UNIVERSITATEA BABEŞ-BOLYAI
BABEŞ-BOLYAI TUDOMÁNYEGYETEM
BABEŞ-BOLYAI UNIVERSITÄT
BABEŞ-BOLYAI UNIVERSITY
TRADITIO ET EXCELLENTIA

FACULTATEA DE FIZICĂ
Str. Mihail Kogălniceanu nr.1
Cluj-Napoca, RO-400084
Tel: +4(0)264-405300 | FAX: +4(0)264-591906
secretariat.phys@ubbcluj.ro
www.phys.ubbcluj.ro

UNIVERSITATEA "BABEŞ-BOLYAI" CLUJ-NAPOCA
FACULTATEA DE FIZICĂ
SPECIALIZAREA FIZICĂ

LUCRARE DE LICENȚĂ

Coordonator științific:
Dr.Lect.Emil Vințeler

Absolvent:
Hosu Emanuel-Claudiu

2022

UNIVERSITATEA BABEŞ-BOLYAI
BABEŞ-BOLYAI TUDOMÁNYEGYETEM
BABEŞ-BOLYAI UNIVERSITÄT
BABEŞ-BOLYAI UNIVERSITY
TRADITIO ET EXCELLENTIA

FACULTATEA DE FIZICĂ
Str. Mihail Kogălniceanu nr.1
Cluj-Napoca, RO-400084
Tel: +4(0)264-405300 | FAX: +4(0)264-591906
secretariat.phys@ubbcluj.ro
www.phys.ubbcluj.ro

UNIVERSITATEA "BABEŞ-BOLYAI" CLUJ-NAPOCA
FACULTATEA DE FIZICĂ
SPECIALIZAREA FIZICĂ

LUCRARE DE LICENȚĂ

GAME THEORY IN QUANTUM PHYSICS

Coordonator științific:
Dr.Lect.Emil Vințeler

Absolvent:
Hosu Emanuel-Claudiu

2022

Contents

Introduction

The twentieth century was a period of incredible intellectual vitality in both science and mathematics. The first half of the century saw the controversies surrounding Quantum mechanics in which Einstein was involved, giving rise to the famous EPR paradox. In the same period in mathematical logic, the very important idea of what does it mean for something to be a computer was defined rigorously and also what are their limits was established. Both lines of thought were developed and eventually met on common ground due to the appearance of physical universal computers which immediately led to the theory of algorithms and complexity due to the limitations of such computers, which can be considered a refinement of concepts of computability and logic that were introduced by pioneers such as Skolem, Gödel and Turing([15][16][17]). These connections led to our days where we have algorithms like Shor's, that can crack important classical encryption schemes by using the concepts developed from EPR paradox by Bell from where many others followed. These lines of thought have through the prism of a seemingly disconnected subject, developed by people like Von Neumann, Nash and Harasanyi, that being the theory of games. The scope of this thesis more specifically is to prove Bell-like theorems using games, to prove results related to a very recent development from 2020 on quantum complexity theory which are based on a class of games and to introduce a new attack scheme on a certain type of protocol, based on methods developed in 2018, which can be formalized as a game.

In the first chapter I introduce what I have thought to be sufficient for a newcomer to get used to the game theoretic concepts that are used in the latter chapters, together with exposition of the history of said results by crediting pioneers such as von Neumann, Nash and Harasanyi. I credit myself with simplifying certain sections of certain arguments of fundamental results such as the Minmax theorem and Nash's theorem, with the idea of better exposition of said results in mind.

In the second chapter I introduced quantum gates and Bell's theorem, together with a bit of context behind it's history and also insights given by Bell himself on his theorems. I give myself the credit of combining different arguments found in different sources in order to give more concise presentation of the main results in the chapter such as the proof of universality of a discrete set of quantum gates and the proof of Bell's theorem.

The third chapter makes use of the concepts that were introduced in the previous two, because I introduce a certain class of games called nonlocal games and show that this type of games can be utilized in order to prove Bell like theorems without requiring certain assumptions that were needed in the classical types of proofs. I credit myself with the conceptualization of the new class of nonlocal games in terms of Harasanyi games and with the combining of certain aspects of different arguments that were used as means to prove the main results such as in the case of the nonlocality of QM through the Mermin-Peres magic square game.

The fourth chapter is the most vast because I introduce concepts from complexity theory such as the notion of formal language, and that of complexity classes, in order to show that

certain games can be used to define the said classes, being especially natural for what are known as quantum complexity classes. I credit myself with a somewhat original introduction to the classes by giving informal algorithms for certain problems and as such making them intuitive with regards to what class they belong to. I also showed through Venn-Euler diagrams the idea of complexity reductions and inclusions of the different classes that we at least briefly go over, to give a better understanding of how are the different classes, both classical and quantum are related. Towards the end of the chapter I introduce the concept of *quasi-observable*, which is just a quantum observable that has certain complexity properties and then I prove using the solution to what was previously known as Tsirelson's problem, that there must exist quasi-observables given certain assumptions related to the resolution to the mentioned problem, which is related to algebraic structures met in quantum field theories. At the end of the chapter I study consequences of models where such observables exist.

The fifth and last contains a very brief analysis of a specific type of quantum protocol called a quantum Byzantine protocol which can be expressed as a game. I introduce the necessary theoretical tools and then analyze the feasibility of a specific type of attack called a "man-in-the-middle" attack on such a protocol, obtaining what I consider to be one of the two main results of this thesis, which can be related to the possible future of quantum servers.

1 Useful game-theoretic notions

Game theory is, informally, the study of choices and outcomes of interactions between agents that are looking to maximize their gains. Historically, as a mathematically well founded field, game theory is a relatively young field, it's genesis being generally attributed to the 1944 monograph by von Neumann and Morgensten[1].However, von Neumann actually proved a famous and fundamental result in game theory much earlier, the minmax theorem, in 1928[2]. Later Nash generalized von Neumann's result to more general games([4]). The games mentioned above however offer quite a lot of freedom to the players, they are what are called *complete* games. The first well received model of a new type of games, which will be called "incomplete games" were introduced years later by Harasanyi in 1967 [5]. In this section we will go briefly through notions from game-theory that will prove useful later, in the exposition of the main results, such as the different ways of defining games and the different types of said games.

1.1 Extended form games and strategies

Definition 1.1 *We call a finite graph directed, if it's a tuple:*

$$G = \langle V, E \rangle \tag{1}$$

whith V a finite set of vertices and $E \subseteq (V \times V)$ are edges of the graph.
A graph has the well known intuitive representation with nodes and arrows(when it comes to directed graphs) where we label the nodes with with numbers for example and even the representations of the edges in certain types of graphs as we will see later. We can visualize a few examples in **Fig 1.1**

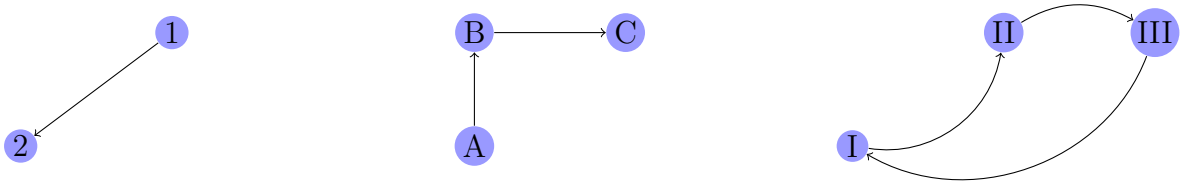


Figure 1.1 Representations of different directed finite graphs.
A more useful tool for our purposes will be a special class of graphs we will call *trees*, but first we must become familiar with the concept of a *path*.

Definition 1.2 *Given the graph $G = \langle V, E \rangle$ and two vertices $v_1, v_k \in V$, we call a path from v_1, v_k through G a sequence of vertices and edges of the form $v_1, e_1, v_2 ... v_{k-1}, e_{k-1}, v_k$, where for any $1 \leq i < k-1$, e_i connects v_i and v_{i+1} and $v_i \neq v_{i+1}$.*

Definition 1.3 *A rooted tree is a tuple*

$$T = \langle V, E, r \rangle \tag{2}$$

where $G = \langle V, E \rangle$ is a what we named a directed graph and $r \in V$ will be called the root of the tree. Also given any vertex $v \in V$, there is a unique path in G from r to v .
From the definition given above, we can see that there is no need for us to be explicit when treating the direction of the edges in a rooted tree. Now we will see that any game can be represented by using a rooted tree, called the game-tree. This will be done in an extensive form of the game.

Definition 1.4 *A game in extensive form is a tuple $\langle V, E, r, N, (V_i)_{i \in N}, O, m \rangle$, where $\langle V, E, r \rangle$ is a finite rooted tree, N is a finite number of players,$(V_i)_{i \in N}$ is a partition of the non-leaf nodes of the tree, O is a set of possible game outcomes and m is map $m : L \to O$, where L is the set of leaf nodes.*
Leaf nodes are terminal nodes of the tree and $(V_i)_{i \in N}$ represents intuitively the accessible game positions to each player throughout a game. We will give a simple example to make it easier to motivate what we meant by the definition above.

Let's imagine a little card game between two players we named generically I and II, where they start with an identical hand of three cards, we will name non-interestingly: c_1, c_2, c_3 where each card has a certain value indicated by it's respective index. The way in which the game is played is that players take turns in placing the cards down in what will become a common stack and the player that added cards with bigger values together wins. The key detail is that the same card cannot be placed twice in the stack. The set of possible game outcomes will be $O = \{I_w, II_w, D\}$.

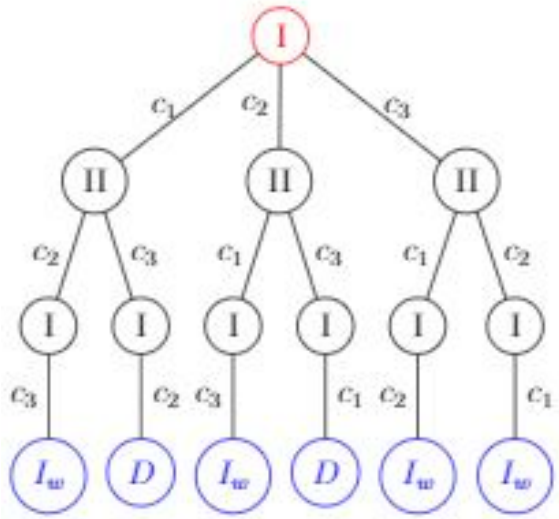


Figure 1.2 The game described above, in extended form.
We will look now at the concept of *strategy* which would be an essential part in any theory of games, but for that we will need to remind ourselves of what we mentioned previously about the set L of leaf nodes, in order to define the set of *children* nodes. We will define it as $C(n_k) = \{n_{k+1} \mid n_k \leq_p n_{k+1}\}$, where by "$\leq_p$" we mean intuitively the path relation, for some path P through the game-tree. In other words, the set of possible continuations from an arbitrary vertex is represented by $C(x)$.

Definition 1.5 *We call a strategy for player i, a function* $s_i : (V_i \setminus L)(x) \to C(x)$.
So for example in the game in extensive form from **Fig. 1.2**, player I has three possible strategies. It is easy now to define the concept of winning strategy based on the idea that if no other for a player, a winning strategy is a strategy such that leads to a favorable outcome regardless of the strategies available to the opponents.

1.2 Strategic games, Payoff and Nash equilibrium

It is more favorable to work in a simplified setting when treating games, called *strategic forms* of the games, which are meant to avoid treatment of nodes and regard only "actions" and outcomes. For that we will need however to learn of a few concepts, such as *utility functions*.

Definition 1.6 *We call an utility function for a player i, a function on the cartesian product of sets of strategies that takes real values, defined* $u_i : S \to \mathbb{R}$*, where* $S = S_1 \times ... \times S_n$*, with* $S_i = \{s_1, ...s_k\}$ *the set of strategies for player i.*
The idea behind the utility function can be visualized with a two player game, where for player I, and two strategies cartesian product $s_1 \times s_2$ that lead to outcome o_1, and another pair $s_3 \times s_4$ that lead to outcome o_2 such that $u_I(s_1, s_2) \leq u_I(s_3, s_4)$, formalizes the idea that the player I "prefers" o_2 over o_1. The values of $u_I(s_1, s_2), u_I(s_3, s_4)$ are called *payoffs*.

Definition 1.7 *We say a game is in strategic or otherwise known as normal form, if it is described by a tuple:*

$$\mathcal{G} = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle \tag{3}$$

where N is a finite number of players, S_i *are the sets of strategies available to them, and* u_i *are the sets of utility functions for each player.*
We can make a simple representation of the definition above by using what is called a *payoff matrix*(see **Fig 1.3**).

		Player 2		
		A	B	C
Player 1	X	(x, a)	(x, b)	(x, c)
	Y	(y, a)	(y, b)	(y, c)
	Z	(z, a)	(z, b)	(z, c)

Figure 1.3 Payoff matrix of a game between two players, which have three available strategies each.
We must now look at the concept of "domination" of strategies. Which will be a concept one must pursue in order to establish the most renowned theorems in finite game theory.

Definition 1.8 *We say a strategy for player i, s_i is strictly dominated if there exists another strategy for the said player l_i such that for any set of strategies of the other players S_{-i}, and any strategy within them $s_{-i} \in S_{-i}$,then*:

$$u_i(s_i, s_{-i}) < u_i(l_i, s_{-i}) \tag{4}$$

Another way to word the definition above is to say that the strategy s_i is strictly dominated by strategy l_i. Now we will need to explore ideas of *stability*.

Definition 1.9 *We call a cartesian tuple of strategies $s* = (s_1*, s_2*, ...s_n*)$, each for the indexed player a Nash equilibrium, if given any player i and strategies available to them:*

$$u_i(s*) \geq u_i(s_i, s_{-i}*) \tag{5}$$

It can be quickly seen that provided a bit of understanding of what a strictly dominated strategy means, that they cannot be part of a Nash equilibrium, but we will prove that in a bit. Before that, it's worth noting that the Nash equilibrium is not a "solution" in the direct sense of the game, as it is not a strategy in itself, but a meta-solution based on the available strategies of the players and the assumption that each player is "rational". Another thing that it makes it insufficient as a tool of analysis of games because there may be games that have no equilibria at all or a multitude of them. It's a concept that intends to capture stability, which will prove unsatisfactory in certain settings where because it lacks in terms of *security*(these are all rich concepts that we will not develop more in this work and the author suggests reading referenced literature). We will give an example of type of game that will make things more clear.

		Player II	
		A	B
Player I	A	$(2, 2)$	$(-1, -1)$
	B	$(-2, -2)$	$(3, 3)$

Figure 1.4 A *coordination game* in which we can see that both ordered pairs of strategies which give us the diagonal of the matrix, are equilibrium points. That is to say, both players are better off "coordinating" their strategies. We can see that there is more than one equilibrium point because both fit relation (5)

One may wonder what is the relationship between strictly dominated strategies as we mentioned in **Def. 1.8** and Nash equilibrium points. A key idea to look at is the fact that we mentioned the fact that Nash equilibrium may be thought of as a way of describing the fact that certain strategies are the best response available to each player for the strategies

available to the others, assuming the others are "rational" in the sense that they would not play something else. This idea of rationality extends to the fact that a rational player would not play dominated strategies. As such we get a connection.

Theorem 1.1 *Strictly dominated strategies cannot be part of a Nash equilibrium.*

Proof. Let's try to prove it through a reductio ad absurdum type fo argument and suppose the contrary holds. Then there exists a strategy for a player j, s_j such that

$$u_j(s_j, s_{-j}*) \geq u_j(x_j, s_{-j}*), \forall x_j \in S_j \tag{6}$$

while at the same time, we know from (4) that we should also have a strategy d_j that gives a better payoff for player j regardless of the strategies with which the other players respond, so even the Nash equilibrium responses $s_{-j}*$. Contradiction.

As we have briefly mentioned before, Nash equilibrium may be a good concept to capture an idea of stability within games, however, it does so at the expense of giving credits of rationality to every player which may not be considered sound. As such much literature was focused on refining and developing new solution concepts that are more suitable to different types of games. For example, let's consider a game in which there may be an equilibrium point, but in which certain aspects of *rationality* assumed for a Nash equilibrium to exist are not taken for granted, one may consider the idea of trying to maximize their profits around these circumstances. As such, one may consider what are the minimum payoffs they may gain by following certain strategies while and try to maximize that. So for starters, to to formalize what is the worst case scenario if some player i chose strategy s_i, we simply denote it in terms of the minimum value of the payoff function for that given strategy, or rather how much would player i lose if the other players chose the most damaging strategy against the said strategy. We write this as

$$\min_{k_{-i} \in S_{-}1} u_i(s_i, k_{-i}) \tag{7}$$

Now the idea is to pick the strategy that maximizes this minimum payoff out of the set of available strategies. Which we will write as

$$v_i = \max_{s_i \in S_i} \min_{k_i \in S_{-}i} u_i(s_i, k_{-i}) \tag{8}$$

It will be useful to have a term for the strategies that satisfy the above condition. We will as such call strategy s_i in (8) as *maxmin strategy* and to the value of the specific payoff as a we noted with v_i what we will call the *maxmin value* of player i and it's taken to formalize the idea of *safety* or *security* for player i.

In a very similar vein, will refer to the idea of *minmax strategy* of a player i and value respectively, which we would refer to as the strategy of player i that minimizes the best case payoff of the other players and mutatis mutandis the value of as the *minmax value*, which we will note as

$$\min_{s_i \in S_i} \max_{k_{-i} \in S_{-}i} u_{-i}(s_i, k_{-i}) \tag{9}$$

This may or may not be useful depending on the game. The reason why we chose to mention it however is the following landmark theorem in game theory. Before that we need to get comfortable with the following particular notion of two-player game.

Definition 1.10 *We call a zero-sum two player game, where the payoff functions meet the following condition*

$$u_1(s_1, s_2) + u_2(s_1, s_2) = 0 \tag{10}$$

As mentioned, we can now state a landmark result first proved by von Neumann in [2].

Theorem 1.2(Minmax theorem) *For any Nash equilibrium point of a zero sum, two players game, the payoffs of each player is equal to both the minmax and the maxmin values of those respective players.*

Proof. First let's keep in mind the following relationship that follows immediately from the definition of Nash equilibrium. We fix an arbitrary strategy $s_1' \in S_1$, then we have:

$$\underset{k_1 \in S_{-}1}{min}\, u_1(s_1', k_{-1}) \leq u_1(s_1', s_2*) \tag{11}$$

Where s_2* is any Nash equilibrium profile strategy for the strategic game. It then follows immediately that:

$$\underset{s_1 \in S_1}{max}\, \underset{k_{-1} \in S_{-}1}{min}\, u_1(s_1, k_{-1}) \leq \underset{s_1 \in S_1}{max}\, u_1(s_1, s_2*) \tag{12}$$

Now we will suppose for the sake of contradiction that it is not a Nash equilibrium and then by a similar argument we will get to the minmax value as well. player 1. *Contradiction.*

$$\underset{s_1 \in S_1}{max}\, \underset{k_{-1} \in S_{-}1}{min}\, u_1(s_1, k_{-1}) = \underset{s_1 \in S_1}{min}\, \underset{k_{-1} \in S_{-}1}{max}\, u_{-1}(s_1, k_{-1}) \tag{13}$$

All that is left to do is to show that these are always Nash equilibrium points. Suppose that there is a maxmin value such that it is not part of an equilibrium point. Then there exists a strategy for at least one player i, such that the payoff value is strictly larger for them, no matter what strategies the other player chooses. But that means it would be bigger even for the tuples where maxmin values occur, which written succinctly would be like

$$\underset{s_1 \in S_1}{max}\, \underset{k_{-1} \in S_{-}1}{min}\, u_1(s_1, k_{-1}) < u_1(s_1, s_{-1}*) \tag{14}$$

But this is again a clear contradiction, and by the equivalence from (13), we finished the proof of theorem. This gives us means of identifying Nash equilibrium points, by finding the maxmin or minmax values or vice-versa in a specific type of game. We will later see a theorem that generalizes this.

When the minmax and the maxmin payoff values in a two-player game are equal, we say that the game has *value*, and it's equal to that common payoff value and the strategies involved in the said payoffs are called *optimal*.

1.3 Mixed strategies and Nash's theorem

Another aspect we have to cover is the fact that what we have been calling up until now as strategies, would actually be referred to in the literature as *pure strategies*. They are called so because they are fully deterministic and there is no aspect of chance explicitly involved in the choice of strategies. A motivation for randomizing aspects of strategy choices when it comes to the set of strategies available to a player is the fact that in many games there is no *pure Nash equilibrium* and then the fact that we can give them a probability distribution for the strategy choices, we may have a mixed type of Nash equilibrium. The idea is the fact that the said strategies are after all just choices of decisions predetermined to be responses to the possibilities of choices employed by the other players. As such, we are not actually creating another type of strategy, just randomizing the pure ones.

Definition 1.11 *Given a game in strategic form $\mathcal{G} = \langle N, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle$, where $(S_i)_{i\in N}$ is finite, a mixed strategy for player i, is a probability distribution $\sigma : S_i \to [0,1]$, with the condition that $\sum_{s_i \in S_i} \sigma(s_i) = 1$.*

In an obvious way then, we would define a set of mixed strategies. We are now faced with another problem, mainly the fact that the results we have mentioned so far are talking about pure strategies and not mixed strategies. It is not much of a problem however, as most results have an equivalent in mixed strategies. To make it easier to foresee what this is about, let's look at a simple example. We can have the

Definition 1.12 *Given a finite game in strategic form $\mathcal{G} = \langle N, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle$, we call it's mixed extension the tuple $\mathcal{G}_m = \langle N, (\Sigma_i)_{i\in N}, (U_i)_{i\in N}\rangle$* , where $(\Sigma_i)_{i\in N}$ is the set of os sets of mixed strategies available of each player i and $(U_i)_{i\in N}$ is the real-valued payoff function that maps the mixed strategies to their respective values and it's defined as

$$U_i(\sigma) = \mathbf{E}(u_i(\sigma)) = \sum_{(s_1,...,s_n)\in S} u_i(s_1, ..., s_n)\sigma_1(s_1)...\sigma_2(s_n) \tag{15}$$

The above definition of the payoff function makes sense if one ponders for a bit, because when considering what player i's expected payoff may be, we need to take into account not only their probability distributions of the strategies, but each other's players probability distributions that form a pure strategy profile, but in a way that it is independent of the the choices of the other players, as it would suggest additional knowledge of the players that they cannot be assumed to possess. It should be also noted that this new dimension or parameter of randomizing the way in which the players pick a strategy, gives rise to cases where the mixed extension of the game has a mixed equilibrium, even if it does not have a pure *equilibrium*. Similar things can be said about the *value* of the game.
We will give a simple example to see things more easily.

		Player II	
		C	D
Player I	A	$(-1, 2)$	$(-1, -1)$
	B	$(0, 1)$	$(-2, 0)$

Figure 1.4 A two players game with no Nash equilibrium in this game with regards to the pure strategies available to the two players.

For example, for the strategy profile AC, while player II would like it, player I would rather choose strategy B in this situation. What is of interest however, is to show that there is indeed an equilibrium point in what we call a mixed extension of this game. To see this, consider the sets of mixed strategies for the two players which must be of the form $[\sigma_I(A), (1-\sigma_I(A))(B)]$ and $[\sigma_{II}(C), (1-\sigma_{II}(C))(D)]$ respectively. Of course, there are infinitely many probability distributions as long as they respect the condition from Definition 1.11. Also, it's important to note that the set of possible mixed strategies for each players is infinite, just needing to be in the closed set [0, 1] so we will just note them generically with two variables x and y as it will make things more aesthetically pleasing. Now by following the equation (15) and making the variable notation as said we obtain

$$U_I(x, y) = -2 + x + 2y - 2xy \tag{16}$$

and

$$U_{II}(x, y) = -x + y + 2xy \tag{17}$$

We can now analyze the best strategies by analyzing monotony aspects of the mixed strategies because we can see them as first degree functions for some particular value of probability distribution for the opponent. Very similarily, the value of the game can be found, which leads us to a generalization of Theorem 1.2. However, the thing we are trying to put into evidence right now is what came to be known and considered as a generazliazion of the MinMax theorem.

Theorem 1.3(Nash, 1951[4]) *For every game that is in finite strategic form, has an equilibrium in it's mixed extension.*

This is what one may argue, a generalization of the Minmax theorem in the sense that it gives us a stronger classification of where the solution concept of equilibrium is present or potentially useful.

1.4 Incomplete games and Bayesian equilibria

What we have been talking about until now has again been a more general setting of games in which players are said to have complete information, in the sense that all players present share the same information regarding the payoffs and strategies available to each other. This may not be always the case however, and in this case we call such settings,

games with *incomplete information*. We will use what is known as the *Harsanyi model* in defining a specific type of incomplete games called Bayesian games. They have the specific characteristic of being good models for most purposes of the intended.

Definition 1.13(Harasanyi game) *We call Harasanyi game a tuple* $< N, (T_i)_{i\in N}, \mu, X >$, *where*

- P *is a finite collection of players.*
- T_i *is a finite collection of types for each player* $i \in N$, *noted as* $t_i \in T_i$.
- μ *is a probability distribution over the collection of vectors of types* $T = \prod_i T_{i\in N}$.
- X *is a collection or set of states of nature, each member of* X *being of the form* $x =< (S_i)_{i\in N}, P, (u_i)_{i\in N} >$, *and it may be called state games.*

For a particular vector of types t, $x_t =< (S_i(t_i))_{i\in N}, P(u_i(t))_{i\in N} >$, we call x_t *the state game for the particular type vector t.*
We can see that for a particular type vector the set of strategies is not dependent on the entire type vector, but only on the particular set of types available to the said player. The payoff function is however dependent on the entire type vector.We now need to get down to the idea of a strategy, which will come as natural

Definition 1.14 *We call a probability distribution* $\sigma_i(t_i) : S_i(t_i) \to [0, 1]$, *a behavioral strategy(or simply strategy) for player i, given type* t_i.

By noting $\sigma = \sigma(s_1, t_1) \times ...\sigma(s_N, t_N)$, we can define the expected payoff for a player i in the Harasanyi game $G_{\mathcal{H}}$ as:

$$U_i(\sigma) = \sum_{t\in T} \mu(\sigma \times u_i(s, t)), \tag{18}$$

where $s = s_i \times ... \times s_N$.
Now we can introduce the notion of Bayesian equilibrium which is equivalent to a Nash equilibrium in finite incomplete games.

Definition 1.15 *We say a strategy profile* $\sigma*$ *is a Bayesian equilibrium if for all players i and all actions* s_i*:*

$$U_i(\sigma * \,|\, t_i) \geq U_i((s_i, \sigma_{-i}*) \,|\, t_i) \tag{19}$$

One may notice that we have called the members of the sets S_i actions, and this is so not to be confused with strategies as we will use them from now on in incomplete games, even though they can be considered pure strategies for said player, given some type for that player.

2 Useful quantum mechanical notions

In this section we will be going over a few particularities that are necessary for all the aspects that go into quantum computing and information theory that we will cover in the rest of the article. We will assume a minimum of familiarity of the reader with concepts of quantum mechanics such as wave functions and quantum states. The important aspects that we will cover are the quantum gates and Bell's theorem, which came as an answer to the famous Einstein–Podolsky–Rosen paradox([5]), that was an attack direced towards quantum mechanics, which should not violate certain restrains imposed by the theory of relativity that was put forward by Einstein years earlier. The original theorem was proved using certain assumptions that would have resulted in what are known as Bell inequalities which are violated by QM. We will see however that using games we can ditch most of those assumptions leaving us with a more concrete proof of the theorem in a certain sense.

2.1 Qubits and quantum gates

Definition 2.1 *We call a qubit a state of the form* $|Q\rangle = c_1|0\rangle + c_2|1\rangle$*, where* c_1 *and* c_2 *are complex amplitudes.*

Definition 2.2 *We call a quantum gate a quantum operator that acts on qubits*

$$\begin{gathered} I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ P = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}, \quad H = 1/\sqrt{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, R_y(\phi) = \begin{bmatrix} cos(\phi/2) & -sin(\phi/2) \\ sin(\phi/2) & cos(\phi/2) \end{bmatrix} \end{gathered} \tag{20}$$

Example 2.1. A set of representations for the most useful unary(acting on a single qubit) quantum gates.

One of the most important aspects that we have to bear in mind is that we need to establish what is known in the literature as *universality*. This is not as straightforward as one may think however and we would need to consider at least two types of universalities, and one may refer to the fact that

Definition 2.3 *We say a set of quantum gates is computationally universal if it is Turing equivalent*

This definition is of course a specialized one, restricted to what we need to concentrate on right now, but it would be in fact easily generalized to any model that can be called a computational model in the sense that it can simulate a Turing machine.

Definition 2.3 *We say a set of quantum gates is unitarily univerasal if we*

Theorem 2.1 *The set of gates* $\{H, Y, Z, R_z(\phi), R_y(\phi)\}$ *is universal*

We will only sketch a proof, by proving certain identities, the details being quite easy to fill

in. First of all what the universality implies is actually two parts now the idea is that we need.

Now, the proof relies on the fact that all of the matrices we used here are unitary matrices and by simple inner product within the gates, any gate can be made into a single unitary gate that has the the dimension of the multiplication of the dimensions of the gates involved. Indeed, one can easily see this by using the Dirac bra ket notation to prove the fact that all of the gates involved in a circuit and as such the circuit itself can be captured within a single unitary matrix. The proof goes as follows: consider the fact that any of the Pauli gates can be turned in one another by repeated applications of the others

$$\begin{aligned} HZH &= X \\ HXH &= Z \end{aligned} \tag{21}$$

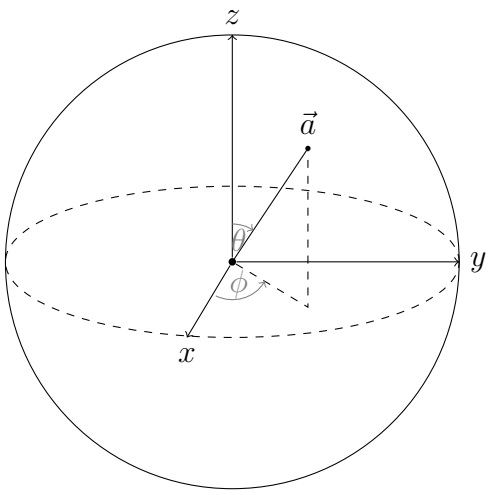


Figure 2.1 The Bloch sphere

Now another thing we have to note is that any unitary transformation or matrix can be written in an exponential form, in terms of a hermitian matrix:

$$U = e^{i\gamma H} \tag{22}$$

This is a quite easy thing to prove with a bit of knowledge about the matrix operatios and the Maclaurin expansion. This is very powerful because universality can be thought of as being able to reach any point on the Bloch sphere(Figure 2.1) with an iterated usage of gates from a finite set of gates, which will all be unitary transformations. As such we can write all of them in the form equation (20). Next we can also write any unitary in a manner similar of the Euler formula as:

$$U = cos(\gamma)\mathbb{I} + (isin(\gamma))H \tag{23}$$

In particular we can for example write

$$\begin{aligned} R_z(\phi) &= e^{i\frac{\phi}{2}Z} = cos(\gamma)\mathbb{I} + (isin(\gamma))Z \\ R_x(\phi) &= e^{i\frac{\phi}{2}Z} = cos(\gamma)\mathbb{I} + (isin(\gamma))X \\ R_y(\phi) &= e^{i\frac{\phi}{2}Z} = cos(\gamma)\mathbb{I} + (isin(\gamma))Y \end{aligned} \tag{24}$$

From which it immediately follows that

$$\begin{aligned} ZR_z(\phi)Z &= R_z(\phi) \\ HR_x(\phi)H &= R_z(-\phi) \\ HR_z(\phi)H &= R_x(-\phi) \end{aligned} \tag{25}$$

This mostly concludes our sketch of a proof, because now we can see that applying the Hadamard gate on the rotation gate we can obtain all the other rotation gates which is easily seen to fulfill our requirement of universality. This is not an optimal result by any means,being known in fact that We have to keep in mind however that we cannot in truth obtain all unitary transformations with just a countable application from a finite set of gates, something one could say to be known since Cantor. However, it is enough to say that we can get arbitrarily close to it.

2.2 Quantum nonlocality and Bell's theorem

The main focus of this section will be Bell's theorem and similar results. For that we will need to get familiar with notions such as *causality* and *locality*.

Definition 2.5 *We say that if an event E_1 is the cause of event E_2, they are causally connected*

This is of course quite informal and not very instructive or clarifying. This is simply because there is no established rigorous definition of what is it meant by causality in the Bell-like theorems is what we can infer from Special Relativity in terms of light-cones. In other words, if an event were to be the effect of another, it must be in the future light-cone of the event that caused or better said influenced it. We can call it a causal influence as limited by Special Relativity.

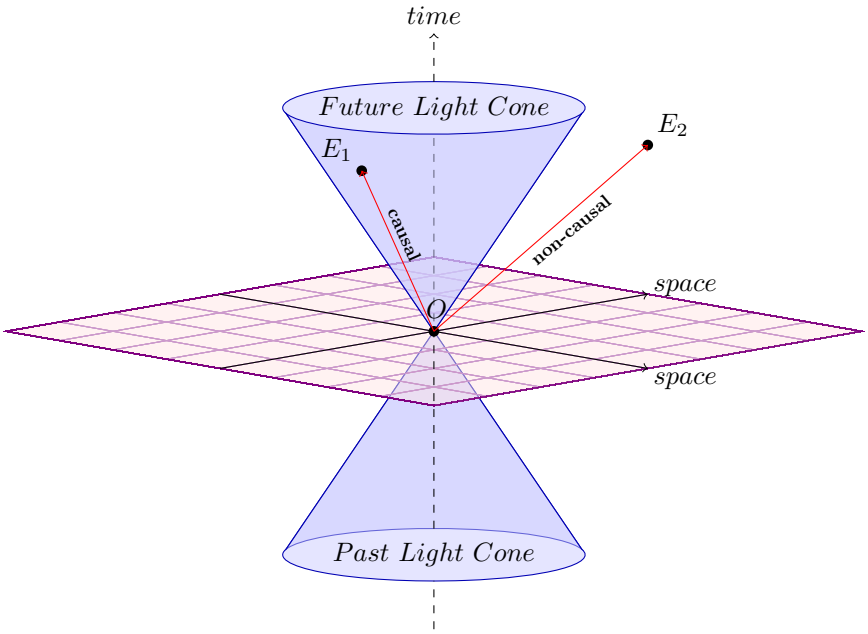


Figure 2.2 Representation of the light cone of an event O.

We can see in the figure above that we can imagine the relations between event O, E_1 and E_2 as vectors which we may call *time-like* and *space-like* respectively. As such we can restate out definition of causality in terms of time-like and space-like vectors. Informally, two events are causally connected if they can be represented by a time-like vector.Just as easily we could say that two event are not causally connected if they are space-like separated

There were several other assumptions being made in the original result which we will mention briefly before stating the theorem itself. One of the more important ones was that of "locality". This concept also has a history of being redefined and modified to form other similar concepts that go into the proofs of Bell-like theorems. We will give the following definition.

Definition 2.6(Bell locality) *We say that an experiment is local if only causally correlated variables can have a statistical influence on the measurement outcomes of an experiment*

This is again quite vague and philosophical. In fact Bell himself([7]) wanted one to proceed with caution regarding this concept, especially given the heavy importance it has in the result that we will cover a bit further ahead. This is because while the idea of causal correlation can be given a quite robust definition as we have seen in Definition 2.5, "correlation" can have encompass a quite vast class relations that may not be taken for granted to be limited in any way by causality or similar ideas.

Another important assumption is that of hidden variables, meaning the fact that there are certain variables that condition the probability that a certain property will be observed upon measurement of an experimental ensemble.

Definition 2.7(Hidden variables) *We call a variable λ a hidden variable if for some random variable X, $P(x = a \,|\, X, \lambda) = 1$, and $P(x = b \,|\, X, \lambda) = 0$ for any other value b.*
The above definition means to convey the fact that for some random variable over an event space, the outcome of the measurement of a variable becomes deterministic in the presence of some hidden variable.

Theorem 2.3 (Bell's theorem) *No local hidden variables model can model Quantum Mechanics.*
Proof. The proof is quite simple, consider three random variables X^i_j, with i= 1,2 and j = a, b , c, such that $X^i_j = -X^k_j$ whenever $i \neq k$. This property is known as the variables being *perfectly anti-correlated*. From the assumption of Bell locality the following identity holds:

$$P(x^1, x^2 \,|\, X^1, X^2, \lambda) = P(x^1 \,|\, X^1, \lambda) P(x^2 \,|\, X^2, \lambda) \tag{26}$$

We ignored the usage of lower indices as a way to tell the fact that they are not relevant in this case. Any of the three variables works for the set of variables associated with both particles. Another thing we can extract from the assumption of hidden variables is that for the same particles, the probability of getting different results, given the same type of

measurement, is null.

$$P(x^1 = 0, x^1 = 1 \,|\, X_j^1, X_j^1 \lambda) = 0 \tag{27}$$

If we invoke locality again, we get:

$$P(x^1 = -1, x^1 = 1 \,|\, X_j^1, X_j^1, \lambda) = P(x^1 = -1 \,|\, X_j^1, \lambda) P(x^1 = 1 \,|\, X_j^1, \lambda) = 0 \tag{28}$$

But we have to note that the two situations are exhaustive over the sample space, which means that one of the two factors needs to be null. Now by a simple pigeonhole principle, we can see that the following inequality must hold:

$$P(X_a^1 \neq X_b^2) + P(X_a^1 \neq X_c^2) + P(X_b^1 \neq X_c^2) \geq 1 \tag{29}$$

Next, given the perfect anti-correlation between the variables, one can see that:

$$P(X_k^1 \neq X_j^2) = P(X_k^1 = X_j^1), \forall l, k (l \neq k) \tag{30}$$

From (29) and (30) we can immediately infer:

$$P(X_a^1 = X_b^1) + P(X_a^1 = X_c^1) + P(X_b^1 = X_c^1) \geq 1 \tag{31}$$

Now we came to the conclusion that a local hidden variable model must satisfy the above inequality, but we will show that this is not true for quantum mechanical systems. Let's consider a system made of two qubits in the Bell state: $|\Phi\rangle = \frac{|00\rangle + |11\rangle}{2}$, and the variables defined as follows:

$$X_a = \begin{cases} |0\rangle \\ |1\rangle \end{cases} \quad X_b = \begin{cases} \frac{1}{2}|0\rangle - \frac{\sqrt{3}}{2}|1\rangle \\ \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle \end{cases} \quad X_c = \begin{cases} \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle \\ \frac{\sqrt{3}}{2}|0\rangle - \frac{1}{2}|1\rangle \end{cases}$$

We left the sample space unspecified as we did up until now when it came to the variables because this has a general character. Now we can treat a single case to see that the above inequality, known as Bell's inequality is not satisfied by this quantum system.

$$P(X_a = X_b) = \left|\frac{1}{2\sqrt{2}}\right|^2 + \left|\frac{1}{2\sqrt{2}}\right|^2 = \frac{1}{4} \tag{32}$$

One can easily verify that the identity

$$P(X_a = X_b) = P(X_a = X_c) = P(X_b = X_c) = \frac{1}{4} \tag{33}$$

As such

$$P(X_a = X_b) + P(X_a = X_c) + P(X_b = X_c) = \frac{3}{4} \leq 1 \tag{34}$$

That means Bell's inequality is violated and this concludes our proof.

3 Games in fundamental QM

In this chapter we explore ways in which game theory can be used to prove results like the Bell theorem we have explored in the previous chapter, making the proof of nonlocality somewhat easier provided one is a bit familiar with game-theoretic notions, as a person reading this article would be. We also explore ideas of actual implementation of quantum games for solving interesting problems in physics. But first of all we would need to introduce what could have been introduced in the first chapter of this thesis but was left here as the author saw it fit given the usefulness of the notions introduced up until now in understanding the concepts that would follow. We talked about games with pure and mixed strategies, but we can consider another type of strategy, what would be called *quantum strategies*.

Definition 3.1 *We call a nonlocal game a tuple* $< X, Y, \mu, A, B, V >$

- X and Y are finite sets of questions
- μ is a probaiblity distribution over $X \times Y$.
- A and B are finite sets of answers for Alice and Bob respectively.
- $V : X \times Y \times A \times B \to \{0, 1\}$

We can see that these games could be modeled as a special case of Harasanyi games by considering X and Y as the sets of types, A and B as the actions within the states of nature and V can be considered an expected payoff function. One may take the last statement with doubt but all we need to consider on top of what I said to make it seem natural is the fact that the game is supposed to be a cooperative game, so the payoff is dependent on the queries and answers that both players give.

We have established now a familiar way in which to characterise nonlocal games. Now we would need to look at what makes them stand-out and that is quantum strategies.

Definition 3.2(Quantum strategies) *A quantum strategy is distirbution* $\sigma : (x, y) \to \langle \Psi | \mathcal{A}_x^a \otimes \mathcal{B}_y^b | \Psi \rangle$, *where*

- $\mathcal{A}_x^a$ and $\mathcal{B}_y^b$ are unitary transformations dependent on the query and answers available to the players.
- $|\Psi\rangle$ is a quantum state of entangled particles shared between Alice and Bob.

The above definition could be read as, a strategy for players Alice and Bob is a probability of them giving answers a and b to queries x and y.

Now another thing we have to consider is the fact that there are many ways in which will allow us to measure the *advantage* of nonlocal games of classical games is the notion of value of the game with which we became familiar in Chapter 1, but which in this case will of course be considered as a unique value for both players as it is essentially a cooperative game.

Definition 3.3(Quantum value) *We call the quantum value of a nonlocal game, noted with $\hat{\omega}(G)$, defined as:*

$$\hat{\omega}(G) = \sum_{(x,y)\in X\times Y} \mu(x,y) \sum_{(a,b)\in A\times B} \langle\Psi|\, \mathcal{A}_x^a \otimes \mathcal{B}_y^b \,|\Psi\rangle\, V(a,b\,|\,x,y) \tag{35}$$

We will see that this value, compared to that in certain cases can be bigger than what could be obtained with mixed strategies for the "reduced" game with no entangled quantum system available.

3.1 Mermin-Peres magic square game

The Magic Peres square is a thought experiment whose variation can be actually implemented to showcase nonlocality without actually appealing to Bell inequalities. The set-up, rules and goals of the game will be presented shortly.

The set-up for the Mermin-Peres square thought experiment that involves a source S that emits four particles, which are sent in pairs of two towards two observers Alice and Bob. Both Alice and Bob have a detector in which the particles received are introduced and a measurement is made out of six possibilities. The way in which the results are shows are through a display that is basically a square separated into 9 equal squares. The squares light up in two different color to give information about the measurement.

The goal of the game is for Alice to fill in a row and Bob a column in such a way that the number of green squares filled in by Alice is even, while the number of red squares filled in by Bob is odd. As such they each have twelve possible strategies that are hidden information from one another(Fig 3.1). The two of them cannot communicate or exchange information about the state of the game once it has started(we can imagine that they are too far apart, outside of each other's light cone).

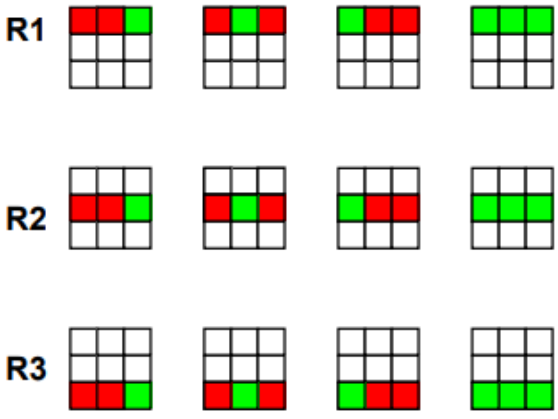


Figure 3.1 The set of strategies available to Alice(image taken from [10]).
We will now show that there is no classical winning strategy for this cooperative game. We can do that by making the appropriate pay-off matrices for a particular row assignment for Alice this game as follows(Fig 3.2-3.4)

		Alice(R1)			
		rrg	rgr	grr	ggg
Bob(C1)	ggr	0	0	1	1
	grg	0	0	1	1
	grg	1	1	0	0
	rrr	1	1	0	0

Figure 3.2.The payoff matrix of the strategies available to Alice when tasked with filling the first row, and the Bob is given the task to fill in the first column. We noted the red and green squares with "r" and "g" respectively, the order being from left to right and from up to down.

		Alice(R1)			
		rrg	rgr	grr	ggg
Bob(C2)	ggr	0	1	0	1
	grg	0	1	0	1
	grg	0	1	0	1
	rrr	1	0	1	0

Figure 3.3.The payoff matrix of the strategies available to Alice when tasked with filling the first row, and the Bob is given the task to fill in the second column.

		Alice(R1)			
		rrg	rgr	grr	ggg
Bob(C3)	ggr	1	0	0	1
	grg	1	0	0	1
	grg	1	0	0	1
	rrr	0	1	1	0

Figure 3.4.The payoff matrix of the strategies available to Alice when tasked with filling the first row, and the Bob is given the task to fill in the third column.

We limited our study to just the task of filling the first row for Alice because the others give similar results. What we can take just from the above matrices is that if we assume this to be the game-state from Alice's perspective once she has been tasked by the arbitrer to fill in the first row, the best strategy ignoring the rationality of Bob is to use strategy "ggg", but even then hew best chances are statistically 2 out of 3. When considering the full game, the best they can do is actually 8/9(see for reference [8]). Now we will show that there is in fact

a quantum solution to the puzzle such that Alice and Bob can win with a hundred percent confidence.

Theorem 3.1 *There is a winning quantum strategy for the Mermin-Peres magic square game*

Proof. Let's assume a pair of qubits being shared between Alice and Bob before going apart and starting the game, such that they are in the following state:

$$|\Psi\rangle = \frac{1}{2}(|0011\rangle - |0110\rangle - |1001\rangle + |1100\rangle) \tag{36}$$

Now the first two qubits are given to Alice, while the other two are given to Bob. We code the green color as "0" and red as "1" .Upon query involving which row and column respectively they have to fill in, Alice and Bob perform the following transformations on their qubits before measurement:

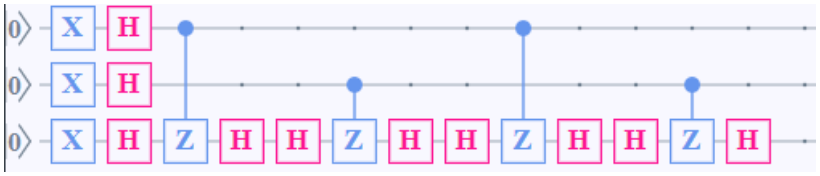
$$\begin{gathered} A_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} i & 0 & 0 & 1 \\ 0 & -i & 1 & 0 \\ 0 & i & 1 & 0 \\ 1 & 0 & 0 & i \end{bmatrix} \quad A_2 = \frac{1}{2}\begin{bmatrix} i & 1 & 1 & i \\ -i & 1 & -1 & i \\ i & 1 & -1 & -i \\ -i & 1 & 1 & -i \end{bmatrix} \quad A_3 = \frac{1}{2}\begin{bmatrix} -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \\ B_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} i & -i & 1 & 1 \\ -i & -i & 1 & -1 \\ 1 & 1 & -i & i \\ -i & i & 1 & 1 \end{bmatrix} \quad B_2 = \frac{1}{2}\begin{bmatrix} -1 & i & 1 & i \\ 1 & i & 1 & -i \\ 1 & -i & 1 & i \\ -1 & -i & 1 & -i \end{bmatrix} \quad B_3 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{bmatrix} \end{gathered} \tag{37}$$

Now for example if the query involved asking Alice to fill row 2 and Bob is asked to fill column 3, this would be equivalent to the transformation

$$(A_2 \otimes B_3)\,|\Psi\rangle = \frac{1}{2\sqrt{2}}(|0000\rangle - |0010\rangle - |0101\rangle + |0111\rangle + |1001\rangle + |1011\rangle - |1100\rangle - |1110\rangle) \tag{38}$$

Now if one takes the probability density of one of any of the states they can see that for example the probability for Alice of measuring $P(|\psi\rangle_A = |00\rangle) = 1/4$ and in fact this the case for all the usual "prefixes" of the strategies available classically to them. All that is left is to fill in the last part of the color scheme for each of them with a color that preserves the parity for their respective color assignment and this concludes the proof.The proof above combines elements of proofs found in [10] and [12].

Next we will implement a run of the magic square game in the Uranium Transylvania platform(https://uranium.transilvania-quantum.org/) for an experimental implementation of the game, which is a slight variation of the implementation found in [14].



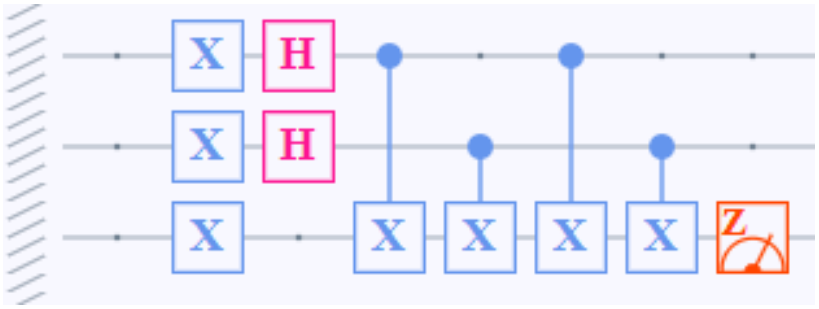


Figure 3.5. The implementation of the first row both in simple and extended form in terms of Clifford gates.

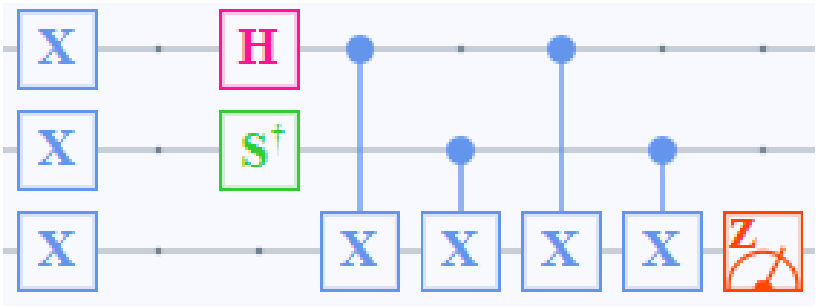


Figure 3.6. The implementation of the second row.

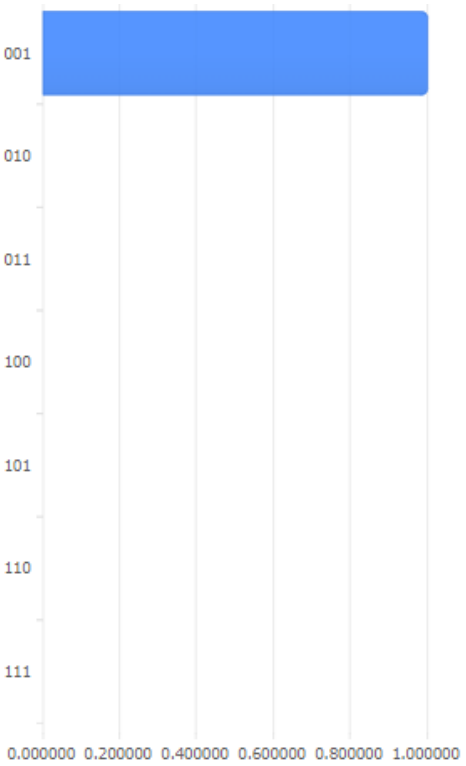


Figure 3.7. The result of measurement in the computational basis for all implementations. The horizontal axis represents the probability and the vertical axis the state.
As we can see, they agree with the theory.

3.2 More nonlocal games

In this section we will cover more briefly other games whose quantum strategies cannot be replicated classically. This is to form a better image of the subject and similar games will

be employed in later chapters for other purposes than show the nonlocality of quantum mechanics.

The CHSH game

The CHSH or short for Clauser-Horne-Shimony-Holt is agame in which the setup involves our established players Alice and Bob. The usual elements that are contained in the nonlocal game tuple are as follows. The values of queries and answers can be described by a function of the form: $f = \{X, Y, A, B\} \to \{0, 1\}$, meaning they all have to take one these two values.

$$\mu(x, y) = \frac{1}{4} \tag{39}$$

Meaning the distribution of the queries is also a constant function or that each pair is equally likely.

$$V(a, b \,|\, x, y) = \begin{cases} 0, \ a \oplus b \neq x \wedge y \\ 1, \ a \oplus b = x \wedge y \end{cases} \tag{40}$$

It's quite easy to see the fact that the classical value of the game is just 3/4. The quantum value of the game however is $\hat{\omega}(G) = \cos^2(\pi/8)$. This game could also be used to put forward the idea of nonlocality of quantum systems.

XOR games

The next class of games we will look at is probably the most researched type of nonlocal games and the reasons will reveal themselves throughout the rest of this thesis. When it comes to the values of the answers must be either 0 or 1 so a function $g = \{A, B\} \to \{0, 1\}$ can describe them. Now unlike the CHSH game, the queries are described by a different kind of function, that takes as input the cartesian pair and returns a single value of the set $\{0, 1\}$. In other words a function $f = \{X \times Y\} \to \{0, 1\}$. The "verifier predicate" has the following form

$$V(a, b \,|\, x, y) = \begin{cases} 0, \ a \oplus b \neq f(x, y) \\ 1, \ a \oplus b = f(x, y) \end{cases} \tag{41}$$

We did not specify the distribution μ because it is uniform on the number of pairs of queries possible.

We can remebmer from chapter 1 that strategies that give the value of the game are called optimal strategies. Now this optimality is in fact a quite deep fact, putting a limit or boundary on how much can it differ from the classical case. In that way we say that the value of a particular nonlocal game is equal to the Tsirelon's bound of that particular system.

4 Games in quantum complexity theory

In this chapter we study both classical and quantum complexity classes. We will see that games can be very useful descriptive tools for different quantum complexity classes given the fact that they represent a homogeneous way to represent the different problems that make up the said complexity classes.

4.1 Some introductory notions for complexity theory

One of the first things we have to establish is what does a class mean, but before that we need to become familiar with the idea of *formal language*.

Definition 4.1 *A formal language is an alphabet Σ together with a grammar $\mathcal{G}$ that produce well-formed formulas based on the alphabet*

This definition is of course highly informal as stated, but the reason is the fact that it is not the goal of this thesis to go into details about such a definition, the details becoming more clear as we move along. We encourage the reader to look into a dedicated textbook that covers such a topic such as ... We will give an simple example to make things more clear. An example of language is

$$L = \{a^n \,|\, n \in \mathbb{N}\} \tag{42}$$

In this case the alphabet is made up of a single symbol $\Sigma = \{a\}$ and the grammar basically has a single production rule $w \to wa$, where w is a word from the language $w \in L$. We could as such rewrite the language as a set in terms of discrete members(words) as:

$$L = \{a, aa, aaa, aaaa...\} \tag{43}$$

Now that we have become familiar with what a formal language is, we can turn back to what we wanted to investigate, that being aspects of complexity. The reason why we introduced the notion of languages is because the problem of finding with a limited amount of resources, whether or not some element is member of a language, also known as the membership problem, is at the core of complexity theory. This may not seem particularly obvious as we move on to treat different problems and get more familiar with what complexity theory is about, but a way in which I may motivate it is by saying that formal languages have language in the term, unlike regular languages they are simply syntactic constructions with no indented meaning, so what classifies as a language may be much broader than one may on first ponder conclude. Now another thing that we have to keep in mind is the fact that there are problems which are more or less as difficult as each other, given a precise definition of what more or less means, makes us able to put problems in classes, which we will come to call *classes of complexity*. Now the way in which we can asses whether or not two languages are in the same complexity class is through what would be called "many-one reduction", but before that we need to define another thing, known as the Kleene star or given the familiarity with the concept of operator, the Kleene operator.

Definition 4.2(Kleene's operator) *Given a finite set of symbols S, Kleene's operator acting on it, denoted as S^*, transforms it into the set of all finite strings definable on the set of strings, together with the empty string $\epsilon = S^0$.*

$$S^* = S^0 \cup S^1 \cup S^2 \cup S^3... \tag{44}$$

Definition 4.3(Many-one reduction) *Given two languages L_1 and L_2, defined on the alphabets Σ_{L1} and Σ_{L1}, we call a map $m : \Sigma^*_{L1} \to \Sigma^*_{L2}$ such that $w \in L_1$ iff $m(w) \in L_2$, the many-one reduction of L_1 to L_2, written as:*

$$L_1 \leq_m L_2 \tag{45}$$

As said, different classes of problems can be delimited by different types of many-one maps.

Many-one reduction is not the only way in which we can delimitate the complexity classes however. There is a less abstract way to do so, that being the "big O" system or notation.

Definition 4.4 *We say a function $g(n) = \mathcal{O}(f(n))$ if for asymptotic n, there exists a constant c such that $g(n) \leq cf(n)$* . We will give an example below(Fig 4.1)

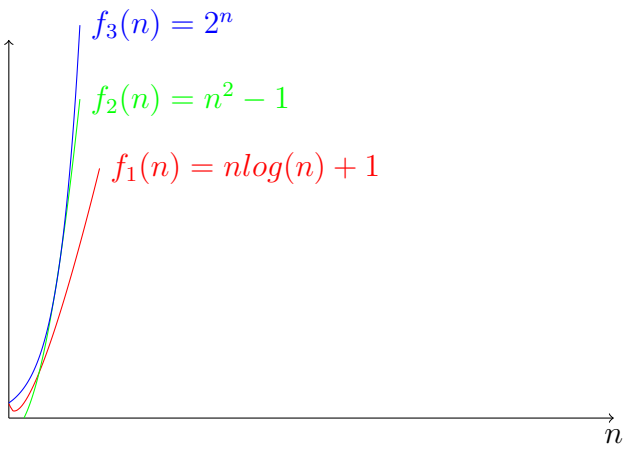


Figure 4.1. The representation of the asymptotic behavior of three different functions, given the input n.

If we were to use the big O notation, what we would say is that $f_1(n), f_2(n) = \mathcal{O}(f_3(n))$, which could be read as saying, the functions $f_1(n), f_2(n)$ grow no faster than $f_3(n)$ given asymptotic n. Another thing one could immediately infer is that the relation $\mathcal{O}(f_1) \leq \mathcal{O}(f_2) \leq \mathcal{O}(f_3)$ holds. That is to say, the big O notation gives us a measure on how the different functions compare to each other asymptotically. For details on such topics I direct the reader to Aror and Barak([18]).

4.2 Classical complexity classes

Now we may move on to study actual complexity classes.Before starting however, we will introduce another notion, that of

Definition 4.4(The class P) *We say a the membership problem of a language is in class* **P** *if the characteristic function* $f(x) = \mathcal{O}(n^c)$*, for some constant* c*.* In other words we can define the class **P** succintly as:

$$\mathbf{P} = \bigcup_c \mathcal{O}(n^c) \tag{46}$$

Examples of problems in $\mathcal{P}$ is what is called the PATH problem, which is the problem of deciding whether or not there is a path between two nodes in a directed graph. Written as a language:

$$PATH = \{(G, a, b) \,|\, \text{G is a directed graph with a path from node a to node b}\} \tag{47}$$

We give below(Fig 4.2) an explicit member of the language PATH.

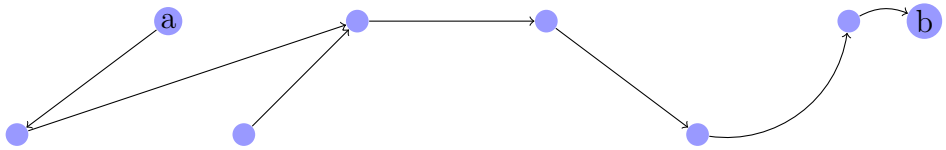


Fig 4.1. A directed graph with a path between vertices a and b.
Now we will give an informal program that decides the language PATH in polynomial time. *PATH program*: First of all imagine that we are working with a machine that can access any segment of the input string and it can mark, delete or write the said input(this is a very crude description of a computational model like a turing machine or register machine).

1. Initialize vertex x, if it is unmarked, mark it and go into the body of instructions below, otherwise go straight into the body of instructions below:
    - Go over all vertices that are at an edge distance from vertex x, (x,y) and if they are unmarked, mark them
    - For each y in the variable above, while y exists, x = y and jump to instruction 1
    - Otherwise jump to instruction 2
2. List all the marked vertices and:
    - If b is in the list accept
    - Otherwise reject

All we have to do now is initialize the program above with $x = a$ and look at the time complexity of the program by looking at how many appeals each instruction makes in the worst case scenario. As such, the first body of instructions makes the makes at most n runs, where n is the number of vertices in the graph. Instruction 2 makes the machine check the entire list of marked vertices so again at most n operations are performed by the machine. We thus have a worst case scenario of $2n$, and $\mathcal{O}(2n) = O(n)$ so we can see that it is as the bottom of the polynomial class **P**, which is what we wanted to prove.

The polynomial class is by far the most useful in real applications of programs, and it is considered that programs that it is the only feasible class of problems that we can decide

in practice. As such, the hope is for there to be programs that can solve as many problems as possible in polynomial time. There is an enormous body of literature that covers these topics, including other examples of other languages that are in this class. For such purposes the author directs the reader towards Sipser[19].

Another important time complexity class is **NP**, which stands for "nondeterministic polynomial", which as the reader may have intuited given the statement we made about the polynomial class as being the real practical class, it is not feasible to solve problems that are in **NP**, unless the two of them are equivalent which is a deep unsolved problem in complexity theory. Now what "nondeterministic" means is that **P** is a deterministic complexity class, which again means that every single program is run in a singular way, meaning for any single relevant parameter like bit of string being currently read and instruction set in which said bit of string is read, the machine can only give a single operation in response to them. That is to say, when seen as a function, it can be taken as a single output function. This is not the case for nondeterministic machines which can have more than one continuation of the program given a bit of string and instruction set being currently in.

Another characterization of the class is that using verifiers, **NP** being what would be called a complexity class *verifiable in polynomial time*. We can thus see the fact that there are many ways. A verifier is simply a program that can tell whether or not some solution given to it is correct or nor, given a polynomial set of set of operations as a function of the size of the solution. We will give a more formal definition of a verifier as follows

Definition 4.5 *Given a language L, a verifier for L is an algorithm V such that*

$$L = \{x \,|\, \text{V accepts (x,c) for some string c}\} \tag{48}$$

The string "c" is called a certificate or what we have mentioned before, a proof or solution. We have to be careful however what does it mean then for a verifier to be polynomial, is it polynomial as a function of the length of x, c or both? The answer is, it is a function only of x. This makes sense if we think about it, because otherwise we would need to somehow be able to give an answer to the problem of what is the length of c given some problem to be solve which would be an unnecessary complication. We can now give a short definition for this new class, based on the definition of a verifier.

Definition 4.6 *Given a language L, it is a member of the class* **NP** *if and only if it has a polynomial time verifier.*

Now there are many problems of interest that are for all we know as of now due to the lack of better algorithms, in the class NP. One interesting problem is again related to graphs and it is the problem of *cliques*.

Definition 4.7 *Given a graph $G =< V, E >$, a clique is a subgraph of G' where given any two vertices $v_m, v_n \in G'$, then $(v_m, v_n) \in E$.*

This is however not the focus of the problem we are going to study, as it needs to be made

specific somehow to the way in which complexity increases with increase in size of. As such we introduce the definition of a k-clique below.

Definition 4.7 *A k-clique is a clique with k vertices*
Now the language that is in **NP** is the following:

$$CLIQUE = \{(G,k) \,|\, \text{there is a k-clique in graph G}\} \tag{49}$$

Now we can give an hand wavy algorithm or verifier by which we can show that CLIQUE is indeed in **NP**. The algorithm goes as follows:

1. V checks the input ((G,k),c), where c is the k-clique, to see if all vertices of the k-clique are in G
   - if true, jump to instruction 2.
   - otherwise reject
2. V compares all edges in c with the edges in G to see if they match
   - if true, accept
   - otherwise reject

Now to make the complexity analysis of the algorithm V, we see that the first instruction is polynomial as a functon of (G,k), since c must be $\mathcal{O}(n^p)$, for some constant p, as a function of the input and the algorithm only runs instruction 1 once. Then it runs instruction 2 just once as well and in the worst case scenario it is $\mathcal{O}(n)$, which gives us $\mathcal{O}(n(n^{c-1}+1))$ which is of course polynomial on the size of the graph/clique, which is what we wanted to arrive at.

4.3 P vs NP

We need to explore another central idea in complexity theory, that of *class complete problems*. Class complete problems are languages for whom if we could find an algorithm that decides them, we can find an algorithm that decides all problems in the said class with at most a polynomial additional complexity. This means for example that if we could find a polynomial time algorithm for an NP-complete problem, we could reduce any single NP problem to a P problem. This is known as the the P vs NP problem and as mentioned in the previous section, it is arguably the most important problem in complexity theory. The reason why it is so important is because many things which now require human aid to be done, could in principle, assuming a few things, be done by computers. For example, computers might be able to find proofs for theorems so arguably, mathematicians would not need to do that anymore. Consequences such as that stated above would be extremely drastic and would change forever the intellectual landscape that we know as of now. However, most people do not actually believe that $\mathbf{P} = \mathbf{NP}$ or even if it is discover that they are equivalent classes, that the consequences would be so dramatic or immediate. For more on such discussions I

recommend the reader to check [20] and [21].

What we neet to do now however is work out what does it mean for a problem to be hard first of all, and we will concentrate strictly on the class **NP** as we became somehwat familiar with it by a stretch of the meaning.

Definition 4.8 *We say a problem membership problem for a language L_h is* $\mathbf{NP-hard}$ *if given any other language in $L_n \in \mathbf{NP}$ there is a polynomial time many-one reduction of L_1 to L_n, written as:*

$$L_n \leq_p L_h \tag{50}$$

What one needs to be careful about is the fact that this is that L_h is not restricted in any way in terms of complexity. That means it might as well be a language from the class of exponential time problems **EXP**, or even the halting problem for Turing machines for all we know. This might be considered to not be a very useful resolution, because after all we just said that **P** and **NP** are the only classes that are at least in some sense efficient or somewhat manageable in practice. As such it would seem to be of no great help that the algorithm for deciding a language in some higher complexity class could be used to solve any problem in **NP**. For that reason, what people working in complexity theory are looking for, are solutions for a subset of the set of $\mathbf{NP-hard}$ languages, that being the set of $\mathbf{NP-complete}$ problems.

Definition 4.9 *We say a language L_c is* $\mathbf{NP-complete}$ *if it is* $\mathbf{NP-hard}$ *and $L_c \in$* **NP***.*

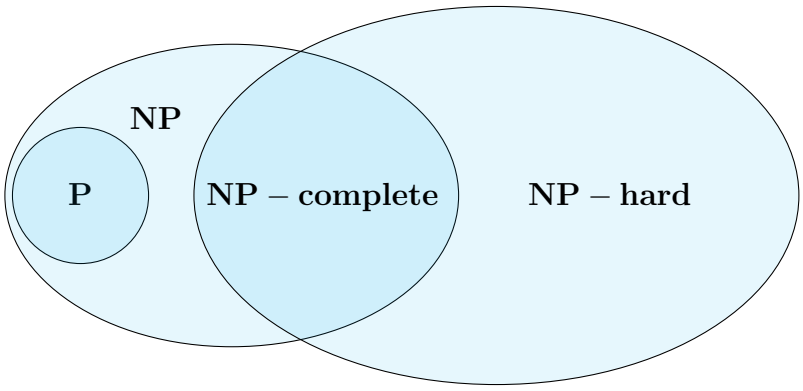


Figure 4.3. A Venn-Euler diagrams representation of the "NP classes".

As we can see by looking at the above figure, we are in other words restricting ourselves to finding polynomial time algorithms for problems that are both $\mathbf{NP-hard}$ while at the same time being in the class of **NP** problems or succinctly we are looking at the NP-complete region on the diagram. If such an algorithm was found as we said at the beginning of this section, the consequences would probably be very significant.

The question the reader may have in mind now is whether or not have we found any $\mathbf{NP-complete}$ problems until now. The answer is yes, quite a few of them in fact from different parts of mathematics or real situations such as problems related to real games. The first NP-complete problem was established by the famous Cook-Levin theorem which

is known as the SAT problem, theorem proved independently by Cook and Levin. For reference see [22], [23]. We will look however at a problem that we became familiar with already, that being the CLIQUE language, which was established to be NP-complete early in the introduction of NP-completeness(see [24]). In fact CLIQUE is NP-complete, which we will encourage the reader to see for themselves in one of the cited sources on the subject.

4.4 Quantum complexity classes

The reader may be wondering where do games fit in this new theory, and the answer again is more related to the quantum versions of the theories.This section is intended to be structured as a parallel to the previous one, as it intends to cover new types of complexity classes, based on quantum algorithms now, instead of classical algorithms(deterministic or not) as we have seen before. What we first need to establish is the fact that there are multiple ways in which we can make the jump from classical to quantum in the sense of the equivalent definitions of real world efficient and non-efficient quantum languages.

Speaking of languages, we know that we could have equivalated the languages with the problems in the classical(deterministic) classes of complexity. However, that is not the case for quantum complexity classes, the reason being that the problems are what will be known throughout this paper now as *promise problems*.

Before covering that however we need to introduce a couple of other notions that will be useful from here on out.

Definition 4.10 *We call a quantum register R, a quantum system such that the possible transformations on the associated hilbert space $\mathcal{H}_R$ is identical with the space of density operators $\mathcal{D}(\mathcal{H}_R)$.*

This means to convey a system in which we hold quantum information when we do computations, just like with classical registers in a classical computer

Now another, even more important concept we need to introduce is that of a *quantum channel*.

Definition 4.11 *Given two quantum registers X_1, X_2, with their associated Hilbert spaces $\mathcal{X}_1, \mathcal{X}_2$, we call a trace preserving linear map $\Phi : \mathcal{L}(\mathcal{X})_1 \to \mathcal{L}(\mathcal{X})_2$,* a quantum channel between the two registers.

Definition 4.12 *Given the language $L = \{0,1\}^*$, we call a pair $P = (S_{yes}, S_{no})$, where $S_{yes}, S_{no} \subseteq L$, a promise problem.*

As we can see, the reason why we cannot make the classes equivalent to languages is because they can be just subsets of said languages called the yes-instances and no-instances of the problem. As such, the classical problems can just be considered as a special case of problems where the yes and no instances exhaust the languages or formally where $S_{yes} \cup S_{no} = L$. We can also immediately infer the fact that there many ways in which we can establish

complexity classes based on promise problems due to the more relaxed contraints on what is a promise problem.

Now to address the way in which games are implemented in quantum complexity theory, the classes themselves can be described as games. But as we have seen until now, slightly different games can have strong influences on the values or other important aspects of the games and as such there are many ways in which games can describe the complexity classes. The main cases that we will explore can be divided into non-interactive and interactive provers games. We will begin with the study of the former, but firstly we have to make a very brief description of what makes up a prover games. A prover game is made up of a verifier V as we have become used to from the previous chapters and a set of provers $P = \{P_1, P_2, ...P_n\}$ which are trying to maximize the value of the game. The game is based on exchange of registers that contain the query from the verifier and the answers from the provers.

The QMA class

This class is one in which there is a single prover and it it basically the quantum equivalent to **NP**, while the equivalent of **P** is a class known as **BQP**. It is easy to see why, because we've seen in the previous secction that nondeterministic time can be described by what a verifier can do in poly-time. As such, this class is described by a non-interactive provers game. On the other hand, as we mentioned previously, quantum complexity classes are described by the promise problems they contain so we can give a definition of **QMA** as a game where the certificates need to be polynomial in the length of the problem.

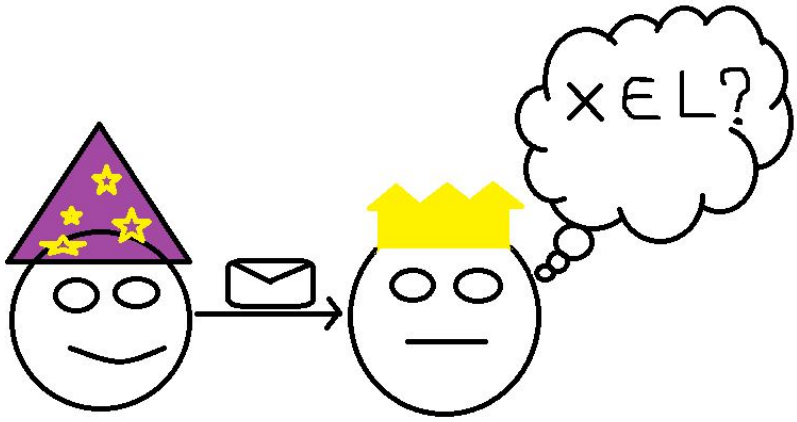


Figure 4.4. Depiction of the Merlin-Arthur game.

We should note that the role Arthur plays is passive, until it is time to verify the proof certificate, so we can think of it as simply being played as a first move by Merlin as depicted in Figure 4.4, and then the verifying stage of Arthur takes place.

Another essential aspect we need to mention is that the probabilistic nature also forces us to certain constraints that need to be fulfilled or in other words some conditions which need to be met in order for the certificate to be considered reliable. We will call such conditions *completeness* and *soundness*.

Definition 4.13 *We say a promise problem* $S = (S_{yes}, S_{no})$ *is in the class* $\mathbf{QMA}_{\frac{2}{3},\frac{1}{3}}$, *if there*

exists a polynomial verifier V such that:

- $\forall x \in S_{yes} \cup S_{no}$, $V(x) : \mathcal{L}(\bigotimes_k |\psi_k\rangle) \to \mathcal{L}(|\psi\rangle)$, which we will denote with Φ_x.
- $\forall x \in S_{yes}$, $\omega(\Phi_x) \geq \frac{2}{3}$. We call this the completeness condition.
- $\forall x \in S_{no}$, $\omega(\Phi_x) \leq \frac{1}{3}$. We call this the soundness condition.

The above definition intends to capture the idea that in case Merlin is malevolent and wants to deceive Arthur when it comes to the validity of the answer to some query, Arthur must be unable to figure out about the deceit in only 1/3 out of all cases. This weakening from the deterministic classes leaves an open problem whether or not there could ever be a *syntactic characterization* of these classes(see [27] for reference).

Now as we have said, there is an equivalent to complete problems from the classical complexity classes, in teems of complete promise problems. Examples include the quantum version of the 3SAT problem, or simply the *quantum 3SAT promise problem*. We now move on to what is known as interactive quantum provers classes.

The QIP class

This class is very similar to **QMA** but unlike it, the game played between the verifier and the prover can take more rounds, where the verifier can send queries to the prover, in the form of registers. This leads to a significant increase in terms of what problems are included inside the class, depending on the amount n of rounds one considers, which is written as $\mathbf{QIP}(\mathbf{n})$. For example, it is easy to see the fact that the identity $\mathbf{QMA} = \mathbf{QIP}(\mathbf{n})$ holds. Another interesting fact is that it has been proven the fact that the class can be reduced to just thee rounds, or formally $\mathbf{QIP} = \mathbf{QIP}(\mathbf{3})$(see [22] for reference). The reader may also wonder how does it compare with classical classes and the answer is given in [23], that being the fact that $\mathbf{QIP} = \mathbf{PSPACE}$. The class **PSPACE** is a what is calle a space complexity class which means the fact that it is the class of problems that require polynomial amount of tape as a function of the input to compute by a deterministic machine.
We can now give a definition of **QIP** in terms of languages and conditions as we did for **QMA**.

Definition 4.13 *We say a promise problem $S = (S_{yes}, S_{no})$ is in the class $\mathbf{QIP}_{\frac{\mathbf{2}}{\mathbf{3}},\frac{\mathbf{1}}{\mathbf{3}}}(\mathbf{m})$, if there exists a polynomial verifier V such that:*

- $\forall x \in S_{yes} \cup S_{no}$, $V(x) : \mathcal{L}(\bigotimes_k |\psi_k\rangle) \to \mathcal{L}(|\psi\rangle)$, which we will denote with Φ_x.
- $\forall x \in S_{yes}$, $\omega(\Phi_x) \geq \frac{2}{3}$. We call this the completeness condition.
- $\forall x \in S_{no}$, $\omega(\Phi_x) \leq \frac{1}{3}$. We call this the soundness condition.

The QMIP* class

This class is a generalization of the previous classes in that it allows for multiple provers that cannot exchange registers but which can share unlimited previous entanglement, similar

to the nonlocal games that we have previously covered. This class is extremely even when compared to the other classes, in fact a recent article([30]) showed the following remarkable result:

Theorem 4.1(Ji, Natarajan, Vidick, Wright & Yuen) *The identity* $\mathbf{QMIP}^* = \mathbf{RE}$ *holds.*

We will discuss what the class of recursively enumerable languages or **RE** is about in a bit, but we first have to mention that the proof of the above statement can be found in the original article([30]) since it would be much too vast for this thesis, however we can still appreciate the profound implications of this theorem, being certain that expository works on the result will follow and in fact some could be found as of the writing of this thesis. It could be said that this single article was the reason for this thesis, as it became instantly interesting just by the title, given my familiarity with the subject of computability theory.

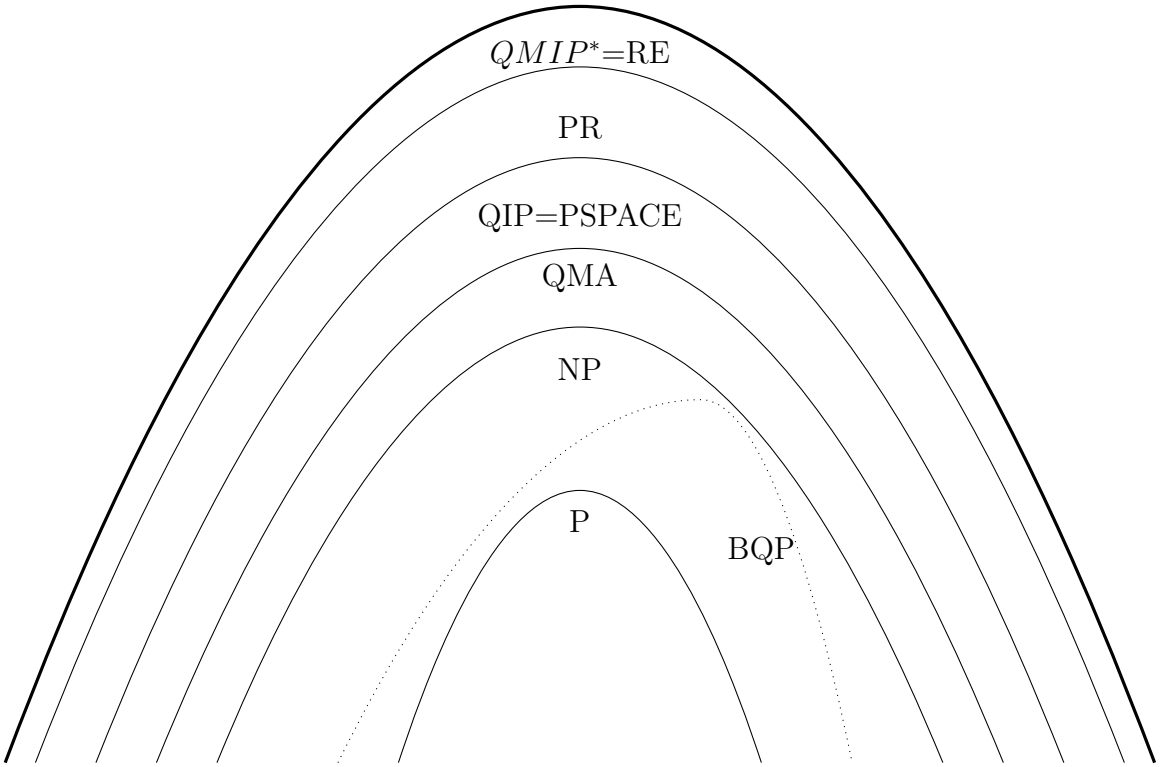


Figure 4.5. The hierarchy of complexity classes mentioned until now(and a couple of new ones) ordered by inclusion(the image is an adaptation from Sardina[34]).

The classes which have not been mentioned until now are **PR** and **RE**. The former was introduced by the logician Skolem in 1922([15]) and it represents the closure of functions that can be defined using for-loops in any universal programming language(Python, C, etc,). To continue this explanation in terms of programs to keep things easy, the class **RE** is the class of those functions for which our Python script will terminate if the function is defined on that value but for which we do not know the behavior of in case the function is not defined on the input. An important problem to put the classes into perspective is the famous *halting problem*, which Turing proved unsolvable in 1936([15]). This problem is in the class **RE** and it is very easy to state in terms of Python or C++ or most other programming languages.

The problem basically says "will some random program p in Python terminate given some random input i?". This problem was as said proven an unsolvable problem as surprisingly as it may seem to a reader which first comes across these concepts. Now I have to clarify that this does not mean we can't solve this problem for some particular set of programs, in fact we always "solve" the problem partially when we see our program return some answer to us. The unsolvability is related to the problem in general, given any program in Python and any input for it.

Now we can move on and mention a very interesting corollary of the theorem 4.1 is that it offers a solution to the famous problem in fundamental quantum mechanics known as *Tsireslon's problem*

Definition 4.14 *Are the game values $\hat{\omega}_c(G)$ and $\hat{\omega}_t(G)$ the same in general for nonlocal games? Where:*

- $\hat{\omega}_t(G) = \sum_{(x,y)\in X\times Y} \mu(x,y) \sum_{(a,b)\in A\times B} \langle\Psi|\, \mathcal{A}_x^a \otimes \mathcal{B}_y^b \,|\Psi\rangle\, V(a,b\,|\,x,y)$
- $\hat{\omega}_c(G) = \sum_{(x,y)\in X\times Y} \mu(x,y) \sum_{(a,b)\in A\times B} \langle\Psi|\, \mathcal{A}_x^a \cdot \mathcal{B}_y^b \,|\Psi\rangle\, V(a,b\,|\,x,y)$

We are familiar with as relation (35) which we simply called the quantum value. However, as we can see, things are not so simple, because we have basically assumed to these two types of nonlocal games to be equivalent, which is true for finite Hilbert spaces(see [26]). The latter corresponds to the case where in case of considering the observables that Alice and Bob posses as expressible with a tensor product(or separable), we take them to commute, hence the index "c". As we said, theorem 4.1 answers this problem, and the answer is in the negative or in other words.

Corollary 4.1.1 $\hat{\omega}_c(G) \neq \hat{\omega}_t(G)$.

We will use the above corollary in what will be one of the main results of this thesis. Firstly however we will introduce a new notion which will be the subject of our result.

Definition 4.15 *We call a quasi-observable Q, an observable defined in terms of projective operators $\{P_1, ..., P_i\}$ such that an effective spectral decomposition for at least one of them, P_k, does not exist.*

What I mean by "effective" is "recursively enumerable".We can now prove the following theorem, which will contain multiple tools from computability theory which we will explain, but for a deeper understanding of them we recommend [33]

Theorem 4.2(Main result) *Assuming the commutative nonlocal games are feasible in practice, there exist quasi-observables.*

Proof. The proof will be structured in two parts, one to extract useful infromation from corollary 4.1.1 and the second will be what I would call "elephant in the room argument" to put the reader at ease when it comes to what follows, for it will be a quite easy journey. First

of all the tensorial and the commutative games are different types of games in one important aspect, in the commutative games unlike in the tensorial games, the provers may exchange infinite entanglement before the game starts. This will prove to be an important difference as one may in fact expect, but we must be careful not to jump straight to conclusions about the fact that $\hat{\omega}_c(G)$ and $\hat{\omega}_t(G)$ are different just because of this fact. Also, in theorem 4.1, the class **QMIP*** is in fact a class of games of the tensorial type, not the commutative type, so the theorem is about tensorial games. Nextly, similarly to how the games in the class **QIP** can be reduced to just three rounds, in the games from the class **QMIP***, we can assume without loss of any important resource that the number of provers is just two. Now we will say something that will allow us to skip a lot of formalities by saying that the values of the games $\hat{\omega}_c(G)$ and $\hat{\omega}_t(G)$ represent the "confidence" the provers of those games have to convince the verifier V that something is or may not be the case. We will also assume that the confidence of the provers is related to the amount of entanglement they share, so that in a tensorial game they have small confidence and as the game progresses their confidence increases. For provers in commutative games however, they can share infinite entanglement so their confidence is maximal at the beginning of the game. Now let's imagine a game in which we have verifier V and two pairs of provers, one pair of tensorial provers which we will call P_t and one pair of commutative type provers we will call P_c. Now let's assume the two types of games are equivalent and if that were the case, the presence of one of the two pairs of provers would be redundant by what we said about the reducibility to just two of them. If that were the case then as what we said about their start of the confidence level, the pair P_t will have low confidence and the pair P_c have maximal confidence. Now by the soundness and completeness conditions on the games, which we introduced in definition 4.13, and the fact that they are equivalent, the two pairs have to arrive at the same level of confidence. Now by assuming they are equivalent, it also means due to the theorem 4.1 that both pairs will be recursively enumerable. Now an important thing to note is that if two problems are both recursively enumerable, the reunion of them or solving both at the same time is also recursively enumerable. This is known as a closure condition for recursively enumearable languages. But what that means is that there could be a program that can simulate the interaction of the verifier with both pairs of provers, which will converge to the same value. However, this means that this would actually make the program able to solve the halting problem which is a contradiction, so our initial assumptions that the two types of games are the same is false. Now one may wonder, does it mean one class of games is not as complex as the class **QMIP***? Well we know due to theorem 4.1 that the tensorial are that complex, so what would be left to assume is that the commutative games are less complex. This is false however, because in finite dimensions the games are identical so the commutative games are not in the class **QMIP***, but are in fact more complex. This finishes our first part of the proof.

The second part is what I was referring to as the elephant in the room argument and it relies on assuming that we can look at the game values as we would look at some functions. Functions on what you may wonder, well of whatever we have on the right side of the

relations in Definition 4.14. We can now pick the function apart and study the complexity by parts. You may agree with me after getting familiar with what complexity refers to, that the complexity of the sum function is not very impressive, and neither is that of products following that line, so they cannot be the "source" of complexity of determining the game values for commutative games, which we say by the first part of the proof that it is not even recursively enumerable. What may seem suspicious is the factors of the verifier function but I will remind you that the verifier is a very economical person and only verifies for a polynomial time on the size of the input, so we have to cross the verifier too as a source of complexity. Next the measure function $\mu(x, y)$ may be the source of troubles but for that I will bring up an axiom for measures which is called the countable additivity condition, which basically translates to us as to say that a search operation performed by some program will eventually approximate it as much as we like, hence it's in the class **RE**, so it is not it. We finally have to adress the "elephant in the room", from where the name I gave to the argument came, that being the eigenvalues $\langle\Psi|\,\mathcal{A}_x^a \cdot \mathcal{B}_y^b\,|\Psi\rangle$. We know the products themselves must be recursively enumerable and as such we must conclude that at least one of the observables Alice or Bob have in possession must not have an effective description, or formally for some answer q, and query x, $A_x^q = Q$. This concludes our proof.

Some consequences

The consequences of the above theorem are many and very interesting, but before we mention that we need to mention the strong connection of our work with work of Murray and von Neumann([41]), where they have classified all possible observable algebra (as bounded operators on hilbert space): von Neumann algebra of type I,II, III. The division of a quantum system into two independent subsystems permits factorizing von Neumann algebra of type I (tensor-product model). In this algebra specific to non-relativistic quantum mechanics there exist minimal projections for observables. Von Neumann algebra of type III describe relativistic quantum field theory where observables are not localized in bounded regions of space-time. Von Neumann algebra of type II is an intermediate case between the two extremes. Out quasi-obsrvables model is related with Von Neumann algebra of type II and III, that have no minimal projections.

Now moving on to interesting consequences. One immediate one would be the following

Corollary 4.2.1 *A model of the universe with quasi-observables violates the Tsirelson correlations with Bell conditions.*

This fact follows from the above theorem in an almost non-constructive manner as it is based on the fact that quasi-particles would be based on type II or III models and experimental evidence we have collected until now suggests that we are actually living in a type 1 universe.In other words, the Bell inequalities of the quantum systems which we investigated in chapter 3 are violated. Models of this kind have been studied(see [42]) being known as *no-signaling* models. We can see a representation based on the *strength of correlations* in Figure 4.6 below, in which a quasi observable model would lie somewhere beyond the purple circle's border. We should note that the image portrays values obtainable

for the CHSH inequality, so not through game-theoretic notions and as such I recommend ignoring the absolute values that appear on the figure and just concentrate on the fact that they do differ, and as we see the absolute value of $2\sqrt{2}$ is indeed violated in a no-signaling model of the universe. This is one reason why we should probably believe a quasi-observable model is not the type of universe we are living in.

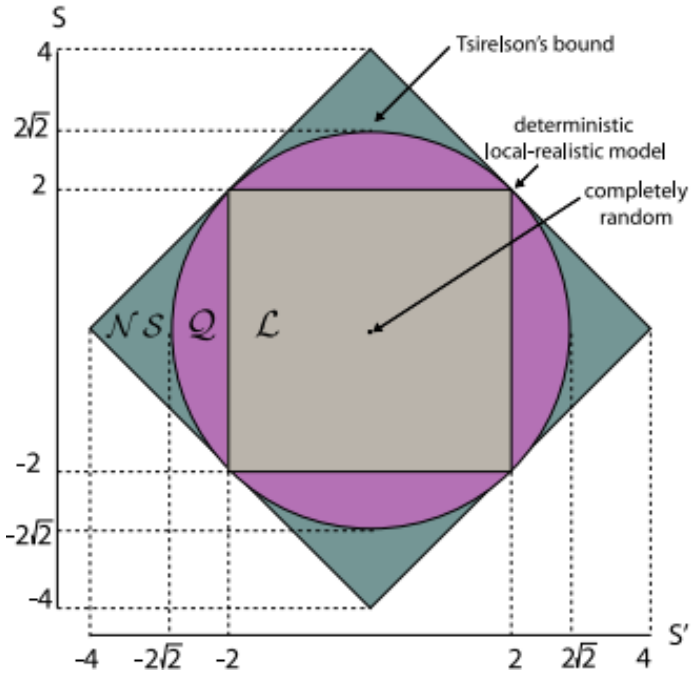


Figure 4.6. The different limits of correlations for different types of models. The purple region or Q is what we think our universe is like.

Another very suggestive reason why we should reject the assumptions of the theorem as being applicable to our universe is related to randomness. Firstly we would have to bring up a concept from mathematical logic known as the Church-Turing thesis, which informally says that any effective means a machine has to solve a problem, are equivalent to what a human could do in principle given countable time. Now going back to randomness. When I say randomness I refer to a technical concept, that of Martin-Löf randomness(see [35] for reference), which informally means that given a Martin-Löf random string, there is no effective way to identify a systematic pattern in the string. In other words, from an effective perspective, the string is random, hence the inclusion of randomness in the concept. Now we can state a corollary of theorem 4.2.

Corollary 4.2.2 *Assuming the Church-Turing thesis and the existence of some quasi-observable Q, from the point of view of an observer, the eigenvalue of Q is random.*

This is due to the fact that by Church-Turin's thesis, whatever we or our machines could do could not tell apart the result of a measurement using somehow the said quasi-observable and some random event.These facts may seem very interesting philosophically the reader may think and may also raise the question is there anything we could actually take from it. The answer I incline to is yes, the above corollary would seem to suggest that there are what we would maybe call *physical Martin-Löf tests*, which would give us insgight into

both the actual nature of randomness in our universe and means to falsify at least some quasi-observable models.

5 Games in quantum information theory

Games have been heavily used in quantum information theory, especially when it comes to cryptography and security protocols. In fact, the author considers that security protocols can be defined without loss of generality as a game. We will be looking at a few notions regarding what does a protocol entail and specifically we will be looking at a security protocol called a Byzantine agreement protocol and investigate whether or not a specific type of attack called a "man-in-the-middle" attack could be used to hack a servers-clients network.

5.1 Quantum cryptography and Byzantine games

As mentioned in the introductory word to this chapter, we can generally consider a protocol in the form of a game.

Definition 5.1 *We call a cryptographic protocol a game* $P = \langle N = \{A, B\}, (\mathcal{A}_i)_{i\in N}, (u_i)_{i\in N}\rangle$.

We will try however to use more informal language in this chapter as we already became familiar with the formal conceptions of a game, making us able to reason more intuitively about them. Most protocols are referred to in literature as a game between Alice and Bob and both of them can take some actions(denoted by A_i) which are mostly about sharing keys. For example, there are what are called key exchange protocols or key distribution protocols which we will denote as KD, where the most important actions are making keys and exchanging them(see [36]).

One simple example and one of the oldest is **the Diffie–Hellman protocol** Is a KD protocol in which Alice and Bob agree beforehand on some large prime number p and some generator $\langle g\rangle$ for the multiplicative group $\mathbb{Z}_p$. Afterwards, they both pick random numbers modulo $\mathbb{Z}_p^*$ which we will call a and b respectively. Now Alice and Bob exchange public keys g^a and g^b, which they use to make the private key $K = g^{ab}$. This type of protocol cannot be broken effectively by a classical computer(it is not in the class **P** as a problem), but it is in the effective quantum class **BQP** due to Shor's algorithm([37]), so it will not be a safe method of encryption when quantum computers will have enough qubits.

Based on the mentioned classical protocol, we can imagine now what a generalization to a quantum key exchange, which like in all the games we have encountered until now, upon the said generalization, the parties involved in the game can share entanglement.

Definition 5.2 *We note it QKD and call it a quantum key distribution protocol, a KD protocol in which the parties involved share quantum entanglement.*

We are interested in a specific type of QKD, that being called the BB84 protocol in which Alice and Bob make a secure communication channel by sending multiple quantum particles(light usually) which will be safe to what is known as "passive attacks", which would involve a third party, we will generically call Eve, to eavesdrop on the messages they share to each other. The reason why the channel is secured is due to what is known as the "no-cloning theorem", which roughly states that Eve would not be able to copy the message encoded in

the the signal sent by the two parties, without destructive interference, which would alert Alice and Bob and they would abort the exchange.

What we are interested to do however is investigate attacks on another type of protocol, called a *Byzantine agreement fault tolerant protocol*. We will give a definition shortly, but first we must define at least on the informal level what a Byzantine fault is.

Definition 5.3(Byzantine fault) *We call a fault in a multi-party system a Byzantine fault if the fault cannot be associated to a single source of error.*
I would like to clarify what I mean through a playful example. Suppose there is a general which has in his charge several lieutenants but the conditions are such that they all must be separated on the battlefield and they have to send messages to communicate what actions to make. It is imperative that most of them take the same action in order to win the battle but there are many ways in which this can fail such as the messenger getting killed, or some of the lieutenants wanting to betray and deceive the general. This is what we would call a Byzantine fault.

Now the idea is to make protocols that are what is called robust against such faults, and indeed such protocol schemes exist(see [39]). The two main ingredients of a fault tolerant Byzantine protocol are:

1. A secure KD channel.
2. A voting game through which one of the members of the multi-party(the general) can propose an action and the action will be accepted based on voting majority with the conditions that:
    - (the Byzantine agreement) the general shares with everyone some instruction x.
    - in case the general is honest and shares with the lieutenants some instruction x all the loyal lieutenants will share the same instructions with their closest comrades.
    - in case the general is not honest all the honest lieutenants still share the same instruction y.

Now what we would like to do is investigate if some form of external foe would be able to disrupt Byzantine agreement of the army, specifically a type of attack that is known in the literature as a *man-in-the-middle attack*

Definition 5.3 *A man-in-the-middle attack is a type of attack against security protocols in which messages between parties are intercepted and manipulated by a third party*

What we really want to look at is a new type of man-in-the-middle attack introduced in [40], which takes place on a provably secure QKD known as the BB84 protocol. The reason why the attack is successful is because what it alters takes place before the secure channel can be established, when what is known as the calibration process occurs, by which it means signals are sent in order to see the time intervals in which messages and replies occur. The

attack is based on intercepting the signal from Alice and then sending to Bob the signal at a slightly modified timing(see Fig. 5.1).

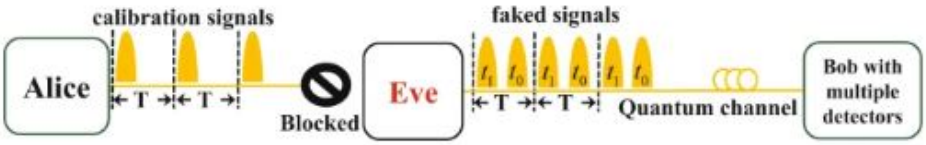


Figure 5.1. The scheme of the man-in-the-middle attack described above(image taken from [40]).

The above attack could be used to provide information about the keys but we will look at other uses for this type of attack, when the goal is to disrupt a Byzantine agreement system, using a man-in-the-middle attack.

A man-in-the-middle attack on a Byzantine system(Main result)

Suppose we have a Byzantine fault tolerant protocol for which the end-to-end QKD between all the parties involved(lieutenants and general) are based on the BB84 encryption scheme. Another assumption we are going to make is that the system is synchronous, which means the broadcasting of messages by the general takes place uniformly towards all.

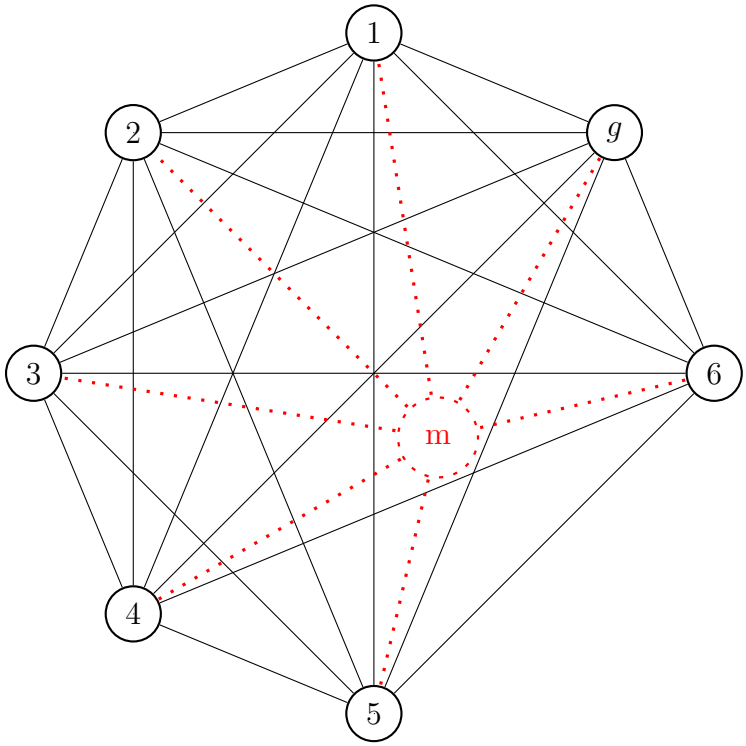


Figure 5.2. The scheme of the attack on the Byzantine system, in which the lieutenants are represented by numbers, the general is noted with g, and the man-in-the-middle with red, noted m. lieutenants.

Now the attack scheme goes as follows:

- wait until the system is expanded with the inclusion of a new lieutenant, with the assumption that new calibrations must be performed
- implement a generalization of the attack found in [40] so that the synchronous condition fails.

That is pretty much it. What would happen is that due to the failure of the synchronous condition, the general will become completely unreliable so the fault tolerance is broken. This type of attack might be of interest in the future, when client to server quantum systems will be implemented. It is only useful given maybe slightly weaker conditions than the one above however so certain authentication schemes would be safe from it, or even certain machine independent QKD's would prevent such attacks.

Conclusions

As a mark for the end of this thesis, I will give my thoughts on the bird-eye view of what I had in mind when writing it an what I think I have managed to do.
The scope of this thesis was to connect seemingly distant subjects of study such as Quantum Mechanics, Complexity theory and Information theory, through the scope of an even more surprising mention besides these, in the form of the theory of games. I think I have managed to bring forth my vision for the most part, as we have obtained non-trivial results in all of those areas using tools from computer science, quantum physics and even mathematical logic.

References

[1] Von Neumann, J., & Morgenstern, O. (1947). Theory of games and economic behavior, 2nd rev.

[2] Neumann, J. V. (1928). Zur theorie der gesellschaftsspiele. Mathematische annalen, 100(1), 295-320.

[3] Zamir, S., Maschler, M., & Solan, E. (2020). Game theory. Cambridge University Press.

[4] Nash, J. (1951). Non-cooperative games. Annals of mathematics, 286-295.

[5] Harsanyi, J. C. (1967). Games with incomplete information played by "Bayesian" players, I–III Part I. The basic model. Management science, 14(3), 159-182.

[6] Einstein, A., Podolsky, B., & Rosen, N. (1935). Can quantum-mechanical description of physical reality be considered complete?. Physical review, 47(10), 777.

[7] Bell, J. S., Bell, J. S. (2004). Speakable and unspeakable in quantum mechanics: Collected papers on quantum philosophy. Cambridge university press.

[8] Dahl, G. B., Landsburg, & S. E. (2011). Quantum strategies. arXiv preprint arXiv:1110.4678.

[9] Dahl, G., & Landsburg, S. (2005). Quantum strategies in noncooperative games. University of Rochester.

[10] Dahl, P. K. (2004). Quantum mysteries revisited again. American Journal of Physics, 72(10), 1303-1307.

[11] Mermin, N. D. (1990). Simple unified form for the major no-hidden-variables theorems. Physical review letters, 65(27), 3373.

[12] Brassard, G., Broadbent, A., & Tapp, A. (2005). Quantum pseudo-telepathy. Foundations of Physics, 35(11), 1877-1907.

[13] Cleve, R., Hoyer, P., Toner, B., & Watrous, J. (2004, June). Consequences and limits of nonlocal strategies. In Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004. (pp. 236-249). IEEE.

[14] Dikme, A., Reichel, N., Laghaout, A., & Björk, G. (2020). Measuring the Mermin-Peres magic square using an online quantum computer. arXiv preprint arXiv:2009.10751.

[15] Skolem, T. (1967). The foundations of elementary arithmetic established by means of the recursive mode of thought, without the use of apparent variables ranging over infinite domains. From Frege to Gödel, 302-333.

[16] Davis, M., Godel, K., & Kleene, S. C. (1990). On Undecidable Propositions of Formal Mathematical Systems. PostscriptumIntroductory Note to 1934. Journal of Symbolic Logic, 55(1).

[17] Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. J. of Math, 58(345-363), 5.

[18] Arora, S., & Barak, B. (2009). Computational complexity: a modern approach. Cambridge University Press.

[19] Sipser, M. (1996). Introduction to the Theory of Computation. ACM Sigact News, 27(1), 27-29.

[20] Cook, S., Krajíček, J. (2007). Consequences of the provability of NP⊆P/poly. The Journal of Symbolic Logic, 72(4), 1353-1371.

[21] Fortnow, L. (2009). The status of the P versus NP problem. Communications of the ACM, 52(9), 78-86.

[22] Cook, S. A. (1971, May). The complexity of theorem-proving procedures. In Proceedings of the third annual ACM symposium on Theory of computing (pp. 151-158).

[23] Levin, L. A. (1973). Universal sequential search problems. Problemy peredachi informatsii, 9(3), 115-116.

[24] Karp, R. M. (1972). Reducibility among combinatorial problems. In Complexity of computer computations (pp. 85-103). Springer, Boston, MA.

[25] Kobayashi, H., Matsumoto, K., & Yamakami, T. (2003, December). Quantum Merlin-Arthur proof systems: Are multiple Merlins more helpful to Arthur?. In International Symposium on Algorithms and Computation (pp. 189-198). Springer, Berlin, Heidelberg.

[26] Vidick, T., & Watrous, J. (2016). Quantum proofs. Foundations and Trends® in Theoretical Computer Science, 11(1-2), 1-215.

[27] Watrous, J. (2008). Quantum computational complexity. arXiv preprint arXiv:0804.3401.

[28] Kitaev, A., & Watrous, J. (2000, May). Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In Proceedings of the thirty-second annual ACM symposium on Theory of computing (pp. 608-617).

[29] Jain, R., Ji, Z., Upadhyay, S., & Watrous, J. (2010). Qip= pspace. Communications of the ACM, 53(12), 102-109.

[30] Ji, Z., Natarajan, A., Vidick, T., Wright, J., & Yuen, H. (2021). Mip*= re. Communications of the ACM, 64(11), 131-138.

[31] Scholz, V. B., & Werner, R. F. (2008). Tsirelson's problem. arXiv preprint arXiv:0812.4305.

[32] Tsirel'son, B. S. (1987). Quantum analogues of the Bell inequalities. The case of two spatially separated domains. Journal of Soviet Mathematics, 36(4), 557-570.

[33] Odifreddi, P. (1992). Classical recursion theory: The theory of functions and sets of natural numbers. Elsevier.

[34] https://texample.net/tikz/examples/complexity-classes/

[35] Martin-Löf, P. (1966). The definition of random sequences. Information and control, 9(6), 602-619.

[36] https://www.win.tue.nl/ berry/CryptographicProtocols/LectureNotes.pdf

[37] Shor, P. W. (1994, November). Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science (pp. 124-134). Ieee.

[38] Driscoll, K., Hall, B., Paulitsch, M., Zumsteg, P., & Sivencrona, H. (2004, October). The real byzantine generals. In The 23rd digital avionics systems conference (IEEE Cat. No. 04CH37576) (Vol. 2, pp. 6-D). IEEE.

[39] Luo, Y., Mao, H. K., & Li, Q. (2022). An Information-theoretical Secured Byzantine-fault Tolerance Consensus in Quantum Key Distribution Network. arXiv preprint arXiv:2204.09832.

[40] Fei, Y. Y., Meng, X. D., Gao, M., Wang, H., & Ma, Z. (2018). Quantum man-in-the-middle attack on the calibration process of quantum key distribution. Scientific reports, 8(1), 1-10.

[41] Yngvason, J. (2005). The role of type III factors in quantum field theory. Reports on Mathematical Physics, 55(1), 135-147.

[42] Gill, R. D. (2014). Statistics, causality and Bell's theorem. Statistical Science, 29(4), 512-528.

DECLARAȚIE PE PROPRIE RĂSPUNDERE

Subsemnatul, **Hosu Emanuel-Claudiu**, declar că Lucrarea de licență/diplomă/disertație pe care o voi prezenta în cadrul examenului de finalizare a studiilor la Facultatea de **Fizică**, din cadrul Universității Babeș-Bolyai, în sesiunea **2022** , sub îndrumarea **Dr.Lect.Emil Vințeler**, reprezintă o operă personală. Menționez că nu am plagiat o altă lucrare publicată, prezentată public sau un fișier postat pe Internet. Pentru realizarea lucrării am folosit exclusiv bibliografia prezentată și nu am ascuns nici o altă sursă bibliografică sau fișier electronic pe care să le fi folosit la redactarea lucrării.

Prezenta declarație este parte a lucrării și se anexează la aceasta.

Data, 22.06.2022

Nume,
Hosu Emanuel-Claudiu
Semnătură