Topics in Theoretical Computer Science May 2, 2016

Lecture 10 (Notes)

Lecturer: Ola Svensson *Scribes: Ola Svensson*

Disclaimer: These notes were written for the lecturer only and may contain inconsistent notation, typos, and they do not cite relevant works. They also contain extracts from the two main inspirations of this course:

1. The book *Computational Complexity: A Modern Approach* by Sanjeev Arora and Boaz Barak;
2. The course *http://theory.stanford.edu/ trevisan/cs254-14/index.html* by Luca Trevisan.

1 Introduction

Recall last lecture:

- Probabilistic Checkable Proofs. **PCP:** A language L is in PCP(r, q) if there exists a probabilistic polynomial time verifier V which, when given an instance x and a proof Π of length $poly(|x|)$, proceeds as follows
  1. Reads x and $O(r)$ random bits.
  2. Based on x and $O(r)$ random bits, queries $O(q)$ bits from Π.
  3. Computes and either accepts or rejects with following properties,
     - **completeness:** If $x \in L$ then $\exists$ proof Π such that $V^{\Pi}(x)$ accepts with probability 1.
     - **Soundness:** If $x \notin L$ the $\forall$ proof Π, the verifier $V^{\Pi}(x)$ accepts with probability at most $\frac{1}{2}$.
- PCP-Theorem $\mathbf{NP} = \mathbf{PCP}(\log n, 1)$.
- Proof that $\mathbf{NP} \subseteq \mathbf{PCP}(poly(n), 1)$ by giving a $\mathbf{PCP}(poly(n), 1)$ verifier for QUADEQ.

Today:

- Connection between PCP-Theorem and (hardness of) approximation algorithms.
- Clique is hard to approximate.
- Introduction to Quantum Computing
  - Two-slit experiment;
  - Qubits, superposition, unitary operations;
  - The EPR paradox.

2 Hardness of Approximation

We now explore the fantastic implications the PCP-Theorem has had on our understanding of approximation algorithms. Indeed, the PCP-Theorem shows that Max-3SAT is hard to approximate and then using that as a starting point you can prove your favorite problem to be hard to approximate. The PCP-Theorem has had a similar impact on our understanding of the complexity of approximation as the Cook-Levin Theorem had on our understanding of the limits of exact algorithms.

2.1 Definition of approximation algorithms

Definition 1 *An α-approximation algorithm for a given optimization problem is an algorithm which can be run in polynomial time and which outputs a solution S such that:*

- $\frac{cost(S)}{cost(Optimal\ solution)} \leq \alpha$ *if the problem is a minimization problem ;*
- $\frac{profit(S)}{profit(Optimal\ solution)} \geq \alpha$ *if the problem is a maximization problem.*

It is clear that we will have $\alpha > 1$ for minimization problems and $\alpha < 1$ for maximization problems.

It is possible to prove hardness of approximation results without using the PCP-theorem, as we are going to see in the following example.

2.2 Hardness of traveling salesman problem (TSP)

An instance of TSP is a graph where each edge is labeled with a cost, that must be a positive or a null real number. The expected output is a Hamiltonian path of minimal total cost in this graph.

What we are going to prove is that TSP falls into the worse category of approximation, that is:

Theorem 2 *For any $\alpha \geq 1$, it is NP-hard to get an α-approximation of TSP.*

Proof To prove this, we are going to introduce HAM, which is the problem of determining whether a graph has an hamiltonian cycle. We will use the fact that HAM is NP-hard (we can notice that this directly implies that TSP is NP-hard, since TSP asks for an Hamiltonian cycle in the graph). What we need is a **gap-introducing reduction** from HAM to TSP, that is, a reduction from HAM to TSP. Namely, from a graph G given as input to HAM, it builds a graph G' such that:

- if G has an Hamiltonian cycle, then G' has an Hamiltonian cycle with "low" cost.
- if G has non Hamiltonian cycle, then all Hamiltonian cycles of G' have "high" cost.

The main idea is that knowing a sufficiently good approximation of the solution of TSP for an input that we know to be in either the "high cost" or "low cost" category will enable us to determine in which of the two categories it belongs. This means that a good approximation for TSP can be used to solve HAM because it can distinguish between the two kind of outputs of our reduction.

To do this, we choose a constant M (that we will want to be very high) and we define our reduction as follows:

- The set of nodes of G' is equal to the set of nodes of G.
- G' is complete (all nodes are connected by an edge)
- The weight of an edge e of G' is 1 if e is also an edge of G, and is M otherwise

We denote by n the number of nodes in G and G'. If there exists a Hamiltonian cycle in G, then the same cycle has cost n in G'. Otherwise, any Hamiltonian cycles in G' use at least one edge that does not exist in G, thus it has a cost greater than M.

Now, suppose that we have an $\frac{M}{n}$-approximation of TSP. We denote by S the solution given by our approximation with input G'. If G has a Hamiltonian cycle, then the optimal solution has cost n. As such the cost of S is less than M. Otherwise, the optimal solution has cost greater than M, which means that S also has a cost greater than M. Therefore, we can decide whether G had a Hamiltonian cycle by looking at the cost of S, so our algorithm solves HAM. If M can be arbitrary, we can choose it as large as we want, which proves our theorem. ■

Observation 3 *Although we did not need the PCP-theorem to conclude, we were somehow "lucky" to be able to manipulate the costs of the edges at will. This trick is not a general method since many NP-hard problems, such as looking for the largest clique in a graph are purely combinatorial.*

Observation 4 *We can sometimes get positive results about existence of approximations. For instance, consider the modified version of TSP where the costs along the edges are required to satisfy the triangular inequality, i.e., for $u, v, w \in V$, $cost(u, w) \leq cost(u, v) + cost(v, w)$. The convention is that the cost between two nodes is infinite if there are no edges between them. This problem remains NP-hard. However, a well-known linear 2-approximation consists in performing a depth-first traversal of a minimum spanning tree of the graph (a sub-graph of G, which is a tree and which contains all nodes of G, and for which the sum of the costs of all edges is minimal among all spanning tree of G). A polynomial 1.5-approximation have also been found, and the theoretical limit has not been reached yet.*

2.3 Maximize accept probability

PCP verifiers, themselves, are a source of NP-hard problems. Given a $\mathbf{PCP}(\log(n), 1)$ verifier V for SAT, and an input φ, one can wonder which proof has a maximal probability of being accepted by V along with input φ.

Theorem 5 *It is NP-hard to approximate this problem within a factor of $\frac{1}{2}$.*

Proof The proof of this assertion is very simple, because the gap we need has already been introduced by the definition of a PCP verifier. Let's assume that we have a $\frac{1}{2}$-approximation for this problem. Let Π be the proof constructed by our algorithm with input φ, and p be the probability that Π is accepted by V along with input φ. We can evaluate p in polynomial time by feeding V with all possible random strings since it asks only for $O(\log(n))$ random bits. If $p \geq \frac{1}{2}$, then by the soundness property of V, φ is satisfiable. Otherwise, we have $p < \frac{1}{2}$ which means that the optimal proof cannot have a probability greater than $2p$ of being accepted, and $2p < 1$. By the completeness property of V, φ cannot be satisfiable.

Therefore, running our approximation algorithm on φ, along with an additional polynomial computation tells us whether φ is satisfiable, which means that our verifier solves SAT. ■

2.4 Hardness of Clique

Here we study the hardness of approximation of the maximum clique problem (or simply the clique problem). The clique problem consists in finding a maximum cardinality clique (i.e., a subset of vertices which forms a complete graph) given an undirected graph $G(V, E)$. We show here that there exists a constant $\epsilon > 0$, such that it is NP-hard to approximate the clique problem within a factor of $\frac{1}{N^\epsilon}$. We start by studying a gap introducing reduction from the SAT problem to a graph which is used to argue on the hardness results.

Lemma 6 *For fixed constants b and q, there is a gap introducing reduction that transforms, in polynomial time, a SAT formula φ of size n to a graph $G(V, E)$, where $N = |V| = n^b 2^q$ such that;*

- ***Completeness**: If φ is satisfiable, $OPT(G) \geq n^b$.*
- ***Soundness**: If φ is not satisfiable, $OPT(G) < \frac{n^b}{2}$.*

Proof Let F be a $\mathbf{PCP}(\log n, 1)$ verifier for SAT that requires $b \log n$ random bits and reads q bits from the proof.
We transform SAT instance φ into G[1] as follows.

[1] This graph is known as FGLSS graph

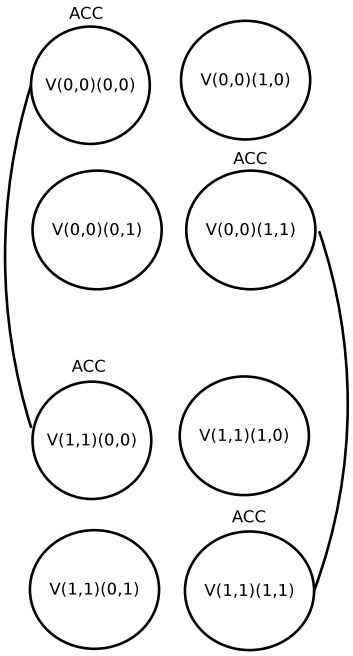

Figure 1: FGLSS graph for $b = 2$ and $q = 2$ (only a subset of the vertices and edges are drawn).

- **Vertices**: For each couple of random string r of $b \log n$ bits and truth assignment τ for q boolean variable, there is a vertex $V_{r,\tau}$ in G.
- **Edges**: Let $Q(r)$ represent the q positions in the proof that F queries given the "random string" r. Now, two vertices V_{r_1,τ_1} and V_{r_2,τ_2} are *consistent*, if τ_1, τ_2 agree on the values of the proof where $Q(r_1)$ and $Q(r_2)$ overlap. Further, we say that $V_{r,\tau}$ is *accepting* if F accepts, given a "random string" r and read τ in the proof. If two vertices are both *consistent* and *accepting*, there is an edge between them.

For example, consider the FGLSS graph defined for 2 random bits and 2 query strings illustrated in Figure 1. Vertices in the graph are labeled as $V(randombits)(querybits)$. Edges thus correspond to the vertices which are both consistent and accepting.
We will argue on the completeness and soundness based on the FGLSS graph.

- **Completeness**: According to the definition, we know that if φ is satisfiable there exists a proof Π such that F accepts with probability 1. Let $S = \{V_{r,\tau} \mid \tau \textit{ consistent with } \Pi\}$. By the definition of the transformation, for every "random string" r, there exists one vertex $V_{r,\tau} \in S$. Therefore, the cardinality of the set S becomes the number of random bits. n^b. We can see that S forms a clique since for each $V_{r_i,\tau_i}, V_{r_j,\tau_j} \in S$ they are accepting vertices as well as consistent (due to the definition of S).
- **Soundness**: Again by the definition of the verifier, if φ is not satisfiable, for all the proofs, F accepts with probability $< \frac{1}{2}$.

  Let C be a clique in G. Since all pairs of vertices in C are consistent, we know that C contains at most one vertex associated with each random string. This also implies that it is possible to build a proof Π that is consistent with every vertex of C: each vertex of C forces the value of up to k bits in the proof but the fact that the vertices are consistent ensures that there will be no contradiction.

Some bits of Π might not be specified by any vertex, i.e., we can freely decide regardless of their value. For instance, we can set them to 0.

Now, we can observe that Π is accepted with probability at least $\frac{|C|}{n^b}$, because each vertex in C corresponds to a different random string that will make our verifier accept C. Because of the soundness of the verifier, this proves that $|C| < \frac{n^b}{2}$.

■

3 Exercises

Exercise 1 *Show that it is NP-hard to approximate the Clique problem within any constant factor.*

Exercise 2 *Consider the problem MAX k-FUNCTION SAT:*

- *Given n Boolean variables $x_1, \ldots, x_n$ and m functions $f_1, \ldots, f_m$, each of which is a function of k (a constant) of the Boolean variables,*
- *Find a truth assignment to $x_1, \ldots, x_n$ that maximizes the number of functions satisfied.*

Show that it is NP-hard to approximate MAX k-FUNCTION SAT within a factor $1/2$.

3.1 Polynomial Hardness of Clique

We shall show the following theorem.

Theorem 7 *For some constant $\epsilon > 0$, it is NP-hard to approximate clique within a factor $1/N^\epsilon$ on a graph with N vertices.*

Note that in Exercise 1 we used the fact that Lemma 6, when given a verifier V that uses $b \log n$ random bits and q bits from the proof and soundness 2^s, says that clique is hard to approximate within a factor $1/2^s$. Moreover, the reduction runs in time $n^b 2^q$ and naively $b = O(s)$ and $q = O(s)$. Therefore, it seems like s can only be chosen to be a constant to have a polynomial time reduction. However, note that the reduction is still polynomial if $q = O(\log n)$ and $b = O(1)$. Therefore, a verifier V for SAT that reads $b' \log n$ random bits and $q' \log n$ bits from the proof with soundness $1/n$ would imply that Clique is NP-hard to approximate within a factor $1/n$. Now using that $N = n^{b'} 2^{q' \log n} = n^{1/\epsilon}$ for some constant $\epsilon > 0$ yields the theorem.

It remains to prove that such a verifier V exists for SAT, i.e., we wish to save random bits :).

- Let F be $\mathbf{PCP}(\log n, 1)$ for SAT which queries $b \log n$ random bits and q bits from the proof.
- We shall show how to obtain V from F.
- Let H be a constant degree expander graph on n^b vertices each vertex having a unique $b \log n$ bit label (one for each possible random string of F).
- Do a random walk on H of length $k \log n$ using $O(\log n)$ ($q' \log n$) random bits.
  - The label of each vertex on this path specifies a $b \log n$ random bit string. It uses these $k \log n + 1$ string as the "random" strings on which it simulates the verifier F.
  - V accepts iff F accepts on all $k \log n + 1$ runs.

Completeness Suppose the verifier is given a formula φ that is satisfiable. Then there exists a proof π that makes F always accept; since V accepts if F accepts on all runs, V accepts this proof with probability 1 as well.

Soundness Suppose the verifier is given a formula φ that is not satisfiable. Then no matter which proof π we consider, verifier F accepts with probability $< 1/2$. Let S denote the set of vertices of H that correspond to random strings on which F accepts. Then $|S| < n^b/2$. Now V accepts iff the random walk remains entirely in S. It is known that there exists a constant k such that

$$\Pr[\text{all vertices of a } k \log n \text{ length random walk lie in } S] < 1/n.$$

4 Introduction to Quantum Computing

- Different model of computing.
- Challenges the strong Church-Turing thesis that all computations can be simulated by a Turing machine with polynomial slow down:
  - Currently, there exists problems (e.g. FACTORING) for which there is efficient quantum algorithms but no efficient classic algorithms.
  - So it is plausible that Quantum Computers could lead to exponential speed-up for important problems.
- Still outstanding open problem to build a quantum computer.

4.1 Quantum weirdness: the two-slit experiment

- Very little physics is required to understand the central results of quantum computing.
- However, let us look at a specific experiment to get some intuition behind the definitions.

In the *two-slit experiment* a photon source is placed between a wall with two slits and a detector array (see Figures 10.1 and 10.2 in the textbook). We measure the number of times each detector lights up during an hour. We have two cases

- *One slit is covered (and one is uncovered):* We expect that the detector that are directly behind the open slit will receive the largest number of hits; this is indeed the case.
- *Both slits are uncovered:* We would expect that the number of times each detector is hit is the sum of the number of times it is hit when the second slit is open and the number of times it is is hit when the second slit is open. In particular, uncovering both slits should only *increase* the number of times each location is hit.

  Surprisingly, this is *not* what happens. In particular, at several detectors the total hit rate is *lower* when both slits are open as compared to when a single slit is open. Photons do not behave as particles or "little balls!"

How can we explain that? According to quantum mechanics, a photon instantaneously explores all possible paths to the detectors through the all open slits. Some paths are taken with positive "amplitude" and some with negative "amplitude".

> "The only difference between a probabilistic classical world and the equations of the quantum world is that somehow or other it appears as if the probabilities would have to go negative." —Richard Feynman in "Simulating Physics with Computers," 1982.

Of course, one may be skeptical about this "path exploration". To check it we could put a detector behind both slits. If a photon is really going through both slits simultaneously, you hope to detect it at both slits. HOWEVER, when you try to make the photon reveal its quantum nature in this way, the interference pattern disappear and it behaves like "little balls!"

The "explanation" is that *observing* and object "collapses" its distribution of possibilities and so changes the result of the experiment.

4.2 Quantum superposition and qubits

Classical computing involves the manipulation of bits. The analogous unit of storage in quantum computing is *qubit*.

- We can think of a qubit as an elementary particle that can be in two different states, which we denote by zero and one (exactly like a bit so far).
- However, unlike a classical bit, this particle can be simultaneously in both basic states.
- Thus, the state of a qubit at any time is called a *superposition* of these basic states.

Formally, we denote the basic states by $|0\rangle$ and $|1\rangle$ (sorry for the notation but it is there because of a long tradition) and a qubit can be in any state of the form

$$\alpha_0|0\rangle + \alpha_1|1\rangle,$$

where α_0, α_1 are called *amplitudes* and are complex numbers satisfying $|\alpha_0|^2+|\alpha_1|^2 = 1$. When the qubit is observed

- with probability $|\alpha_0|^2$, it is revealed to be in state $|0\rangle$;
- and with probability $|\alpha_1|^2$, it is revealed to be state $|1\rangle$.

Example 1 *Two different state vectors for a one-qubit quantumt system are:*

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

Although these states have the same measurement probabilities, they are considered distinct states.

Because states are always unit vectors, we often drop the normalization factor and use $|0\rangle - |1\rangle$ to denote the state $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$.

Also, the above discussion can be generalized to systems of many qubits. For example, the state of a two-qubit system at any time is described by a superposition of the type

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

where $\sum_{b_1,b_2} |\alpha_{b_1,b_2}|^2 = 1$. When this system is observed, its state is revealed to be $|b_1b_2\rangle$ with probability $|\alpha_{b_1,b_2}|^2$.

To manipulate the state of a qubit, we have to use a *quantum operation*, which is a function that maps the current state to a new state. Today, we only use operations on single qubits. Quantum mechanics allows only *unitary* operations, which are linear operations that preserve the invariant $|\alpha_0|^2 + |\alpha_1|^2$.

Example 2 *In the case of single qubit operations with real coefficients, this means that the allowed operations involve either a* reflection *of the state vector about a fixed vector in* $\mathbb{R}^2$ *or a* rotation *of the state vector by some angle* $\sigma \in [0, 2\pi)$.

4.3 The EPR paradox

- The EPR paradox, named after its proposers, Einstein, Podosky, and Rosen'1935, is a though experiment that shows that quantum mechanics allows systems in two far corners of the univers to instantaneously coordinate their actions, seemingly contradicting the axiom of Einstein's special theory of relativity that nothing can travel faster than light.
- In 1964, John Bell showed how to turn the EPR thought experiment into an actual experiment. Two systems far away from each other in the universe have a shared quantum state (actually, a two-qubit system). This shared state allows them to to coordinate tehir actions in a way that is provably impossible in a "classical" system.
- Since then Bell's experiment has been repeated many times in various settings always with the same conclusion: the predictions of quantum mechanics are correct.
- Today, it is not seen as a paradox, since the systems involved do not *transmit* information faster than the speed of light — they merely act upon information that was already shared, albeit in the form of a quantum superposition.

4.3.1 The Parity Game

We start by describing a game that seems to involve no quantum mechanics at all. The game involves two players Alice and Bob isolated from each other and an experimenter. It proceeds as follows:

1. The experimenter chooses two random bits $x, y \in \{0, 1\}$.
2. He presents x to Alice and y to Bob.
3. Alice and Bob respond with bits a, b, respectively.
4. Alice and Bob win if and only if $a \oplus b = x \wedge y$, where $\oplus$ denotes the XOR operation.

Observe that there is an easy strategy for Alice and Bob so that they win with probability 3/4. Bot always responds with 0 bits. This makes them win in every case except when $x = y = 1$ which happens with probability 3/4. The following theorem is also easy to prove and left as an exercise (see also book).

Theorem 8 *No (deterministic or probabilistic) strategy used by Alice and Bob can cause them to win with probability more than* 3/4.

4.3.2 The parity game with sharing of quantum information

We show that if Alice and Bob can share a two-qubit system (which they created in a certain state, and split between them before they were taken light year apart), then they can circumvent Theorem 8 and win the parity game with probability better than 3/4 using the following strategy:

1. Before the game begins, Alice and Bob prepare a two-qubit system in the state $|00\rangle + |11\rangle$, which we call the *EPR state*.
2. Alice and Bob split the qubits: Alice takes the first qubit, and Bob takes the second qubit.

   (Quantum mechanism does not require the individual bits of a two-qubit quantum system to be physically close to one another. It is important that Alice and Bob have not measured these qubits yet.)
3. Alice receives the qubit x from the experimenter, and if $x = 1$, then she applies rotation by $\pi/8$ (22.5 degrees) operation to her qubit.

4. Bob receives the qubit y from the experimenter, and if $y = 1$, then he applies rotation by $-\pi/8$ (-22.5 degrees) to his qubit.

5. Both Alice and Bob measure their respective qubits and output the values obtained as their answers a and b.

Some remarks:

- We remark that the order in which Alice and Bob perform their rotations and measurements does not matter: It can be shown that all orders yield exactly the same distribution.
- While splitting a two-qubit system and applying unitary transformations to the different parts may sound far fetched, this experiment has been performed several times in practice, verifying the theorem below (which is the prediction of quantum mechanics).

Theorem 9 *With the above strategy, Alice and Bob win with probability at least* 0.8*.*

Proof Alice and Bob win the game if they output a different answer when $x = y = 1$ and the same otherwise.

The intuition behind the proof is that unless $x = y = 1$, the states of the two qubits will be close to one another (with an angle of at most $\pi/8$) and otherwise the states will be "far" (having an angle of $\pi/4$). Specifically, we will show that

1. If $x = y = 0$, then $a = b$ with probability 1.
2. If $x \neq y$, then $a = b$ with probability $\cos^2(\pi/8) \geq 0.85$.
3. If $x = y = 1$, then $a = b$, with probability $1/2$.

This implies the theorem as then the overall acceptance probability is at least $\frac{1}{4} \cdot 1 + \frac{1}{2} \cdot 0.85 + \frac{1}{4} \cdot \frac{1}{2} = 0.8$.

Let us now prove the three different cases.

- In case (1) both Alice and Bob perform no operation on their qubits and so when measured it will be either in the state $|00\rangle$ or $|11\rangle$, both resulting in Alice and Bob outputting the same answer and winning the game.
- To analyze case (2), it suffices to consider the case $x = 0, y = 1$ (the other case is symmetrical). In this case, Alice applies no transformation but Bob rotates his qubit in a $-\pi/8$ angle. So after the rotation of Bob, the two-qubit system has the state

$$|0\rangle \cdot (\cos(\pi/8)|0\rangle - \sin(\pi/8)|1\rangle) + |1\rangle \cdot (\sin(\pi/8)|0\rangle + \cos(\pi/8)|1\rangle$$

  implying that the probability that they measure the same value is $\cos^2(\pi/8)$.[2]
- In case (3) we again use direct computation. After both rotations are performed, the two-qubit system has the state

$$\begin{aligned}&(\cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle)(\cos(\pi/8)|0\rangle - \sin(\pi/8)|1\rangle)\\ &\quad + (-\sin(\pi/8)|0\rangle + \cos(\pi/8)|1\rangle)(\sin(\pi/8)|0\rangle + \cos(\pi/8)|1\rangle)\end{aligned}$$

  which equals (by collecting the terms)

$$\begin{aligned}&\left(\cos^2(\pi/8) - \sin^2(\pi/8)\right)|00\rangle - 2\sin(\pi/8)\cos(\pi/8)|01\rangle\\ &+ 2\sin(\pi/8)\cos(\pi/8)|10\rangle + \left(\cos^2(\pi/8) - \sin^2(\pi/8)\right)|11\rangle\end{aligned}$$

  Now since $(\cos^2(\pi/8) - \sin^2(\pi/8)) = \cos(\pi/4) = \sin(\pi/4) = 2\cos(\pi/8)\sin(\pi/8$ we have that all the coefficients in this state have the same absolute value; hence, when measured, the two-qubit system will output two qubits of same value with probability $1/2$.

[2]We remark that here we have also used the notation $|x\rangle|y\rangle$ to denote the state $|xy\rangle$. An operation that is easily checked to respect the distributive law.

■ Some remarks:

- Exist games with more dramatic differences between classical and quantum cases.
- The ideas behind the EPR's and Bell's experiments have recently been used to device quantum encryption schemes whose security depends only on the principles of quantum mechanics; rather than any unproven conjecture such as $\mathbf{P} \neq \mathbf{NP}$.

4.4 Exercises

Exercise 3 *Prove Theorem 8, i.e., that no classic (deterministic or probabilistic) strategy used by Alice and Bob can cause them to win the parity game with probability more than* $3/4$.

Exercise 4 *Can you come up with a quantum strategy that allows Alice and Bob to win the game with probability greater than* 0.8*?*