Electronic Colloquium on Computational Complexity, Report No. 32 (2007)

Quantum deduction rules (preliminary version)

Pavel Pudlák [*]

March 27, 2007

Abstract

We define propositional quantum Frege proof systems and compare it with classical Frege proof systems.

1 Introduction

In this paper we shall address the question whether quantum circuits could help us prove theorems faster than conventional devices. For some computational problems, quantum circuits have been designed that are much smaller than all known classical circuits. In particular, polynomial size quantum circuits were found for the problem of factoring integers and researchers believe that there are no polynomial size classical circuits for this problem. Therefore our aim in this paper is to initiate the study of quantum concepts in proof complexity. We should stress that we shall only study the proof complexity of *classical logic,* we shall *not* consider *quantum logic* or any other nonclassical logic.

In proof complexity one studies the complexity of proofs of mathematical theorems and the strength of the axioms they need. Many concepts in proof complexity are naturally related to concepts in computational complexity. For instance, the concept of a proof system for propositional logic corresponds to the concept of a complexity class. One of the fundamental concepts in proof theory and proof complexity is the concept of a Frege system [5]. Frege systems are proof systems for propositional logic based on axiom schemas and deduction rules. The main question that we study in this paper is how to define a quantum counterpart of Frege systems.

The key step in defining such a concept is to define a *quantum deduction rule.* Roughly speaking, a quantum deduction rule is a unitary operation that acts on strings of a fixed length in such a way that it preserves truth. Given a finite set of quantum deduction rules, a quantum Frege proof is a sequence of proof lines, the first line being empty and each next line is obtained from the previous one by applying one of the quantum deduction rules. Thus each proof line is a quantum superposition of strings of formulas. We say that the proof proves a given proposition ϕ if ϕ occurs in the last proof line with amplitude α, $|\alpha|^2 \geq 1/2$ (ie., if we measure the last state we shall see a string of propositions which includes ϕ with probability $\geq 1/2$).

We shall show that given such a quantum proof one can extract from it a classical proof of the same proposition that has essentially the same size. Thus quantum Frege proofs do not allow us to present proofs in a more compact way than classical ones. Yet it could happen that one can prove more using quantum proofs. The reason is that given a quantum proof there does not seem to be an efficient algorithm to extract a classical proof from it. We shall show, that it is impossible to extract classical proofs from quantum Frege proofs in polynomial time using classical Turing machines if factoring integers is hard.

We shall only consider the propositional calculus. (Thus we shall omit the adjective *propositional* when talking about proof systems etc.) However, we believe that similar results can be proven for predicate logic,

[*]Supported by the research project MSM 0021620838 of the Ministry of Education of the Czech Republic.

ISSN 1433-8092

because the essential combinatorial concepts, namely, rewriting and substitution, are present already in the propositional calculus.

In computational complexity the concept of a quantum proof has a much more general meaning. It is a quantum state that witnesses the presence of a string in a set. It is a means to define sets, which is an extension of nondeterminism and randomness to the quantum setting. The most important complexity class considered is **QMA** (quantum Merlin-Arthur) an extension of the classical **MA** (Merlin-Arthur). In a relativized world **QMA** is strictly larger than **MA**. Our results are related to this area of research, but do not imply anything new about these classes.

2 Basic definitions and notation

In this section we shall introduce notation and recall some basic concepts from proof complexity and quantum computation. However, we do suppose that reader is familiar with basics of these fields (as presented eg. in [8, 6]).

Let a complete basis of connectives be fixed. $TAUT$ will denote the set of all propositional tautologies. We shall use a slightly modified definition of a propositional proof system.

Definition 1 *A* proof system *is a set $A \subseteq \Sigma^*$ and a mapping $\Pi : A \to TAUT$ such that*

1. *A is decidable in polynomial time, ie., $A \in \mathbf{P}$;*
2. *Π is computable in polynomial time;*
3. *$\Pi(A) = TAUT$.*

For $a \in A$, we say that *a is a proof of* $\Pi(a)$, or *a proves* $\Pi(a)$.

Proof systems are compared using the following quasiordering relation: (Π, A) *polynomially simulates* (Σ, B) if there exists a polynomial time computable function $f : B \to A$ such that for every $b \in B$, $\Sigma(b) = \Pi(f(b))$. I.e. f transforms proofs in (Σ, B) into proofs in (Π, A) of the same tautologies.

To formalize quantum computations we shall use the standard notation. We shall consider a finitely dimensional complex space $\mathbb{C}^N$. *States* will be unit vectors, and they will be denoted by

$$|\text{description or label}\rangle$$

The scalar product of two vectors $|a\rangle$ and $|b\rangle$ is denoted by $\langle a|b\rangle$. The symbol $\langle u|V|w\rangle$ denotes the product of a row vector $\langle u|$, linear operator or a matrix U and a column vector $|w\rangle$.

Transformations of the states are determined by unitary operators. If U is unitary, we write

$$|a\rangle \mapsto U|a\rangle.$$

In order to represent a state or a transformation we fix an orthonormal basis of the space. Given a basis $\mathcal{B}$, a state $|a\rangle$ is represented by the vector

$$\{\langle b|a\rangle\}_{b\in\mathcal{B}},$$

and a unitary operator U is represented by its matrix

$$\{\langle b|U|c\rangle\}_{b,c\in\mathcal{B}}.$$

We shall say that $\langle b|a\rangle$ is *the amplitude of b in state $|a\rangle$* and $\langle b|U|c\rangle$ is *the amplitude of the transition from b to a.*

When computing we assume that the states of the basis are labeled by strings of 0's and 1's. So the basis is:

$$\{|a\rangle\ ;\ a \in \{0,1\}^m\},$$

where we assume that the dimension of the space of states is 2^m. The elements of the space are called m *quantum bits.*

A *quantum gate* is a unitary transformation $g : \mathbb{C}^{2^k} \to \mathbb{C}^{2^k}$, where $k = 1, 2$. Such a transformation can be extended to the unitary transformation $U_g : \mathbb{C}^{2^m} \to \mathbb{C}^{2^m}$ by defining

$$U_g = I_{m_1} \otimes U_g \otimes I_{m_2}$$

where I_j denotes the identity transformation on $\mathbb{C}^{2^j}$, $m_1 + k + m_2 = m$ and $\otimes$ denotes the tensor product. More generally, we allow the quantum gate to act on an arbitrary pair of bits, not necessarily on a pair of consecutive bits. (This would be harder to express using a formula.)

A *quantum circuit* is $C = (U_{g_1}, \dots, U_{g_t})$, where $g_1, \dots, g_t$ are quantum gates. This circuit computes the unitary transformation

$$U_C = U_{g_1} \circ \dots \circ U_{g_t}$$

In most computations we need auxiliary qubits for computations. Thus for some n, l such that $n + l = m$, the computation starts with a boolean input vector a of length n and the rest, ie. l bits, are fixed to 0s. The result of the computation $U|a\bar{0}\rangle$ is in general a linear superposition of basis states

$$U_C|a\bar{0}\rangle = \sum_{b \in \{0,1\}^n} \beta_b |b\rangle$$

(where $\sum_b |\beta|^2 = 1$). Again, we consider the output to be a specified subset of $n' \leq m$ bits and we ignore the rest. If measurements give always the same n' output bits (ie., if for all $\beta_b \neq 0$ the first n' of b are the same) for a fixed input, we say that it is an *exact computation.* The circuit then computes a boolean function. In *bounded error computation* of a function f, we only require that, given an argument $a \in \{0,1\}^n$, the output value $f(a)$ occurs with large probability, where "large" means $> 1/2$. (The 1/2 is not essential, we can stipulate any positive constant.) One can compute *factoring of integers* with bounded error in polynomial time [14]. *Discrete logarithm* can be computed exactly in polynomial time if one is allowed to choose an appropriate gate for the given finite field. For these functions no classical randomized polynomial time algorithms are known.

It should be stressed that when we compose quantum computations, we cannot ignore the remaining $m - n'$ bits of the output. In many cases we must eliminate the "junk qubits", ie., we need to have those bits to be equal to 0. This makes a huge difference: the function must be one-to-one, and it must be reversible—we can compute the inverse by turning the circuit upside-down.

Circuits are best suited for defining nonuniform complexity classes. If we want to define uniform complexity classes we either have to use Turing machines or uniform families of circuits. One can define Quantum Turing Machines, but it is more convenient to work with uniform families of quantum circuits. We shall only need **P**-uniformity which is defined as follows. A family of quantum circuits $\{C_n\}$ is **P**-*uniform,* if there exists a classical Turing machine which, given a string of n zeros, constructs C_n in polynomial time (notice that in particular the size of C_n is polynomial in n). Here we denote by n the number of input bits of C_n.

The most important quantum complexity class is **BQP**; it is defined as the class of languages L for which there exist a **P**-uniform family of quantum circuits $\{C_n\}$ such that for every n, C_n accepts $a \in \{0,1\}^n$

1. with probability $\geq 2/3$, if $a \in L$,
2. with probability $\leq 2/3$, if $a \notin L$.

We say that C_n accepts a, if the measurement of the state $C_n(a)$ produces a string starting with 1.

Quantum proof systems

There are several ways in which one can generalize the concept of a proof system using quantum computations. We are not going to study generic quantum proof systems, we shall rather focus on one and compare it with the corresponding classical one. Nevertheless, it is worthwhile to list the main properties that the system possesses. Since think that these properties are natural, we propose to define in this way quantum proof systems.

In order to obtain a concept more general than classical proof systems we have to relax at least one of the conditions of Definition 1. We shall replace condition 2 by a weaker condition that only requires the tautologies proved by the proof to be computable by **P**-uniform quantum circuits. Furthermore, we shall allow one proof to prove more than one tautology. This would make essentially no difference for the original concept, but in case of quantum proof systems it seems too restrictive to require each proof to prove only one tautology.

Definition 2 *A* quantum proof system *is a set $A \subseteq \Sigma^*$ and a family of circuits $\{C_n\}_{n\in\mathbb{N}}$ such that*

1. *A is decidable in polynomial time, ie., $A \in \mathbf{P}$;*
2. *$\{C_n\}$ is $\mathbf{P}$-uniform;*
3. *given $P \in A$ and $n = |P|$, C_n produces a superposition $C_n(P)$ of* strings *of tautologies;*
4. *for every tautology ϕ, there exists P such that ϕ occurs in the superposition $C_n(P)$.*

Let us explain the last two conditions in detail. Let $C_n(P)$ be a superposition of strings

$$\sum_{i\in I} \alpha_i|S_i\rangle,$$

where all $\alpha_i \neq 0$ and $S_i \neq S_j$ for $i \neq j$. Then Condition 3 says that a distinguished part of each S_i is a string of tautologies. Condition 4 says that for every tautology ϕ, there exists a proof $P \in A$ such that in the superposition of strings $C_n(P)$ a string containing ϕ occurs with a nonzero amplitude.

In the sequel we shall always use the words "*occurs in the superposition*" with this meaning. Namely, *a basis state $|a\rangle$ occurs in $B = \sum_b \alpha_b|b\rangle$ iff $\alpha_a \neq 0$ iff the measurement of B gives the string a with nonzero probability.* Here the sum is over basis states, which always will be all strings of zeros and ones of some fixed length.

For a given proof $P \in A$ and a tautology ϕ, we shall say that P *weakly proves* ϕ, if ϕ occurs in $C_n(P)$. If P weakly proves ϕ, the probability that we obtain ϕ by measuring $C_n(P)$ is nonzero, but it can be very small. This is a rather theoretical concept, since in practice it would not be possible to test this relation. Therefore we reserve the words P *proves* ϕ to the case when the probability that we obtain ϕ from $C_n(P)$ is is at least 1/2. (The constant 1/2 is not essential, as the usual amplification methods apply to this concept.) Furthermore, we define that P *strongly proves* ϕ if the probability that we obtain ϕ is 1.

Remarks. 1. In this definition we adhere to the original meaning of the word *proof* by ensuring absolute certainty that a proposition ϕ is a tautology once we establish that P is a quantum proof of it. Also having a string of bits representing a proof enables us to copy the proof and use it repeatedly, which would not be possible if proofs were general quantum states. In other words, the experiment demonstrating that ϕ is a tautology is not a one time experiment, it can be repeated. What may seem counterintuitive is that ϕ is not a part of P. But notice that being a part of P means that one can easily compute ϕ from P (namely, to compute in polynomial time). If we had both $A \in \mathbf{P}$ and the function $P \mapsto \phi$ computable in polynomial time, then we would just obtain the definition of *classical* proof systems. We have to relax at least one of the two conditions in order to obtain a more general concept.

2. Our definition of quantum Frege proof systems will satisfy a condition slightly stronger than 2. of Definition 2. Namely, in quantum Frege systems every string S occurring in $C_n(P)$ will consist only of several tautologies and the rest filled by the default value 0. So there will be no "junk bits".

We shall extend the concept of simulation to quantum proof systems. We say that a classical (quantum) proof system (A, Π) (resp. $(A, \{C_n\})$) *polynomially simulates* a classical (quantum) proof system (B, Σ) (resp. $(B, \{D_n\})$), if there exists a function f

$$f : B \times TAUT \to A$$

which is computable in polynomial time and such that for every $P \in B$ and $\phi \in TAUT$, if P proves ϕ in (B, Σ) (resp. $(B, \{D_n\})$), then $f(P, \phi)$ proves ϕ in (A, Π) (resp. $(A, \{C_n\})$).

We shall also need the following concept.

Definition 3 *Let $\mathcal{B}$ be an orthonormal basis of a finitely dimensional Hilbert space $\mathcal{H}$. Let $K = (U_1, \dots, U_t)$ be a string of unitary operators.*

1. *For $a_0, a_1, \dots, a_t \in \mathcal{B}$, we shall say that $(a_0, a_1, \dots, a_t)$ is a* history *of K, if for all $i = 1, \dots, t$, $\langle a_i | U_i | a_{i-1} \rangle \neq 0$.*
2. *The* amplitude of a history $h = (a_0, a_1, \dots, a_t)$ *is the complex number*

$$\alpha_h = \langle a_t | U_t | a_{t-1} \rangle \langle a_{t-1} | U_{t-1} | a_{t-2} \rangle \dots \langle a_1 | U_1 | a_0 \rangle.$$

Lemma 2.1 *Let $U = U_t \dots U_1$ be the product of a string of unitary transformations $K = (U_1, \dots, U_t)$ and $a, b \in \mathcal{B}$ such that $\langle b | U | a \rangle \neq 0$. Then*

$$\langle b | U | a \rangle = \sum_h \alpha_h,$$

where the sum is over all histories h of the form

$$(a = a_0, a_1, \dots, a_t = b).$$

In particular, there exists at least one history of this form.

Proof. Easy—from the definition of the multiplication of matrices. ■

We shall use this concept for the Hilbert space $\mathcal{H}$ of n qubits and for U_i the unitary transformation defined on $\mathcal{H}$ by the gates of a quantum circuit. More generally, we may consider any stratification of the circuit and define fibers with respect to this stratification. Clearly, histories in a coarser stratification are subsequences of histories of a finer stratification.

Given a quantum circuit C we can easily construct a circuit that produces the superposition of histories with the assigned amplitudes. However, in general, this does not help us to obtain a history starting and ending in particular states. Even if $\langle b | U | a \rangle = 1$, the sum $\sum_h |\alpha_h|^2$ over all histories of the form $(a = a_0, a_1, \dots, a_t = b)$ may be exponentially small.

Example. Let H be a unitary matrix on an N-dimensional space whose matrix in a basis $\mathcal{B}$ is a normalized Hadamard matrix. (For instance, let $N = 2^n$ and H be the tensor product of n 2-by-2 normalized Hadamard matrices.) Let $t = 2$, $U_1 = U_2 = H$ and let $a \in \mathcal{B}$ be a fixed element. Then $\langle a | U_2 U_1 | a \rangle = 1$, since $U_2 U_1$ is the identity. For every $b, c \in \mathcal{B}$, we have

$$|\langle c | U_2 | b \rangle \langle b | U_1 | a \rangle| = \frac{1}{\sqrt{N}} \frac{1}{\sqrt{N}} = \frac{1}{N}.$$

Hence

$$\sum_b |\alpha_{(a,b,a)}|^2 = \frac{1}{N}.$$

3 Frege proof systems

A *Frege system,* [5], is determined by a finite set of (usual) axiom schemas and rules. Recall that an axiom schema can be viewed as a rule with no premises. We assume that the system is sound and complete—it proves exactly all tautologies in $TAUT$.

A *Frege proof* is a finite sequence of propositions such that each proposition is either an instance of an axiom schema or follows from previous propositions by an application of a rule. A proof proves a proposition ϕ, if ϕ is the last proposition in it.

Example. [Hilbert and Ackermann]
Connectives: $\neg, \vee$.
Axiom schemas:

1. $\neg(A \vee A) \vee A$
2. $\neg A \vee (A \vee B)$
3. $\neg(A \vee B) \vee (B \vee A)$
4. $\neg(\neg A \vee B) \vee (\neg(C \vee A) \vee (C \vee B))$

Rule:
5. *From A and $\neg A \vee B$ derive B.*

It is important to realize that in Frege systems we do not use the general substitution rule, we only use substitution instances of axioms and rules. Therefore we shall use different variables in rules and in propositions that we are proving:

- $A, B, C, \ldots$ will be variables in the rules, ie., for which we can substitute propositions;
- $u, v, w, \ldots$ will be variables in propositions in proofs, ie., they represent the truth values *true* and *false.*

We shall call $A, B, \ldots$ *metavariables* and $u, v, \ldots$ *variables.* We shall denote by $PROP$ propositions that contain only variables. We have been assuming that a complete basis of connectives is fixed and we are considering only propositions in this basis.

4 Quantum deduction rules

Our aim is to generalize the concept of a Frege system using quantum computations. The crucial step is to define quantum deduction rules. We shall start with a general definition and show that every such a rule must have a special form.

Let $\mathcal{S}$ be a finite set of finite strings of propositions. Let $\mathcal{H}$ be the Hilbert space with an orthonormal basis $\{|S\rangle;\ S \in \mathcal{S}\}$.

Definition 4 *A* quantum deduction rule defined on $\mathcal{S}$ *is a unitary transformation $D : \mathcal{H} \to \mathcal{H}$ such that*

1. *If*
$$D : \sum_i \alpha_i |S_i\rangle \ \mapsto \ \sum_j \beta_j |T_j\rangle$$
where all $\alpha_i, \beta_j \neq 0$, then every proposition ϕ in every T_j is a logical consequence of the set of propositions $\bigcup_i S_i$. Formally,
$$\bigcup_i S_i \ \vdash \ \bigcup_j T_j.$$

2. *For every $S, T \in \mathcal{S}$, if $S \neq T$, then no string of propositions is a substitution instance of both S and T.*

The second condition concerns only the set of strings on which D is defined. It is needed to ensure that applications of quantum rules are unitary. (An example of an ambiguous pair of strings is the following: $(A \wedge B, C)$ and $(A, B \vee C)$ have a common substitution instance $(u \wedge v, u \vee v)$.)

Classical deduction rules can have several assumptions but they always have one conclusion. In order to be able to compare them with quantum deduction rules, we shall assume that also classical rules can have multiple conclusions (each proposition being a logical consequence of the assumptions). The following proposition is a trivial consequence of the definition. It shows that in a sense a quantum deduction rule is a combination of classical rules.

Proposition 4.1 *The first condition of the definition of a quantum deduction rule is equivalent to:*

$$\langle T|D|S\rangle \neq 0 \;\Rightarrow\; S \;\vdash\; T. \tag{1}$$

Proof. Suppose $\langle T|D|S\rangle \neq 0$. Then T occurs in $D|S\rangle$, hence the propositions in T logically follow from propositions in S.

For the converse observe

$$D\sum_i \alpha_i|S_i\rangle = \sum_i \alpha_i \sum_T \langle T|D|S_i\rangle|T\rangle.$$

Hence T can occur in the superposition only if $\langle T|D|S_i\rangle \neq 0$ for some i. ■

Proposition 4.2 *Every quantum deduction rule D is reversible, ie., D^{-1} is also a quantum deduction rule. Thus*

$$\langle T|D|S\rangle \neq 0 \;\Rightarrow\; \bigwedge S \;\equiv\; \bigwedge T.$$

For example, the following rule cannot be extended to a quantum rule:

$$D : A \wedge B \;\mapsto A \vee B\ .$$

Proof. The essence of this proposition is the following simple fact. Let A be a doubly stochastic matrix and let $\begin{pmatrix} B & C \\ E & F \end{pmatrix}$ be its block decomposition where B is a square submatrix. Then $C = 0$ iff $E = 0$.

Let D be a quantum deduction rule with a set of strings of propositions $\mathcal{S}$. Let G be the directed graph with the set of vertices $\mathcal{S}$ and $S \to T$ an arrow iff $\langle T|D|S\rangle \neq 0$. Since D^{-1} is the conjugate transposed of D, it suffices to prove that if $S \to T$, then $T \vdash S$. To this end it suffices to show that if $S \to T$ in G, then there exists a directed path from T to S in G.

Suppose, by contradiction, that $S \to T$, but there is no directed path from T to S. Let $\mathcal{T} \subseteq \mathcal{S}$ be all vertices of G reachable from T by directed paths. Let A be the doubly stochastic matrix whose entries are the squares of the absolute values of the entries of D. Let B be the square submatrix of A determined by $\mathcal{T}$ and C, E, F matrices of the decomposition of A determined by B. Then $C \neq 0$ whereas $E = 0$, which is impossible. ■

We shall say that a *quantum deduction rule is simple* if $S \equiv T$ for every $S, T \in \mathcal{S}$. The previous proposition implies:

Corollary 4.3 *Every quantum deduction rule is a direct sum of simple rules.*

We shall call a quantum deduction rule *proper*, if the set of metavariables occurring in strings S is the same for all $S \in \mathcal{S}$.

Examples. 1. The following is a simple proper quantum deduction rule.

	A, B	$A \wedge B$	$A, B, A \vee B$
A, B	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
$A \wedge B$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	$-\frac{1}{2}$
$A, B, A \vee B$	$-\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	$-\frac{1}{2}$

Notice that we may derive $A \vee B$ from $A \wedge B$, but we must keep A and B in order to have a string of formulas of the same logical strength.

It is convenient to assume the same number of propositions in every string. Therefore, we shall use the constant true for the missing propositions.

	$A,\ B,\ \top$	$A \wedge B,\ \top,\ \top$	$A,\ B,\ A \vee B$
$A,\ B,\ \top$	0	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
$A \wedge B,\ \top,\ \top$	$\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	$-\frac{1}{2}$
$A,\ B,\ A \vee B$	$-\frac{1}{\sqrt{2}}$	$\frac{1}{2}$	$-\frac{1}{2}$

If the number of formulas in the strings is k, we shall say that it is a k-ary rule. So the above rule is ternary.

2. A classical axiom schema $\Phi(A, B, \ldots)$ can be presented as a quantum rule as follows:

	$\top$	$\Phi(A, B, \ldots)$
$\top$	0	1
$\Phi(A, B, \ldots)$	1	0

This rule is not proper.

3. In the same manner one can transform every classical deduction rule into a quantum deduction rule, we only have to keep some premises in case the conclusion is weaker than the conjunction of premises. Thus the version of *modus ponens* used in the proof system of Hilbert and Ackermann can be presented as follows.

	A, B	$A, \neg A \vee B$
A, B	0	1
$A, \neg A \vee B$	1	0

Since quantum concepts are usually more delicate than classical ones, we have to define precisely how quantum rules are applied. This concerns details that are usually ignored when defining applications of classical rules. When defining applications of quantum rules, we need to prove that they are unitary transformations, and then these things become important.

Let D be a k-ary quantum deduction rule on the set of strings of propositions $\mathcal{S}$. Let $\Gamma \subseteq PROP^k$ be a finite set of strings of length k of propositions. Let Σ be a set of substitutions. Then we can use D to define a unitary transformation Δ on Γ as follows.

1. Assume that the same metavariables $A_1, \ldots, A_n$ occur in every $S \in \mathcal{S}$. We need to define amplitudes for transitions $s \rightarrow t$ for every $s, t \in \Gamma$.

(i) If there is no $S \in \mathcal{S}$ such that s is a substitution instance of S where we only consider substitutions from Σ, then we put $\langle t|\Delta|s\rangle = 1$ if $s = t$, otherwise we put it equal to 0.

(ii) If there exist $S \in \mathcal{S}$ and $\sigma \in \Sigma$ such that $s = \sigma(S)$, then S and σ are determined uniquely, according to condition 2. of Definition 4. Let $t \in \Gamma$ be such that $t = \sigma(T)$ for some $T \in \mathcal{S}$. Then we put

$$\langle t|\Delta|s\rangle = \langle T|D|S\rangle.$$

For all other $t \in \Gamma$ we put $\langle t|\Delta|s\rangle = 0$.

To ensure that the resulting transformation Δ is unitary we need to assume the following condition.

$$\textit{if } \sigma(S) \in \Gamma \textit{ for some } S \in \mathcal{S} \textit{ and some substitution } \sigma \in \Sigma, \textit{ then for all } T \in \mathcal{S},\ \sigma(T) \in \Gamma. \tag{2}$$

Lemma 4.4 *If condition (2) is satisfied, then Δ is unitary.*

Proof. Let σ be a substitution such that $\sigma(S) \in \Gamma$ for some $S \in \mathcal{S}$. Let $\mathcal{S}_\sigma = \{\sigma(T); T \in \mathcal{S}\}$. If σ' is another such a substitution, then $\mathcal{S}_\sigma \cap \mathcal{S}_{\sigma'} = \emptyset$. That is because of condition 2. Definition 4 and because $\sigma \neq \sigma'$ implies $\sigma(T) \neq \sigma'(T)$ (we are tacitly assuming that substitutions are defined only on $A_1, \dots, A_n$). Thus Γ can be partitioned into blocs $\mathcal{S}_\sigma$ and a block of strings that do not belong to any $\mathcal{S}_\sigma$. Therefore the matrix of Δ is a direct sum of copies of the matrix of D and an identity matrix. This proves that Δ is unitary. ■

2. Now assume that D is a general quantum rule. Then it can happen that some variable A occurs T but not in S. If σ is a substitution such that $\sigma(S) = s$ for some $s \in \Gamma$, then σ is uniquely determined only for variables occurring in S. Thus the pair S, s does not uniquely determine what we should substitute for A. Therefore we have to fix the values of substitutions on A independently on Γ.

In more detail, we shall do it as follows. Let $A_1, \dots, A_n$ be the variables that occur in all $S \in \mathcal{S}$; we shall call them *proper metavariables.* Let $B_1, \dots, B_m$ be the variables that occur in some elements of $\mathcal{S}$, but not in all; we shall call them *improper.* Let $\phi_1, \dots, \phi_m \in PROP$ be some fixed propositions. We shall consider substitutions σ defined on variables $A_1, \dots, A_n, B_1, \dots, B_m$ which are defined arbitrarily on proper variables and such that $\sigma(B_i) = \phi_i$ is fixed for $i = 1, \dots, m$. Then we proceed as in 1; in particular, in condition (2) the substitutions have the form just described. The same argument as in 1. shows that the resulting transformation Δ is unitary also in general.

The fact that an application of a quantum deduction rule defines a unitary transformation justifies its name, but we need more, because we want to be able to compute quantum proofs using quantum computers. We need to show that quantum circuits computing these transformations can be constructed in polynomial time. We shall assume that Γ is the set of all strings of k propositions where each proposition is coded by n_2 bits. Thus the input size is $n = kn_2$. Σ will be the maximal set of substitutions satisfying condition (2).

Proposition 4.5 *It is possible to construct in polynomial time a quantum circuit that computes the unitary transformation defined by D on $\{0,1\}^n$.*

Proof. Instead of a circuit, we shall describe a quantum algorithm which can be transformed into circuits using standard methods.

Let $s \in \{0,1\}^n$ be given as input. Deciding whether s is a string of propositions and whether there exists a string $S \in \mathcal{S}$ and a substitution $\sigma \in \Sigma$ such that $\sigma(S) = s$ and if so computing the string and the substitution the string can be done in polynomial time. We shall keep s during the computation, hence we can use a reversible computation (which is a quantum computation).

1. If there is no such pair S, σ, this is the end of the computation.

2. Otherwise the pair is uniquely determined. Now we can erase s since it is determined by σ and S. Then we apply the unitary transformation D to S. In general we obtain a superposition of elements of $\mathcal{S}$, but we only need to describe what we shall do with a specific single element $T \in \mathcal{S}$: we apply the substitution σ to T; let $t = \sigma(T)$. Now we have t, σ, T. Since the pair σ, T is uniquely determined by t, we can erase it.

Here is a diagram of the computation:

$$\begin{array}{ccc}
s & & \\
s & \sigma & S \\
 & \sigma & S \\
 & \sigma & T \\
t & \sigma & T \\
t & &
\end{array}$$

■

5 Quantum Frege proofs

In classical computations we can keep all data that we produce during the computation, but in quantum computations it is often very important to dispose of some data that are not needed in the rest of the computation. We shall see the same relation between classical and quantum proofs. Classical proofs are defined as sequences of *all* formulas produced during the deduction process. However, when showing a proof on a blackboard, we erase (due to the space limitations) propositions that we do not need anymore. Our definition of quantum Frege proofs will be more like the proofs on blackboards—we shall add and *erase* propositions during the proof. Furthermore, at each step we shall not have just a set of propositions, but rather a quantum superposition of sets of propositions. Since superpositions may consists of exponentially many observable states, we cannot represent them explicitly. Therefore, a quantum proof will be a sort of a quantum program to produce such superpositions.

Definition 5 *A* quantum Frege proof *is a sequence* $(n_1, n_2, J_1, \dots, J_m,)$*, where* n_1 *is the number of propositions in proof lines,* n_2 *is the maximal size of propositions, and* $J_1, \dots, J_m$ *are instructions specifying applications of quantum rules.*

An instruction *is a string* $(D, i_1, \dots, i_k, \psi_1, \dots, \psi_l)$*, where* D *is a* k*-ary quantum deduction rule* $i_1, \dots, i_k$ *are indices of formulas to which the rule should be applied and* $\psi_1, \dots, \psi_l$ *are propositions that should be substituted for improper metavariables.*

Such a proof is associated with a computation that produces a sequence of $m+1$ *proof-lines. A* proof line *is a quantum superposition of strings of* n_1 *propositions, each proposition of size at most* n_2*. The* 0*-th proof line is always the string of* n_1 *formulas* $\top$*. For* $i = 1, \dots, m$*, the* i*-th proof-line is obtained by applying the quantum rule* D_i *according to the* i*-th instruction.*

We say that $(n_1, n_2, J_1, \dots, J_m)$ proves *(resp.* weakly proves*, resp.* strongly proves*)* ϕ*, if the following holds true. If we measure the last proof line, then* ϕ *occurs in the string of the propositions obtained from from the measurement with probability* $\geq 1/2$ *(resp.* > 0*, resp.* $= 1$*).*

Definition 6 *A* quantum Frege system *is determined by a finite set of quantum deduction rules such that all tautologies are provable using only these deduction rules.*

By Proposition 4.5, we can construct in polynomial time a quantum circuit that computes the proof lines of a given quantum Frege proof. Thus the concept of proving (resp. weakly/strongly proving) is well defined. To show that the definition of quantum Frege systems satisfies our general definition of quantum proof systems (Definition 2), we only need to show that all propositions that appear in some proof-lines are tautologies. This follows from Proposition 5.1 below.

Since every classical Frege proof system can be turned into a quantum Frege system, the class of quantum Frege systems is nonempty. A classical result [5] states that Frege systems polynomially simulate each other. We shall show that the corresponding statement for quantum Frege proof systems is unlikely, since classical Frege systems probably do not polynomially simulate some quantum Frege systems. However, we conjecture that there exist "universal" quantum Frege systems in the following sense.

Conjecture 1 *There exists a quantum Frege system that polynomially simulates all other quantum Frege systems.*

The conjecture is supported by the fact there are finite sets of quantum gates that are universal for quantum circuits. In particular, it has been shown that it suffices to take two gates, the Toffoli gate and the Hadamard gate, [12, 2].

We can view the sequence of proof-lines of a quantum proof as a linear superposition of classical proofs, hence a classical Frege proof is always contained in a quantum Frege proof.

Proposition 5.1 *For every quantum Frege proof system there exists a classical Frege proof system such that if* $(n_1, n_2, J_1, \dots, J_m)$ *is a quantum proof which weakly proves* ϕ*, then there exists a classical proof of* ϕ *whose size is at most* $n_1 n_2$*.*

Thus, in particular, we cannot use quantum Frege proofs to construct shorter proofs.

Proof. Decompose the quantum deduction rules into classical ones. Consider a history the computation of proof-lines that leads to a proof-line containing ϕ. The history consists of $m+1$ strings s_i of propositions. According to Proposition 4.1, for $i = 1, \dots, m$, every proposition of s_i follows from some propositions of s_{i-1} by an application of a classical rule. ■

We shall need to simulate quantum proofs. To this end it is useful to simplify the form of instructions of quantum proofs.

First we observe that it suffices to use propositional variables as formulas substituted for improper metavariables. This is a well-known fact in sequent calculi and we shall use the same idea. Suppose we need to substitute a proposition ψ for an improper variable B. Then we first construct a tautology containing ψ, for example, we can take $\psi \vee \neg\psi$. We shall start with the axiom schema that introduces such a tautology and apply it to all variables of ψ. These will be the only applications of an improper rule. Then we use proper rules to combine $\psi_1 \vee \neg\psi_1$ and $\psi_2 \vee \neg\psi_2$ into $(\psi_1 \wedge \psi_2) \vee \neg(\psi_1 \wedge \psi_2)$, and so on for other connectives. Once we have such a proposition we may use $B \vee \neg B$ as a "side formula" of the rule, and thus B becomes a proper variable.

Secondly, we observe that the information about positions of formulas to which a rule should be applied can be coded in the index of the proof line. The same can be done with the specification of variables to which improper rules should be applied. In more detail, if we use at most k-ary deduction rules with at most l improper variables, we shall pick an enumeration of all k-element subsets of n_1 and l-element multisets of n_2 by numbers $1, \dots, p$; note that p is bounded by a polynomial in n_1 and n_2. Then we can require that at the i-th proof line we apply the rule to the subset of formulas and the string of variables specified by the number $q \equiv i \mod p$. To simulate an application of a rule D, we shall apply the identity rule everywhere except at the required position, where we apply D.

We shall call such proofs *oblivious*. An oblivious proof is simply a string of quantum deduction rules $(D_1, \dots, D_m)$ (and the pair of numbers (n_1, n_2) which we shall omit in the sequel). Thus we have shown the following:

Proposition 5.2 *For every quantum Frege system P there exists another quantum Frege system Q which is oblivious and which polynomially simulates P.*

6 Error propagation in quantum proofs

We have shown that using sound quantum deduction rules we can only derive tautologies from tautologies. The proof was based on the fact that for every string of propositions that occur (with nonzero amplitude) in the last state, there exists a history in the computation that can be interpreted as a classical Frege proof. The proof used the distinction between zero and non-zero amplitudes in an essential way. Such a proof is not sufficient if one wants to argue that quantum Frege proofs could be realized, if we had a quantum computer. In practice we can never construct an instrument with absolute precision, hence it is unrealistic to assume that non-tautologies always have amplitudes zero. The argument used in that proof does not exclude that once a non-tautology occurs in the computation, just with a tiny nonzero amplitude, it could spread so that non-tautologies occurring later would have large amplitudes. However, it is not difficult to see that this cannot happen.

Consider the space $\mathcal{H}$ spanned by strings S consisting n_1 propositions of length at most n_2. Let D be a quantum deduction rule and U the unitary transformation that it induces on $\mathcal{H}$. Consider a state

$$|a\rangle + |b\rangle,$$

where $|a\rangle$ is a superposition of strings S in which every proposition is a tautology and $|b\rangle$ is a superposition of strings S such that in each S at least one proposition is not a tautology. Since quantum deduction rules preserve truth and are invertible, in

$$U(|a\rangle + |b\rangle) = U|a\rangle + U|b\rangle$$

$U|a\rangle$ is the superposition of strings in which every proposition is a tautology and $U|b\rangle$ is the superposition of strings such that in each string at least one proposition is not a tautology. The lengths of the vectors $U|b\rangle$ and $|b\rangle$ are the same, which shows that a small error remains small throughout the computation.

This shows that we can ignore errors whose probabilities are very small. We do not know exactly what kind of errors will occur in quantum computers, as it will surely depend on the particular construction, but it seems reasonable to assume that only errors that change a small number of bits will have significant probabilities. Those errors can be eliminated using quantum error correcting codes.

7 How difficult is to extract a classical proof?

Although quantum proofs are not shorter than classical ones, they still may be useful. It can happen that we find a short quantum proof without being able to find the short classical proof that is hidden in it. Assuming that factoring integers is hard, we shall prove even more: in general, it is not possible to extract a *any* classical proof from a quantum Frege proof in polynomial deterministic time. We shall state it formally as a theorem about polynomial simulations.

Theorem 7.1 *Assuming that it is not possible to factor integers in probabilistic polynomial time, there exists a quantum Frege system that cannot be polynomially simulated by any classical Frege system.*

In more detail, there exists a quantum Frege system Q, and a set of propositions $\gamma_{a,0}, \gamma_{a,1}$, $a \in A$ such that

1. *exactly one of $\gamma_{a,0}$ and $\gamma_{a,1}$ is a tautology for every $a \in A$;*
2. *given an $a \in A$, it is possible to construct in polynomial time propositions $\gamma_{a,0}$ and $\gamma_{a,1}$ and a quantum proof P_a in Q which proves one of them;*
3. *there exists no classical proof system (B, Γ) and a polynomial time computable function f, such that, for every $a \in A$, $f(a)$ is a proof in (B, Γ) of either $\gamma_{a,0}$, or $\gamma_{a,1}$.*

We shall show that 3. implies that no classical proof system can simulate the quantum proof system Q. Suppose, by contradiction, that (B, Γ) is a classical proof system and g is a polynomial simulation of Q by (B, Γ). In order to compute $f(a)$ we first compute $L_0 = g(P_a, \gamma_{a,0})$ and $L_1 = g(P_a, \gamma_{a,1})$ and check for which $i = 0, 1$, $\Gamma(L_i) = \gamma_{a,i}$. Then we put $f(a) = L_i$ for this i. This produces a function f that violates 3.

Notice that we do not claim that A is computable in polynomial time. We must also say that 3. is a consequence of the the stronger statement that 1. is not decidable in polynomial time assuming factoring is hard. Indeed, suppose 3. were false, ie., we had such a proof system (B, Γ) and a function f. Then we could decide which of the two $\gamma_{a,0}$ and $\gamma_{a,1}$ is a tautology by computing $\Gamma(f(a))$.

The basic idea of the proof is to encode quantum computations as quantum Frege proofs. To encode a quantum circuit by a quantum Frege proof is very easy. Represent bits 0 and 1 by two tautologies τ_0 and τ_1, say $\tau_0 \leftrightarrow u \vee \neg u$ and $\tau_1 \leftrightarrow \neg u \vee u$. Thus a string of bits $(a_1, \ldots, a_n) \in \{0,1\}^n$ will be coded by the string of propositions $(\tau_{a_1}, \ldots, \tau_{a_n})$. Then quantum gates will be represented by quantum deduction rules in a natural way. For example, the NOT gate will correspond to the rule that interchanges $u \vee \neg u$ with $\neg u \vee u$.

However, it should be stressed that the simulation of quantum circuits alone does not suffice. If a quantum proof proves only simple tautologies (such as $u \vee \neg u$), then it is trivial to simulate it by a classical Frege proof. What we need are tautologies that are difficult to recognize without a quantum computer. To this end we shall use some concepts and results from cryptography and proof complexity.

A *weak one-way function* is a sequence of functions $\{F_n\}$ such that

1. $F_n : \{0,1\}^n \to \{0,1\}^n$;
2. $x \mapsto F_n(x)$ is computable in polynomial time;

3. the inverse function is *not* computable in polynomial time (ie., there exists no polynomial time computable function f such that $F_n(f(F_n(x))) = F_n(x)$ for all $x \in \{0,1\}^n$ and all n).

Remark. In cryptography the inverse of a one-wayfunction is required not to be computable with *non-negligible probability for a random input* using a *randomized* algorithm and polynomial time. Therefore we call our concept "weak". The same concerns the next concept that we are going to use.

A *weak hard-core predicate* for a weak one-way function $\{F_n\}$ is a sequence of functions $\{h_n\}$ such that for all n

1. $h_n : \{0,1\}^n \to \{0,1\}$ and satisfies
$$\forall x, y \in \{0,1\}^n \ (F_n(x) = F_n(y) \to h_n(x) = h_n(y)); \tag{3}$$
2. $x \mapsto h_n(x)$ is computable in polynomial time;
3. $y \mapsto h_n(F_n^{-1}(y))$, for $y \in Rng(F_n)$, is *not* computable in polynomial time.

Lemma 7.2 ([3, 14, 7, 13, 10]) *Assuming factoring is hard, there exist a weak one-way function $\{F_n\}$ and its weak hard-core predicate $\{h_n\}$ such that*

1. *F_n and h_n can be defined in the propositional calculus so that the formalizations of the sentences (3) above have Frege proofs constructible in polynomial time;*
2. *$h_n(F_n^{-1}(y))$, for $y \in Rng(F_n)$, can be computed by a $\mathbf{P}$-uniform family of quantum circuits with bounded error.*

This lemma follows from results of the papers [3, 14, 7, 13, 10]. We shall briefly comment on how these results are applied.

The idea of the proof of 1. goes back to the paper [9], where this lemma is implicitly shown for a stronger proof system, *Extended Frege,* using the cryptosystem RSA. Constructing Frege proofs is much more involved, therefore we shall use the result of [3] where the formalization has been done in detail. Their approach is based on the Diffie-Hellman system; the corresponding one way function is defined by

$$F(N, g, a, b) = (N, g, g^a \bmod N, g^b \bmod N),$$

and the hard-core predicate they used is

$$h(N, g, a, b) = (g^{ab} \bmod N) \bmod 2,$$

where N, g, a, b are positive integers, $g, a, b < N$, $gcd(g, N) = 1$. Notice that we do not require g to be a generator of the multiplicative group $\mathbb{Z}_N^*$.

In [13] it was proven that if h can be computed in polynomial time, then one can factor integers in randomized polynomial time.

In [3] Bonet, Pitassi and Raz constructed Frege proofs of formalizations of

$$\forall N, g, a, b, c, d \ (g^a \equiv g^c \wedge g^b \equiv g^d \to g^{ab} \equiv g^{cd}),$$

where $\equiv$ is the congruence mod$N$, from which a Frege proof of (3) can easily be constructed. They formalize the following argument. Assuming $g^a \equiv g^c$ and $g^b \equiv g^d$, we have:

$$g^{ab} \equiv (g^a)^b \equiv (g^c)^b \equiv g^{cb} \equiv g^{bc} \equiv (g^b)^c \equiv (g^d)^c \equiv g^{dc} \equiv g^{cd}.$$

2. A polynomial time bounded-error quantum algorithm for the discrete logarithm modulo a prime was found in the seminal paper of P. Shor [14]. This was generalized to the discrete logarithm in any finite group

in [7], which includes the case of the discrete logarithm modulo a composite number. Using this algorithm we can compute a, b from from N, g, g^a, g^b, hence also $(g^{ab} \bmod N) \bmod 2$.

Remark. Recently, Mosca and Zalka a showed that the discrete Fourier transform can be computed exactly in quantum polynomial time, which implies that also the discrete logarithm can be computed in this way [10]. Nevertheless, they assume that a suitable unary quantum gate is constructed for each particular N. In our definition of a quantum Frege system we require the set of rules to be finite. So we *cannot* use their result to strengthen our Theorem 7.1 by saying that the quantum Frege system strongly proves $\gamma_{a,0}$, or $\gamma_{a,1}$. To do that we would have to relax the condition of the finite basis of rules in a similar fashion.

Proof of Theorem 7.1. Assume that factoring is hard and let F_n be the function from Lemma 7.2 and h_n its weak hard-core predicate. Define $G_n : \{0,1\}^n \to \{0,1\}^{n+1}$ by

$$G_n(x) =_{df} (F_n(x), h_n(x)).$$

For $b \in \{0,1\}^{n+1}$, let propositions γ_b be formalizations of the following sentences:

$$\forall x \; G_n(x) \neq b.$$

Thus γ_b is a tautology iff $b \notin Rng(G_n)$. Furthermore, for $b' \in Rng(F_n)$, exactly one of the $b'0$ and $b'1$ is in $Rng(G_n)$, hence exactly one of $\gamma_{b'0}$ and $\gamma_{b'1}$ is a tautology. So our set of tautologies will be $\gamma_{b'0}$ and $\gamma_{b'1}$ for $b' \in Rng(F_n)$. (As noted above, which of the two is a tautology cannot be decided in deterministic polynomial time, but it can be decided in polynomial time by a quantum machine.)

Lemma 7.3 *Let a such that $F_n(a) = b'$ and $i \in \{0,1\}$ such that $b = (b'i) \notin Rng(G_n)$ be given. Then one can construct in polynomial time a Frege proof of γ_b.*

Proof. We shall use Lemma 7.2 part 1. to formalize the following argument:

> *Suppose that γ_b is false, ie., $G_n(x) = b = (b'i)$. Then $F_n(x) = b'$. Since $F_n(a) = b'$, we get $h_n(x) = h_n(a)$ from (3). Whence $G_n(x) = G_n(a) \neq b$. Contradiction.*

The formalization of this proof presents no problem. The only thing that we should comment on is the application of (3). As we do not have the substitution rule in Frege systems, we cannot obtain

$$F_n(x) = F_n(a) \to h_n(x) = h_n(a)$$

by substituting a in the formula (3). Instead we must substitute the bits of a (more precisely, the truth constants representing these bits) for the propositional variables representing y *in the entire proof* of (3).

■

Lemma 7.4 *Given $b' \in Rng(F_n)$, it is possible to construct in polynomial time a quantum Frege proof P which proves either $\gamma_{b'0}$ or $\gamma_{b'1}$.*

Proof. The first part P_1 of the quantum proof P will encode a quantum computation of a from b' such that $F_n(a) = b'$, which is possible because the discrete logarithm can be computed in polynomial time using quantum circuits. Then it will simulate the computation of an encoding of the proof constructed in Lemma 7.3. This proof is a classical Frege proof, but we shall represent it as an oblivious quantum proof $(D_1, \ldots, D_m)$. It will be encoded, as all other bits of the computation, by a string of tautologies τ_0 and τ_1.

The second part P_2 of P will be a sequence of instructions that will actually produce the proposition that we need, $\gamma_{b'0}$ or $\gamma_{b'1}$, from the string of tautologies τ_0 and τ_1 that encode $(D_1, \ldots, D_m)$. We shall assume that each rule D_j is represented in P_1 by a string of length d of tautologies τ_0 and τ_1. The instructions in P_2 will use one "universal" rule U controlled by these strings of τ_0 and τ_1. Suppose w.l.o.g. that all D_j are

k-ary rules. Then U is the $(k+d)$-ary rule defined as follows. If $\tau_{i_1}, \ldots, \tau_{i_d}$ is the string coding a rule D, and D transforms a string of formulas s to a string of formulas t, then U transforms

$$\tau_{i_1}, \ldots, \tau_{i_d}, s \;\rightarrow\; \tau_{i_1}, \ldots, \tau_{i_d}, t.$$

(We simulate a classical proof, so the transitions will be with amplitudes 0 or 1.)

The registers in which we keep formulas will also be divided into two blocks. In the first block we shall do the simulation of the quantum computation and then keep the the formulas that code $(D_1, \ldots, D_m)$. In the second part we shall simulate the Frege proof using m applications of the rule U. As we do not have to construct an oblivious quantum proof, we can use the rich format of instructions. Hence the j-th instruction of P_2 will specify that U is applied to the string of d formulas in the first block that codes the rule D_j and to a string of k formulas in the second block; since $(D_1, \ldots, D_m)$ is oblivious, the positions of the k formulas are determined by the index j.

Thus the first part of the circuit defined by the quantum proof P produces a linear superposition of strings of formulas in which an encoding of the classical proof of $\gamma_{b'0}$ or $\gamma_{b'1}$ occurs with an amplitude α, $|\alpha|^2 \geq 1/2$. The second part of the circuit is a deterministic reversible circuit, hence $\gamma_{b'0}$ or $\gamma_{b'1}$ will occur in the last proof line also with the amplitude α.

We note that the number of quantum deduction rules that we need is a fixed constant. This constant is determined only by the number of gates of quantum circuits that we simulate in the first part. Apart from those, we need the rule U and a few rules to make the system complete. Hence the constructed quantum proofs are proofs in some fixed quantum proof system. ■

Now we can finish the proof of Theorem 7.1. By Lemma 7.4, given $b' \in Rng(F_n)$, we can construct a quantum Frege proof of $\gamma_{b'0}$ or $\gamma_{b'1}$ in polynomial time from b'. The propositions $\gamma_{b'0}, \gamma_{b'1}$ are also constructible in polynomial time from b'. By Lemma 7.2, h_n is a hard-core predicate, hence it is not possible to decide in polynomial time which of the two propositions $\gamma_{b'0}, \gamma_{b'1}$ is a tautology. As noted earlier, this implies that we cannot polynomially simulate the quantum proofs using a classical proof system. ■

A better example?

The theorem above is not quite satisfactory, since it is based on the hardness of the decision problem whether a quantum Frege proof proves one proposition or the other. We feel that extracting classical proofs from quantum Frege proofs is hard for some deeper reasons. Therefore we would like to show that even if we have a set of quantum Frege proofs and a set of tautologies for which it is decidable in polynomial time which proof proves which tautology, we still may not be able to extract classical proofs from the quantum proofs. This can be formalized as follows.

Problem 1 *Let C be a set of pairs (P, ϕ), with P a quantum Frege proof and ϕ a tautology such that P proves ϕ. Suppose C is decidable in deterministic polynomial time (ie., $C \in \mathbf{P}$). Is it possible to polynomially simulate C by classical Frege proofs? Ie., does there exist a polynomial time computable function f such that for all $(P, \phi) \in C$, $f(P, \phi)$ is a Frege proof of ϕ ?*

Let us remark that now it is essential that we only ask about polynomial simulations by classical *Frege proof systems.* There are surely classical proof systems that polynomially simulate C (it suffices to take any proof system and extend it by proclaiming every $(P, \phi) \in C$ to be a proof of ϕ).

We conjecture that the answer to the problem above is negative. However, it seems impossible to prove it using only assumptions such as that factoring integers is hard, because the problem is much more connected with Frege proofs. Below we shall describe a specific candidate for a set C.

Let $\pi_n(x, y, z)$ be a proposition formalizing the statement *"n is a prime number"*. A natural formalization of this sentence is obtained as follows. Take a boolean circuit $c(x, y)$ for multiplication of numbers presented in binary. Then introduce variables z for the values of gates in order to transform $c(x, y) = n$ into a formula. Finally add formulas expressing that $x > 1$ and $y > 1$.

If n is prime, then $\pi_n(x, y, z)$ is a tautology. We do not know how to construct a polynomial size Frege proofs of these tautologies, so they cannot be used for our purpose. Instead we shall use propositions $\nu_{n,m}$ that formalize the following statements for n composite:

u is not a Frege proof of π_n

where u represents a string of length m. More formally, let $Prf_{\phi,m}(u)$ be a natural formalization of the predicate that u is a Frege proof of ϕ (we have been assuming that one classical Frege proof system has been fixed). Then

$$\nu_{n,m}(u) =_{df} \neg Prf_{\pi_n,m}(u)$$

The set of these tautologies is clearly in **P**.

Lemma 7.5 *Having a factorization of n one can construct a Frege proof of $\nu_{n,m}$ in polynomial time.*

Proof-sketch. Recall that it is possible to construct sentences expressing the reflection principle for a Frege system and their Frege proofs in polynomial time. This was essentially shown in [4]. In that paper Buss only considered the consistency of Frege systems, but he was aware of the fact that proofs of the tautologies expressing the reflection principle can easily be obtained using his results.

The reflection principle says that if we are given a proof of a proposition, then it is a tautology; formally

$$Prf_{\phi,m}(u) \to \phi.$$

Hence the contrapositive sentence says: if we are given a falsifying assignment to ϕ, the string u that should represent a proof is not an actual proof. Taking π_n for ϕ we have

$$\neg\pi_n(x, y, z) \to \neg Prf_{\phi_n,m}(u).$$

A falsifying assignment for $\pi_n(x, y, z)$ are factors of n, ie., a, b such that $ab = n$ and the bits of the computation the product ab. Thus we can obtain a proof of $\nu_{n,m}$ by substituting the bits of the factors and the bits of the computation of the product in the proof of the reflection principle. ■

Corollary 7.6 *Quantum Frege proofs of $\nu_{n,m}$, for n composite, can be constructed in polynomial time.*

Proof-sketch. Proceed as in the proof of Theorem 7.1. Namely, the first part of the quantum proof will be an encoding of the circuit computing a factorization. In the second part use the Frege proofs from Lemma 7.5. ■

We conjecture that classical Frege proofs of $\nu_{n,m}$, for n composite, cannot be constructed in polynomial time. The intuition is that in Frege system we have limited means and thus the only way to prove that there are no proofs of the primality of n is to provide a witness in the form of a factor of n.

Extracting classical proofs using quantum algorithms

Problem 2 *Given a quantum Frege proof P of ϕ, can one construct a Frege proof of ϕ in polynomial time using* quantum *circuits?*

For the quantum proofs of Theorem 7.1 and Corollary 7.6 it was possible. In order to construct these quantum proofs we actually used the fact that such classical proofs can be computed in polynomial time by quantum machines. Thus to answer the problem negatively, one would need to use a different approach.

This problem is a very intriguing question, since if the solution were negative, then it would open the possibility of having theorems proven by quantum devices, but for which we would not be able to construct classical proofs.

How likely is this possibility? Extracting a classical Frege proof from a quantum Frege proof is closely related to the problem of constructing a history from a given quantum for given starting and ending states.

No polynomial time bounded error quantum algorithm is known for the latter problem. One can easily see that the problem of finding a history is **QP**-hard, but we do not know much more than that. Aaronson [1] studied related problems in some hidden-parameter theories. He showed that an oracle producing samples of histories in these theories gives more power to quantum computations. This supports our feeling that the problem of extracting a classical proofs from quantum Frege proofs is hard for quantum circuits, but we do not see how to apply Aaronson's results to this problem.

8 Conclusions and open problems

In this paper we have proposed a quantum version of the concept of a Frege propositional proof system. It is based on generalizing the concept of a logical rule to a unitary transformation that can combine several classical rules. This concept needs further investigations. We do not know how robust this concept is—one should check whether there are modifications that may change the power of the system. We do not know if a universal quantum Frege proof system (in the sense of Conjecture 1) exists. Constructing such a system would clarify what is essential in the definition of quantum Frege proof systems.

We know that quantum Frege proofs are not shorter than classical Frege proofs, but, unless factoring integers is easy, we cannot produce a classical Frege proof from a quantum Frege proof in polynomial time. The problem of extracting a classical proof from a quantum proof also needs more investigation. The most interesting question is whether it is possible to efficiently extract classical proofs from quantum ones using quantum circuits (Problem 2). Solving this problem (at least conditionally, using a plausible complexity-theoretical assumption) requires better understanding of the relation of quantum proofs and quantum computations.

Acknowledgment

I would like to thank to Scott Aaronson, Sam Buss and Jan Krajíček for useful remarks about this paper.

References

[1] S. Aaronson, *Quantum Computing and Hidden Variables.* Physical Review A 71:032325, March 2005.

[2] D. Aharonov, *A simple proof that Toffoli and Hadamard are quantum universal.* arXiv.org/quant-ph/0301040.

[3] M.L. Bonet, T. Pitassi, R. Raz, *On Interpolation and Automatization for Frege Systems.* SIAM Journal of Computing 29(6) (2000), 1939-1967.

[4] S.R. Buss, *Propositional consistency proofs.* Annals of Pure and Applied Logic 52 (1991) 3-29.

[5] S.A. Cook and A.R. Reckhow, *The relative efficiency of propositional proof systems.* J. of Symbolic Logic 44(1), 36-50.

[6] I. L. Chung and M. A. Nielsen, *Quantum Computing and Quantum Information.* Cambridge University Press, Cambridge, UK, 2000.

[7] D. Boneh, and R. Lipton, *Quantum cryptanalysis of hidden linear forms.* In Proc. Crypto '95, Lecture Notes in Computer Science, Vol. 963, Springer-Verlag, 1995, 424-437.

[8] J. Krajíček, *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge, UK, 1995.

[9] J. Krajíček and P. Pudlák, *Some consequences of cryptographical conjectures for S_2^1 and EF.* Information and Computation 142, (1998), 82-94

[10] M. Mosca, Ch. Zalka, *Exact quantum Fourier transforms and discrete logarithm algorithms.* quant-ph/0301093, 2003.

[11] R.Raz, A.Shpilka, *On the power of Quantum Proofs.* In Proc. Computational Complexity, 2004, 260-274.

[12] Y. Shi, *Both Toffoli and controlled-Not need little help to do universal quantum computation.* arXiv.org/quant-ph/0205115.

[13] Z. Shmuley, *Composite Diffie-Hellman public-key generating systems are hard to break.* Technical Report No. 356, Computer Science Department, Technion, Israel, 1985.

[14] P. Shor, *Polynomial time algorithms for prime factorization and discrete logarithms.* SIAM J. Comput., 26(5), 1997, 1484-1509.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/