Certified Quantum Computation in Isabelle/HOL

Anthony Bordg[1] · Hanna Lachnitt[2] · Yijun He[3]

Abstract

In this article we present an ongoing effort to formalise quantum algorithms and results in quantum information theory using the proof assistant Isabelle/HOL. Formal methods being critical for the safety and security of algorithms and protocols, we foresee their widespread use for quantum computing in the future. We have developed a large library for quantum computing in Isabelle based on a matrix representation for quantum circuits, successfully formalising the no-cloning theorem, quantum teleportation, Deutsch's algorithm, the Deutsch–Jozsa algorithm and the quantum Prisoner's Dilemma. We discuss the design choices made and report on an outcome of our work in the field of quantum game theory.

Keywords Isabelle/HOL · Certification · Quantum computing · No-cloning · Quantum teleportation · Deutsch's algorithm · Deutsch–Jozsa algorithm · Quantum prisoner's dilemma

Mathematics Subject Classification 03B35 · 03B15 · 81P68 · 68Q12

1 Introduction

On January 4th 2017 the computer scientist László Babai retracted a claim he made in a preprint[1] after the mathematician Andrés Helfgott spotted an error in his work. Back in 2015, Babai's result, dealing with the so-called graph isomorphism problem, a central problem in the field of computer algorithms, was dubbed "the theoretical computer science advance of the decade". In the meantime Helfgott had spent months reviewing Babai's algorithm in

---

[1] https://arxiv.org/abs/1512.03547

✉ Anthony Bordg
apdb3@cam.ac.uk

Hanna Lachnitt
lachnitt@stanford.edu

[1] Department of Computer Science and Technology, University of Cambridge, Cambridge, UK

[2] Computer Science Department, Stanford University, Stanford, USA

[3] University of Cambridge, Cambridge, UK

preparation for a talk at the Bourbaki seminar to report on Babai's major result. On January 9th 2017 in a new (last?) unexpected twist, Babai announced a fix for his error and he restored his claim. It seems that Helfgott is confident that the fix is correct, but Babai's paper is still unpublished as of September 23, 2020.

This story exemplifies the difficulty making sure that an algorithm obeys its specification. This challenge might be even harder with respect to quantum algorithms, since our intuition weakens when one moves from the classical world to the quantum realm. Fortunately, formal methods can help with the task of certifying that a quantum algorithm obeys its specification. Recent work in that direction include the formalisation of Grover's algorithm in Isabelle by Liu *et al.* [1]. However, for their formalisation of Grover's algorithm the authors use a tailored quantum Hoare logic. One should also mention the QWIRE project by Rand *et al.* [2] for quantum circuits, although the authors use a different approach since their work is an embedding of the QWIRE quantum circuit language in the proof assistant Coq to formally prove properties of those circuits. The closest to the present work is the work of Boender *et al.* [3] which culminates in the formalisation of the quantum teleportation protocol using the proof assistant Coq, this algorithm becoming the de facto benchmark in the field. This benchmark is successfully reached and surpassed in our work.

In this paper we present a large formalisation of results in quantum computation and quantum information theory developed in the proof assistant Isabelle. Our library[2] includes the quantum teleportation protocol, the no-cloning theorem, Deutsch's algorithm, the Deutsch–Jozsa algorithm and the quantum Prisoner's Dilemma. To the best of our knowledge a formalisation of these last four classic results has never been done before. We start with the basics of quantum computing in Sects. 2, 3 and 4 . We then introduce the aforementioned results formalised in the library in Sect. 5. Throughout the article we discuss the design choices made. Finally, we outline an unexpected outcome of our formalisation in Sect. 5.4.

2 Qubits and Quantum States

In the classical model of computation the bit is the fundamental unit of information. There are two classical states for a bit, namely 0 and 1. In quantum computing the bit is superseded by the quantum bit, the so-called *qubit*, which becomes the fundamental unit of information. In the same way, any qubit has a (quantum) state, but the situation is more involved.

For the sake of simplicity, let us start with a 1-qubit system. In that case, the quantum state of our qubit is a normalised vector in a 2-dimensional complex vector space. Using the Dirac notation introduced in quantum mechanics, a column vector in that space is denoted by $|\psi\rangle$, where ψ is a mere label, and the vector $|\psi\rangle$ is called a *ket*. In that context the two elements of the computational basis, namely

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

are denoted by $|0\rangle$ and $|1\rangle$, respectively. Actually, the label for the *n*th element of the computational basis corresponds to the binary expression of *n*, hence $|0\rangle$ should not be confused with the zero vector, namely

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

```
locale state =
  fixes n:: nat and v:: "complex mat"
  assumes is_column [simp]: "dim_col v = 1"
    and dim_row [simp]: "dim_row v = 2^n"
    and is_normal [simp]: "‖col v 0‖ = 1"
```

Fig. 1 The locale *state* in *Quantum.thy*

```
definition state_qbit :: "nat ⇒ complex vec set" where
"state_qbit n ≡ {v| v:: complex vec. dim_vec v = 2^n ∧ ‖v‖ = 1}"
```

Fig. 2 The definition of the set *state_qbit* in *Quantum.thy*

```
definition ket_vec :: "complex vec ⇒ complex mat" ("|_⟩") where
"|v⟩ ≡ mat (dim_vec v) 1 (λ(i,j). v $ i)"
```

Fig. 3 Dirac's ket notation

The zero vector being not normalised, it is not a quantum state.

So, in the computational basis a state of our qubit is a linear combination $\alpha_0 \, |0\rangle + \alpha_1 \, |1\rangle$ such that α_0 and α_1 are complex numbers satisfying the normalisation constraint

$$|\alpha_0|^2 + |\alpha_1|^2 = 1.$$

In the quantum world, the coefficients α_0 and α_1 are called *amplitudes*, and one sometimes uses the word *superposition* instead of the phrase *linear combination*.

For a 2-qubit system the state of a qubit becomes a normalised vector in a 4-dimensional complex vector space. If $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$ denote the elements of the computational basis, then such a state is a superposition

$$\alpha_{00} \, |00\rangle + \alpha_{01} \, |01\rangle + \alpha_{10} \, |10\rangle + \alpha_{11} \, |11\rangle \, ,$$

with $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$.

In order to model the quantum states of qubits we exploit Isabelle's system module for dealing with a hierarchy of parametric theories, the so-called *locales* [4]. In our library the locale *state* provides the context for talking about the quantum states of a n-qubit system (Fig. 1).

In this locale v is a complex matrix, but the condition *is_column* ensures it is a column matrix, *i.e.* a column vector. We choose to model quantum states by column matrices instead of vectors, since this design choice will come in handy when applying quantum gates to quantum states (more on that later). The condition *dim_row* relates to the dimension of the ambient vector space and the condition *is_normal* provides the normalisation constraint. We also introduce the corresponding set of quantum states of a given dimension, where this time we can directly use complex vectors (Fig. 2).

Of course, given the context provided by the locale *state* we can prove that v (or rather the first column of v) belongs to *state_qbit n*.

```
lemma (in state) state_to_state_qbit [simp]:
  shows "col v 0 ∈ state_qbit n"
  using state_def state_qbit_def by simp
```

One can also go the other way around, *i.e.* from vectors to column matrices. We take this opportunity to introduce in our library Dirac's ket notation, since Isabelle allows some syntactic sugar (Fig. 3).

In our library we implicitly work in the computational basis, hence the amplitudes have to be understood accordingly.

```
definition dagger :: "complex mat ⇒ complex mat" ("_†") where
  "M† ≡ mat (dim_col M) (dim_row M) (λ(i,j). cnj(M $$ (j,i)))"

definition unitary :: "complex mat ⇒ bool" where
  "unitary M ≡ (M†) * M = 1ₘ (dim_col M) ∧ M * (M†) = 1ₘ (dim_row M)"
```

Fig. 4 The definitions of the Hermitian conjugate of a matrix and the *unitary* predicate, respectively (see *Quantum.thy*)

Fig. 5 The locale *gate* in *Quantum.thy*

```
locale gate =
  fixes n:: nat and A:: "complex mat"
  assumes dim_row [simp]: "dim_row A = 2^n"
    and square_mat [simp]: "square_mat A"
    and unitary [simp]: "unitary A"
```

```
lemma unitary_is_length_preserving [simp]:
  fixes U:: "complex mat" and v:: "complex vec"
  assumes "unitary U" and "dim_vec v = dim_col U"
  shows "‖U * |v⟩‖ = ‖v‖"
  using assms unitary_is_sq_length_preserving
  by (metis cpx_vec_length_inner_prod inner_prod_csqrt of_real_hom.injectivity)
```

Fig. 6 The statement and the proof that unitary matrices preserve length

3 Quantum Gates

Like their classical counterparts, quantum gates are used to manipulate information. More exactly, quantum gates are ways of manipulating the quantum states of qubits. Usually there are two kinds of representations for quantum gates, namely circuit representations and matrix representations. Since it is not possible to directly work in Isabelle with circuits, we choose in that context the more convenient matrix representations. In this process we take advantage of the nice library for matrices developed by Thiemann and Yamada [5]. Then, a quantum state $|\psi\rangle$ being in particular a column matrix, the action of a quantum gate U on $|\psi\rangle$ is simply given by the matrix multiplication $U |\psi\rangle$, denoted $U * |\psi\rangle$ in Isabelle.

However, not every matrix is a quantum gate. Quantum gates belong to a specific class of complex matrices. Actually, given a n-qubit system the quantum gates are exactly the $2^n \times 2^n$ matrices that are unitary. In order to explain what *unitary* means, we need to introduce the Hermitian conjugate of a complex matrix. Let U be a complex matrix, its Hermitian conjugate $U^\dagger$ is the complex conjugate of its transpose, namely $(U^t)^*$. In different contexts people use different notations for the Hermitian conjugate, but the dagger operator is commonly used in quantum mechanics, and we keep this notation in the library. A complex square matrix U is said to be unitary if $U^\dagger U = U U^\dagger = I$, *i.e.* if its inverse is given by its Hermitian conjugate (Fig. 4).

In Isabelle we encapsulate the definition of a quantum gate inside a dedicated locale (Fig. 5).

What is the idea behind unitarity? Unitary matrices are length-preserving. One has $\|U |v\rangle\| = \||v\rangle\|$ for every unitary matrix U and every ket $|v\rangle$ such that their multiplication is well defined. Given the normalisation constraint in quantum states, it is no wonder quantum gates should be unitary matrices (Figs. 6 and 7).

Actually, unitary matrices are the only matrices that preserve length. To prove this result one needs the following key lemma.

$$(M |\psi\rangle)^\dagger = \langle\psi| M^\dagger,$$

```
definition bra :: "complex mat ⇒ complex mat" ("⟨_|") where
"⟨v| ≡ mat 1 (dim_row v) (λ(i,j). cnj(v $$ (j,i)))"

lemma dagger_of_mat_on_ket:
  fixes v:: "complex vec" and A :: "complex mat"
  assumes "dim_col A = dim_vec v"
  shows "(A * |v⟩ )† = ⟨v| * (A†)"
  using assms by (metis bra_mat_on_vec dagger_of_ket_is_bra mat_on_ket)
```

Fig. 7 A key lemma to prove that length-preserving matrices are unitary

```
lemma length_preserving_is_unitary:
  fixes U:: "complex mat"
  assumes "square_mat U" and "∀v::complex vec. dim_vec v = dim_col U ⟶ ‖U * |v⟩‖ = ‖v‖"
  shows "unitary U"
```

Fig. 8 Length-preserving matrices are unitary

```
definition H ::"complex mat" where
"H ≡ 1/sqrt(2) ·m (mat 2 2 (λ(i,j). if i≠j then 1 else (if i=0 then 1 else -1)))"
```

Fig. 9 The definition of the Hadamard gate in *Quantum.thy*

Fig. 10 The formal proof in Isabelle that H is a gate and is self-adjoint

```
lemma H_is_gate [simp]:
  "gate 1 H"
  apply(simp add: gate_def unitary_def)
  apply(simp add: H_def)
  done
lemma dagger_of_H [simp]:
  "H† = H"
  using dagger_def by (auto simp: H_def cong_mat)
```

where $\langle\psi|$ is Dirac's *bra* notation, namely if $|\psi\rangle$ is the column vector

$$\begin{pmatrix} a_1 \\ \vdots \\ a_{2^n} \end{pmatrix},$$

then its bra is the corresponding row vector with conjugate coefficients

$$\begin{pmatrix} a_1^* \cdots a_{2^n}^* \end{pmatrix}.$$

Using this lemma and the many results on the dagger operator and unitary matrices provided in the library, one can eventually prove that length-preserving matrices are unitary (Fig. 8).

Now, we introduce our first quantum gate, namely the Hadamard gate H (Fig. 9). It is a single-qubit gate, *i.e.* a 2×2 unitary matrix,

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

One can easily check that H is unitary and self-adjoint, *i.e.* $H^\dagger = H$ (Fig. 10). The action of H on the basis elements is given as follows.

$$|0\rangle \mapsto \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle)$$

$$|1\rangle \mapsto \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle)$$

As a consequence, H maps a state $\alpha_0 |0\rangle + \alpha_1 |1\rangle$ to $\frac{\alpha_0 + \alpha_1}{\sqrt{2}} |0\rangle + \frac{\alpha_0 - \alpha_1}{\sqrt{2}} |1\rangle$.

Fig. 11 The formal proof that cNOT is a 2-qubit gate

```
definition CNOT ::"complex mat" where
"CNOT ≡ mat 4 4
  (λ(i,j). if i=0 ∧ j=0 then 1 else
   (if i=1 ∧ j=1 then 1 else
    (if i=2 ∧ j=3 then 1 else
     (if i=3 ∧ j=2 then 1 else 0))))"

lemma CNOT_is_gate [simp]:
  "gate 2 CNOT"
  apply(simp add: gate_def unitary_def)
  apply(simp add: CNOT_def)
  done
```

Since H creates a superposition, it is truly a quantum gate. Maybe somewhat puzzling for the beginner is the fact that H is sometimes described as a "square-root of NOT" gate. One simply means that it turns $|0\rangle$ (*resp.* $|1\rangle$) into $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ (*resp.* $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$), so "halfway" between $|0\rangle$ and $|1\rangle$.

Before introducing our first example of a 2-qubit gate, we need to say a few words on the initial states of a 2-qubit system. Actually, such states are given by the tensor product of the states of each qubit. For instance, if the first qubit is in the initial state $|0\rangle = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ and the second one is in the initial state $|1\rangle = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$, then the initial state of the combined system is

$$|0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix}$$
$$= \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$
$$= |01\rangle.$$

Now, an interesting 2-qubit quantum gate is the controlled-NOT gate (cNOT). Its matrix representation is given by

$$cNOT = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix},$$

and one easily checks that the cNOT gate is unitary and it is again self-adjoint (Fig. 11).

The cNOT gate maps the basis elements $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ to $|00\rangle, |01\rangle, |11\rangle, |10\rangle$, respectively. In other words, the cNOT gate flips the second qubit (the so-called target qubit) if the first qubit (the so-called control qubit) is 1 and does nothing otherwise. One summarizes the action of the cNOT gate with the following handy piece of notation

$$|xy\rangle \mapsto |x \ x \oplus y\rangle,$$

where $\oplus$ denotes the addition modulo 2.

The cNOT gate can be used to perform non-classical computations. For instance, starting with the $|00\rangle$ state and applying a Hadamard gate to the first qubit followed by a cNOT gate, one creates the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, which is a highly non-classical state, a so-called Bell's state (more on that later).

To put everything together, let us assume that we have a 3-qubit system. Moreover, assume that we want to apply an Hadamard gate to the first qubit and a cNOT gate to the second and third qubits. The initial state of the combined system is given by $|x\rangle \otimes |y\rangle \otimes |z\rangle$ which is a 8-dimensional column vector with $|x\rangle$ (*resp.* $|y\rangle, |z\rangle$) denoting the initial state of the first (*resp.* second, third) qubit. Since the tensor product is associative, we omit the parentheses

```
lemma tensor_gate [simp]:
  assumes "gate m G1" and "gate n G2"
  shows "gate (m + n) (G1 ⊗ G2)"
proof
  show "dim_row (G1 ⊗ G2) = 2^(m+n)"
    using assms dim_row_tensor_mat gate.dim_row by (simp add: power_add)
  show "square_mat (G1 ⊗ G2)"
    using assms gate.square_mat by simp
  thus "unitary (G1 ⊗ G2)"
    using assms unitary_def by simp
qed
```

Fig. 12 The Kronecker product of two quantum gates is a quantum gate (see *More_Tensor.thy*)

```
definition bell00 ::"complex mat" ("|β₀₀⟩") where
"bell00 ≡ 1/sqrt(2) ·ₘ |vec 4 (λi. if i=0 ∨ i=3 then 1 else 0)⟩"

definition bell01 ::"complex mat" ("|β₀₁⟩") where
"bell01 ≡ 1/sqrt(2) ·ₘ |vec 4 (λi. if i=1 ∨ i=2 then 1 else 0)⟩"

definition bell10 ::"complex mat" ("|β₁₀⟩") where
"bell10 ≡ 1/sqrt(2) ·ₘ |vec 4 (λi. if i=0 then 1 else if i=3 then -1 else 0)⟩"

definition bell11 ::"complex mat" ("|β₁₁⟩") where
"bell11 ≡ 1/sqrt(2) ·ₘ |vec 4 (λi. if i=1 then 1 else if i=2 then -1 else 0)⟩"
```

Fig. 13 The Bell's states (see *Quantum.thy*)

in $|x\rangle \otimes |y\rangle \otimes |z\rangle$. Then one can sum up the two gate applications using only one 8×8 matrix, namely $H \otimes cNOT$, where $\otimes$ denotes the *Kronecker product* between two matrices. With this in mind we needed to formalize the Kronecker product in our library and proved that the Kronecker product of two gates is a gate as shown in the snippet of code above. This essentially amounts to proving that the Kronecker product of two unitary matrices is a unitary matrix (Fig. 12).

At that point we faced a design choice connected to the important issue of legacy code in formal mathematics. Indeed, there is already a formalisation of the Kronecker product in [6] but for a legacy notion of matrix which is not the one developed in [5] and used in our library. So, we could either restart the formalisation of the Kronecker product from scratch or we could build a bridge between the two formalisations of matrices available and reuse as much as possible the code in [6]. We chose the latter, using the code available as a convenient scaffolding (*cf.* our theory *Tensor.thy*). This choice may ease in the future the reuse of formalisations based on legacy matrices.

We come back to the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ obtained as the result of the application of the Hadamard gate followed by the cNOT gate to the state $|00\rangle$. Actually, this state is part of a set of four states known as the *Bell's states* (Fig. 13) or sometimes the *EPR states* (EPR stands for Einstein, Podolsky and Rosen):

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$|\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) .$$

```
definition prod_state:: "nat ⇒ complex Matrix.mat ⇒ bool" where
"prod_state m u ≡ if state m u then ∃ p::nat.∃v w. state n v ∧ state p w ∧
   n < m ∧ p < m ∧ u = v ⊗ w else undefined"
definition entangled:: "nat ⇒ complex Matrix.mat ⇒ bool" where
"entangled n v ≡ ¬ (prod_state n v)"
```

Fig. 14 The property of being entangled (see *Entanglement.thy*)

Fig. 15 The proof that $|\beta_{00}\rangle$ is entangled (see *Entanglement.thy*)

```
lemma bell00_is_entangled2 [simp]:
"entangled2 |β₀₀⟩"
```

The peculiarity of these states resides in the fact that they cannot be written as the tensor product of two 1-qubit states. These states are said to be *entangled*. *Entanglement*, one of the key concepts in quantum mechanics, is simply the fact that not every state is a tensor product of smaller states (Fig. 14).

In the case of the Bell's state $|\beta_{00}\rangle$ for instance it is very easy to prove that it cannot be written as $(\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\alpha_0'|0\rangle + \alpha_1'|1\rangle)$ using the distributivity of the tensor product (Fig. 15).

Finally, in the library many other quantum gates are introduced like the Pauli matrices X, Y and Z, the phase gate S, the T gate[3].

4 Measurements

Given a n-qubit system, the state of a qubit involves 2^n amplitudes linked by a normalisation constraint. Can we determine those amplitudes? For instance take $n = 1$, a quantum state has the form $\alpha_0|0\rangle + \alpha_1|1\rangle$ with $|\alpha_0|^2 + |\alpha_1|^2 = 1$. Can we determine α_0 and α_1? The answer is no. A quantum state cannot be directly observed and the amplitudes cannot be directly determined.

Actually, the outcome of any measurement of our qubit through an apparatus is a classical bit of information. Moreover, the measurement will disturb the state of the qubit. Indeed, the outcome will be either 0 with probability $|\alpha_0|^2$ or 1 with probability $|\alpha_1|^2$. The sum of the probabilities should be 1, hence the normalisation constraint of quantum states. Moreover, if the outcome happens to be 0 (*resp.* 1), then the post-measurement state is $|0\rangle$ (*resp.* $|1\rangle$) and the amplitudes vanish.

The generalisation to a multiple qubits system is straightforward. For a system of two qubits, assuming the state of our system is

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \,,$$

one has

$$pr(00) = |\alpha_{00}|^2$$
$$pr(01) = |\alpha_{01}|^2$$
$$pr(10) = |\alpha_{10}|^2$$
$$pr(11) = |\alpha_{11}|^2,$$

where pr(00) (*resp.* pr(01), pr(10), pr(11)) denotes the probability of the outcome being 0 for both qubits (*resp.* 0 for the first one and 1 for the second one, 1 for the first one and 0 for the second one, 1 for both qubits), and the post-measurement state is $|00\rangle$ (*resp.* $|01\rangle$, $|10\rangle$, $|11\rangle$).

---

[3] See the theory *Quantum* of the library, especially the subsection entitled A *Few Well-Known Quantum Gates*.

```
definition select_index ::"nat ⇒ nat ⇒ nat ⇒ bool" where
"select_index n i j ≡ (i≤n-1) ∧ (j≤2^n - 1) ∧ (j mod 2^(n-i) ≥ 2^(n-1-i))"
```

Fig. 16 The *select_index* predicate (see *Measurement.thy*)

```
definition prob0 ::"nat ⇒ complex mat ⇒ nat ⇒ real" where
"prob0 n v i ≡ ∑j∈{k| k::nat. (k<2^n) ∧ ¬ select_index n i k}. (cmod(v $$ (j,0)))²"

definition prob1 ::"nat ⇒ complex mat ⇒ nat ⇒ real" where
"prob1 n v i ≡ ∑j∈{k| k::nat. select_index n i k}. (cmod(v $$ (j,0)))²"
```

Fig. 17 Computing the probabilities of outcomes (see *Measurement.thy*)

Now, what does happen if one has a 2-qubit system and one makes a partial measurement, *i.e.* one measures the first qubit for instance (but not the second one)? What are the probabilities pr(0) and pr(1) of the outcome being 0 and 1, respectively? One has

$$pr(0) = pr(00) + pr(01) = |\alpha_{00}|^2 + |\alpha_{01}|^2$$
$$pr(1) = pr(10) + pr(11) = |\alpha_{10}|^2 + |\alpha_{11}|^2.$$

In other words, we sum over the probabilities of measuring the whole system and getting 0 (*resp.* 1) for the first qubit. What is the post-measurement state of the system? To get the answer we first rewrite

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

as

$$|0\rangle \otimes (\alpha_{00}|0\rangle + \alpha_{01}|1\rangle) + |1\rangle \otimes (\alpha_{10}|0\rangle + \alpha_{11}|1\rangle).$$

If the outcome of measuring only the first qubit happens to be 0 (*resp.* 1), then the post-measurement state of the system is

$$|0\rangle \otimes \frac{\alpha_{00}|0\rangle + \alpha_{01}|1\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}} \quad (resp. \ |1\rangle \otimes \frac{\alpha_{10}|0\rangle + \alpha_{11}|1\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}).$$

In particular, the state of the second qubit after measuring 0 (*resp.* 1) for the first qubit is

$$\frac{\alpha_{00}|0\rangle + \alpha_{01}|1\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}} \quad (resp. \ \frac{\alpha_{10}|0\rangle + \alpha_{11}|1\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}}),$$

namely the normalised vector of $\alpha_{00}|0\rangle + \alpha_{01}|1\rangle$ (*resp.* $\alpha_{10}|0\rangle + \alpha_{11}|1\rangle$).

To translate measurements in Isabelle we first need a predicate *select_index* such that *select_index n i j* outputs true if the *j*th element of the computational basis has a 1 at the *i*th spot of its label and false otherwise (Fig. 16).

Then given the state of a *n*-qubit system, we can compute the probability[4] of the outcome being 0 (*resp.* 1) when measuring the *i*th qubit (Fig. 17).

If the outcome of measuring the *i*th qubit is 0 (*resp.* 1), then *post_meas0* (*resp. post_meas1*) gives the new state of the system (Fig. 18).

Entanglement has some interesting consequences with respect to measurement. In quantum mechanics measurements of physical properties, such as momentum, position or spin, on entangled particles are perfectly correlated. In quantum computing this phenomenon can be illustrated through the Bell states. Given a Bell state, if one makes one measurement, then one gets either 0 with probability 1/2 or 1 with probability 1/2 whatever the qubit being measured (either the first or the second one). Moreover, in the case of two successive

---

[4] We do not use any of the probability theory developed in Isabelle, we use ad hoc definitions instead.

```
definition post_meas0 ::"nat ⇒ complex mat ⇒ nat ⇒ complex mat" where
"post_meas0 n v i ≡
of_real(1/sqrt(prob0 n v i)) ·ₘ |vec (2^n) (λj. if ¬ select_index n i j then v $$ (j,0) else 0)⟩"

definition post_meas1 ::"nat ⇒ complex mat ⇒ nat ⇒ complex mat" where
"post_meas1 n v i ≡
of_real(1/sqrt(prob1 n v i)) ·ₘ |vec (2^n) (λj. if select_index n i j then v $$ (j,0) else 0)⟩"
```

Fig. 18 The new states of the system after outcome 0 and 1, respectively (see *Measurement.thy*)

```
locale quantum_machine =
  fixes n:: nat and s:: "complex Matrix.vec" and U:: "complex Matrix.mat"
  assumes dim_vec [simp]: "dim_vec s = 2^n"
    and dim_col [simp]: "dim_col U = 2^n * 2^n"
    and square [simp]: "square_mat U" and unitary [simp]: "unitary U"
```

Fig. 19 A quantum machine in Isabelle (see *No_Cloning.thy*)

```
definition inner_prod :: "complex vec ⇒ complex vec ⇒ complex" ("⟨_|_⟩") where
"inner_prod u v ≡ ∑ i ∈ {0..< dim_vec v}. cnj(u $ i) * (v $ i)"
```

Fig. 20 The inner product of two complex vectors (see *Quantum.thy*)

measurements of the first and second qubit, the outcomes are correlated. Indeed, in the case of $|\beta_{00}\rangle$ or $|\beta_{10}\rangle$ (*resp.* $|\beta_{01}\rangle$ or $|\beta_{11}\rangle$)) if one measures the second qubit after a measurement of the first qubit (or the other way around) then one gets the same outcomes (*resp.* opposite outcomes), *i.e.* the probability of measuring 0 for the second qubit after a measurement with outcome 0 for the first qubit is 1 (*resp.* 0).

5 Theorems and Quantum Algorithms

We present briefly the main theorems and algorithms formalized in the library. For a detailed presentation the reader is invited to consult a standard reference like [7].

5.1 The No-Cloning Theorem

A notable theorem in quantum computation and quantum information is the so-called *no-cloning theorem* articulated by Wootters and Zurek [8] and by Dieks [9]. It is one of the earliest results in the field. Roughly, the no-cloning theorem states it is impossible to make an exact copy of an unknown quantum state. Since classical information can be copied exactly, this no-go theorem[5] is one of the main differences between classical and quantum information. More precisely, given two non-orthogonal quantum states $|\phi\rangle$ and $|\psi\rangle$, there does not exist a quantum device that, when input with $|\phi\rangle$ (*resp.* $|\psi\rangle$), outputs $|\phi\rangle \otimes |\phi\rangle$ (*resp.* $|\psi\rangle \otimes |\psi\rangle$). First, we use Isabelle's locale mechanism to define a quantum machine. A quantum machine consists of a natural number n, a complex vector s, and a complex matrix U, plus the assumptions that s has dimension 2^n and U is a $2^{2n} \times 2^{2n}$ unitary matrix (Fig. 19).

Second, we need to introduce the inner product $\langle v|w\rangle$ of two complex vectors v, w (Fig. 20).

Recall that for every complex vector v one has $\|v\|^2 = \langle v|v\rangle$ (Fig. 21), and two complex vectors v, w being orthogonal means their inner product $\langle v|w\rangle$ is 0.

Thus, in Isabelle the no-cloning theorem is formalised as follows (Fig. 22).

---

[5] In physics a no-go theorem states that a particular situation is physically impossible.

```
lemma cpx_vec_length_inner_prod [simp]:
  "‖v‖² = ⟨v|v⟩"
proof -
  have "‖v‖² = (∑i<dim_vec v. (cmod (v $ i))²)"
    using cpx_vec_length_def complex_of_real_def
    by (metis (no_types, lifting) real_sqrt_power real_sqrt_unique sum_nonneg zero_le_power2)
  also have "... = (∑i<dim_vec v. cnj (v $ i) * (v $ i))"
    using complex_norm_square mult.commute by (smt of_real_sum sum.cong)
  finally show ?thesis
    using inner_prod_def by (simp add: lessThan_atLeast0)
qed
```

Fig. 21 The squared length of a complex vector is equal to its inner product with itself (see *Quantum.thy*)

```
theorem (in quantum_machine) no_cloning:
  assumes [simp]: "dim_vec v = 2^n" and [simp]: "dim_vec w = 2^n" and
    cloning1: "⋀s. U * ( |v⟩ ⊗ |s⟩) = |v⟩ ⊗ |v⟩" and
    cloning2: "⋀s. U * ( |w⟩ ⊗ |s⟩) = |w⟩ ⊗ |w⟩" and
    "⟨v|v⟩ = 1" and "⟨w|w⟩ = 1"
  shows "v = w ∨ ⟨v|w⟩ = 0"
```

Fig. 22 The no-cloning theorem (see *No_Cloning.thy*)

Fig. 23 The Cauchy-Schwarz inequality (see *No_Cloning.thy*)

```
lemma cauchy_schwarz_ineq:
  assumes "dim_vec v = dim_vec w"
  shows "(cmod(⟨v|w⟩))² ≤ Re (⟨v|v⟩ * ⟨w|w⟩)"
```

In other words, if someone has built a quantum machine which is able to copy two quantum states (*i.e.* two normalised complex vectors), then these two states are either identical or orthogonal. The proof relies on the Cauchy-Schwarz inequality:

$$|\langle v|w\rangle|^2 \leq \langle v|v\rangle\langle w|w\rangle$$

for every complex vectors v and w (Fig. 23).

In the snippet above one needs to take the real part of $\langle v|v\rangle\langle w|w\rangle$, since Isabelle is not able to notice immediately that $\langle v|v\rangle\langle w|w\rangle$ is a real number and so the real part is required for type-checking.
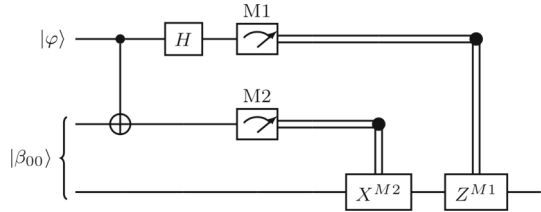
5.2 Quantum Teleportation

The quantum teleportation protocol has already been formalised with the proof assistant Coq [3]. We follow closely this previous formalisation to give a counterpart in Isabelle.

First, we outline the protocol introduced in the seminal paper of Bennett *et al.* [10]. The quantum teleportation allows the transmission of an unknown quantum state between a sender and a receiver in the absence of a quantum channel using only an entangled pair and a classical channel. Let us assume that Alice in London wants to send Bob in Paris an unknown quantum state $|\varphi\rangle$. By sharing an EPR pair, each taking one qubit of the EPR pair, this feat can be achieved. Indeed, Alice can take the tensor product of $|\varphi\rangle$ with her half of the EPR pair to apply a cNOT gate (using $|\varphi\rangle$ as the control qubit) and then apply an Hadamard gate on $|\varphi\rangle$. Finally, she measures her two qubits, obtaining one of the four possible results: 00, 01, 10 or 11. She sends these two classical bits to Bob using the classical channel at her disposal. Depending on Alice's two bits, Bob performs one of four predetermined operations on his half of the EPR pair. More precisely, if Alice's two bits are 00 (*resp.* 01, 10, 11) then Bob applies the identity (*resp.* Pauli's X gate, Pauli's Z gate, Pauli's X gate followed by Pauli's Z gate). It can be shown that as a result Bob recovers $|\varphi\rangle$!

In the quantum circuit below the single lines denote qubits, the top two lines being Alice's qubits while the last one is Bob's qubit. The first gate represents a cNOT gate, H denotes an

Fig. 24 Circuit implementing the quantum teleportation protocol


```
lemma teleportation:
  assumes "state 1 φ" and "state 3 q" and "List.member (alice_meas φ) (p, q)"
  shows "∃r. state 2 r ∧ bob q (alice_out φ q) = r ⊗ φ"
```

Fig. 25 The quantum teleportation (see *Quantum_Teleportation.thy*)

```
definition bob:: "complex Matrix.mat => bit × bit ⇒ complex Matrix.mat" where
"bob q b ≡
if (fst b, snd b) = (zero, zero) then q else
  if (fst b, snd b) = (zero, one) then (Id 2 ⊗ X) * q else
    if (fst b, snd b) = (one, zero) then (Id 2 ⊗ Z) * q else
      if (fst b, snd b) = (one, one) then (Id 2 ⊗ Z * X) * q else
        undefined"
```

Fig. 26 Bob's decoding function

Hadamard gate, the meters represent measurements, the double lines are classical channels carrying the classical bits $M1$ and $M2$ obtained after the measurements. This circuit gives a concise description of the protocol outlined above (Fig. 24).

The formal specification of the protocol can be written in Isabelle as follows (Fig. 25).

The function *alice_out* φ q outputs the two classical bits sent by Alice after the measurements.

```
definition alice_out:: "complex Matrix.mat => complex Matrix.mat => bit × bit" where
"alice_out φ q ≡
if q = mat_of_cols_list 8 [[φ$$(0,0),φ$$(1,0),0,0,0,0,0,0]] then (zero,zero) else
  if q = mat_of_cols_list 8 [[0,0,φ$$(1,0),φ$$(0,0),0,0,0,0]] then (zero,one) else
    if q = mat_of_cols_list 8 [[0,0,0,0,φ$$(0,0),-φ$$(1,0),0,0]] then (one,zero) else
      if q = mat_of_cols_list 8 [[0,0,0,0,0,0,-φ$$(1,0),φ$$(0,0)]] then (one,one) else
        undefined"
```

The decoding function *bob* q *(alice_out* φ $q)$ corresponds to the state of a 3-qubit system whose first and the second qubits are Alice's qubits after measurement and third qubit is Bob's qubit after performing his predetermined operation given the two classical bits sent by Alice (Fig. 26).

Then the formal specification 25 asserts that the final state of Bob's qubit is nothing but $|\varphi\rangle$, namely the state given as argument and representing the unknown state Alice started with. The quantum state $|\varphi\rangle$ has been "teleported" from the first to the third position, *i.e.* from Alice to Bob. The existential quantification in the statement asserting that whatever Alice's two classical bits sent to Bob the state of the combined system always "factors" through $|\varphi\rangle$.

5.3 The Deutsch–Jozsa Algorithm

Deutsch in [11] was the first to demonstrate that a quantum computer could perform a task faster than any classical computer. His algorithm was improved later by numerous researchers. We explain below the purpose of Deutsch's algorithm.

A function taking values in {0, 1} is *balanced* if it outputs 0 for half of its inputs and 1 for the other half. We start with a function $f : \{0, 1\} \rightarrow \{0, 1\}$.
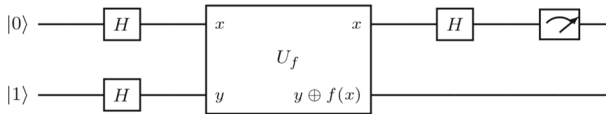
Fig. 27 Circuit implementing Deutsch's algorithm

```
definition (in deutsch) deutsch_algo:: "complex Matrix.mat" where
"deutsch_algo ≡ (H ⊗ Id 1) * (Uᶠ * ((H * |zero⟩) ⊗ (H * |one⟩)))"
```

Fig. 28 Deutsch's algorithm (see *Deutsch.thy*)

```
theorem (in deutsch) deutsch_algo_is_correct:
  shows "deutsch_algo_eval = 0 ⟶ is_const" and "deutsch_algo_eval = 1 ⟶ is_balanced"
  using deutsch_algo_eval_is_sum_mod_2 sum_mod_2_cases by auto
```

Fig. 29 Deutsch's algorithm (see *Deutsch.thy*)

Classically one needs two evaluations of f to determine if the function f is constant or balanced. Deutsch's quantum algorithm determines if f is constant or balanced using only one evaluation of f. This feat is made possible by quantum parallelism, *i.e.* the ability to evaluate a function $f(x)$ for many values of x simultaneously. The quantum circuit implementing Deutsch's algorithm is drawn above (Fig. 27).

Two qubits are prepared, one in the state $|0\rangle$ and another one in the state $|1\rangle$. A Hadamard gate is then applied to each of them followed by the unitary U_f. Afterward the second qubit remains unchanged while the first one is subject to another application of the Hadamard transform. Finally, the first qubit is measured.

In Isabelle the set-up is provided by the following locale.

```
locale deutsch =
  fixes f:: "nat ⇒ nat"
  assumes dom: "f ∈ ({0,1} →ₑ {0,1})"
```

Then we translate the algorithm in Isabelle, the last gate operation being translated by $H \otimes Id\ 1$ since it leaves the second qubit untouched. Note that if time flows from left to right in the circuit, the code should be read from right to left, since the first matrix applied in a matrix multiplication is the one on the right (Fig. 28).

Finally, we check the correctness of the algorithm (Fig. 29).

where *deutsch_algo_eval* is equal to $f(0) \oplus f(1)$, namely $f(0) + f(1)$ modulo 2.

```
lemma (in deutsch) deutsch_algo_eval_is_sum_mod_2:
  shows "deutsch_algo_eval = (f 0 + f 1) mod 2"
  using deutsch_algo_eval_def f_cases is_const_sum_mod_2 is_balanced_sum_mod_2
prob1_deutsch_algo_const prob1_deutsch_algo_balanced by auto
```

Deutsch's algorithm has a generalisation, the so-called Deutsch–Jozsa algorithm, where the domain of f has 2^n values.

Let us assume that we have a function $f : \{0, \ldots, 2^n - 1\} \to \{0, 1\}$ which is either constant or balanced. In the following circuit for the Deutsch–Jozsa algorithm the wire annotated with $/^n$ represents a set of n qubits (Fig. 30). For n = 1 one recovers the particular case of Deutsch's algorithm.

The set-up in Isabelle is given by two locales where Bob promises Alice that he will use a function which is either constant or balanced.
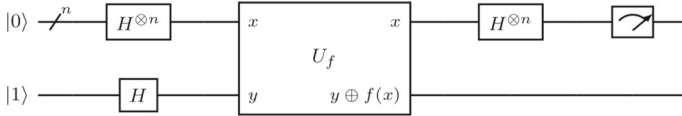
Fig. 30 Circuit implementing the Deutsch–Jozsa algorithm

```
definition (in jozsa) jozsa_algo:: "complex Matrix.mat" where
"jozsa_algo ≡ ((H ⊗ₙₛ) ⊗ Id 1) * (Uꜰ * (((H ⊗ₙₛ) * ( |zero⟩ ⊗ₙₛ)) ⊗ (H * |one⟩))))"

definition (in jozsa) jozsa_algo_eval:: "real" where
"jozsa_algo_eval ≡ prob0_fst_qubits n jozsa_algo"
```

Fig. 31 The Deutsch–Jozsa algorithm (see *Deutsch_Jozsa.thy*)

```
theorem (in jozsa) jozsa_algo_is_correct:
  shows "jozsa_algo_eval = 1 ⟷ is_const"
    and "jozsa_algo_eval = 0 ⟷ is_balanced"
  using prob0_jozsa_algo_of_const_1 prob0_jozsa_algo_of_const_0 jozsa_algo_eval_def
prob0_jozsa_algo_1_is_const is_const_def balanced_prob0_jozsa_algo
prob0_jozsa_algo_of_balanced by auto
```

Fig. 32 The Deutsch–Jozsa algorithm (see *Deutsch_Jozsa.thy*)

```
locale bob_fun =
  fixes f:: "nat ⇒ nat" and n:: "nat"
  assumes dom: "f ∈ ({(i::nat). i < 2^n} →ᴇ {0,1})"
  assumes dim: "n ≥ 1"
locale jozsa = bob_fun +
  assumes const_or_balanced: "is_const ∨ is_balanced"
```

Classically in the worst-case scenario Alice needs $\frac{2^n}{2} + 1$ queries to determine if Bob's function f is constant or balanced. Indeed, Alice can get $\frac{2^n}{2}$ 0s before getting a 1. However, using the Deutsch–Jozsa algorithm Alice can decide if f is constant or balanced using only one evaluation of f.

The translation in Isabelle is similar to the one of Deutsch's algorithm except that the evaluation of the algorithm now requires the measurement of the first n qubits (Fig. 31).

Then one can certify the correctness of the Deutsch–Jozsa algorithm which outputs 1 (*resp.* 0) if and only if f is constant (*resp.* balanced) (Fig. 32).

5.4 The Quantum Prisoner's Dilemma

We will assume that the reader is familiar with the Prisoner's Dilemma and the basic concepts of non-cooperative game theory [12]. The quantum version of the Prisoner's Dilemma was introduced by Eisert, Wilkens and Lewenstein in their classic article [13].

The strategic space of the quantum game is given by the set of unitary 2×2 matrices of the form

$$\hat{U}(\theta, \varphi) = \begin{pmatrix} e^{i\varphi} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & e^{-i\varphi} \cos(\theta/2) \end{pmatrix}$$

with $0 \leq \theta \leq \pi$ and $0 \leq \varphi \leq \pi/2$. As noted in [14] the strategic space used by Eisert *et al.* consisting of these 2-parameter unitary matrices is only a subset of $SU(2)$ and as a consequence is unlikely to reflect any reasonable physical constraint. However, this subset already exhibits interesting properties arising in the quantum regime and as a consequence is worth studying.

Fig. 33 Our two players, Alice and Bob, and their parameters defining their strategies

```
locale prisoner =
  fixes γ:: "real"
  assumes "γ ≤ pi/2" and "γ ≥ 0"

locale strategic_space_2p = prisoner +
  fixes ϑA:: "real"
    and φA:: "real"
    and ϑB:: "real"
    and φB:: "real"
  assumes "0 ≤ ϑA ∧ ϑA ≤ pi"
    and "0 ≤ φA ∧ φA ≤ pi/2"
    and "0 ≤ ϑB ∧ ϑB ≤ pi"
    and "0 ≤ φB ∧ φB ≤ pi/2"
```

Fig. 34 The final state of the game

```
lemma (in strategic_space_2p) equation_one:
  shows "(J†) * ((UA ⊗ UB) * (J * |unit_vec 4 0⟩)) = ψf"
  using psi_one psi_two psi_f by auto
```

```
definition (in strategic_space_2p) alice_payoff :: "real" where
"alice_payoff ≡ 3 * (prob00 ψf) + 1 * (prob11 ψf) + 0 * (prob01 ψf) + 5 * (prob10 ψf)"
```

```
definition (in strategic_space_2p) bob_payoff :: "real" where
"bob_payoff ≡ 3 * (prob00 ψf) + 1 * (prob11 ψf) + 5 * (prob01 ψf) + 0 * (prob10 ψf)"
```

Fig. 35 Alice's and Bob's expected payoffs

The quantization scheme is parametrized by a real $\gamma \in [0, \pi/2]$ which is a measure of the game's entanglement. For $\gamma = 0$ one recovers the classical game while $\gamma = \pi/2$ corresponds to a maximally entangled game (Fig. 33).

Then one defines a unitary operator $\hat{J}$ as $\hat{J} := exp\{i \ \gamma \ \hat{D} \otimes \hat{D}/2\}$, where $\hat{D} := \hat{U}(\pi, 0)$ is the strategy to defect while $\hat{C} := \hat{U}(0, 0)$ is the strategy to cooperate.

```
abbreviation (in prisoner) J :: "complex Matrix.mat" where
"J ≡ mat_of_cols_list 4 [[cos(γ/2), 0, 0, i*sin(γ/2)],
                          [0, cos(γ/2), -i*sin(γ/2), 0],
                          [0, -i*sin(γ/2), cos(γ/2), 0],
                          [i*sin(γ/2), 0, 0, cos(γ/2)]]"
```

If $\hat{U}_A := \hat{U}(\theta_A, \varphi_A)$ (*resp.* $\hat{U}_B := \hat{U}(\theta_B, \varphi_B)$) denotes Alice's (*resp.* Bob's) strategy, then the final state of the game is given by (Fig. 34).

$$|\psi_f\rangle := \hat{J}^\dagger (\hat{U}_A \otimes \hat{U}_B) \hat{J} |00\rangle .$$

Finally, Alice's expected payoff is calculated according to the following formula

$$\$_A := 3P_{00} + P_{11} + 5P_{10} ,$$

while Bob's expected payoff is obtained by

$$\$_B := 3P_{00} + P_{11} + 5P_{01} ,$$

where $P_{xy} := |\langle xy|\psi_f\rangle|^2$ (Fig. 35).

To formalise in Isabelle the main results of [13], we need to introduce formal definitions for Nash equilibriums and Pareto optimality in the context of our game and its restricted strategic space (Fig. 36).

In the classical game ($\gamma = 0$, also called the separable case) it is well known that both players defecting (*i.e.* playing the strategy $\hat{D}$, namely $\varphi_A = \varphi_B = 0$ and $\theta_A = \theta_B = \pi$) is a Nash equilibrium.

```
lemma (in strategic_space_2p) separable_case_DD_is_nash_eq:
  assumes "γ = 0"
  shows "φA = 0 ∧ ϑA = pi ∧ φB = 0 ∧ ϑB = pi ⟶ is_nash_eq"
  using is_nash_eq_def separable_case_DD_alice_opt separable_case_DD_bob_opt assms by auto
```

```
definition (in strategic_space_2p) is_nash_eq :: "bool" where
"is_nash_eq ≡
(∀tA pA. strategic_space_2p γ tA pA ϑB φB ⟶
    alice_payoff ≥ strategic_space_2p.alice_payoff γ tA pA ϑB φB)
∧
(∀tB pB. strategic_space_2p γ ϑA φA tB pB ⟶
    bob_payoff ≥ strategic_space_2p.bob_payoff γ ϑA φA tB pB)"
definition (in strategic_space_2p) is_pareto_opt :: "bool" where
"is_pareto_opt ≡ ∀tA pA tB pB. strategic_space_2p γ tA pA tB pB ⟶
((strategic_space_2p.alice_payoff γ tA pA tB pB > alice_payoff ⟶
  strategic_space_2p.bob_payoff γ tA pA tB pB < bob_payoff) ∧
 (strategic_space_2p.bob_payoff γ tA pA tB pB > bob_payoff ⟶
  strategic_space_2p.alice_payoff γ tA pA tB pB < alice_payoff))"
```

Fig. 36 The formal definitions of Nash equilibrium and Pareto optimality, respectively

```
lemma (in strategic_space_2p) max_entangled_QQ_is_nash_eq:
  assumes "γ = pi/2"
  shows "φA = pi/2 ∧ ϑA = 0 ∧ φB = pi/2 ∧ ϑB = 0 ⟶ is_nash_eq"
  using max_entangled_alice_opt max_entangled_bob_opt is_nash_eq_def assms by blast

lemma (in strategic_space_2p) max_entangled_QQ_is_pareto_optimal:
  assumes "γ = pi/2"
  shows "φA = pi/2 ∧ ϑA = 0 ∧ φB = pi/2 ∧ ϑB = 0 ⟶ is_pareto_opt"
  using coop_is_pareto_opt max_entangled_QQ assms by blast
```

Fig. 37 In the quantum regime a new Nash equilibrium appears which is Pareto optimal

First, the authors prove that in the maximally entangled quantum game ($\gamma = \pi/2$) both players defecting is no longer a Nash equilibrium.

```
lemma (in strategic_space_2p) max_entangled_DD_is_not_nash_eq:
  assumes "γ = pi/2"
  shows "φA = 0 ∧ ϑA = pi ∧ φB = 0 ∧ ϑB = pi ⟶ ¬is_nash_eq"
```

Second, the authors introduce a new quantum strategy when $\gamma = \pi/2$, coined the *quantum move* and denoted $Q := \hat{U}(0, \pi/2)$, with a high payoff (namely 3) for both players resolving the prisoner's dilemma. They prove that both players playing Q is a Nash equilibrium which is also Pareto optimal (Fig. 37).

Finally, in the last part of their article Eisert *et al.* study an unfair version of the Prisoner's Dilemma where one player is restricted to classical strategies while the second player is not subject to such a restriction, *i.e.* Alice can play any strategy, either classical or quantum, while Bob can only play classical strategies. However, in the next section we point out a flaw in their treatment of the unfair version of the game. Indeed, we will see it is not true that the so-called *miracle move* as defined in [13] always gives quantum Alice a large reward against classical Bob and outperforms the so-called *tit-for-tat* strategy in an iterated game.

6 The Unfair Version of the Quantum Prisoner's Dilemma

Below we show the section in [13] on the quantum-classical version of the Prisoner's Dilemma, where Alice may use a quantum strategy while Bob is restricted to a classical strategy, is flawed.

In particular, the claim that the so-called miracle move, defined as $\hat{M} := \hat{U}(\pi/2, \pi/2)$, gives Alice

at least reward $r = 3$ as pay-off, since $\$_A(\hat{M}, \hat{U}(\theta, 0)) \geq 3$ for any $\theta \in [0, \pi]$, leaving Bob with $\$_B(\hat{M}, \hat{U}(\theta, 0)) \leq \frac{1}{2}$ [13, p.3079]

is false. Indeed, for a maximally entangled game $\gamma = \frac{\pi}{2}$, for $\theta = \frac{\pi}{2}$ one has

$$\frac{1}{2} < \$_A(\hat{M}, \hat{U}(\frac{\pi}{2}, 0)) = \$_B(\hat{M}, \hat{U}(\frac{\pi}{2}, 0)) = 1 < 3.$$

In the situation where Alice plays the miracle move while Bob is restricted only to classical strategies, for $0 \leq \gamma \leq \frac{\pi}{2}$ we have

$$\$_A(\hat{M}, \hat{U}(\theta, 0)) = \frac{1}{8}(21 + \cos(\gamma)^2(-3 + 14\cos\theta) + 3\sin(\gamma)^2 - 16\sin\gamma\sin\theta) \quad (1)$$

$$\$_B(\hat{M}, \hat{U}(\theta, 0)) = \frac{1}{8}(11 + \cos(\gamma)^2(7 - 6\cos\theta) - 7\sin(\gamma)^2 + 4\sin\gamma\sin\theta) . \quad (2)$$

So, pluging $\gamma = \frac{\pi}{2}$ in equations (1) and (2) gives

$$\$_A(\hat{M}, \hat{U}(\theta, 0)) - \$_B(\hat{M}, \hat{U}(\theta, 0)) = \frac{5}{2}(1 - \sin\theta)$$

admitting a minimum of 0 when $\theta = \frac{\pi}{2}$.

In other words, the dilemma is not removed in favor of the quantum player contrary to the claim in [15, III.C] which reproduced the error in [13] supported by erroneous computations (the authors found $\$_A = 3 + 2\sin\theta$ and $\$_B = \frac{1}{2}(1 - \sin\theta)$ instead of $\$_A = 3 - 2\sin\theta$ and $\$_B = \frac{1}{2}(1 + \sin\theta)$).

Indeed, Bob can immunize himself against Alice's miracle move by playing the *down-to-earth* move $\hat{E}$

$$\hat{E} \equiv \hat{U}(\frac{\pi}{2}, 0) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} ,$$

the outcome being a draw $\$_A = \$_B = 1$.

Assuming $\gamma = \frac{\pi}{2}$, $\phi_B = 0$, we get the following pay-off matrix.

Alice	Bob		
	$\hat{C}$	$\hat{D}$	$\hat{E}$
$\hat{C}$	$(3, 3)$	$(0, 5)$	$(\frac{3}{2}, 4)$
$\hat{D}$	$(5, 0)$	$(1, 1)$	$(3, \frac{1}{2})$
$\hat{Q}$	$(1, 1)$	$(5, 0)$	$(3, \frac{1}{2})$
$\hat{M}$	$(3, \frac{1}{2})$	$(3, \frac{1}{2})$	$(1, 1)$

So, if Alice plays $\hat{M}$, the dominant strategy of Bob becomes $\hat{E}$, thereby doing substantially worse than if they would both cooperate, reproducing the dilemma. Moreover, nothing supports the claim that Alice

may choose "Always-$\hat{M}$" as her preferred strategy in an iterated game. This certainly outperforms *tit-for-tat* [...] [13, p.3079].

In conclusion, the "miracle move" as defined in [13] is of no advantage.

7 Conclusions and Future Work

Our work demonstrates that an extensive formalisation of quantum algorithms and quantum information theory in Isabelle/HOL is possible and not a fruitless exercise. Indeed, the Letter [13] of Eisert *et al.* is a pioneering and highly cited article published in Physical Review Letters, a high-profile physics journal. The error uncovered therein is a notable unexpected outcome of our library. Indeed, this error had gone unnoticed in the field until our work and we found at least one subsequent published paper that reproduced it. After a private communication Eisert *et al.* acknowledged their error and they actually found a fix to re-establish their conclusions regarding what they call the "miracle move". An erratum was published by Physical Review Letters [16].

Possible future applications of our library could include the verification of quantum cryptographic protocols, Isabelle having been successfully used in the past by Lawrence Paulson for the verification of cryptographic protocols using inductive definitions [17]. A related work is the formalisation in Isabelle of parts of the Quantum Key Distribution algorithm by Florian Kammüller using a framework extending attack trees to probabilistic reasoning on attacks [18].

Last, there is ongoing work in our library to formalise the quantum Fourier transform and unlock the potential formalisation of a wide range of more advanced quantum algorithms relying on it.

Compliance with ethical standards

References

1. Liu, J., Zhan, B., Wang, S., Ying, S., Liu, T., Li, Y., Ying, M., Zhan, N.: Formal verification of quantum algorithms using quantum hoare logic. In: Dillig, I., Tasiran, S. (eds.) Computer Aided Verification, pp. 187–207. Springer, Cham (2019)
2. Rand, R., Paykin, J., Zdancewic, S.: QWIRE practice: formal verification of quantum circuits in coq. EPTCS **266**, 119–132 (2018)
3. Boender, J., Kammüller, F., Nagarajan, R.: Formalization of Quantum Protocols using Coq. In: *QPL* (2015)

4. Ballarin, C.: Locales: a module system for mathematical theories. J. Autom. Reason. **52**(2), 123–153 (2014)
5. Thiemann, R., Yamada, A.: Matrices, Jordan normal forms, and spectral radius theory. Archive of Formal Proofs (2015). http://isa-afp.org/entries/Jordan_Normal_Form.html, Formal proof development
6. Prathamesh, T.V.H.: Tensor product of matrices. Archive of Formal Proofs (2016). http://isa-afp.org/entries/Matrix_Tensor.html, Formal proof development
7. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th edn. Cambridge University Press, New York (2011)
8. Wootters, W.K., Zurek, W.H.: A single quantum cannot be cloned. Nature **299**, 802–803 (1982)
9. Dieks, D.: Communication by EPR devices. Phys. Lett. A **92**, 271–272 (1982)
10. Bennett, C.H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., Wootters, W.K.: Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. Phys. Rev. Lett. **70**, 1895–1899 (1993)
11. Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proc. R. Soc. Lond. A **400**, 97–117 (1985)
12. Osborne, M.J., Rubinstein, A.: A Course in Game Theory, vol. 1. MIT Press, Cambridge (1994)
13. Eisert, J., Wilkens, M., Lewenstein, M.: Quantum games and quantum strategies. Phys. Rev. Lett. **83**, 3077–3080 (1999)
14. Benjamin, S.C., Hayden, P.M.: Comment on quantum games and quantum strategies. Phys. Rev. Lett. **87**, 069801 (2001)
15. Flitney, A.P., Abbott, D.: An introduction to quantum game theory. Fluct. Noise Lett. **02**(04), R175–R187 (2002)
16. Eisert, J., Wilkens, M., Lewenstein, M.: Erratum: quantum games and quantum strategies [Phys. Rev. Lett. 83, 3077 (1999)]. Phys. Rev. Lett., 124, 139901 (2020)
17. Paulson, L.C.: The inductive approach to verifying cryptographic protocols. J. Comput. Secur. **6**(1–2), 85–128 (1998)
18. Kammüller, F.: Attack trees in Isabelle extended with probabilities for quantum cryptography. Comput. Secur. **87**, 101572 (2019)