as a single determinant. We shall now show that the same holds for the Pfaffian. We define the $n \times n$ matrix $\Lambda^{(n)}$ as follows:

$$\Lambda^{(n)}(i,j) = \begin{cases} (-1)^{j-i+1}\lambda_i\lambda_j & \text{if} \quad i < j, \\ (-1)^{i-j}\lambda_i\lambda_j & \text{if} \quad i > j, \\ 0 & \text{if} \quad i = j. \end{cases}$$

Also for an $n \times n$ matrix B we define B^+ to be the $(n+1) \times (n+1)$ matrix of which the first n rows and columns equal B itself, and the $(n+1)^{st}$ row and column entries are all zero.

Pfaffian Sum Theorem. *For an $n \times n$ skew-symmetric matrix B*

$$\mathrm{PfS}(B) = \begin{cases} \mathrm{Pf}(B + \Lambda^{(n)}) & \text{if} \quad n \quad \text{is even,} \\ \mathrm{Pf}(B^+ + \Lambda^{(n+1)}) & \text{with} \lambda_{n+1} \text{ set to 1,} \quad \text{if } n \text{ is odd.} \end{cases}$$

Lieb [23] gave a proof for the instance of this relationship in which the λ_i are equal for all i, and n is even. One can derive our result from that instance. We shall, however, give a direct proof of this fact that makes the associated combinatorial structures more explicit.

Lemma 1. *If matrix B' is obtained from skew-symmetric matrix B by first interchanging the elements of rows i and j, and then interchanging the elements of columns i and j, then* $\mathrm{Pf}(B) = -\mathrm{Pf}(B')$.

Proof. The specified matrix operation clearly leaves the determinant function unchanged. By Theorem 1 it follows that the operation either leaves the Pfaffian invariant, or multiplies it by -1. Assume without loss of generality that $i < j$. Then each monomial in $\mathrm{Pf}(B)$ that contains $B(i,j)$ as a factor will have an identical monomial in $\mathrm{Pf}(B')$ except that the latter will contain $B'(i,j)$ where $B'(i,j) = -B'(j,i) = -B(i,j)$. Therefore at least some terms in $\mathrm{Pf}(B)$ and $\mathrm{Pf}(B')$ have opposite signs, and it follows that they all must have opposite signs. □

Lemma 2. *If skew-symmetric matrices B, B' are identical except that for some $i \in \{1, 2, \ldots, n\}$ for all j $B(i,j) = -B'(i,j)$ and $B(j,i) = -B'(j,i)$ then* $\mathrm{Pf}(B) = -\mathrm{Pf}(B')$.

Proof. The monomials of $\mathrm{Pf}(B)$, $\mathrm{Pf}(B')$ can be paired so that they are identical, except that they contain one factor $B(k,i)$ or $B'(k,i)$, respectively, or one factor of $B(i,k)$ or $B'(i,k)$, respectively, for some k. In either case the two factors will have opposite signs by assumption, and hence so will the monomials. □

Lemma 3. *Let $\Lambda^{(n)}$ be the $n \times n$ skew-symmetric matrix defined above with n even. Then*

$$\mathrm{Pf}(\Lambda^{(n)}) = \lambda_1 \ldots \lambda_n.$$

Proof. We prove this by induction on even values of n. Clearly $\mathrm{Pf}(\Lambda^{(2)}) = \lambda_1\lambda_2$. Now assume that the result holds for some even $n-2$. Consider the monomials in the expansion for $\Lambda^{(n)}$. In all the monomials that contain the element $\Lambda^{(n)}(n-1,n) = \lambda_{n-1}\lambda_n$ the complementary factors clearly sum to $\Lambda^{(n-2)}$, which by induction equals $\lambda_1 \ldots \lambda_{n-2}$. It is therefore sufficient to observe that all the other monomials in $\Lambda^{(n)}$ cancel in pairs: Any such other monomial contains a product $\Lambda^{(n)}\ (i,n-1)\Lambda^{(n)}(j,n)$ and has a companion monomial that is identical except that this product is replaced by $\Lambda^{(n)}(i,n)\Lambda^{(n)}(j,n-1)$, which is equal in value but gives rise to the opposite sign because of the extra transposition $(n-1,n)$. □

Proof of Theorem. First we assume that n is even. For some fixed $A \subseteq \{1, \ldots, n\}$ consider the monomial set N of $\mathrm{Pf}(B+\Lambda^{(n)})$ that involve λ elements in all rows and columns indexed by A, and involve B elements in the remaining rows and columns. We claim that the monomials in N sum to

$$\mathrm{Pf}(B[A])\mathrm{Pf}(\Lambda^{(A)}) \tag{1}$$

where $\Lambda^{(A)}$ is the $\Lambda^{|A|}$ matrix in which for each λ_i there has been substituted the indeterminate λ_j where j is the i^{th} smallest element of A. By applying Lemma 3 to the second term here and summing over all choices of A, the Theorem then follows.

To establish this claim we consider a sequence of σ transpositions or swaps on the indices (i.e., row and column numbers) of the original matrix that results in the indices A migrating to $n-|A|+1$, $n-|A|+2, \ldots, n$, and an accompanying process of sign changes to the elements described below. We claim that:

(a) The sequence of transpositions and sign changes to the elements causes all monomials in N to remain unchanged in sign in the Pfaffian.

(b) The matrix that is the outcome of the process has the B entries in the first $n-|A|$ rows/columns and these contribute a multiplicand of $\mathrm{Pf}(B([A])$ (with positive sign) to the sum of N.

(c) The matrix that is the outcome of the process has the λ entries in the last $|A|$ rows/columns and these contribute a multiplicand of $\mathrm{Pf}(\Lambda^{(A)})$ to the sum of N.

The conjunction of these three claims clearly implies that the quantity (1) above equals the original sum of N as needed. □

4. MATCHGATES

We shall simulate each quantum or other gate by what we call a matchgate.

A *matchgate* Γ is a quadruple (G, X, Y, T) where G is a graph (V, E, W), $X \subseteq V$ is a set of *input* vertices, $Y \subseteq V$ is a set of *output* vertices, and $T \subseteq V$ is a set of *omittable* vertices such that (i) X, Y and T are all disjoint, and (ii) $\forall i \in T$ if $j \in X$ then $j < i$ and if $j \in Y$ then $j > i$.

The matchings we consider will be those that saturate all the unomittable nodes, i.e. $V - T$, and also some, possibly empty, subset of T. When we define the Pfaffian Sum of a matchgate we shall perform the substitutions $\lambda_i = 1$ if $i \in T$, and $\lambda_i = 0$ otherwise.

We call $X \cup Y$ the *external* nodes. For $Z \subseteq X \cup Y$ we define the *character* $\chi(\Gamma, Z)$ of Γ with respect to Z to be the product

$$\mu(\Gamma, Z)\mathrm{PfS}(G')$$

where: (a) $G' = (V - Z, E', W')$ where further E' is the restriction of E to edges with both endpoints in $V - Z$, and W' is the corresponding restriction of W, and (b) the *modifier* $\mu(\Gamma, Z)$ is a multiplier of 1 or -1 that counts the parity of the number of overlaps between matching edges in E' and external edges. The external edges are the edges that link each matchgate to the rest of the circuit. We consider there to exist one external edge from each node in $X \cap Z$ and from each node in $Y \cap Z$. The other endpoint of each of the former is some node of lower index than any in V, and of each of the latter is some node of index higher than any in V. Figure 1 gives an illustrative example.

The character of a matchgate, therefore, takes into account overlaps between its internal edges and the external edges that link its external nodes to the rest of the circuit. The significance of condition (ii) in the definition of matchgates is that it guarantees that the modifier $\mu(\Gamma, Z)$ is always well defined: for any fixed Z the external edges that arise are uniquely defined, but it has to be guaranteed that the parity of the overlap of any one such external edge with *every* matching of E' that saturates all the unomittable nodes is the same. Condition (ii) achieves this by not allowing an omittable node in the gate to be numbered intermediate between the endpoints of an external edge. (That case might produce different overlap parity for the given external edge and the various internal matchings depending on whether the omittable node was in the matching.) To verify this note that if for $i \in X \cap Z$ there are r nodes $j < i$ where $j \in V - Z$, then the parity of the overlap of the external edge from i with the internal edges is the parity of r.

We define the *character* $\chi(\Gamma)$ of Γ to be the vector of $2^{|X \cup Y|}$ values of $\chi(\Gamma, Z)$ for the various $2^{|X \cup Y|}$ possible choices of Z. Often it is useful to think of a character as a $2^{|X|} \times 2^{|Y|}$ matrix where the rows represent the subsets of the inputs X, and the columns the subsets of the outputs Y. Matchgates with $|X| = |Y| = k$ can then be regarded as matrix transformations defined by the character matrix. For example $k = 1$ corresponds to one bit 2×2 matrix transformations and $k = 2$ corresponds to two bit 4×4 transformations. In all cases we need to specify a correspondence between subsets of X and the rows of the matrix, and another correspondence between subsets of Y and the columns of the matrix. We insist that there is the following consistency between the row and column orderings. We assume that there is a bijection f that maps the elements of X to elements of Y such that if the i^{th} row corresponds to subset X' of X and the i^{th} column to subset Y' of Y then Y' is the image of X' under f. The *character matrix* is therefore defined as a matrix of the elements $\chi(\Gamma, Z)$ with some such consistent row/column ordering.

The character of the matrix can be thought of as determining the nondeterministic steps allowed by a matchgate. If $\{0,1\}$ vector $\underline{x}$ is specified for the bits of $X \cap Z$ then for each possible $\{0,1\}$ vector $\underline{y}$ for $Y \cap Z$, the corresponding element in the character matrix gives the *value* to be associated with nondeterministic output $\underline{y}$ for input $\underline{x}$. The overall computations of the circuits composed of these matchgates are like counting Turing machines [31] or quantum computations [3,13] in that the values are multiplied over the steps of individual nondeterministic computation branches, and then added over all possible branches to obtain the numerical value of the computation.

First we shall show that all one-bit or 2×2 matrix transformations can be simulated by matchgates.

Proposition 1. *For any field F and any 2×2 matrix B of elements from F there exists a four node matchgate Γ with weights drawn from F such that $\chi(\Gamma) = B$.*

Proof. Consider the matchgate (G, X, Y, T) where G is the complete graph on nodes $V = \{1, 2, 3, 4\}$, $X = \{1\}$, $Y = \{4\}$, $T = \{2\}$, and W is as variously specified below.

We shall identify row 1 of the matrix with $\emptyset$, and row 2 with $\{1\}$. Similarly we identify column 1 with $\emptyset$ and column 2 with $\{4\}$. Let us abbreviate $B(1,1) = a$, $B(1,2) = b$, $B(2,1) = c$, and $B(2,2) = d$. Now it is immediate from the definitions that the modifier is always 1 since in this case external edges cannot overlap with internal ones. The character is simply the (0,1)-Pfaffian Sum for the gate with the external nodes corresponding to the variables set to 1 removed, and with T as the omittable nodes. In particular,
$\chi(\Gamma, \emptyset) = w(2,3)w(1,4) - w(1,3)w(2,4) + w(1,2)w(3,4)$,
$\chi(\Gamma, \{1\}) = w(3,4)$,
$\chi(\Gamma, \{4\}) = w(1,3)$, and
$\chi(\Gamma, \{1,4\}) = w(2,3)$.
Suppose first that at least one of b, c, or d is nonzero. Then by fixing $w(1,3) = b$, $w(3,4) = c$, $w(2,3) = d$, and setting the remaining weights so that $-bw(2,4) + cw(1,2) + dw(1,4) = a$, the four components of B can be set to the arbitrary values a, b, c, and d as desired.

In the special case that $b = c = d = 0$ we consider the same matchgate but make $T = \emptyset$. Then setting $w(2,3) = 0$ ensures that $b = c = d = 0$. The expression for $\chi(\Gamma, \emptyset)$ is unchanged and therefore by setting the remaining weights one can set $B(1,1)$ to the arbitrary value of a. □

To model two-bit or 4×4 matrices we consider two-bit matchgates, i.e., where $|X| = 2$, $|Y| = 2$. The characters of these can be viewed as above, with the rows representing subsets of X and the columns subsets of Y.

Proposition 2. *For any field F and any 4×4 matrix B of elements from F where all the off diagonal elements are zero and $B(1,1)B(4,4) = B(2,2)B(3,3)$ there exists a matchgate Γ with weights drawn from F such that $\chi(\Gamma) = B$.*

Proof. Consider $V = \{1,2,3,4,5,6\}$, $X = \{1,2\}$, $Y = \{5,6\}$, and $T = \emptyset$. The rows of the matrices have ordering $\emptyset$, $\{1\}$, $\{2\}$, $\{1,2\}$, and the columns have ordering $\emptyset$, $\{6\}$, $\{5\}$, $\{5,6\}$.

Suppose first that $B(4,4) \neq 0$. Then let the only nonzero weights be: $w(1,6) = B(3,3)/B(4,4)$, $w(2,5) = B(2,2)/B(4,4)$, and $w(3,4) = B(4,4)$. It can be verified that this matchgate has the required character.

If $B(4,4) = 0$ then we have various special cases: (a) $w(1,6)$, $w(2,3)$, $w(4,5) \neq 0$ solves the case $B(1,1) \neq 0$, $B(2,2) \neq 0$, $B(3,3) = 0$, $B(4,4) = 0$. (b) $w(2,5)$, $w(1,3)$, $w(4,6) \neq 0$ solves the case $B(1,1) \neq 0$, $B(2,2) = 0$, $B(3,3) \neq 0$, $B(4,4) = 0$. (c) $w(2,3)$, $w(4,5) \neq 0$ solves the case $B(1,1) = 0$, $B(2,2) \neq 0$, $B(3,3) = 0$, $B(4,4) = 0$. (d) $w(1,3)$, $w(4,6) \neq 0$ solves the case $B(1,1) = 0$, $B(2,2) = 0$, $B(3,3) \neq 0$, $B(4,4) = 0$. To solve the remaining case of $B(1,1) \neq 0$, $B(2,2) = 0$, $B(3,3) = 0$, $B(4,4) = 0$ we can consider the eight node matchgate with $X = \{1,2\}$, $Y = \{7,8\}$ and nonzero edges $w(1,3)$, $w(2,4)$, $w(5,7)$ and $w(6,8)$. □

We now turn to more general 4×4 matrices:

Proposition 3. *For any field F and any 4×4 matrix B of elements from F satisfying the nine equations $B(1,2) = B(1,3) = B(2,1) = B(2,4) = B(3,1) = B(3,4) = B(4,2) = B(4,3) = 0$, and $B(1,1)B(4,4) - B(2,2)B(3,3) = B(1,4)B(4,1) - B(2,3)B(3,2)$, there exists a matchgate Γ with weights drawn from F such that $\chi(\Gamma) = B$, provided $B(4,4) \neq 0$.*

Proof. Consider $V = \{1,2,3,4,5,6\}$, $X = \{1,2\}$, $Y = \{5,6\}$, $T = \emptyset$, and row/column ordering as in Proposition 2. We let $t = 1/B(4,4)$ and consider the following set of nonzero weights: $w(1,6) = tB(3,3)$, $w(2,5) = tB(2,2)$, $w(1,2) = tB(1,4)$, $w(5,6) = tB(4,1)$, $w(1,5) = -tB(3,2)$, $w(2,6) = -tB(2,3)$, $w(3,4) = B(4,4)$. □

Now consider the following set of what we call the *matchgate identities* for 4×4 character matrices:

$$B(1,1)B(4,4)-B(2,2)B(3,3)-B(1,4)B(4,1)+B(2,3)B(3,2)=0$$
$$B(2,1)B(4,4)-B(2,2)B(4,3)-B(4,1)B(2,4)+B(2,3)B(4,2)=0$$
$$B(3,1)B(4,4)+B(3,3)B(4,2)-B(4,1)B(3,4)-B(3,2)B(4,3)=0$$
$$B(1,3)B(4,4)+B(3,3)B(2,4)-B(1,4)B(4,3)-B(2,3)B(3,4)=0$$
$$B(1,2)B(4,4)-B(2,2)B(3,4)-B(1,4)B(4,2)+B(3,2)B(2,4)=0$$

Proposition 4. *For any field F and any set of values of the 11 entries of the 4×4 matrix B other than $R = \{B(1,1), B(1,2), B(1,3), B(2,1), B(3,1)\}$ such that $B(4,4) \neq 0$, there exists a matchgate Γ with weights drawn from F such that $\chi(\Gamma)$ equals B in the eleven entries other than R, and in each of the R entries $\chi(\Gamma)$ equals various polynomials in terms of these other 11 entries and $1/B(4,4)$. In particular, $B = \chi(\Gamma)$ satisfies the five matchgate identities.*

Proof. Consider the matchgate of Proposition 3 but modify it by making $T = \{4\}$ and adding four more nonzero edge weights: $w(1,3) = -B(3,4)$, $w(2,3) = B(2,4)$, $w(3,5) = B(4,2)$, $w(3,6) = -B(4,3)$. This is illustrated in Figure 2.

By inspection, for any $Z \subseteq X \cup Y$ one can write down $\chi(\Gamma, Z)$: If $|\ Z\ |$ is even then all the matchings that contribute to $\chi(\Gamma, Z)$ saturate the omittable node 4, and do so necessarily with edge $\{3,4\}$. Hence this case reverts back to Proposition 3 since the newly added edges are all incident to node 3, and can, therefore, play no part. Hence in the corresponding eight positions in B we can regard seven as being arbitrary, and the eighth, namely $B(1,1)$ as being constrained by the first matchgate relation in terms of the other seven exactly, as in Proposition 3.

We now consider the eight entries of $\chi(\Gamma, Z)$ where $|Z|$ is odd. By inspecting Figure 1 and recalling the definition of $\chi(\Gamma, Z)$ we obtain immediately that:

$$\chi(\Gamma, \{2,5,6\}) = B(3,4)$$
$$\chi(\Gamma, \{1,5,6\}) = B(2,4)$$
$$\chi(\Gamma, \{1,2,6\}) = B(4,2)$$
$$\chi(\Gamma, \{1,2,5\}) = B(4,3)$$
$$\chi(\Gamma, \{1\}) = tB(2,2)B(4,3)+tB(4,1)B(2,4)-tB(2,3)B(4,2)$$
$$\chi(\Gamma, \{2\}) = -tB(3,3)B(4,2)+tB(4,1)B(3,4)+tB(3,2)B(4,3)$$
$$\chi(\Gamma, \{5\}) = -tB(3,3)B(2,4)+tB(1,4)B(4,3)+tB(2,3)B(3,4)$$
$$\chi(\Gamma, \{6\}) = tB(2,2)B(3,4)+tB(1,4)B(4,2)-tB(3,2)B(2,4)$$

The first equality states simply that the arbitrary chosen value $B(3,4)$ is in fact the value of the character $\chi(\Gamma, \{2,5,6\})$ for the given matchgate, exactly as desired. The same holds for $B(2,4), B(4,2)$ and $B(4,3)$. Hence in addition to the seven entries in the character that we said could be fixed arbitrarily, here are a further four.

The last four statements describe the remaining four entries of the character in terms of the eleven arbitrarily fixed, and of $B(4,4)$, which, as we saw, is constrained by the first matchgate identity. These four statements assert nothing other than that: if $B(2,1), B(3,1), B(1,3)$ and $B(1,2)$ do equal the corresponding character entries of the specified matchgate, then these quantities obey the last four matchgate identities, respectively. □

A related question is to characterize the subclass of the feasible character matrices that are also unitary. For matrices in which the only nonzero entries are those appearing in the first matchgate identity this is easy. We note that in that case restricting the matrix to rows/columns $\{1,4\}$ imposes a unitary constraint on the four remaining elements, as does the restriction to rows/columns $\{2,3\}$. But unitary 2×2 matrices can be written as [5]:

$$e^{i\alpha} \begin{bmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{bmatrix} \begin{bmatrix} \cos\gamma/2 & -\sin\gamma/2 \\ \sin\gamma/2 & \cos\gamma/2 \end{bmatrix} \begin{bmatrix} e^{-i\delta/2} & 0 \\ 0 & e^{i\delta/2} \end{bmatrix}$$

where $\alpha, \beta, \gamma, \delta$ are real valued. In other words the 2×2 matrix with elements $B(1,1), B(4,1), B(1,4), B(4,4)$ can be written in this form with appropriate $\alpha_1, \beta_2, \gamma_1, \delta_1$, as can the 2×2 with elements $B(2,2), B(2,3), B(3,2), B(3,3)$ for appropriate $\alpha_2, \beta_2, \gamma_2, \delta_2$. But the first matchgate identity is equivalent to saying that the difference of the determinants of these two 2×2 matrices is zero. However, the determinant of the above product expression for 2×2 unitary matrices is $e^{i\alpha}$. It follows that, except for the restriction that $e^{i\alpha_1} - e^{i\alpha_2} = 0$, and B(4,4) $\neq 0$, the space of matchgate character matrices is spanned as the values of $\alpha_1, \beta_1, \gamma_1, \delta_1, \alpha_2, \beta_2, \gamma_2$ and δ_2 range over the reals..

Theorem 2 *Given any matchgate Γ there exists another*

matchgate Γ' that has the same character as Γ, and has an even number of nodes, at most one of which is omittable.

The following is proved in [33]:

Theorem 3 *Suppose that for a matchgate Γ with inputs $X = \{i, j\}$ and outputs $\{k, l\}$ with $i < j < k < l$, B is the character matrix for row ordering $\phi, \{i\}, \{j\}, \{i, j\}$ and column ordering $\phi, \{l\}, \{k\}, \{k, l\}$. Then B obeys the five matchgate identities.*

5. MATCHCIRCUITS

We now discuss how matchgates can be combined in large numbers to form circuits. In the first instance the global properties of circuits that we seek are of the same nature as those of individual gates: we define input and output nodes for a circuit and wish to establish the character of the circuit from the characters of the constituents and the particulars of how the gates are connected together. These latter particulars need special attention if the characters of the constituents are to be composed so that no untoward sign effects arise. The only difference between a matchgate and a matchcircuit is that modifiers are assumed to be $+1$ for the latter since we do not consider a circuit to be externally connected.

One important purpose of combining matchgates is to derive matchgates with new characters. A most basic operation is that of putting two 2-bit gates in sequence, as illustrated in Figure 3.

It should be clear that if gate G is formed from constituent gate G_1 appended by gate G_2 as shown, and if the characters of G1 and G_2 are B_1 and B_2, respectively, then the character of G will be $B = B_1 B_2$ where matrix multiplication is implied and provided the row and column orderings of B_1, B_2, and B are appropriate. To see this consider the entry $B(2, 3)$ which corresponds to input $\{1\}$ and output $\{5\}$. In the matrix product $B = B_1 B_2$ the corresponding value is

$$B_1(2,1)B_2(1,3) + B_1(2,2)B_2(2,3) + B_1(2,3)B_2(3,3) + B_1(2,4)B_2(4,3).$$

It is easy to verify that the four terms correspond, respectively, to sets of matchings in G that (i) contain neither edge $e1$ nor edge $e2$, (ii) have $e1$ but not $e2$, (iii) have $e2$ but not $e1$, and (iv) have both $e1$ and $e2$. In the Pfaffian expression for G these terms all have positive signs since the potential overlap between $e1$ and $e2$ is always even, and their overlap with the internal edges of G_1 and G_2 are already accounted for in the characters of G_1 and G_2.

More generally let $\Gamma_1, \ldots, \Gamma_m$ be a set of matchgates $\Gamma_i = (G_i, X_i, Y_i, T_i)$ where $G_i = (V_i, E_i, W_i)$ and $|X_i| = |Y_i|$. A *matchcircuit* $\Gamma = (G, X, Y, T)$ with $G = (V, E, W)$, is a composition of $\Gamma_1, \ldots, \Gamma_m$ that can be obtained in the following way: (1) The Γ_i are first reordered as necessary. (2) Each external node of Γ_i has one external edge. An output node of Γ_i can be linked to an input node of Γ_j if $i < j$, via a *linking chain* of an odd number of edges and in that case that output node and that input node are considered as members of a set D. (3) Nodes in X_i or Y_i that are not in D are the endpoints of linking chains of odd length, the other endpoints of which are considered as the members of X or Y, the input and output nodes of the circuit, respectively. An odd length chain between a node in X and a node in Y represents a bit on which the circuit does not act. (4) T is the union of the T_i. (5) The node set is now renumbered as necessary so that for each i all the nodes in V_i are *contiguous* in V (i.e., if j, k are nodes originating from V_i, and $j < l < k$ for some node l in V, then l originates from a node in V_i and is not a node on a linking chain or from a different V_i.)

Figure 4 shows an example of a matchcircuit with $|X| = |Y| = 6$ and with three matchgates.

Clearly, in general, if $|X_i| = |Y_i|$ for each i then $|X| = |Y|$. Hence in that case just as the characters of the gates can be represented as square matrices, so can the character of the matchcircuit. Also if each constituent matchgate has a unitary character then so has the matchcircuit. To see this we note that we can regard any matchgate as acting on all $n = |X| = |Y|$ bits and regard the individual gates then as composed in sequence. Figure 3, for example, illustrates three two-bit gates, each viewed as an n-bit gate composed in sequence. If a gate has a unitary character over two bits, it has a unitary character also over the n bits, and any composition of such gates also has a unitary character since unitary matrices are closed under matrix product.

Matchcircuit Theorem. *If matchcircuit Γ is a composition of matchgates $\Gamma_1, \ldots, \Gamma_m$ then*

$$\mathrm{PfS}(G) = \sum_S \epsilon_S \prod_{1 \le i \le m} \mathrm{PfS}(G_i - S_i),$$

where (1) S is a mapping that determines for each node in D whether it is to be matched by an edge internal or external to the matchgate from which it originated, (2) S_i is the set of external nodes in Γ_i that are assigned by S to be matched by an external edge, and (3) $\epsilon_S \in \{-1, 1\}$ (and depends on the choice of S and on the nature of the composition Γ.)

Proof. The different choices of the assignment S partition the matchings of G. In order to establish the theorem it suffices to show that the matchings of G that respect S contribute to $\mathrm{PfS}(G)$ a term

$$\prod_{1 \le i \le m} \mathrm{PfS}(G_i - S_i) \tag{2}$$

up to a multiplicative factor of 1 or -1.

It is clear that for any sets $U_1 \subseteq T_1$, $U_2 \subseteq T_2, \ldots, U_m \subseteq T_m$, and $U = U_1 \cup U_2 \cup \ldots \cup U_m$ each matching in $\mathrm{PfS}(G)$ that omits U corresponds one to one to the union of matchings from $\Gamma_1, \ldots, \Gamma_m$ that omit $U_1, \ldots, U_m$, respectively. Hence the monomials in $\mathrm{PfS}(G)$ that are consistent with S are identical to the monomials in (2) above. It remains therefore only to establish that either all the corresponding pairs of terms have equal signs (and hence $\epsilon_S = 1$) or they all have unequal signs (and hence $\epsilon_S = -1$).

For all the matchings in G that are consistent with a single S the external edges (or other chain edges) matched are identical. Hence the matchings for the fixed S differ only as far as the edges that are entirely internal to the various Γ_i separately. Within any one such Γ_i the signs of the various terms in $\mathrm{PfS}(G_i - S_i)$ are whatever they are. When they form subterms of $\mathrm{PfS}(G)$ their signs are not affected relative to each other. This is because for any two edges from distinct gates, say edge (u, v) of E_i and (w, x) of E_j, either $u < v < w < x$ or $w < x < u < v$ and therefore there is no overlap in either case. It follows that the contributions from the various $\mathrm{PfS}(G_i - S_i)$ simply multiply together.

It remains to observe that the external and other linking edges produce no overlap. This is simply because the overlap between the external edges and those within gates is, as previously observed, fixed by virtue of condition (ii) in the definition of matchgates. The overlap among the external edges and linking chain edges is clearly fixed also since there is just one matching being considered for these. It remains to observe that the fixed matching in the linking edges that are not external (i.e., not directly incident to a gate) have no overlap with internal gate edges by virtue of condition (5) in the definition of matchcircuits that imposes an adequate constraint on their numbering. □

We shall think of a matchcircuit as acting on a sequence of bits $x_1, \ldots, x_n$, and of a k-bit matchgate as acting on a subset of these n bits. We say that a matchcircuit is in *standard description* if when the character of a gate is described to be B it is the case that any one diagonal element of B refers to exactly the same subset of $\{x_1, \ldots, x_n\}$ for the inputs as for the outputs. Thus if, for example, $B(2,2)$ refers to an input subset of $\{1\}$ and an output subset of $\{6\}$, as in Proposition 2, then a standard description would insist that output 6 of this gate be the same x_i bit as input 1 of this gate. Clearly, a circuit not in standard description can be put into standard description at the expense only of reordering the outputs (and renaming the bits operated on inside the circuit.)

Main Theorem. *For any matchcircuit composed of 2-bit matchgates any entry in its character can be expressed as a Pfaffian of the circuit with the corresponding external nodes removed, if the circuit when described in standard form is composed of (a) matchgates with any character B such that $B(i,j) = 0$ for all i, j such that $i \neq j$, (b) matchgates operating on any two bits with indices differing by one (e.g., x_i, x_{i+1}) that have character B in which the only nonzero entries are among $B(1,1)$, $B(2,2)$, $B(3,3)$, $B(4,4)$, $B(1,4)$, $B(4,1)$, $B(2,3)$, $B(3,2)$, and (c) any matchgate operating on bits x_1 and x_2.* □ We note that if we take into account the matchgate identities then case (a) can be simulated by case (b) and is technically redundant in the above Theorem. To see this note that a case (a) matrix is a diagonal matrix with some (a,b,c,bc/a) on the diagonal. This can be simulated by two diagonal 1-bit gates, the first with diagonal (a,b) acting on the first bit, and the second with diagonal $(1, c/a)$ acting on the second bit. Each of these 1-bit gates can then be viewed as a 2-bit gate that ignores the other bit, and is feasible for a matchgate. Hence a case (a) 2-qubit gate acting on bits i, j can be replaced by two 1-bit gates acting on bits $i, i+1$ and $j, j+1$ respectively.

We call Ω the class of matchcircuits defined in the above Theorem and limited to the classes of matchgates that can be constructed. In particular the gates that can be used in unrestricted ways include those of Proposition 2. Those that can be applied to neighboring bits include those of Proposition 3. Those that can be applied to the end bits x_1, x_2 include those of Proposition 4.

Now consider a member M of this class Ω. For any input variable x_i or output variable y_j that is to be fixed to have value 0 we delete this input/output node and the edges incident to it. This forces the circuit to have value 0 for this variable since the corresponding $X \cup Y$ node will not be saturated. If we wish to set the variable to 1 we retain the corresponding nodes and edges. (This $\{0,1\}$ interpretation of the variable values is the complement of that considered for the character, but this is an inessential technicality.) For $\underline{x} \in \{0,1\}^n$, $\underline{y} \in \{0,1\}^n$ we define $M_{\underline{x},\underline{y}}$ to be the matchcircuit M with these modifications. In the case that the matchgates are unitary, we can regard the circuit as a quantum matchcircuit.

Corollary 1. *For a quantum matchcircuit $M \in \Omega$ started in state $\underline{x}$ the probability that it terminates in state $\underline{y}$ is $|\mathrm{PfS}(M_{\underline{x},\underline{y}})|^2$.*□

A critical aspect of our theory, which we have not mentioned and which goes beyond the standard quantum computational model, is that it is capable of simulating in polynomial time *nondeterministic summation* over the *inputs* of the simulated circuit evaluations in addition to the nondeterministic branches. Thus if $\underline{x}^*, \underline{y}^*$ assign subsets of the inputs and outputs fixed values, we first excise the external nodes that are so fixed to have value 0 and retain those fixed as 1. We then apply to the resulting matchcircuit M the various transformations described in the following:

Theorem 4. *For any matchcircuit $M \in \Omega$, if x^*, y^* are partial assignments to the inputs and outputs then each of the three quantities*

$$(\sum_{\underline{x},\underline{y}} \mathrm{PfS}(M_{\underline{x},\underline{y}}))^2, \quad \sum_{\underline{x},\underline{y}} \mid \mathrm{PfS}(M_{\underline{x},\underline{y}}) \mid^2, \quad \textit{and} \quad \sum_{\underline{x},\underline{y}} (\mathrm{PfS}(M_{\underline{x},\underline{y}}))^2,$$

where summation is over all $\underline{x}$ and $\underline{y}$ that agree with $\underline{x}^*$ and $\underline{y}^*$, respectively, can be evaluated in polynomial time. □

Proof. For the first quantity we define the nondeterministic input and output nodes in M as omittable nodes in the matchcircuit and appeal to the Pfaffian Sum Theorem.

For the second quantity we construct for M its complex conjugate matchcircuit M^* which is identical to M except that corresponding weights are complex conjugates of each other, and the node indices are from disjoint intervals and in reverse order. For each nondeterministic input or output node we join the corresponding nodes of M and M^* by an edge of weight one. We claim that the Pfaffian of the result is the desired quantity. This follows from the observation that each subset of the joining edges that is matched represents a distinct truth assignment to the nondeterministic nodes, and its contribution to the Pfaffian will be the product of some $\mathrm{PfS}(M_{\underline{x},\underline{y}})$ and its complex conjugate.

The third quantity is computed in the same way as the second except that instead of M^* we simply use a second copy of M. □

This result, clearly, has applications broader than the quantum computation model, since it is not restricted to unitary matrices. We note, however, that if we do restrict ourselves to the quantum model, or any other model with a similar probabilistic interpretation, then results can be derived that are strongly suggestive of a computational interpretation of quantum physics. In particular, the last part of the above Theorem can be used to generate in polynomial time, an output state according to the probability distribution that the quantum mechanical interpretation of a quantum matchcircuit specifies. This is precisely equivalent to the idea of performing a measurement in quantum mechanics.

Measurement Theorem. *For any quantum matchcircuit $M \in \Omega$ if $\underline{x}$ is a total assignment to the inputs, and $\underline{y}^*$ a partial assignment to the outputs then there is a randomized polynomial time procedure that generates an output $\underline{y}$ consistent with $\underline{y}^*$ with probability*

$$|\mathrm{PfS}(M_{\underline{x},\underline{y}})|^2 / \sum |\mathrm{PfS}(M_{\underline{x},\underline{z}})|^2$$

where summation in the denominator is over all $\underline{z}$ consistent with $\underline{y}^*$.

Proof. We proceed in the manner described in (19) for generating a random combinatorial structure when we can count their number. The procedure has m stages if there are m nondeterministic variables. In each stage we consider one such variable y_i and fix its value as follows: We evaluate for $y_i = 0$, and separately for $y_i = 1$, the quantity

$$\sum |\mathrm{PfS}(M_{\underline{x},\underline{y}})|^2$$

where the summation is over the $\underline{y}$ variables that have not been fixed yet. Each of the two evaluations is a call of the procedure in the second part of Theorem 5. If we obtain the values p_0 and p_1 for these two calls then with probability $p_0/(p_0+p_1)$ we fix $y_i = 0$ and with probability $p_1/(p_0+p_1)$ we fix $y_i = 1$. It is clear that the procedure has the desired result. □

In conclusion we note that we can generalize the Measurement Theorem to sum nondeterministically over unspecified inputs also. There is an obvious restriction, however, on how the various nondeterministic choices can be correlated. If they are all independent then nothing special needs to be done. Otherwise one option is to relate pairs of them by joining such a pair by an edge or a chain of two edges, and this corresponds to *read-twice circuits*. Controlling the overlaps of the added edges becomes problematic in general, however. The next section takes this route under circumstances when these overlaps do not matter.

6. MATCHNETS

We shall compose matchgates now in a different way. For quantum computation we needed to be careful in constructing matchcircuits so that the effects of the constituent match gates simply added, without any uncontrollable sign effects. Thus our simulation of quantum computation was limited by (a) what individual matchgates could achieve, and by (b) the constrained composition rules that were adopted to maintain control of the signs. In the two applications in this section we shall be limited by (a) alone. In the first application we shall be detecting the *existence* of solutions, and in the second the *parity* of the number of solutions.

We shall consider the following classification of Boolean functions:

$B01 = \{1, 0\}$
$B11 = \{x\}$
$B21 = \{xy, x+y, x \oplus y = 1, x \oplus y = 0\}$
$B31 \cup B32 \cup B33$ where
$B31 = \{xyz, x(y=z), x'yz + xy'z + xyz', x(y+z),$
$\quad x \oplus y \oplus z = 0, x \oplus y \oplus z = 1, xy + yz + zx,$
$\quad x + (y=z), xy + (y=z)\},$
$B32 = \{x+y+z, \neg(xyz + x'y'z')\}$, and
$B33 = \{xyz + x'y'z', xyz + x'y', xy + y'z,$
$\quad x + yz, z = x+y,\ x'y'z' + xy + yz + zx\}.$

A matchnet is a set of matchgates with no output nodes where pairs of input nodes from different matchgates may be joined (a) by a single edge if the input nodes are to have the same value, or (b) by a chain of two edges if the input nodes are to have opposite values. We can then define the matchnet of a read-twice Boolean formula as follows: if the formula is a conjunction of a number of three-argument Boolean expressions then the matchnet is simply the union of matchgates, one for each such expression, in addition to single edges or chains of length two to constrain pairs of variables in the expressions to have equal or unequal values.

More formally a *matchnet* is a matchcircuit having no input or output nodes, and composed of matchgates with no outputs, where condition (1) in the definition of matchcircuits is redundant, and conditions (2) and (3) are replaced by the above condition, that inputs of different matchgates may be linked by edge chains of length one or two:

Theorem 5. *If F is a read-twice formula consisting of gates from $B01 \cup B11 \cup B21 \cup B31 \cup B32$ then the matchnet of F has a matching iff F is satisfiable. If F is a read-twice formula F consisting of gates from $B01 \cup B11 \cup B21 \cup B31$ then the matchnet of F has an odd number of matchings iff F has an odd number of satisfying assignments. The existence of a solution for the formula in the first case and the parity of the number of solutions in the second case can both be computed in deterministic polynomial time.* □

A variety of special cases of these results were previously known and some related problems are known to be NP-hard. One such case is read-twice formulae consisting only of $x+y+z$ gates (from $B32$), for which it was known that satisfiability is polynomial time solvable [22, p. 207]. For that case it was further known that counting the number of solutions is #P-complete [6]. The parity problem for it is apparently open. The construction in [18] for read-twice formulae (of the form of conjunctions of disjunctions of conjunctions) implies that the existence problem for read-twice formulae constructed from $z = x + y$ functions is NP-complete. Also one can deduce that the corresponding counting problem is #P-complete and, using [30], that the corresponding parity problem is NP-hard via randomized reduction.

Theorem 6. *None of the six functions in $B33$ has a matchgate where either the parity of the number of matchings or the existence of matchings defines the required function. Neither of the functions in $B32$ has a matchgate where the parity of the number of matchings defines the required function.* We observe finally that numerous open problems remain regarding read-twice formulae composed of gates from the 24 gates that we enumerated.

7. CONCLUSION

The exact power of matchgate techniques for deriving polynomial time classical algorithms remains to be resolved. Are there indirect methods not yet identified for mapping richer classes computations into matchgates? Can our particular class of polynomially simulatable circuits be extended by allowing in addition arbitrary use of one-bit gates? Do circuits based on k-bit matchgates for $k > 2$ have greater power for encoding computations?

We have defined a class of quantum computations for which the outcome can be predicted by linear algebra com-

putations in polynomial time by classical computers. While the standard quantum mechanical formulation is linear also, it is linear in exponentially more dimensions than is ours. This raises the question as to whether physical limitations restrict scalable quantum devices to those that are polynomial time simulatable classically, as nonquantum devices are believed to be.

References

1. A. Barenco, A universal two-bit gate for quantum computation, *Proc. R. Soc. Lond.* A449, 679–683, (1995).
2. P. A. Benioff, Quantum mechanical Hamiltonian models of discrete processes that erase their own histories: Application to Turing machines, *Int. J. Theor. Phys.*, 21:6/7, 177, (1982).
3. E. Bernstein and U. V. Vazirani, Quantum complexity theory, *SIAM J. Computing*, 26:5, 1411–1473, 1997.
4. A. Borodin, S. A. Cook, and N. J. Pippenger, Parallel computation for well-endowed rings and space bounded probabilistic machines, *Information and Control*, 58:1–3, 113–136, (1983).
5. R. A. Brualdi and H. J. Ryser, *Combinatorial Matrix Theory*, Cambridge University Press, 1991.
6. R. Bubley and M. Dyer, Graph orientations with no sink and an approximation for a hard case of #SAT, *8th Ann. ACM-SIAM Symp. on Discrete Algorithms*, (SIAM Press, 1997), pp248-257.
7. P. Bürgisser, M. Clausen, and M.A. Shokrollahi, *Algebraic Complexity Theory*, Springer Verlag, (1996).
8. P. Bürgisser, *Completeness and Reduction in Algebraic Complexity Theory*, Springer Verlag, (2000).
9. A. Cayley, Sur les déterminates gauches, *Crelle's J.*, 38, 93, (1854).
10. S. A. Cook, The complexity of theorem proving procedures, *Proc. 3rd ACM Symp. On Theory of Computing*, (ACM Press, New York, 1971), 151–158.
11. D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, *J. Symbolic Comput.*, 9, 251–280, (1990).
12. D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, *Proc. R. Soc. Lond.* A400, 97–117, (1985).
13. D. Deutsch, Quantum computational networks, *Proc. R. Soc. Lond.* A425, 73–90, (1989).
14. D. Deutsch, A. Barenco, and A. Ekert, Universality of quantum computation, *Proc. R. Soc. Lond.* A449, 669–677, (1995).
15. D. P. DiVincenzo, Two-bit gates are universal for quantum computation, *Phys. Rev. A*, 51:2, 1015–1022, (1995).
16. P. van Emde Boas, Machine models and simulations, in *Handbook of Theoretical Computer Science*, Vol. A, J. van Leeuwen, ed. (Elsevier, Amsterdam, 1990), 1–61.
17. R. P. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.*, 21:6/7,467–488, (1982).
18. D. Gottesman, The Heisenberg representation of quantum computers, quant-ph/9807006, (1998).
19. L.K. Grover, Fast quantum mechanical algorithm for database search, *Proc. 28th ACM Symp. on Theory of Computing*, ACM Press, New York, 212-218 (1996)
20. H. B. Hunt, and R. E. Stearns, The complexity of very simple Boolean formulas with applications, *SIAM J. on Comput.*, 19:1, 44–70, (1990).
21. M. R. Jerrum, L. G. Valiant, and V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution, *Theor. Comp. Sci.* 46, 169–188, (1986).
22. K. de Leeuw, E. F. Moore, C. E. Shannon, and N. Shapiro, Computability by probabilistic machines, in *Automata Studies*, C. E. Shannon and J. McCarthy, eds. (Princeton Univ. Press, 1956), pp. 183–212.
23. E. H. Lieb, A theorem on Pfaffians, *J. Comb. Theory*, 5, 313–319, (1968).
24. K. Murota. *Matrices and Matroids for Systems Analysis*, Springer-Verlag, Berlin, 2000.
25. C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley, Reading, MA, (1994).
26. R.W. Robinett. *Quantum Mechanics*, Oxford University Press, New York, 1997.
27. P. W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, *SIAM J. Computing*, 26:5, 1484–1509, 1997.
28. V. Strassen, Gaussian elimination is not optimal, *Numer. Math.*, 13, 354–356, (1968)
29. L. Troyansky and N. Tishby. Permanent uncertainty: On the quantum evaluation of the determinant and the permanent of a matrix. In *Proceedings of Phys. Comp.* 96.
30. A. M. Turing, On computable numbers, with an application to the *Entscheidungsproblem*, *Proc. Lond. Math. Soc.*, Ser. 2, 42, 230–265 (1936).
31. L. G. Valiant, The complexity of computing the permanent, *Theor. Comp. Sci.*, 8(2), 189–201, (1979).
32. L. G. Valiant, Completeness classes in algebra, *Proc. 11th ACM Symp. On Theory of Computing*, (ACM Press, New York, 1979), 249–261.
33. L. G. Valiant, Expressiveness of matchgates, submitted for publication, February 2001.
34. L. G. Valiant and V. V. Vazirani, NP is as easy as detecting single solutions, *Theor. Comp. Sci.*, 47, 85–93, (1986).
35. A. C.-C. Yao, Quantum circuit complexity, *Proc 34th Symp. On Foundations of Computer Science*, (IEEE Computer Society, Los Alamitos, CA, 1993), pp. 352–360.

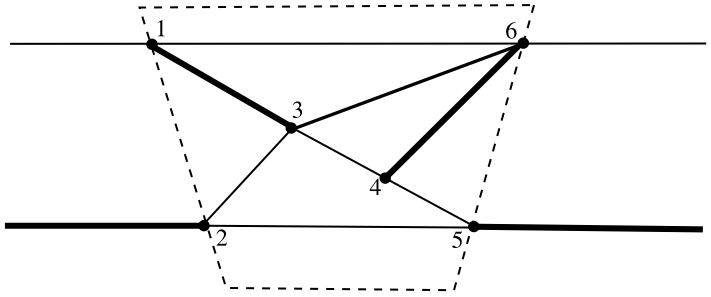

Figure 1: Illustration of a matchgateΓ with inputs X={1,2} and outputs Y={5,6}, with the external nodes Z={2,5} matched. The character $\chi(\Gamma,Z)$ is the product of the modifier $\mu(\Gamma,Z)$ and of the Pfaffian Sum of the matchgate after nodes Z have been removed. Now $\mu(\Gamma,Z) = 1$ since the externally matched edges (i.e. those from nodes 2 and 5) overlap exactly twice with the internally matched edges, whether the edges {1,3} and {4,6} shown in this example, or any alternative ones, such as {1,6}, {3,4}.

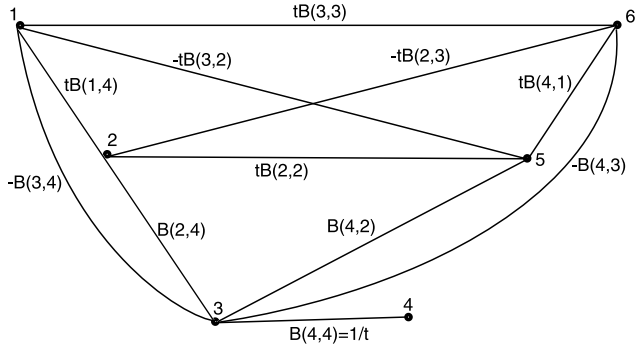

Figure 2: A matchgate with inputs {1,2} and outputs {5,6}, designed to have character that equals the given matrix B in eleven of the sixteen entries, as required by Proposition 4. The quantity t denotes 1/B(4,4).

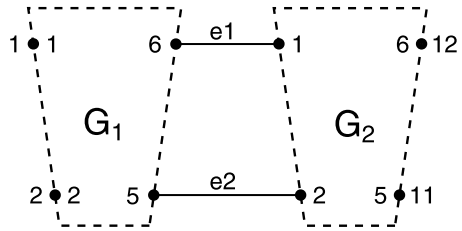

Figure 3: A composition of two six node matchgates G_1 and G_2, that forms a twelve node matchgate G. The original labeling {1,2,5,6} of the external nodes of G_1, G_2, as well as the new labeling {1,2,11,12} of the external nodes of G are shown. In the composition all the twelve nodes are renumbered {1, ..., 12} from left to right in the diagram.

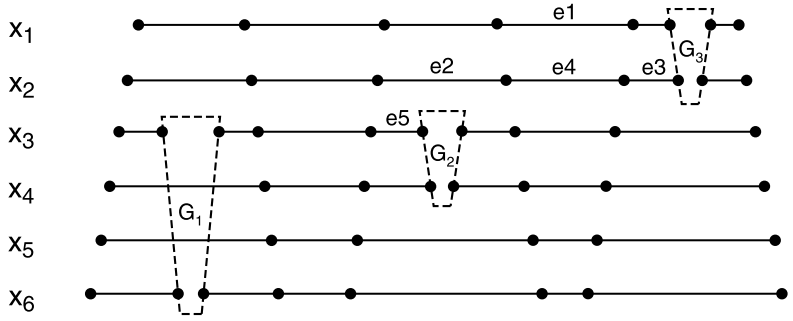

Figure 4: An example of a circuit composed of gates G_1, G_2, and G_3 acting on bits $\{x_3, x_6\}$, $\{x_3,x_4\}$ and $\{x_1,x_2\}$, respectively. In this example e3 and e5 are external edges, e2 is a parallel edge, e1 and e4 are connecting edges, and the internal edges within G_1, G_2, and G_3 are not shown. The nodes in the overall circuit are numbered so as to be increasing from left to right. The input and output node sets X,Y are the leftmost and rightmost six nodes, respectively.