

A  
Computational  
Model for  
Observation in  
Quantum  
Mechanics

Guillermo Juan Rozas

MIT Artificial Intelligence Laboratory

*This blank page was inserted to preserve pagination.*

# A Computational Model for Observation in Quantum Mechanics

Guillermo Juan Rozas<sup>1</sup>

## Abstract

A computational model of observation in quantum mechanics is presented. The model provides a clean and simple computational paradigm which can be used to illustrate and possibly explain some of the unintuitive and unexpected behavior of some quantum mechanical systems. As examples, the model is used to simulate three seminal quantum mechanical experiments. The results obtained agree with the predictions of quantum mechanics (and physical measurements), yet the model is perfectly deterministic and maintains a notion of locality.

This paper describes research done at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology, supported in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-86-K-0180.

Revised version of a dissertation submitted to the Department of Electrical Engineering and Computer Science in December, 1986, in partial fulfillment of the requirements for the degree of Master of Science.

Thesis Supervisor: Gerald J. Sussman  
Title: Professor of Electrical Engineering

---

<sup>1</sup>I.B.M. Graduate Fellow.

*This empty page was substituted for a  
blank page in the original document.*

## Acknowledgments

I would not have been able to complete this work without the help of many people to whom I am grateful. In particular, I would like to thank:

The I.B.M. Corporation, for its generous funding.

The Hewlett Packard Company, for its generous donation of equipment.

Professor Gerald J. Sussman, for his guidance, for the environment that he has created around him, for being excited when I am depressed, for knowing where I am going way before I find out, and most importantly, for having more confidence in me than I do myself.

Dr. Yekta Gürsel, for patiently explaining over and over sophomore-level concepts quantum mechanics, for knowing about everything, and for “living” the philosophy that it is always better to do something than to talk about it.

Michael R. Douglas, Michael Eisenberg, and Professor Jack Wisdom, for reading what nobody else bothered to read.

David A. McAllester, for providing a friendly and knowledgeable ear.

Jim and Barb Miller, for encouraging me on, for making sure that I was fed and that I slept enough hours, for helping me keep my sanity or whatever is left of it, and generally, and most importantly, for being there.

Chris Hanson, for implementing “Gator” Scheme and the “real” Edwin, where most of this work was done, and for providing a friendly ear when I needed a break.

The Scheme Team, for being my coreligionists, and for constantly providing ideas and encouragement.

The 4th floor AI community and the Educational Computing Group for listening to me and providing valuable criticism.

Finally, my family. Short words cannot express my feelings and how much I owe them.

**Te...**

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Context: Three relevant experiments</b>	<b>8</b>
2.1	The Stern-Gerlach experiment . . . . .	8
2.2	The Two Slit Interferometer experiment . . . . .	17
2.3	The EPR Paradox experiment . . . . .	22
<b>3</b>	<b>The Computational Model, an Overview</b>	<b>28</b>
<b>4</b>	<b>Implementation</b>	<b>34</b>
4.1	Linguistic issues . . . . .	34
4.2	General Description . . . . .	37
4.3	Code for the Stern-Gerlach experiment . . . . .	40
4.4	Code for the EPR paradox experiment . . . . .	46
4.5	Code for the double slit interferometer experiment . . . . .	50
<b>5</b>	<b>Conclusions</b>	<b>59</b>
<b>A</b>	<b>Results</b>	<b>64</b>
<b>B</b>	<b>Reimplementation</b>	<b>67</b>

## List of Figures

1	A single Stern-Gerlach device. . . . .	9
2	Stern-Gerlach device with no paths blocked. . . . .	10
3	Stern-Gerlach device with one path blocked. . . . .	12
4	Stern-Gerlach device with a detector on one path. . . . .	13
5	2 slit experiment setup. . . . .	18
6	Patterns in the absence and presence of a detector. . . . .	20
7	Device for the EPR experiment. . . . .	25
8	Spin directions which violate Bell's inequality. . . . .	26
9	A Stern-Gerlach apparatus and some of its time lines. . . . .	29
10	Device requiring complicated backtracking. . . . .	60



# 1 Introduction

I have developed a computational model of observation in quantum mechanics that presents a clean and simple computational paradigm, which can be used to simulate and possibly explain some of the unintuitive and unexpected behavior of some quantum mechanical systems. As examples, the model is used to simulate three seminal quantum mechanical experiments. The results obtained agree with the predictions of quantum mechanics (and physical measurements), yet the model is perfectly deterministic and maintains a limited notion of locality.

Although quantum mechanics has been around for quite a while, and its success is unquestionable in terms of predicting and explaining phenomena, especially in the microscopic world, it is still considered esoteric and to some degree unsatisfying. The mathematical complexity usually associated with the theory is not to be blamed for this problem. It is often the case that the mathematics involved in the quantum mechanical description of some phenomenon is simpler than the mathematics involved in its classical analogue<sup>2</sup>. As a whole, the usual mathematical formulation of quantum mechanics<sup>3</sup> (vector spaces and hermitean operators) is conceptually exceedingly simple, requiring only some simple algebraic concepts, rather than the large analytical baggage necessary for classical mechanics. Yet the mathematical formulation of quantum mechanics fails to be completely satisfactory because it does not really capture or explain adequately the most striking aspect of what can be called quantum behavior.

The key component missing from the mathematical formulation is an adequate model of observation. Observation (measurement) is an integral part of the qualitative theory, yet it is not handled well by the mathematics. As a matter of fact, observation is usually described in a purely qualitative way, even when the behavior of a system between observations is handled formally. The customary explanation of observation consists of a variation

---

<sup>2</sup>eg. The momenta, which are quantities independent from the positions in classical mechanics, are dependent quantities in quantum mechanics, and thus the phase space (see [Landau76]) has half the dimension in the latter case.

<sup>3</sup>See, for example, [Baym73], [Sakurai85], or any textbook on quantum mechanics.

of the sentence “A measurement always causes the system to jump<sup>4</sup> into an eigenstate of the dynamical variable that is being measured, the eigenvalue this eigenstate belongs to being equal to the result of the measurement<sup>5</sup>.” Sometimes analogues are offered, but often observation is not explained further. Observation and its consequences are often left to metaphysics, and informal arguments and models abound.

The problem lies in the following: Complicated systems are described by some global state vector (or wave function) which takes all the (apparent) components of the system into account. Although computations with the global state present few problems mathematically, they present intuitive problems. It is desirable to decompose the global state into local states which correspond to individual parts of the total system. For example, it is natural to consider each of the particles of a system as an individual local subsystem. Unfortunately, it then becomes hard, if not impossible, to deal with these local states consistently. When an observation occurs, the global state changes in a relatively simple way (note that the global state includes the object performing the observation). The local states, on the other hand, do not present such nice behavior. It is often the case that local states are coupled in the global state, that is, there are some constraints which must be satisfied among them, and which become apparent only when the local states are considered as parts of the whole global state. But then, an observation which affects a local state (and which would seem to act only on that local state), is potentially also affecting other local states, thus becoming non-local. The obvious answer is that our decomposition into parts was incorrect. We have not drawn the right boundaries around the components of the system. Unfortunately the decomposition seems natural and intuitive. Indeed, similar decompositions are consistently used in classical mechanics (and indeed, in every day life) and no problems arise. Yet such decompositions must be incorrect, for they lead to wrong conclusions or extremely complicated non-local interactions. But it is not easy to give up this time-proven paradigm. Furthermore, the change in the state of the system which occurs when an observation happens<sup>6</sup> does not fit well in the

---

<sup>4</sup>In wave mechanics, this jump of the wave function is called “collapsing”.

<sup>5</sup>P. A. M. Dirac, in [Dirac58].

<sup>6</sup>This problem only arises in the Copenhagen interpretation, not in the multiple world

mathematical model. The mathematical model accounts only for linear, incremental, instantaneous changes of state, but the change in state which occurs at an observation is much more drastic. Collapsing violates not only Schrödinger's equation, but Special Relativity as well<sup>7</sup>.

The fact that observation apparently consists of changes in state, is what leads me to believe that computational models, with their detailed notions of state, sharing, and change, might be appropriate to describe the processes involved. While mathematics cannot conveniently deal with non-invertible non-local changes (except by giving up the notion of locality), a computational model might be able to do better. Shared objects could implement the relevant state, and the non-local interactions would appear from side effects (changes) to these objects shared by the correlated states. Unfortunately straight-forward approaches do not work. Simple models in which particles have some internal state which determines the behavior of the particle run counter to fact. The EPR paradox experiment (see section 2.3) is hard to resolve with this class of models, collectively called hidden variable models. Experiment seems to imply that it does not make sense to talk about the state before an observation, because there is no consistent state the system could be in which would explain the outcome. While we can, for every particular trial, assign a consistent value to the state prior to the observation, the statistics turn out wrong when this is done for large numbers of particles.

We need a model which has the locality properties desired, but at the same time is powerful enough to incorporate cleanly the non-local aspects which reality presents. In the rest of this paper I explain one such possibility.

---

interpretation, which has other conceptual problems.

<sup>7</sup>See [Aharonov80].

## 2 Context: Three relevant experiments

In this chapter<sup>8</sup> I describe three simple but important experiments that illustrate the peculiar problems associated with observation in quantum mechanics. I will also point out why they are disturbing (they run counter to our intuition derived from the classical world) and where the problems in modeling arise.

These experiments are *Gedankenexperimente* (thought experiments<sup>9</sup>). They are often used in text books for precisely the same purpose: illustrating “quantum behavior”.

### 2.1 The Stern-Gerlach experiment

Electrons are lumped particles with a small (in absolute value) negative charge. By lumped we mean that they are localized in space: a single electron is not in two relatively distant places at the same time, and we cannot detect parts of electrons, only whole electrons.

Electrons also have a property called spin angular momentum, or spin in short. Spin behaves almost exactly like classical angular momentum, but it is quantized. That is, while classical angular momentum can assume any value in some continuous range, spin can only take on a value from a discrete set. In classical mechanics, the total angular momentum is the maximum of the projections of the angular momentum into all possible axes. Electron spin, on the other hand, when projected into any axis, will present only one of two values, opposites of each other (conventionally called  $+1/2$  and  $-1/2$  or spin up and spin down). The Stern-Gerlach experiment shows that this is the case and draws startling consequences of this seemingly innocuous statement.

Spin detection is performed in the following way: Particles are fired (in a narrow beam) into a magnetic field with a very large gradient. Classically, in a similar situation, particles would diverge from the original path

---

<sup>8</sup>Readers familiar with quantum mechanics can skip this chapter.

<sup>9</sup>All three of them have been carried out in one form or another.

according to the angle formed between the total angular momentum vector and the magnetic field. Keeping the magnitude of the total angular momentum constant, but letting the direction vary, particles would come out of the device anywhere inside of a given angle. Surprisingly electrons do not behave like this at all. Electrons come out of the device in only two directions corresponding to opposite values of the angular momentum vector. The particles in the beam can then be merged again by using a similar device, with its magnetic field aligned in the same direction, backwards. A Stern-Gerlach device is one such electron splitter and merger, with potential detectors and blocks on the paths leading from the splitter to the merger. Figure 1 shows such a device. There are only two paths inside the device, one corresponding to each value of the spin. We talk about a device aligned in some particular direction when the magnetic field is parallel to that direction.

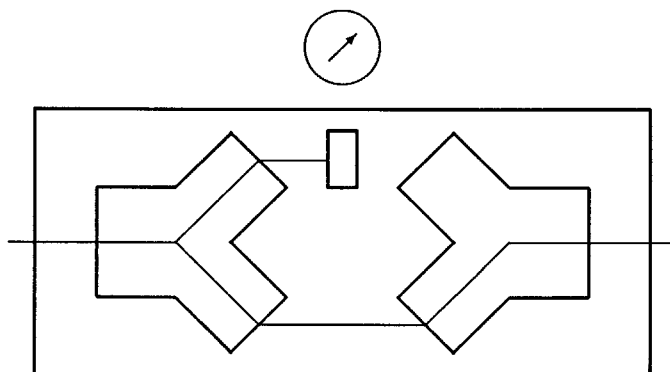


Figure 1: A single Stern-Gerlach device.

In the Stern-Gerlach experiment, particles travel through three such devices (see figure 2):

The first (the order is given by the order in which the electrons traverse them) is used to polarize the electrons. One of the two paths the electron can follow inside the device is blocked, thus letting through only the electrons whose spin is such that they choose the unblocked path. Their spin will therefore be parallel or anti-parallel to the magnetic field, according to which branch is blocked. If we place a particle counter at the output of the device, some fraction of the incoming number of particles will make it

through. Cascading an identical device (magnetic field aligned in the same direction and the corresponding path blocked) has no further effect. The number of particles that make it through the second device is the same as the number of particles that make it through the first, thus re-enforcing our belief that the particles have in effect been divided by the first device into two groups: the ones that make it through, having their spin in a particular direction, and the ones that do not, which have their spin in the opposite direction. We can consider a particle source in conjunction with one such device as a source of particles with uniform spin, and as such it is used in this experiment.

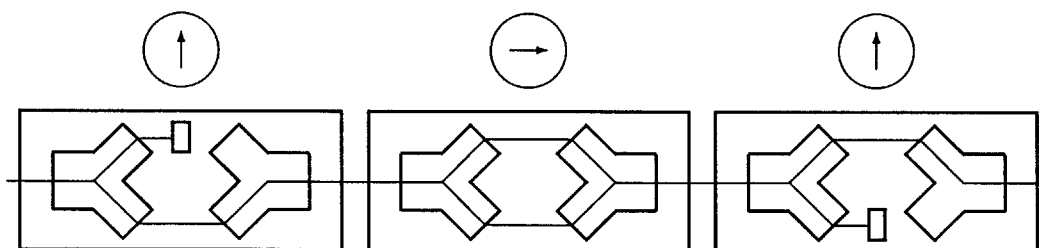


Figure 2: Stern-Gerlach device with no paths blocked.

The middle device has its magnetic field aligned in a direction perpendicular to the first. The experiment consists of blocking or placing counters on the paths internal to this device.

The last device has its magnetic field aligned in the same direction as the first. It also has one path blocked, but this time it is the opposite one from the one blocked in the first device.

There is a particle counter at the output of each device.

Initially no paths in the middle device are blocked or observed by placing counters. The number of particles coming out of the middle device is the same as the number of particles coming out of the first, and no particles come out of the last. This is not surprising since we can reason as follows: The first device is only letting through those particles whose spin is aligned in some particular direction. The second device is letting all the particles go through, given that both paths are free. The last device is blocking

particles whose spin is precisely that which the first device allows through. Thus any particular particle is blocked either by the first or the last device, depending on its spin component along the chosen direction. No particle can then make it through the whole apparatus. The middle device is not really performing any function and the results do not change if it is removed. We can thus say that the first device guarantees that the spin is in some direction, and that the second device does not affect this constraint. After going through the middle device the particles behave as if they had not, and thus blocking the opposite path in the last device blocks all the remaining particles. This is already a bit surprising: Why should particles that are split and merged along a second axis remember their previous polarization state? See below for the “amazing” answer.

We then block one of the paths in the middle device, as depicted in figure 3. It does not matter which path we block, since the chosen directions for the first and middle devices are perpendicular. The number of particles which make it through the middle device is one half of the number of particles which make it through the first device. Observing the output of the third device (the one whose magnetic field is aligned parallel to the first device) produces completely unexpected (insane?) results. While in the first situation, when no paths were blocked in the middle device, we observed no particles in the output, we now have some particles coming through! In fact, the number of particles coming through the third device is one half of the number of particles coming out of the middle device. By blocking some possible paths, we have increased the number of particles which make it through the apparatus! This can be explained by saying that measuring the component along the “middle” direction effectively destroys the polarization state caused by the first device, and thus when the last device is reached, the particles are no longer polarized along the direction which the device measures. If, as in the case above, we do not measure this component, we do not destroy this polarization state. This explanation is extremely disturbing because it does not allow us to think that each particle follows a well defined path through each device. If particles chose a spin component every time they reached a splitter, destroying previous polarization information, we would observe some particles coming out of the first setup. Particles would have been repolarized in the middle direction

losing their first state of polarization irrelevant of whether some path in the middle device was blocked or not. This clearly cannot be the case, for the behavior which the first device presents is different. On the other hand, if the splitter does not make the particle choose a component, how is it that in the second setup we observe particles with a definite choice? It is almost as if the particle lost its polarization state only if we could somehow tell which path it had followed (by virtue of the fact that only the ones with some component are allowed through). In other words, the particle only manifests the effects of following a particular path if we can distinguish between the paths. Merely observing causes a decision to be made!

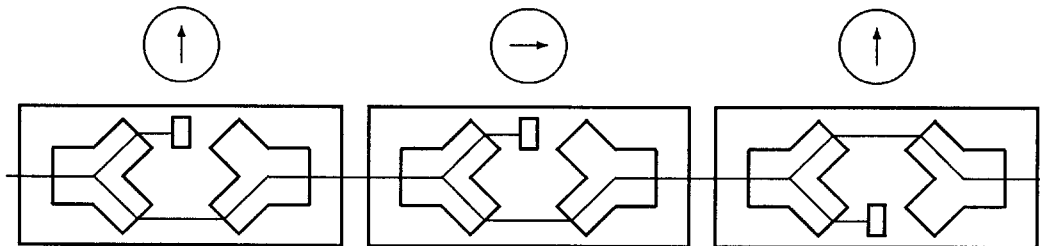


Figure 3: Stern-Gerlach device with one path blocked.

To further test this extremely frustrating behavior, we modify the second setup so that no path is blocked in the middle device, but rather a counter is placed along one or both of the branches. This setup is sketched in figure 4. The number of particles coming out of the second device increases to the number of particles coming out of the first device, which is not surprising, since no paths are blocked just as in the initial case. However, we again observe particles coming out of the last device, in fact, one half of the number coming out of the middle device, as in the previous case. This is extremely annoying. We are letting all of the particles go through the middle device, as in the first case, but the mere fact that we can tell which path they followed inside the device destroys the polarization state created by the first device and makes particles come out of the last device. The observation is enough to disturb the situation sufficiently to ruin some information which would otherwise still be available at the output of the second device.



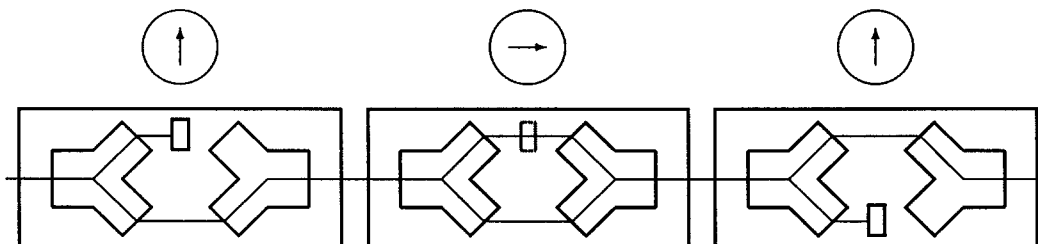


Figure 4: Stern-Gerlach device with a detector on one path.

There is more than one explanation (interpretation) for this bizarre behavior, and the more common elements are as follows:

- It makes no sense to talk about the particle as an individual object. It only makes sense to talk about the state of the whole system, which includes the apparatus. But then it is clear that all three situations are different, since the apparatus are not the same in each case, and thus it is no surprise that the outcome is so different. The “intelligent” or prescient aspect of the particles’ behavior is only an illusion caused by our mistaken attempt to impose some locality in a situation where there is little or none. We are decomposing the problem incorrectly and it is no surprise that inconsistencies arise from erroneous decomposition.
- The particle is in an undetermined state inside the middle device and does not really choose one path or the other. When an effective observation is made (by blocking, counting, etc), the particle then picks one of the possibilities and the world acts accordingly.

These two elements can be integrated in different ways to yield different interpretations, the most common of which are<sup>10</sup>:

1. The Copenhagen interpretation<sup>11</sup>: When the observation occurs, the

<sup>10</sup>See [Davies86] for an informal discussion of many of the most common alternatives.

<sup>11</sup>Properly speaking, there is no single Copenhagen interpretation, it is rather a collection of different possibilities encompassing almost everything between a die-hard positivist attitude which admits no other analysis than the mere statement and recording of results, and the informal mechanism suggested here. See for example [Bohr35].

observed object chooses one of the possible outcomes. The previously undetermined state, a superposition of the states corresponding to the various options, instantaneously and globally becomes the state which is consistent with this observation. Any correlated states change accordingly. In wave mechanics, this behavior is called “collapsing” (of the wave function).

2. The multiple worlds interpretation<sup>12</sup>: The observing and observed object are correlated after the observation. There is no collapsing of states, but rather the state evolves in a uniform way. An observation consists of a coupling of the observer and observed objects. The global state changes at the observation in such a way that the components we identify with the individual coupled objects are correlated. Thus if the initial state was a superposition of various possibilities, the final state will be a superposition of the outgoing states corresponding to each of the initial possibilities. For example, if the incoming state is a superposition of two states whose respective possible outcomes are yes and no, the outgoing state is a superposition of two states, one of which represents an observed object resulting in yes and an observer which observed yes, and the other of which represents an observed object resulting in no and an observer which observed no. In other words, at every choice point the universe splits into as many possibilities as there are options, and the state of the universe is just a superposition of these possibilities.

Both interpretations have problems because they give up important notions:

In the Copenhagen interpretation locality is lost. Correlated, distant, states will also “feel” the effects of the observation. The consequences of the loss of locality are catastrophic. In the same way that the particle is affected at the decision point by a detector it has not yet met (the mere fact that it is there is enough), we cannot guarantee that in any experiment we perform, some distant and external (to us) event is not altering the results in an analogous way. One of the basic premises of experimental science in

---

<sup>12</sup>See [DeWitt73] for a collection of articles describing this interpretation in detail.

general is that systems can be isolated so that the behavior observed is a property of the observed systems rather than of the surrounding environment (behavior is being caused by outside interference). Locality is too powerful a notion to give up so easily. Without it we cannot study systems unless the complete state of the universe is known, which is clearly infeasible. This interpretation also has mathematical problems: The collapse implies an instantaneous change in the state, which violates the normal, infinitesimal, time evolution of the system, described by Erwin Schrödinger's famous equation<sup>13</sup>, and Special Relativity<sup>14</sup>. Determinism is also lost since the state prior to the observation is truly in a superposition of states, the disambiguation occurring only at the observation point, and the choice is truly arbitrary.

The multiple worlds interpretation does not have these mathematical problems since there is a single state which describes the state of the complete system, and this state just evolves in time in a continuous and orderly way. Determinism suffers a greater blow than in the Copenhagen interpretation, where there is at least a single universe, parts of which may be in an undetermined state. In the multiple worlds interpretation it is the whole universe that is in an undetermined state. Determinism is only an illusion. The universe is in a superposition of states, yet this ensemble is not visible. There are few options to explain this:

- The whole universe splits at the choice point to produce consistent universes for each of the alternatives.
- The universe oscillates between the alternatives.

In both cases each alternative is internally consistent, and this multitude of possible realities cannot be observed by one of its components. Thus the universe might change drastically from one moment to the next, and there would be no traces left of the “previous” configuration. Locality is also lost

---

<sup>13</sup>In the words of Schrödinger himself ([Schrödinger35]), “Now it was previously stated and explained that any *measurement* suspends the law that otherwise governs continuous time-dependence of the  $\psi$ -function and brings about in it a quite different change, not governed by any law but rather dictated by the result of the measurement.”

<sup>14</sup>See [Aharonov80].

in this interpretation since the complete universe splits every time anything makes an arbitrary choice. Correlated states are split consistently even if spatially separated. This interpretation might almost seem like a sleight of hand. The end result is the same. Along each time line, the universe is just as non-local and unpredictable as in the Copenhagen interpretation. Arguing over whether there is a larger Universe, which we cannot observe, in which the mathematics are more elegant, may just be a philosophical squabble.

My main objection to both explanations is not the philosophical problems they present, but rather that the explanations are too vague. They are open to far too much interpretation. What exactly does it mean to say that the state of the particle collapses? What does it mean to say that the universe splits? How is this behavior accomplished? In this report I present a model which incorporates aspects of both interpretations, yet is completely deterministic, and preserves a notion of locality. It is, nevertheless, sufficiently detailed that it allows us to easily simulate this and other experiments on a computer.

## 2.2 The Two Slit Interferometer experiment

This experiment illustrates the wave-like and particle-like properties of electrons. In the Stern-Gerlach experiment we have already seen that we run into conceptual problems in equating electrons with classical particles. In contrast to the classical case, in some circumstances it is not possible even in principle to specify which path (if any in particular) the electron followed through some device. This particular aspect is further explored by this experiment, where given some results it is easier to view the electron as a wave than as a classical particle.

The experiment is easily described: The objects involved are a particle source, a target wall, and a detector placed on yet another wall parallel to the target wall. The source emits particles, some of which make it to the target wall. There are some holes (slits) in this wall, and some particles make it through to the region beyond, where they might impinge on the detector. See figure 5 for a rough sketch of the situation.

For simplicity, assume the source to be pointwise and the distribution to be spherical (the particles come out of the source in all possible directions, and with equal probability). Consider the line perpendicular to the target wall and going thorough the source. We make two holes on the wall at equal distances from the source, and on opposite sides of this line. The size of the holes is small relative to the “wavelength of the electron”<sup>15</sup>, the distance between the holes is large compared to this same parameter, and the distance between both walls is considerably larger.

Initially we cover one of these holes, so that only one remains open. We then observe particles by moving the detector around on the detector wall. The distribution of electrons hitting the detector wall matches exactly the distribution which we would observe if the electrons were classical particles (eg. bullets). No surprises here. The distribution is inversely proportional

---

<sup>15</sup>This statement assumes that we know of the wave-like aspect of electrons. We can eliminate this assumption by repeating the experiment with different values of the three parameters. If we experiment with a large enough range, we will in fact discover a set of values which cause the behavior described below. This set of values would allow us in turn to compute the wave-length of the electron. This has fortunately already been done, so we do not need this search at all.

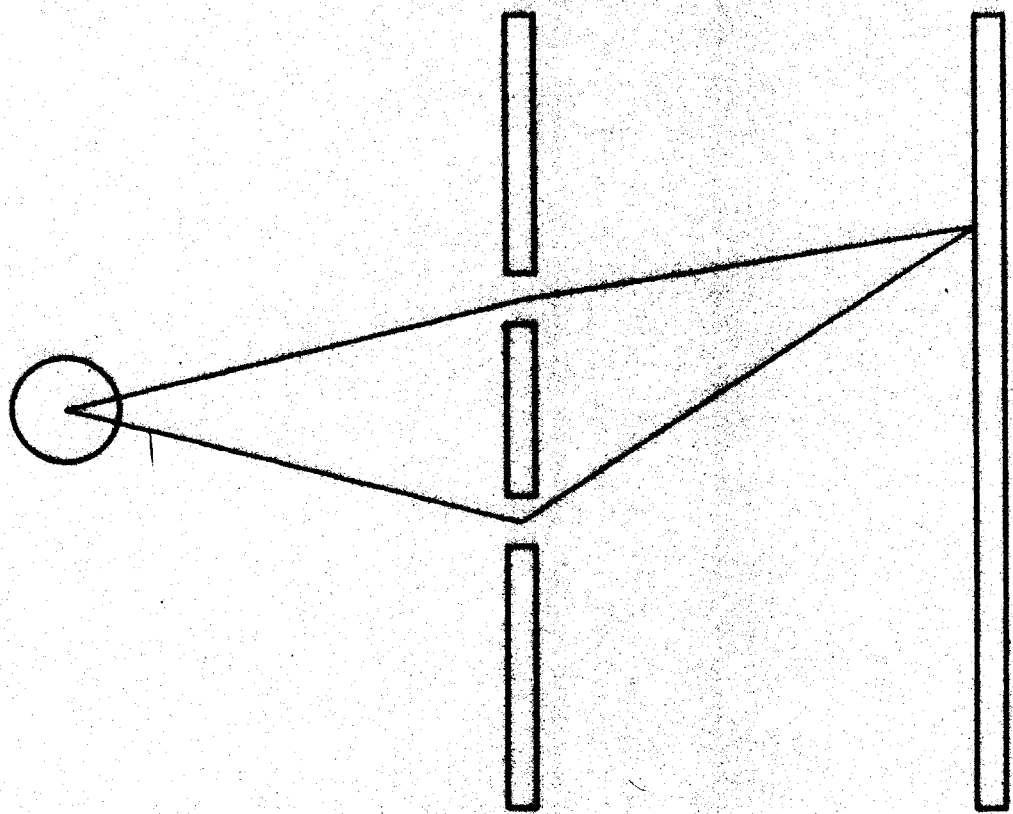


Figure 5: 2 slit experiment setup.

to the distance between the hole and the point where the detector is placed, so there is a maximum at the point “right across” the hole, and it decreases from there. Note that neither this behavior nor the behavior described below depend on cooperation between the electrons. The behavior remains unchanged if, for example, we send the electrons through at the rate of one per second.

We now uncover the covered hole, and cover the uncovered one, and we obtain a similar distribution although centered at the projection from the previously covered hole.

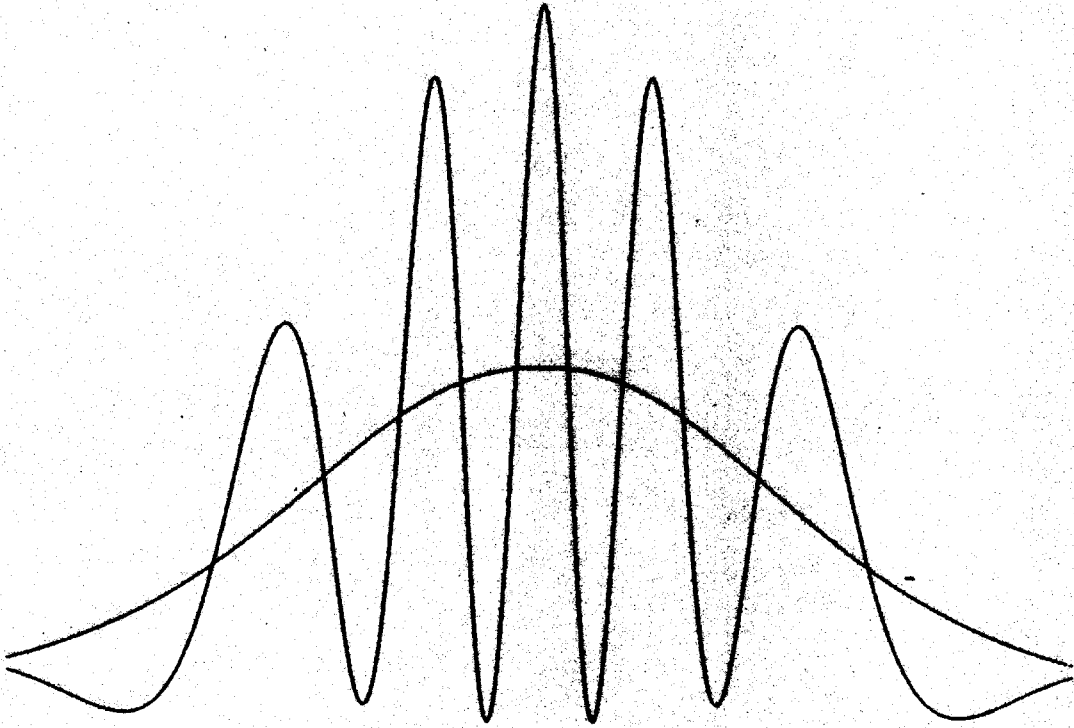
We then leave both slits open. Classically we would expect the distribution to be the sum of both distributions above, since when both slits are open, the particle can go through either, and thus the probability that it hits a particular point on the detector wall will be the probability that it hits it given that it goes through one hole, plus the probability that it hits it if it has gone through the other. This reasoning is valid in the classical case because the events described are incompatible: both are not possible at the same time.

Nature destroys our expectations again and the observed distribution is quite different. While the initial distributions have no nodes (places where no particles are detected, given a large enough sample), and thus the expected distribution has no nodes either, we find places where no matter how large the sample (number of particles we try), no electrons are ever detected. We also find some places (in particular, the point “right across” from the source) where the number of observed electrons is much higher than that predicted from the expected distribution. Plotting the observed distribution we obtain a graph similar to that shown in figure 6. This is the interference pattern which we would have observed if we had been doing the whole experiment with waves, and we had been measuring the intensity of the waves reaching the wall rather than counting the number of particles.

We now place a detector very close to one of the slits. We place in such a position, that the probability that the electron goes through the appropriate hole and not through the detector<sup>16</sup> (which lets the electrons

---

<sup>16</sup>We are using an idealized detector here. A different possibility is to use a source of



**Figure 6: Patterns in the absence and presence of a detector.**



go through) is negligible. The distribution this time is what was originally expected, namely the sum of the distributions corresponding to each of the holes. Again, the mere fact that we can tell which path the electron follows destroys the interference, and the results of the experiment change.

Although spin plays no part in this experiment, and geometry is not involved in the Stern-Gerlach experiment, we see that the situation is completely analogous. The change in behavior in the system caused by observation is the same. The problems and explanations presented above are thus the same in this case.

---

high-frequency light placed between the slits. The photon-electron scattering will produce flashes which will tell us which path individual electrons followed. This is the actual experiment described in the [Feynman63], where the experiment is extended to examine the effect that the frequency of the light (and thus its resolution) has on the distribution observed.

## 2.3 The EPR Paradox experiment

Although the previous two experiments are qualitatively similar, this experiment is different. The setting is the same as in the Stern-Gerlach experiment, where the observable is electron spin, and the devices used are the same for the purposes of this discussion.

The EPR<sup>17</sup> paradox experiment is somewhat subtle. Einstein, Podolsky, and Rosen published a paper [Einstein35] where they presented objections to the orthodox interpretation of some measurements presenting correlations between distant objects. As a result, some have claimed that the statistics predicted by quantum mechanics could also be exhibited by a system where particles have internal state (variables) hidden from the rest of the world. This state would determine how the particles would react and choose paths, and thus the world would be perfectly deterministic.

Until recently it was assumed that this hypothesis was untestable. No matter how hard it seemed to explain some particular set of results given this assumption, the model could be modified further to accommodate the new data.

In 1964, John S. Bell devised a simple test [Bell64] which would differentiate between standard quantum mechanics and the hidden variable theories. He pinpointed the basic assumptions of hidden variable theories (actually a wider class, local realistic theories, see [d’Espagnat79]), and using them as premises, derived a simple inequality relating certain observation frequencies. This inequality would always hold if local realistic theories were correct, while there existed particular instances of the experiment for which quantum mechanics predicted that the inequality would not hold. The experiment has been carried out, and alas, the inequality is found not to hold for some cases.

The basic premises of local realistic theories are the following:

- The particles would act the same way even if there were nobody to observe them. In other words, the particles have a reality independent

---

<sup>17</sup>The acronym is formed from the three authors’ initials.

of the observer, and the observer is not modifying the situation by merely observing.

- Inductive inference is a valid form of reasoning. Inductive reasoning allows extrapolation from a large enough sample into the rest of the population. That is, if some property is found to be true of a large enough (and arbitrary) subset of some set, it can be concluded that the property is true of all members of the set, including the ones not tested. Although this might seem a shaky assumption at first, it is freely used in experimental science. Inductive reasoning allows us to assume, for example, that all humans have red blood, even those whose blood we have not actually seen. It is hard to envision an experimental science without a similar premise.
- The detection happens simultaneously and so far apart that the particles do not have time to communicate their decision to each other<sup>18</sup> and thus conspire to fool the experimenter. This property is Einstein separability, an aspect of locality. Each particle must then make its own decision according to its own local (possibly hidden) state. The experiment as described above is not sufficiently “paranoid” about this point. Since the detectors are in place for a long time, the particles could “know” about them and thus conspire when they are first brought together. This can be remedied by placing the detectors far from the source, and changing the direction which the detectors observe dynamically. The particles would then have no way of guessing what component would be measured in each, and thus would have no possibility of conspiring. Some of the real experiments carried out have actually gone to this extreme.

The conditions for the experiment are described below, but I will not derive the inequality formally from the assumptions in local realistic theories. For this, the reader is referred to [d’Espagnat79] or [Sakurai85]. I will merely justify it informally.

Two electrons can be joined into what is called a spin singlet state. A spin singlet state is a state in which the electrons always have opposite

---

<sup>18</sup>In Special Relativity, two such events are said to be related in a “space-like” fashion.

spin: if any spin component of both electrons is measured (eg. in the manner of the Stern-Gerlach experiment), both results are always opposite. The results are always correlated perfectly when the component measured is the same.

The situation where local realistic theories and quantum mechanics disagree arises when we choose different axes for each electron in the pair. A source of spin singlet electron pairs could be constructed by forcing electrons from a standard source into the spin singlet state<sup>19</sup>. Assume that both electrons come out of the source in opposite directions (left and right), and that we measure a spin component of each electron in the pair by placing a Stern-Gerlach style device on each of the paths. This experimental setup is sketched in figure 7. Consider three different axes,  $A$ ,  $B$ , and  $C$ . Let  $f[AB]$  be the relative frequency of the event in which the electron on the left path is found to have spin  $+1/2$  when measured in the  $A$  direction, and the electron on the right path has spin  $+1/2$  when measured in the  $B$  direction. Similarly for  $f[AC]$  and  $f[CB]$ . Bell's inequality is simply  $f[AB] \leq f[AC] + f[CB]$ , which hidden variable theories require to hold for all choices of axes  $A$ ,  $B$ , and  $C$ . It is a simple exercise to show that quantum mechanics disagrees for some choices of axes (the corresponding probabilities violate the inequality, and thus, after a large enough sample, the relative frequencies will not satisfy the inequality). A particular choice for which they disagree is when  $A$  and  $B$  are perpendicular, and  $C$  is their bisector, as shown in figure 8.

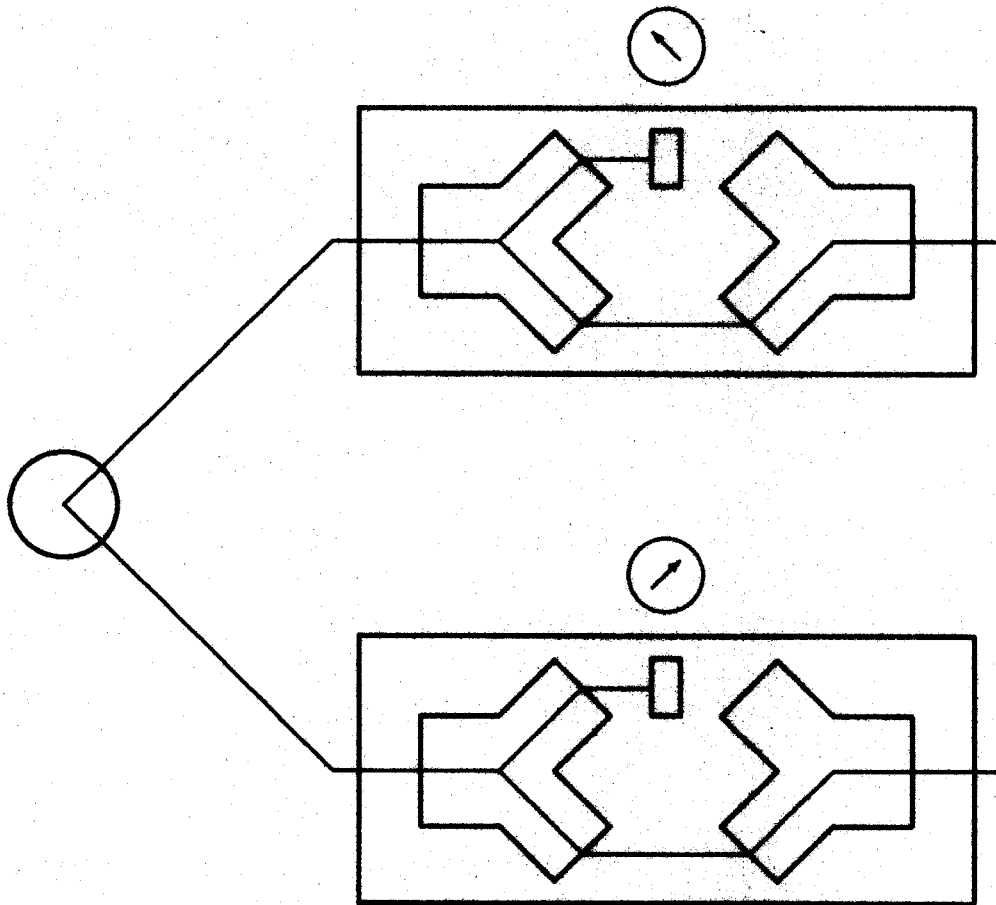
It is not intuitive at all how the inequality follows from local realistic theories. The informal explanation is as follows:

Consider the particles which take the left branch. When each electron reaches the detector, it has to make the decision of which path to follow depending only on its internal (potentially hidden) state. It has no time to communicate with its twin on the right branch.

Define the set of particles  $\{A+ B- C+\}$  to contain exactly those particles which would have chosen the spin  $1/2$  path if measured in the  $A$  direction,

---

<sup>19</sup>See [d'Espagnat79] for a list of the actual experiments carried out. The most similar to the one described here involves protons in the singlet state, rather than electrons.



**Figure 7: Device for the EPR experiment.**

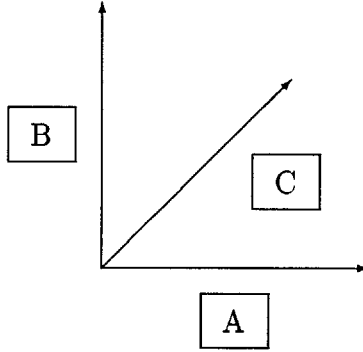


Figure 8: Spin directions which violate Bell's inequality.

the  $-1/2$  path if measured in the  $B$  direction, and the  $1/2$  path if measured in the  $C$  direction. The other combinations of axes and letters are defined similarly.

Particles are completely deterministic according to hidden variable theories, thus,

$$\{A_+ B_-\} = \{A_+ B_- C_+\} \cup \{A_+ B_- C_-\}$$

since a particle belonging to  $\{A_+ B_-\}$  would have either chosen  $C_+$  or  $C_-$  had its  $C$  component been measured. Similarly,

$$\{A_+ C_-\} = \{A_+ B_+ C_-\} \cup \{A_+ B_- C_-\}$$

$$\{B_- C_+\} = \{A_+ B_- C_+\} \cup \{A_- B_- C_+\}$$

but then

$$\{A_+ B_-\} \subseteq \{A_+ C_-\} \cup \{B_- C_+\}$$

since

$$\{A_+ B_- C_+\} \subseteq \{B_- C_+\}$$

$$\{A_+ B_- C_-\} \subseteq \{A_+ C_-\}$$

.

Therefore

$$|\{A_+ B_-\}| \leq |\{A_+ C_-\}| + |\{B_- C_+\}|$$

Now  $f[AB]$  is proportional to  $|\{A_+ B_-\}|$ , because if measured in the  $A$  direction, an  $\{A_+ B_-\}$  particle would result in a  $+1/2$  component. On the other hand, the particle following the right branch must be in  $\{A_- B_+\}$ , since when measured along the same direction they always show opposite components. But then it would result in a  $+1/2$  component along the  $B$  direction, and thus the pair would be counted in  $f[AB]$ . Similarly  $f[AC]$  is proportional to  $|\{A_+ C_-\}|$  and  $f[CB]$  is proportional to  $|\{B_- C_+\}|$ , with the proportionality constants being the same (depending on the fraction of particles actually tested in each of these directions, assumed equal).

But then it must be the case that

$$f[AB] \leq f[AC] + f[CB]$$

which is precisely Bell's inequality.

The fact that Bell's inequality is violated implies that at least one of our premises (or our reasoning system) is incorrect. Locality is often considered to be the erroneous premise. Furthermore, the proof of Bell's inequality depends on the assumption (hidden variable theory) that a consistent state can be assigned to each of the electrons in a pair before detection. This presents obvious problems for models, since it seems that if this is true of a model, Bell's inequality will hold for it, and the model will therefore be incorrect.

### 3 The Computational Model, an Overview

As stated above, in terms of the effects of observation, the Stern-Gerlach experiment and the double slit interferometer experiment are essentially identical. Given this, I will refer below only to the Stern-Gerlach experiment, pointing out the differences as necessary. The EPR paradox experiment is different, and we attack it later. As we will see, the most important idea used to model the first two experiments is powerful enough to give us a solution for the third.

Intuitively we can explain the behavior observed in the Stern-Gerlach experiment by using various time lines. A time line is a consistent sequence of events in time. Along each time line the past and the present (which is just a distinguished point in the time line) are perfectly consistent. The pasts and presents of different time lines might be quite different, however. As an extreme macroscopic example we might think of the time line where the history we know is valid, and a different time line where Hannibal destroyed Rome. Along this hypothetical time line there would be a present (corresponding to “today”), but it might be very different due to this difference in a past event. Before this bifurcation point both time lines might have been identical, and they might merge (or have merged) at some point after the bifurcation also. Note that if a system were able to switch time lines, there would be no way of telling within the system, since the result of any difference would have been erased in favor of the result consistent with the new time line.

In the Stern-Gerlach experiment, the electron is “originally” (these words are not appropriate when dealing with multiple time lines, but we have no others) in some unrestricted state (more about this below) which determines the probabilities for any observation of the electron. Invertible operators transform these probabilities, but the particle remains in the same time line. Observation has a radically different effect: Once a decision is made, the electron (and the system making the observation) abandons the current time line and settles in a time line which is consistent with the result of the observation. By consistent with the observation I mean a time line where the result of the observation is not aleatory, but rather deter-



ministic. In the “new” time line, the probability for the observed event is either 0 or 1, depending on the decision made on the “old” time line. Thus there is no “change” in the state of the electron at the point of the observation. The “change” occurs at the moment the unrestricted state was created (the ambiguity arose), by merely having followed a different time line where the state was the desired one to begin with. Within each time line the system is perfectly consistent and evolves deterministically, since any potential inconsistency would be resolved by switching to a time line where the inconsistency did not arise in the first place. See figure 9 for a graphical representation of the possible behavior of an electron going through a Stern-Gerlach device.

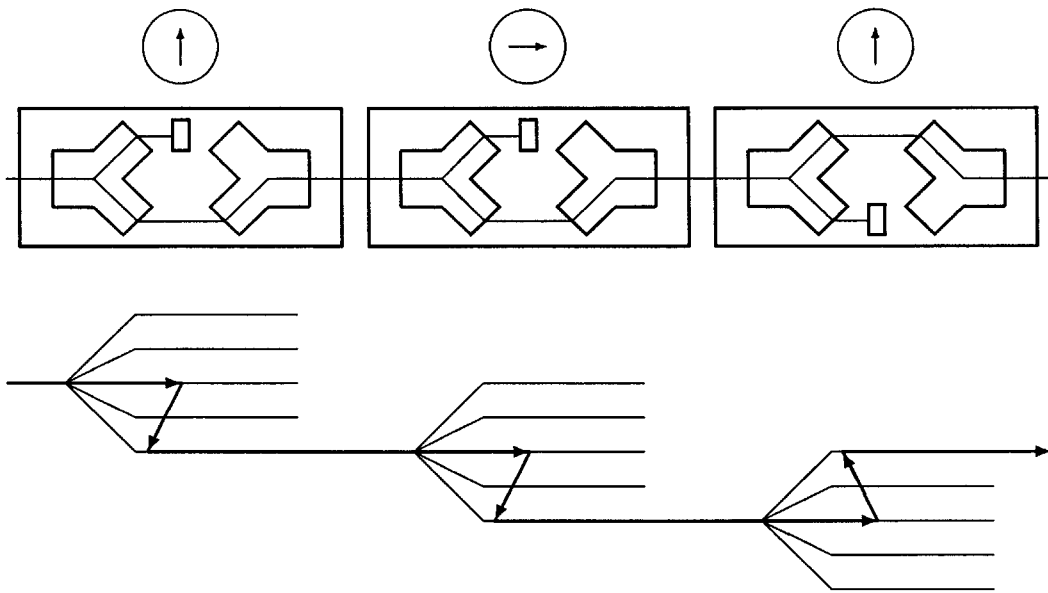


Figure 9: A Stern-Gerlach apparatus and some of its time lines.

The result of an observation may be positive or negative. A positive result determines completely some aspect of the state of the system immediately before the observation. A negative result merely constrains it. In the case of the double slit experiment, a positive result would be to detect the particle somewhere. This would determine the position aspect of the

state completely (up to the accuracy of the experiment) at the time of the observation. A negative result would be not to detect it at some particular position. It would still constrain the system to be consistent with the result, but would not give nearly as much information. The value (yes or no) of a result is not what determines whether the result is positive or negative, but rather how much it limits the ambiguity still remaining in the system. For example, if only two paths are possible for a particle, an observation on either path will be a positive result, since the state of the system (which path was taken) will be completely determined irrelevant of whether the result of the observation is yes or no.

An unrestricted state is one for which there exists more than one time line which produces the correct results at the observation points. Thus a positive result causes an unrestricted state to become restricted, while a negative result prunes the set of time lines still available for an unrestricted state, without removing the ambiguity completely. A positive result causes the system to enter a state which is restricted until the moment of the last observation, but for which there is a maximum amount of ambiguity after the observation. Thus after a positive result, the system will only change to time lines which are identical until the point of the observation.

How does this solve the Stern-Gerlach experiment? Particles are represented by a state vector, as in the matrix formulation of quantum mechanics. Consider first a single device. When the electron enters the device, it “splits” into two – there is a state vector corresponding to each of the paths. The probability of following either branch is given by the projection of the incoming spin vector and the polarization direction corresponding to the branch. These polarization directions are represented by orthogonal vectors, which form a basis in the 2-dimensional vector space of spin states. If no path is blocked or detected, both parts merge at the point where both paths join. The merge consists of adding the state vectors corresponding to each of the parts, and this reconstructs the initial state since both parts where just the components along the basis vectors. Thus the middle device has no effect if there are no paths blocked.

If instead there is a block (or detector, they only differ in whether they let the particle through or not) on one of the paths, when the particle

meets the block it first decides what the result of this observation is (the probability is given by the amplitude of the state at the point of the observation), and then if it is not already in a time line consistent with the observation (the probability is one or zero), it switches to a time line where the result of the observation is inevitable. It accomplishes this by choosing a time line where the incoming spin vector to the overall device is such that its projection along the branch with the block gives a probability of one or zero. Thus if the paths can be distinguished in any way (by means of blocks or detectors), the system will always effectively align its spin along the direction of the magnetic field on entry to the device, while otherwise it will not align it at all. Thus in the case where there is a block in the middle device, the electron will have its spin vector aligned with the appropriate magnetic field, losing the information that it had been aligned with the magnetic field of the first device. If there is no block in the middle device, no such realignment will occur and the information will be available at the input to the third device, so the split will occur in the same way as in the first, as expected.

In the double slit interferometer the situation is similar. The particle propagates by Huygens's principle, with its phase growing linearly with time. Thus for the region beyond the target wall we effectively have two sources, one per slit, and the difference in phase for each of the components gives the desired interference pattern as the probability distribution. When a particle is observed, the probability of a positive outcome is thus given by this interference pattern, and the correct statistics are obtained. If a detector is placed near one of the slits, the particle will change time lines at the time of detection to be consistent with the chosen outcome. If the outcome is positive (the particle is detected), in the new time line the particle will have gone through the slit close by, and the probability distribution beyond will be that corresponding to a single source. The situation is similar if the particle is not detected, and therefore the overall distribution will be the sum of the single slit distributions.

We now turn our attention to the EPR paradox experiment. In this experiment we have qualitatively different behavior. Instead of a single particle being observed, and obtaining different results depending on whether some intermediate stage is observed or not, we have particle pairs which are

correlated in particular ways. In other words, we have correlations within the system being observed, besides correlations between the system and the observer, as in the other two cases.

The solution follows the same paradigm: When one of the particles is observed, a new time line is chosen so that the particle's spin is aligned with the observation axis. In previous cases only the particle being observed switched time lines, but in the case of the electrons in the spin singlet case, both switch time lines simultaneously so that their spins are opposite. There is one more crucial idea: The time line both particles switch to is such that there is no longer a link between them (they will no longer simultaneously switch time lines).

How are the “interesting” correlations obtained? The answer is simple: Assume that the particle on the left branch is observed “first”, and both spin vectors become aligned along the appropriate direction. “Later”, when the particle on the right branch is observed, its spin is no longer arbitrary, but rather it is aligned in the direction along which the particle on the left branch was observed. Thus, instead of having a uniform distribution, it now has a biased distribution. For example, if the spin “component” measured on the particle following the left branch was the  $X$  component, and the result was “up” (+1/2), the particle on the right branch would have a low probability of resulting in an “up” component on directions close to  $X$ , and a higher probability on directions close to  $-X$ . On the other hand, since the a-priori probability of obtaining an “up” measurement on the right path was 50% to start with, the overall probability of obtaining an “up” measurement on the left path is also 50%. The difference from the unrestricted case –where the two particles are uncorrelated to start with– is that they will have a different probability, higher or lower depending on the directions chosen, of resulting in “up” or “down” simultaneously.

In other words, if we measure the same spin component on the second particle as on the first, we will obtain the desired negative correlation, since the system was already in a time line where the spin was aligned in this direction. If we measure it along a different direction, the second particle's spin will align itself in the new direction, but since it is now uncoupled from the first, the first will not change. Furthermore, the probabilities for the

two possible outcomes at the new direction are given by the overlap between the incoming state vector and the observation direction, but the incoming state vector is biased (it is aligned along the direction of the first particle's device), so the probabilities are biased and the frequencies observed are those predicted by quantum mechanics.

In chapter 4 I explain the details of the implementation of the above informal solution.

## 4 Implementation

### 4.1 Linguistic issues

For the implementation of the above ideas I have chosen a language which is almost functional. It is a subset of pure Scheme<sup>20</sup> which includes `call-with-current-continuation`, and which is extended with a non-functional procedure, namely `random`<sup>21</sup>.

`(random n)` returns a pseudo-random integer between 0 and  $n - 1$  (both inclusive). An approximation to a continuous random quantity in the range  $[0, 1]$  can be obtained by the expression `(/ (random (+ m 1)) m)` where  $m$  is a large integer. A binary choice with probability  $p$  can be approximated by the expression `(< (/ (random (+ m 1)) m) p)`. Pseudo-random numbers are used to generate arbitrary initial state vectors and to make decisions according to some probability.

`call-with-current-continuation` is a powerful Scheme primitive which can be used to build arbitrary control structure.

```
(call-with-current-continuation
 (lambda (the-current-computation)
   <Some Expression>))
```

results in evaluation of `<Some Expression>` in an environment where `the-current-computation` is bound to an escape procedure. If the value of `the-current-computation` is ever invoked on some argument, the whole `call-with-current-continuation` expression will abort and return this argument. Thus, `call-with-current-continuation` can be used to implement non-local `gotos`, Lisp's `catch` (from which it was derived), etc. But it is more powerful than this. The object to which `the-current-computation` is bound has unlimited extent. That is, its lifetime is not

---

<sup>20</sup>Pure Scheme is the subset of Scheme which contains no side effects. See [Abelson85] for a description of the full language.

<sup>21</sup>There is another implementation which is even more functional in that it does not need `random`. I present this other implementation in Appendix B.

limited to the time taken to evaluate `<Some Expression>` as with Lisp's `catch`, but rather it can survive this evaluation and be invoked later. Its effect in such circumstances would be to abort the computation in whatever state it was, and return again from the `call-with-current-continuation` with the new argument as the value for the whole expression. `call-with-current-continuation` corresponds to Landin's J-operator [Landin65] and, in semantic terms, its effect is to make the implicit continuation of the expression available to its argument as a special kind of procedure.

The most common use of `call-with-current-continuation` does not involve invoking these continuations more than once, and some Scheme implementations have at various points disabled this feature (usually called "re-entrant" continuations), but it will be seen that this feature is crucial to the implementation of the model.

There should be no doubt that `random` is not a functional procedure, but the issue is not so clear as far as `call-with-current-continuation` is concerned. There are various possible definitions of functionality, and `call-with-current-continuation` violates some but not others. For example, it violates normal order beta substitution, but it does not violate applicative order beta substitution. It falls in a strange category, and irrelevant of whether it is functional or not, it has unexpected consequences. See [Felleisen86] for a formal treatment of the consequences of adding first class continuations to a language.

All the code could be written in a completely and unambiguously functional language by explicitly passing stores and continuations around, but doing so would clutter the programs and make them much less clear. Even though the language is not completely functional, it is not a fully unrestricted language. `random` constitutes the only clear violation of functionality, and it is of a benign variety. It is hard to construct general purpose side effects from a pseudo random number generator, and its use in the code at hand is relatively constrained<sup>22</sup>.

The advantages of functional languages are varied and many. The code

---

<sup>22</sup>Even if not benign, this violation is made unnecessary by the modification to the code explained in Appendix B.

is usually clearer because there are no non-local interactions<sup>23</sup>, which also makes it easier to debug. But, beyond these stylistic and almost “syntactic” differences, code written in a functional (and therefore procedural) style presents obvious advantages in modeling physical situations. Devices can be modeled by functions (procedures), and paths and flow are easily expressed in the code by using functional composition. The expression `(f (g (h x)))` means give the value of `x` to function `h`, then give the result of this to function `g`, and finally pass the new result along to `f`. There is, therefore, an implicit notion of “flow” in functional composition. If a particle, for example, is to go through three devices, and we model the particle by some object transformed by the devices, the path of the particle can be expressed as the procedure which is the composition of the individual devices.

```
(define (compose f g)
  (lambda (x)
    (f (g x))))
```

```
;; The following is the path which consists of three component
;; devices -> 3 -> 2 -> 1 ->
```

```
(define path
  (compose device-1
            (compose device-2 device-3)))
```

---

<sup>23</sup> `call-with-current-continuation` violates this claim.



## 4.2 General Description

Particle states, as implied above, are modeled by data structures which contain some representation of the state vector, plus an upgrade function<sup>24</sup>. This upgrade function is invoked when an observation occurs. The argument supplied to this function is the state which the particle should have had to make the result of this observation deterministic (probability either zero or one).

```
(define (collapse state)
  (let ((v (state-vector state)))
    (let ((n (norm v)))                ;Prob. amplitude
      (let ((new-v
              (if (choose (square n))  ;Probability
                  (scale v (/ 1 n))    ;Increased to 1
                  (make-zero-vector)))) ;Decreased to 0
            (update state new-v))))))

(define (update state new-vector)
  (let ((upfunc (upgrade-function state)))
    (upfunc (make-state upfunc new-vector))))
```

The upgrade function guarantees consistency between the different related states<sup>25</sup> when an observation occurs. This function takes care of making sure that the related states are updated consistently. This might, a priori, seem hard to accomplish in a functional language, or in an almost functional language, given that there are no side effects. Side effects would provide the option of mutating the related states, but they might also run into problems because other related states might have already been transformed, for further computations might have already used the values generated by the split.

---

<sup>24</sup>The examples are drawn from the Stern-Gerlach experiment, the others having more complicated code but being essentially similar. Some of it has been rewritten slightly to increase readability without changing any of the essential aspects. Some more comments have been added.

<sup>25</sup>States are related if they resulted from some split at a previous point in time.

The solution is the following: at every point at which an ambiguity arises, and an arbitrary state is chosen, the continuation for the expression creating the state is obtained by using `call-with-current-continuation`. The upgrade function for this initial state is then just the continuation for this expression. When an observation occurs on a state, a new state vector consistent with the result of the observation is chosen unless the arbitrarily chosen state was, by chance, already consistent with it. The upgrade function is then invoked on this new state. Since the upgrade function is the continuation existent at the point when the state was created, the current computation will be aborted. In other words, the upgrade function, and therefore `update` and `collapse` above, do not in general return to their callers. Instead the computation which followed the creation of the arbitrary state will be re-started, but now with a value consistent with the observation. Note that the computation corresponding to the measurement, and any other computation which has taken place before it but after the choice point, will be redone but with a new value for the “previously” arbitrarily chosen state.

The way consistency along different branches is maintained is as follows: When a “non-observational” transformation is applied to a state, the resulting state contains not only a transformed state vector, but also a transformed upgrade function. This new upgrade function is not merely a continuation, as at the particle creation time, but rather it is somewhat “cleverer”. When an observation occurs, and this new function is invoked, it in turn invokes the upgrade function of the incoming state, but with a different state. The state propagated to the incoming upgrade function has the transformation taken into account, by, for example, applying the inverse transformation to the state vector. Thus, if the transformation consists of rotating the state vector by some angle  $\alpha$ , the outgoing state will have an upgrade function which will take this rotation into account. Namely, when an observation occurs, the resulting state vector will be rotated by  $-\alpha$  so that when the transformation is applied again after restarting the appropriate computation, the value obtained will be the desired one.

A split is a special case of transformation. The upgrade function for each of the resulting parts “knows” that the corresponding part arose from a split, that is, either upgrade function constructs the appropriate incoming

(“un-split”) state when an update occurs on its branch. In this way, when the split happens “again”, the result along the branch where the observation happened will be the desired one. Since this happens at each transformation and split point, all the states which can be traced to an arbitrary choice will be computed again, but this time starting with a state which takes the various “twists and turns” into account. This process guarantees, therefore, not only that the observation will be deterministic the second time around, but also that the other states which are correlated with the one observed have appropriate values. The analogy with time lines is straightforward: each possible computation path corresponds to a different time line, and observation triggers a switch to a time line where a state vector consistent with the observation had been chosen to begin with.

### 4.3 Code for the Stern-Gerlach experiment

The most important piece of code in the Stern-Gerlach experiment is the code which implements a single Stern-Gerlach apparatus:

```
;; APPARATUS generates a Stern-Gerlach device from a direction
;; used to test the spin components, and a description of the
;; devices in each of the paths.

(define (apparatus spin-direction path1 path2)
  (lambda (particle-state)
    (split particle-state spin-direction
           (lambda (state1 state2)
             (join spin-direction
                   (path1 state1)
                   (path2 state2)))))))
```

The various possibilities for the paths, and the use of apparatus are described below.

The code for a particle source is:

```
;; The source: a procedure of no arguments which returns a new
;; particle state every time it is invoked. MAKE-UNIT-VECTOR
;; returns a random unit vector (informal spinor).

(define (make-particle-state)
  (make-simple-state (make-unit-vector)))

;; MAKE-SIMPLE-STATE constructs a state whose upgrade function
;; performs a consistency check and then restarts the
;; computation suspended in HERE.

;; >=? should really be =. However, numerical inaccuracies
;; force us to check for "nearness", rather than for exact
;; equality.
```

```

(define (make-simple-state vector)
  ;; Vector must be normalized. Only called from places which
  ;; guarantee this.
  (call-with-current-continuation
   (lambda (here)
     (make-state
      (lambda (new-state)
        (if (not
             (>=<? (norm vector)
                    (norm (state-vector new-state))))
            (error "Simple-state: particle number!")
            (here new-state))
          vector))))))

```

The crux of the matter is the use of `call-with-current-continuation`. The continuation obtained by its use corresponds to the complete computation awaiting the particle state. The upgrade function provided is the result of the inner lambda expression. It is a simple wrapper, which performs some consistency checks, and then passes the updated state along to the continuation. Thus if this upgrade function is ever invoked, `make-simple-vector` will return “again”, but now with the “correct” state.

A detector is simply the following:

```

;; A detector. It depends on COLLAPSE shown above.
;; NORMALIZED? checks whether the norm of the vector (whose
;; square is the probability of detection) is either 0 or 1.

```

```

(define (detector state)
  (let ((v (state-vector state)))
    (if (normalized? v)
        (make-simple-state v)
        (collapse state))))

```

The top level code is as follows:

```

;; DETECT returns true or false depending on whether the
;; particle represented by state was observed or not.

```

```

(define (detect state)
  (not (zero-vector?
        (state-vector (detector state)))))

;; FREE-PATH is a possible path for APPARATUS. It leaves the
;; state unmodified. Other possible paths are DETECTOR above
;; and BLOCK below.

(define (free-path state)
  state)

;; A block. Notice the similarity with DETECTOR above. They
;; are basically identical, but BLOCK always passes along a
;; state with probability 0, even when COLLAPSE results in an
;; observed particle.

(define (block state)
  (if (normalized? (state-vector state))
      (make-simple-state (make-zero-vector))
      (collapse state)))

;; These are the actual directions and devices used. NORMALIZE
;; scales its input to be a unit vector (spinor).

(define z-spin-up (normalize (make-vector 1 0)))
(define x-spin-up (normalize (make-vector 1 1)))

;; The convention for the names of the devices is simple: The
;; first letter after APP is the direction along which the
;; magnetic field is aligned. The following 2 letters
;; represent the devices in each of the paths, the first
;; corresponding to the "up" direction. F represents a free
;; path, B a block, and D a detector. Thus APPXFB is a
;; Stern-Gerlach device whose magnetic field is aligned along
;; the X axis, which allows the "up" component through, since
;; its path is Free, and which Blocks the "down" component.

(define appzfb (apparatus z-spin-up free-path block))
(define appzbf (apparatus z-spin-up block free-path))

```

```

(define appxff (apparatus x-spin-up free-path free-path))
(define appxfb (apparatus x-spin-up free-path block))
(define appxdf (apparatus x-spin-up detector free-path))

;; The actual complete apparatus corresponding to the three
;; situations described in section 2.1. NULL is the apparatus
;; which produces no output. WEIRD is the apparatus which
;; results from blocking one of the paths in the middle device,
;; and WEIRDER is the apparatus which results from placing a
;; detector in one of the branches.

(define null (compose appzfb (compose appxff appzbf)))
(define weird (compose appzfb (compose appxfb appzbf)))
(define weirder (compose appzfb (compose appxdf appzbf)))

```

join is relatively simple:

```

(define (join spin-direction state1 state2)
  ;; Note that, as in SPLIT below, the outgoing state is always
  ;; normalized.
  (let ((result
        (vector+ (state-vector state1)
                  (state-vector state2))))
    (if (not (normalized? result))
        (error "Join: Unnormalized outgoing state"
              result))
    (make-simple-state result)))

```

The most interesting point about it is that, like `detector` and `block`, it uses `make-simple-state` rather than `make-state` to generate the outgoing state. This guarantees the independence of measurements along different components, since when the spin attempts to be realigned along a particular direction, the only computation which will be redone is that starting at the last observation. In particular, although the spin is perfectly determined coming into the detector, it is completely unrestricted coming out. It so

happens that the initial outgoing value is precisely the incoming vector, thus if two devices along the same direction are placed right after another, all the states coming into the second device will have their spins aligned along the correct direction, making the choice deterministic in the second device (all the vectors will be normalized?, that is they will have magnitude one or zero). Thus it is only by “accident” that sequential spin measurements along the same direction give the same results.

```
(define (split-one incoming spin-direction receiver)
  (let ((other-direction (orthogonal-direction spin-direction)))
    (define (into first second)
      (make-state
       (lambda (new-state)
         (let* ((nv (state-vector new-state))
                (n (norm nv)))
           ;; Consistency check
           (if (not (parallel? nv first))
               (error "Split into: Bad direction"
                      first nv)
               (update
                incoming
                ;; The else clause in the conditional is conceptually
                ;; sufficient. If there were no numerical accuracy
                ;; problems, it would compute the correct vector, but
                ;; round-off errors make the other clauses necessary.
                (cond ((zero? n) second)
                      ((one? n) first)
                      (else
                       (let* ((second-factor
                              (sqrt
                               (- 1 (square (norm nv))))))
                         (second-component
                          (scale second
                                 second-factor)))
                        (vector-+ nv
                                 second-component)))))))
        (project-into first (state-vector incoming))))
      (receiver (into spin-direction other-direction)
                (into other-direction spin-direction))))))
```



```

(define (split incoming spin-direction receiver)
  (let ((v (state-vector incoming)))
    (cond ((not (normalized? v))
           (error "Split: Unnormalized incoming state"
                  incoming))
          ((one? (norm v))
           (split-one incoming spin-direction receiver))
          (else ; norm is zero
           (let ((components (make-non-existent-state)))
             (receiver components components))))))

(define (make-non-existent-state)
  (make-state
   ;; This state should never be updated
   (lambda (updated)
     (error "Split: unnecessary update")))
  (make-zero-vector))

```

The code for `split` presented above does not handle the case where the state vector incoming to a Stern-Gerlach device is not normalized?. This is a serious restriction, which can be solved, but which would make the code much more complicated. The code above works in the Stern-Gerlach experiment and in the EPR paradox experiment (see below) because the incoming states to the devices in those experiments are always normalized? either by originating in the source, or by being the result of a previous observation. This point is further discussed in chapter 5.

## 4.4 Code for the EPR paradox experiment

The model for the EPR paradox experiment is a simple extension of the above. The spin measurement devices are the same, and the rest of the representation is shared. We only need a source which creates pairs in the singlet state, and the appropriate devices. Thus the additional code is:

```
;; MAKE-PARTICLE-STATE-PAIR returns two correlated particles.
;; Since Scheme does not have multiple value returns, this is
;; accomplished by continuation passing:
;; MAKE-PARTICLE-STATE-PAIR receives an argument, which is a
;; procedure of two arguments. These two arguments will be the
;; two particles in the pair, and the procedure will be invoked
;; at the end of MAKE-PARTICLE-STATE-PAIR, which will not
;; return directly. Effectively it is as if
;; MAKE-PARTICLE-STATE-PAIR had returned the two particles, and
;; RECEIVER had been immediately invoked with these two
;; particles as arguments. This could also have been achieved
;; by returning a list (or pair) of the two particles. In
;; fact, this is exactly how MAKE-COUPLED-PARTICLE-STATES
;; works. Note that the initial, shared continuation is
;; obtained here. Receiver here is similar to the receiver argument
;; to SPLIT above.
```

```
(define (make-particle-state-pair receiver)
  (let ((result
        (call-with-current-continuation
         (lambda (here)
           (make-coupled-particle-states
            here
            (make-unit-vector))))))
    (receiver (car result) (cdr result))))
```

```
;; MAKE-COUPLED-PARTICLE-STATES creates two states with related
;; upgrade functions. They share a continuation, obtained in
;; the procedure above, but the upgrade functions are not just
;; simply the continuation as in the Stern-Gerlach case above.
```

```

(define (make-coupled-particle-states cont vector)
  (define (coupled-state my-direction make-compound)
    (make-state (lambda (new-state)
                  (let* ((direction
                        (state-vector new-state))
                        (other
                        (orthogonal-direction direction)))
                    (cont
                     (make-compound
                      (make-simple-state direction)
                      (make-simple-state other))))))
    my-direction))
  (cons
   (coupled-state vector cons)
   (coupled-state (orthogonal-direction vector)
                   (lambda (x y) (cons y x)))))

;; This procedure is the one used to compute successes
;; and failures. It returns a list of the outcomes: If the
;; norm is 1, the particle was seen, if 0, it was not. There
;; are no other possibilities. Thus a particular test was
;; successful if both norms came out 1.

(define (bell-apparatus apparatus-1 apparatus-2)
  (lambda (state-1 state-2)
    (list (norm (state-vector (apparatus-1 state-1)))
          (norm (state-vector (apparatus-2 state-2)))))

;; These use the definitions for the Stern-Gerlach devices
;; above.

(define a-spin-up z-spin-up)
(define b-spin-up x-spin-up)
(define c-spin-up      ; bisector
  (normalize (vector+ a-spin-up b-spin-up)))

;; These are Stern-Gerlach devices, and the convention is the
;; same as above.

```

```
(define appafb (apparatus a-spin-up free-path block))
(define appbfb (apparatus b-spin-up free-path block))
(define appcfb (apparatus c-spin-up free-path block))
```

The key idea is that initially both particles in the pair have related upgrade functions. These initial upgrade functions have the continuation for the expression which created the particle pair available to them. When any of the particles of the correlated pair is observed –assume in the following that the one observed is the one on the left branch– both particles are updated simultaneously. The upgrade function for each of the individual particles computes the appropriate states for *both* particles, and the computation is restarted with a new particle pair at the point at which the original pair was created. The new component states, corresponding to each of the particles in the pair, will contain spin vectors in opposite directions, but their upgrade functions will no longer be related. Each particle will now have its own independent upgrade function which will no longer upgrade the state corresponding to the other particle<sup>26</sup>.

Although the upgrade functions are no longer related, and both particles will not be realigned together any more when further observations occur, the spin vectors they have are opposite to each other. Therefore, if the particle on the right branch is now measured along the same axis as the particle on the left branch, its spin will *already* be aligned in that direction, and the result will be completely consistent with the result observed on the other branch. The 100% opposite correlation of measurements in the same direction is guaranteed in this way.

Assume instead, that the particle on the right path is measured along a different axis. Then, when the new desired state is computed as a result of the new observation, the upgrade will only “affect” that particle, and not its sibling on the other branch. The reason for this is that the new upgrade functions, which are again mere continuations, will only update the individual state corresponding to their particle, and not both. A fact to note is that, if the update functions tried to update both states again,

---

<sup>26</sup>In the code this is achieved by using `make-simple-state` once the update occurs. Only the original state pair uses `coupled-pair`.

the program would go into an infinite loop. Each particle would be trying to align both particles' spin along its own direction, but this is impossible since both directions are different. The uncoupling of the particles' spin becomes a necessity, not an accident.

The interesting correlations are obtained by the fact that the state vector on the right path is biased according to the observation on the left path, as explained in chapter 3.

## 4.5 Code for the double slit interferometer experiment

The code here shares the overall structure with the code for both experiments above, but the details are far more complicated for a variety of reasons:

- The first complication is geometry: The amplitudes and probabilities depend on positions and distances, not only on the “flow” between devices. Particle states must therefore incorporate something which allows amplitudes to be computed differently for different points in space.
- The second, and worst complication, is the appearance of negative observations. In the two experiments above, any observation would completely determine the state of the particle being observed immediately before the observation. This is no longer the case here: A detector placed at a particular position may not detect a particle, but this does not restrict the state sufficiently to determine completely the state of the particle prior to the observation. Particle states need to incorporate some mechanism to specify progressive stages of restriction, until they eventually become determined. This problem is analyzed further in chapter 5.

Most of the geometry is handled by the representation of surfaces: Surfaces are represented as collections of “gaps”. A gap is an infinitesimal oriented surface whose attributes are, position, direction, and area.

```
(define (make-gap position dir area)
  (make-gap-internal position (normalize dir) area))

(define (invert-gap gap)
  (make-gap-internal (position gap)
                    (scale (direction gap) -1)
                    (area gap)))
```

The fundamental operation on gaps is to find the spherical angle covered by a gap when viewed from a particular position. The `mabs` procedure is used because gaps are oriented, thus the angle covered is 0 if the gap points “the wrong way”.

```
(define total-angle (* 4 pi))
(define half-angle (/ total-angle 2))

(define (mabs x)
  (if (positive? x) x 0))

(define (gap-angle origin gap)
  (let* ((v (vector-- (position gap) origin))
        (rr (dot v v)))
    (if (zero? rr)
        half-angle
        (min half-angle
              (/ (* (area gap)
                    (mabs (dot (direction gap)
                               (normalize v))))
                 rr))))))
```

The probability that a particle is observed on a surface (collection of gaps) is computed by adding the probabilities that it is seen on any of the gaps. The probability on a single gap is computed by “squaring” the amplitude.

```
(define (probability particle . gaps)
  (sum-of-map (lambda (gap)
                (amplitude->probability (amplitude particle gap)))
              gaps))

(define (sum-of-map f . args)
  (sum (apply map f args)))

(define (amplitude->probability amplitude)
  (* amplitude (conjugate amplitude)))
```

Particle states are not as simple, or static as in both experiments above. A particle state consists of the following elements:

- An upgrade (replace) function. This is, as above, the central piece to the observation mechanism.
- An amplitude function. This takes a gap as its argument, and computes the amplitude that the particle is seen on that gap.
- A forbid function. After an observation in which the particle was not seen on a set of gaps, this constructs a new particle state for which the amplitudes for those gaps are zero. The new state is often immediately used to “update” the current particle state.
- A restrict function. This is used after a positive observation, where the particle was observed on a set of gaps. Again, the most frequent case is when it is immediately used to “update” the current particle state.

Observation is handled by `determine`, which uses the update procedure contained in the particle state to update the state to accommodate the result of the observation. If the particle was in fact observed, `determined` generates the new particle state, otherwise the “forbid” procedure from the particle state is used to generate the replacement.

```
(define (determined original cgaps)
  (let* ((probs (map (lambda (gap)
                     (probability original gap))
                    cgaps))
        (total (sum probs)))
    (forbidden-determined-state
     original
     cgaps
     (map (lambda (prob) (/ prob total)) probs)
     (map (lambda (gap) '()) cgaps)
     make-simple-particle-state)))
```



```

(define (determine particle . gaps)
  (let ((prob (apply probability particle gaps)))
    (cond ((deterministic? prob)
           particle)
          ((choose prob)
           (replace particle (determined particle gaps)))
          (else (replace particle (forbid particle gaps))))))

```

The simplest state is a spherically symmetric state, ie., a state in which the probability of finding the particle on any surface covering the same spherical angle when viewed from the source is the same. Such a state is generated by a spherical source, and constructed by `forbidden-particle-state`. This procedure constructs states somewhat more general than simple spherical states. The states returned by `forbidden-particle-state` are mostly spherically symmetric, except that they are forbidden from appearing in a given set of gaps. A spherically symmetric state is therefore just a forbidden particle state whose set of forbidden gaps is empty.

```

(define (make-spherical-source origin #!optional intensity)
  (lambda ()
    (forbidden-particle-state
     (if (unassigned? intensity) 1 intensity)
     origin
     '()
     make-simple-particle-state)))

(define (forbidden-particle-state intensity origin cgaps maker)
  (let ((factor
        (max 0
            (- 1
              (/ (sum-of-map (lambda (gap) (gap-angle origin gap))
                           cgaps)
                 total-angle))))
    (if (or (zero? intensity) (zero? factor))
        ;; MAKER ignored, better not try to update it
        (make-null-particle-state)
        (inner-forbidden-state intensity factor origin cgaps maker))))

```

```
(define (make-simple-particle-state amplitude forbid restrict)
  (call-with-current-continuation
    (lambda (replace)
      (make-particle-state replace amplitude forbid restrict))))
```

factor is used in forbidden-particle-state to adjust the intensity of the source so that the probability over the complete spherical angle is not lowered. All of the work is done in inner-forbidden-state.

```
(define (inner-forbidden-state intensity factor origin cgaps maker)
  (define (ifs-amplitude gap)
    (* intensity
      (common-amplitude
        origin
        (/ (max 0
            (- (gap-angle origin gap)
              (sum-of-map (lambda (cgap)
                            (path-angle origin gap cgap))
                          cgaps))))
          factor)
      gap)))
```

```
(define (ifs-forbid myself gaps)
  (forbidden-particle-state
    intensity
    origin
    (merge-forbidden-gaps origin gaps cgaps)
    (updater myself)))
```

```
(define (ifs-restrict gaps probs)
  (restricted-spherical-state origin
    intensity
    gaps
    probs))
```

```
(maker ifs-amplitude ifs-forbid ifs-restrict))
```

`inner-forbidden-state` calls `forbidden-particle-state` when the particle is not observed in a given set of gaps. The new set of gaps from which the particle is forbidden is constructed by `merge-forbidden-gaps`.

If the particle is observed on a set of gaps, the new state is generated by `restricted-spherical-state`, which constructs a spherical state which is allowed only to be observed on the given set of gaps.

```
(define (restricted-spherical-state origin intensity cgaps probs)
  (let ((total-prob (sum probs)))
    (if (zero? total-prob)
        (make-null-particle-state)
        (make-restricted-particle-state
         (let ((factors
                (map (lambda (prob cgap)
                       (/ prob (amplitude->probability
                                (simple-amplitude origin cgap))))
                     probs cgaps)))
           (named-lambda (rss-amplitude gap)
             (make-polar
              (sqrt
               (sum-of-map (lambda (cgap factor)
                             (* factor
                                (amplitude->probability
                                 (path-amplitude origin gap cgap))))
                           cgaps factors))
              (angle (* intensity (simple-amplitude origin gap))))))))))
```

The most general state is created by `forbidden-determined-state`. This procedure divides space into two regions: The region before the gaps, ie. the region between the source and the gaps where the particle was observed, and the region beyond the gaps. `forbidden-determined-state` also generates a version of the original particle state which is “restricted” to travel through these gaps. It then generates and returns a new state which behaves differently in each of these regions.

```

(define (forbidden-determined-state original cgaps iprobs iflist maker)
  (define (generate-state flist maker)
    (let* ((probs (new-determined-probs iprobs cgaps flist))
           (before (restrict original cgaps probs))
           (huygens-particles
            (map (lambda (gap forbidden)
                  (forbidden-particle-state
                   (amplitude before gap)
                   (position gap)
                   (cons (invert-gap gap) forbidden)
                   make-fixed-particle-state)))
                cgaps
                flist)))
      (define (fds-amplitude gap)
        (if (before? cgaps (list gap))
            (sum-of-map (lambda (huey) (amplitude huey gap))
                        huygens-particles)
            (amplitude before gap)))
        (define (fds-forbid myself gaps)
          (if (before? cgaps gaps)
              (generate-state
               (map (lambda (cgap old)
                     (merge-forbidden-gaps (position cgap) gaps old))
                   cgaps flist)
               (updater myself))
              (error
               "Forbidden-determined-state: Forbidding determined particle")))
          (define (fds-restrict gaps nprobs)
            (if (before? cgaps gaps)
                (restricted-determined-state original cgaps gaps
                (probability-transfer-table probs cgaps flist gaps nprobs))
                (error
                 "Forbidden-determined-state: Restricting determined particle")))
            (maker fds-amplitude fds-forbid fds-restrict)))
      (generate-state iflist maker))

```

This new state generated by `forbidden-determined-state` behaves like the restricted version of the original particle state in the region before the gaps, but behaves like a superposition of states in the region beyond the gaps. The component states in this superposition are spherical states given by Huygens's principle: Their origins are the constraint gaps, and their amplitudes at their respective origins are the amplitudes given by the restricted state at the constraint gaps. Thus the amplitude for the compound state on a gap in the region beyond the constraint gaps is given by the sum of the amplitudes for each of the individual Huygens components. If the new particle state is not observed in a gap, the new particle state is constructed by forbidding all of the Huygens components from "appearing" on this gap.

`restricted-determined-state` generates the states which result from restricting superposition states. This procedure is similar in structure to `forbidden-determined-state`, but ignores anything not on the direct path between the "sources" of the Huygens particles and the gaps to which the superposition state has been restricted.

```
(define (restricted-determined-state original cgaps1 cgaps2 prob-table)
  (let* ((before (restrict original cgaps1 (sum-columns prob-table)))
        (huygens-particles
         (map (lambda (gap restricted)
               (restricted-spherical-state (position gap)
                                           1
                                           cgaps2
                                           restricted))
              cgaps1 prob-table)))
        (define (rds-amplitude gap)
          (if (before? cgaps1 (list gap))
              ;; This needs some phase information
              (sqrt
               (sum-of-map (lambda (huey)
                           (amplitude->probability (amplitude huey gap)))
                          huygens-particles))
              (amplitude before gap))))
```

```
(make-restricted-particle-state rds-amplitude)))
```

The actual devices used to test the above code are defined as follows:

```
(define ((wall . slits) particle)
  ;; Simplification: for now it decides to go through the wall or not
  ;; when it hits it, rather than when an observation is made.
  ;; For the current purposes this makes the code simpler but should
  ;; not change the results.
  (apply determine particle slits))

(define ((detector gap) particle)
  (determine particle gap))

(define gap+ (make-gap '(2 1) '(1 0) .1))
(define gap- (make-gap '(2 -1) '(1 0) .1))

(define the-source
  (make-normalized-spherical-source '(0 0) gap+ gap-))

(define the-wall (wall gap+ gap-))

(define the-detector (detector gap+))

(define device-1 the-wall)
(define device-2 (compose the-detector the-wall))
```

The remaining code is support code for the above procedures.

## 5 Conclusions

Problems with the model and future work:

As noted above, the simplicity of the code arises from the simplicity of the experiments. For instance, if in the Stern-Gerlach experiment we were to have non-normalized states incoming to particular devices, the representation of states would have to change drastically. Using the terminology defined above, the code is simple only when all results are positive, but becomes considerably more complex when some of the results are negative, as the code for the double slit interferometer experiment demonstrates.

The basic mechanism of restarting the computation at the appropriate point after an observation can still be used, but in some instances the states recomputed must have special knowledge about the devices they had already encountered before the previous computation was aborted. Such a situation arises, for instance, in a device like the one depicted in figure 10.

If an observation was made on path 1, and the particle was observed there (positive result), there would be no problem. On the other hand, if the particle was not observed there, the computation would have to be restarted at the  $x$  split point. Yet there is no single state vector we can use to restart the computation, since the  $x$  and  $y$  splits are independent (inconsistent?). Thus the state with which the computation is restarted must have built into it the fact that there can be no component along path1 when it eventually reaches the upper  $y$  splitter. The code in section 4.5 incorporates such a mechanism, but this is the source of most of its complexity.

A fact to notice is that the model only attempts to “explain” observation, not other aspects of quantum mechanics. Thus it is not applicable to situations where the quantum mechanical behavior is of a different sort. In particular, results like the Aharonov-Bohm effect<sup>27</sup>, insofar as observation is not concerned, lie outside the scope of the model. It should not be very hard (in principle, at least) to incorporate effects like these into the model given that the state-vector component of a system state, as described in

---

<sup>27</sup>This is the name given to the dependence of the phase of the amplitude along a path on the magnetic vector potential. See [Baym73] for a derivation and discussion.

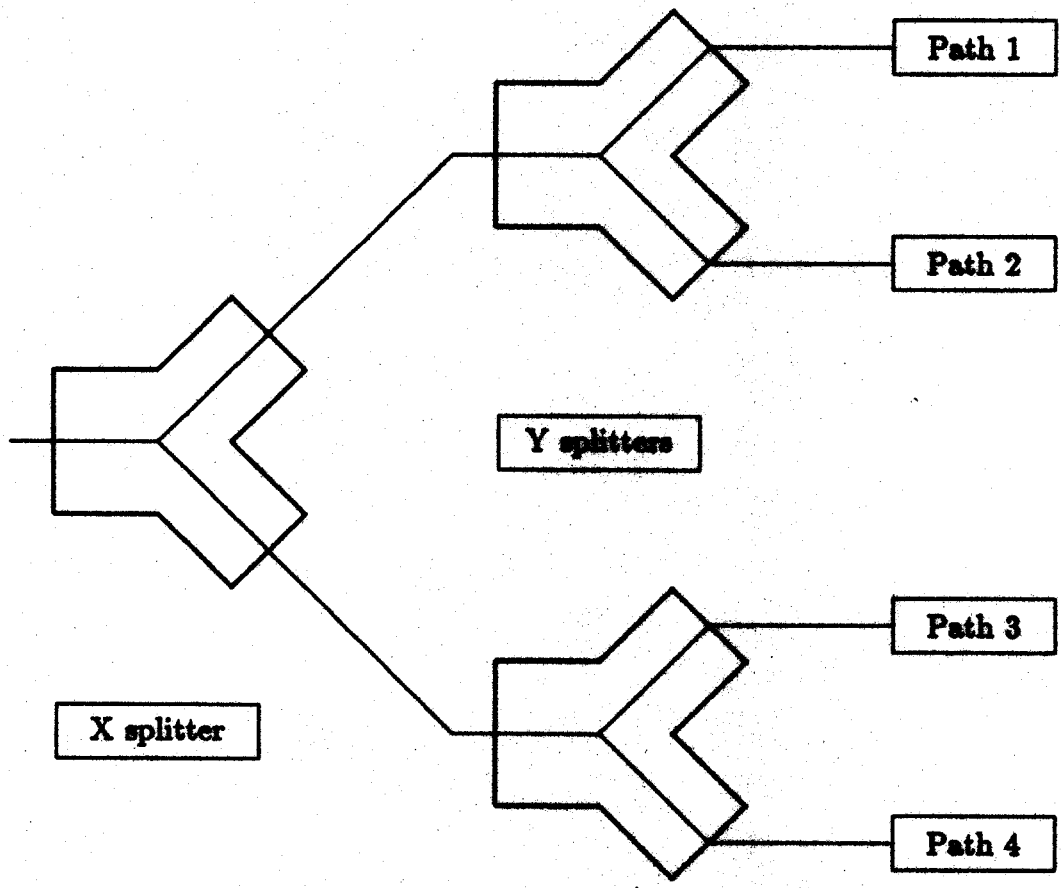


Figure 10: Device requiring complicated backtracking.



previous sections, is closely related to the state vector in traditional formulations. Effects like Aharonov-Bohm would involve just transformations of the state vector with trivial transformations on the upgrade functions. Further observation would be handled by the same mechanism.

There is a problem even when this is taken into account, namely the pervasive assumption that the “split” and observation are localized: The ambiguity arises at a particular location, and at a particular point in time, and observation is similarly constrained. In the general case, where the situation may not be so clear cut, the model may not be so easily adaptable. In particular, a continuous “split”, as in the dispersion of a wave packet given by Schrödinger’s equation, will be particularly hard to accommodate. The main problem is that it is not clear to what point the system should backtrack when an observation happens.

A related conceptual problem is that the model does not explain which interactions result in observations and which do not. For example, in the Stern-Gerlach experiment, interaction with a block or a detector results in observation, but interaction with a “splitter” does not. The present model differentiates implicitly between observational and non-observational interactions, and what kind each device performs must be built into the model. A more comprehensive model would eliminate this distinction or have a coherent theory from which this information could be deduced.

A Computer Science-theoretical point to explore is the possible connection between the implementation of the model and the issue of consistent choice in general non-deterministic programming. For example, the following program, which uses McCarthy’s non-deterministic AMB operator

```
((lambda (x) (+ x x))
 (amb -1 1))
```

can return the values 0, 2, or -2, depending on the particular semantics involved<sup>28</sup>. The value 0 can be considered to be an interference between the two consistent possibilities for the value of  $x$  (-1 and 1). It may be possible to build a model for quantum mechanics based on the above observation.

---

<sup>28</sup>See [Clinger82] for a description of the possibilities.

In fact, it was thoughts along these lines that originally led me to the model under discussion, although the connection, if any, with the model at hand is not obvious.

There are also problems of a pragmatic nature with the model: Unrestricted recomputation is extremely expensive. It is often the case that independent, unmodified branches of a computation have to be recalculated although there was nothing in the observation that affected them. The recomputation mechanism used is too coarse, much like a sledge-hammer. Everything which occurs after a split is recomputed, even those things that do not depend at all on the nature of the split! This is especially noticeable in the presence of negative results. In their absence, there is never a need to backtrack to a decision point twice. The first observation completely determines the state at the choice point, and further backtracking to that point would only result from errors in the code. In the presence of negative results (as in the example above, or the code for section 4.5), a chosen state is only constrained by observation, not determined, so further backtracking may be necessary to further constrain it as more observations occur. When there are multiple dependent choice points, the backtracking can become prohibitive. This is related to a well known problem in Artificial Intelligence programming. My model uses a technique closely related to simple chronological backtracking, but dependency-directed backtracking is often considerably more efficient<sup>29</sup>.

Programmers must also be somewhat careful when coding their programs using the paradigm described above. The recomputation caused by observation must be isolated from pieces of code which have nothing to do with the mechanism manipulating particle states. For example, if a statistics gathering function were written in a poor way, a recomputation of some state could cause the whole procedure to be restarted and lose the information already gathered about previous experiments. This isolation could be achieved easily by having more than one process. A recomputation would only affect the particular process being observed, and potentially others whose computation depended on some preliminary value of the process being observed. In fact, this can be extrapolated into a system where

---

<sup>29</sup>See [Stallman76] for a description of this technique.

each object is implemented as a process. An observation is then just an interaction between processes, where the result of the interaction has to be deterministic. Thus at the time of the observation, the observer and the observed object agree on the result of the observation, and both upgrade their own state to be consistent with the result. This provides a model where observation is a particular case of interaction, much more symmetric, and more in line with many physicists' beliefs.

In general, the model is simple and can be used to explain or illustrate the behavior of simple quantum mechanical systems under observation. It is not a panacea, however, and there are many details which need to be worked out before the variations of this model become powerful enough to be used for significant modeling and/or prediction. The model is, nevertheless, flexible enough that it can accommodate many possibilities. I think that the idea of using recomputation to guarantee consistency should be explored further to develop even more powerful models of quantum mechanical phenomena. Detailed mechanistic models like the one under discussion should be helpful in gaining intuition about the microscopic world. In particular, it may be possible to integrate the ideas which give rise to this model into a more complete interpretation of quantum mechanics. The resulting interpretation would probably be similar to the transactional interpretation presented in [Cramer86], which uses advanced waves to accomplish what this model achieves with recomputation, but has a very similar overall flavor.

## A Results

The following are the results of the Stern-Gerlach experiment on two different runs. 20000 particles are driven through each apparatus on each run. The numbers given are the relative frequencies (number of successful events divided by total number of tests) of detecting a particle at the output of the apparatus.

Null: 0  
Weird: .1256  
Weirder: .25005

Null: 0  
Weird: .1261  
Weirder: .2485

The corresponding probabilities assigned by quantum mechanics (trivial exercise) are:

Null: 0  
Weird: .125  
Weirder: .25

The following are the results of the EPR Paradox experiment. Again, the relative frequencies are given, and each run consisted of 20000 particles. A test is successful (and the event counted) if a particle is observed at the output of each device, i.e., both particles in the pair make it through their respective devices.

AB: .2436  
AC: .0724

CB: .0751

AC + CB = .1475

AB: .2523

AC: .07625

CB: .0731

AC + CB = .14935

The corresponding probabilities assigned by quantum mechanics (see [Sakurai85] for this calculation, but remember to include the factors of 1/2 dropped there.), are:

AB: .25

AC: .0732

CB: .0732

AC + CB = .1464

The actual results can be seen to be very close to the predicted results, as expected. The above results took 4 days to compute, running interpreted MIT "Gator" Scheme on an HP9000, 237 (Gator) computer.

Unfortunately, the code for the double slit experiment is even slower, and the lack of a compiler which can optimize this sort of code painfully apparent. Thus I was not able to run the emulation long enough to generate graphs like those in chapter 2.2. I have run the experiment for the most significant points, however, both in the absence and the presence of a detector close to one of the slits. These situations correspond to devices 1 and 2, respectively, in section 4.5. The most significant points are those where a maximum and a minimum occur when there is interference. The results are as follows, expressed as relative frequencies:

Number of particles: 10000

Device-1 at maximum: .0032

Device-1 at minimum: 0

Device-2 at maximum: .0012

Device-2 at minimum: .0011

The corresponding probabilities predicted by quantum mechanics are:

Device-1 at maximum:  $3.01975e-3$

Device-1 at minimum:  $1.59275e-5$

Device-2 at maximum:  $1.50988e-3$

Device-2 at minimum:  $1.48113e-3$

The actual results are not as close to the probabilities predicted as I would like, but this is probably due to the small number of particles processed in each run. I believe that a faster version of the code, hopefully available soon, will alleviate this problem. The pattern is, however, the desired one, and the results are within 1 standard deviation of the predicted probabilities.

## B Reimplementation

As mentioned in section 4.1, the code can be reorganized somewhat to eliminate the need for `random`. A non-functional procedure remains, but this procedure is used only each time a particle is generated. Even this non-functionality could be eliminated if instead of generating a new particle every time, we were to use the same particle over and over, in sort of a loop.

The main idea is that particle states can be extended to include a stream<sup>30</sup> of random numbers. Every time a state is updated the “new” state contains a new random number stream, derived from the old one, by, for example, taking its tail. This corresponds to the random number generator changing its state, so that it will return a different number on the next invocation. In this way the “generation” of random numbers can be done in a purely functional style. The random number streams are generated at particle “creation” time, and since the rest of the code is functional (and therefore deterministic), these streams must in general be different, so the particles created will not all behave the same. Thus the only non-functional procedure in the code is precisely the procedure that generates the random number streams. This stream is then used to generate the initial state, and to choose which path to follow at the observation points.

In this implementation the particle state contains all the relevant state, and examining it immediately after generation allows us to predict exactly what the particle’s behavior will be when confronted with a given set of devices. This modification, therefore, removes all non-determinism during the execution of the actual experiment, and the only non-determinism that remains is at particle generation time.

This implementation has even one more advantage: A source of “globality” is removed. In the implementation described in chapter 4, the procedure `random` is used throughout the code, and thus there are potential correlations between different and independent particles and states caused by this sharing. In this implementation, this sharing disappears, since each particle will have no effect on the rest. Each particle has its own ran-

---

<sup>30</sup>See [Abelson84].

dom number stream, and the rate at which it “consumes” random numbers cannot affect what values other particles will see.

As an example, here is the new version of some of the code in sections 4.2 and 4.3:

```
(define (make-state upfunc state-vector rand-stream)
  (list upfunc state-vector rand-stream))

(define (make-particle-state)
  (generate-new-state-info make-simple-state))

;; This procedure depends on the only global state (which is inside
;; new-random-stream). The rest is functional. It creates different
;; particles by giving them a different random stream.

(define (generate-new-state-info make-state)
  (let ((rand-stream (new-random-stream)))
    (make-state (make-unit-vector (->angle (head rand-stream)))
                (tail rand-stream))))

(define (update state new-vector new-stream)
  (let ((upfunc (upgrade-function state)))
    (upfunc (make-state upfunc
                        new-vector
                        new-stream))))

(define (collapse state)
  (let ((v (state-vector state))
        (s (rand-stream state)))
    (let ((n (norm v)) ;Prob. amplitude
          (new-v
            (let ((if (choose (square n) (head s)) ;Probability
                          (scale v (/ 1 n))
                          (make-zero-vector))))
            (update state new-v (tail s))))))
```



```
(define (detector state)
  (let ((v (state-vector state)))
    (if (normalized? v)
        (make-simple-state v (rand-stream state))
        (collapse state))))
```

The rest of the code, including that for the other experiments, can be transformed in a similar way.

## References

- [Abelson84] Abelson, H., and Sussman, G., with Sussman, J.  
*Structure and Interpretation of Computer Programs*  
M.I.T. Press, Cambridge, Massachusetts, 1984.
- [Abelson85] Abelson, H. *et al.*  
*The Revised Revised Report of Scheme or An UnCommon Lisp.*  
AI Memo 848, M.I.T. Artificial Intelligence Laboratory,  
Cambridge, Massachusetts, August 1985.
- [Aharonov80] Aharonov, Y., and Albert, D. Z.  
*States and observables in relativistic quantum field theories.*  
in *Physical Review D*, Vol. 21, no. 12, 3316-3324, June 1980.
- [Baym73] Baym, G.  
*Lectures on Quantum Mechanics*  
Benjamin/Cummings, Menlo Park, California, 1973.
- [Bell64] Bell, J. S.  
*On the Einstein Podolsky Rosen Paradox*  
in *Physics*, Vol. 1, No. 3, 195-200, Nov./Dec. 1964.  
Also in [Wheeler83].
- [Bohr35] Bohr, N.  
*Can Quantum-Mechanical Description of Physical Reality  
be Considered Complete?*  
in *Physical Review*, Vol. 48. 696-702, 1935.  
Also in [Wheeler83].
- [Clinger82] Clinger, W.  
*Nondeterministic Call by Need is Neither Lazy Nor by Name*  
in *Proceedings, 1982 ACM Symposium on Lisp and Functional  
Programming.*
- [Cramer86] Cramer, J. G.  
*The transactional interpretation of quantum mechanics*  
*Reviews of Modern Physics*, Vol. 58, No. 3, July 1986.
- [Davies86] Davies, P. C. W., and Brown, J. R.  
*The Ghost in the Atom*  
Cambridge Univ. Press, Cambridge, 1986.

- [d’Espagnat79] d’Espagnat, B.  
*The Quantum Theory and Reality*  
in Scientific American, Nov. 1979.
- [DeWitt73] DeWitt, B. S., and Graham, N.  
*The Many-Worlds Interpretation of Quantum Mechanics*  
Princeton Univ. Press, Princeton, New Jersey, 1973.
- [Dirac58] Dirac, P. A. M.  
*The Principles of Quantum Mechanics*, 4th ed.  
Oxford Univ. Press, London, 1958.
- [Einstein35] Einstein, A., Podolsky, B., and Rosen, N.  
*Can Quantum-Mechanical Description of Physical Reality  
be Considered Complete?*  
in Physical Review, Vol. 47, No. 10, 777-780, May 15, 1935.  
Also in [Wheeler83].
- [Felleisen86] Felleisen, M., Friedman, D., Kohlbecker, E., and Duba, B.  
*Reasoning With Continuations*  
Indiana University Computer Science Technical Report No. 191,  
May 1986.
- [Feynman63] Feynman, R. P., Leighton, R. B., and Sands, M.  
*The Feynman Lectures on Physics*  
Addison-Wesley, Reading, Massachusetts, 1963.
- [Landau76] Landau, L. D., and Lifshitz, E. M.  
*Mechanics (Course of Theoretical Physics; v. 1)*  
Pergamon Press, Elmsford, New York, 1976.
- [Landin65] Landin, P.  
*A Correspondence between Algol 60 and  
Church’s Lambda Notation: Part I*  
in Communications of the ACM, 8(2), 89-101, Feb. 1965.
- [Sakurai85] Sakurai, J. J.  
*Modern Quantum Mechanics*  
Benjamin/Cummings, Menlo Park, California, 1985.
- [Schrödinger35] Schrödinger, E.  
*Die gegenwärtige Situation in der Quantenmechanik*

in *Naturwissenschaften* 23: pp. 807-812; 823-828; 844-849, 1935.  
Also in [Wheeler83].

[Stallman76] Stallman, R. M., and Sussman, G. J.  
*Forward Reasoning and Dependency-Directed Backtracking  
in a System for Computer-Aided Circuit Analysis*  
MIT Artificial Intelligence Laboratory Memo 380, Sept. 1976.

[Wheeler83] Wheeler, J. A., and Zurek, W. H.  
*Quantum Theory and Measurement*  
Princeton Univ. Press, Princeton, New Jersey, 1983.

**CS-TR Scanning Project**  
**Document Control Form**

Date : 10/19/95

Report # AI-TR-928

Each of the following should be identified by a checkmark:  
Originating Department:

- Artificial Intelligence Laboratory (AI)  
 Laboratory for Computer Science (LCS)

Document Type:

- Technical Report (TR)     Technical Memo (TM)  
 Other: \_\_\_\_\_

**Document Information**

Number of pages: 74 (80-images)  
Not to include DOD forms, printer instructions, etc... original pages only.

Originals are:

- Single-sided or  
 Double-sided

Intended to be printed as :

- Single-sided or  
 Double-sided

Print type:

- Typewriter     Offset Press     Laser Print  
 InkJet Printer     Unknown     Other: \_\_\_\_\_

Check each if included with document:

- DOD Form     Funding Agent Form     Cover Page  
 Spine     Printers Notes     Photo negatives  
 Other: \_\_\_\_\_

Page Data:

Blank Pages (by page number): \_\_\_\_\_

Photographs/Tonal Material (by page number): \_\_\_\_\_

Other (note description/page number):

Description :	Page Number:
<u>IMAGE MAPS (1-74) UN# AND TITLE &amp; BLANK PAGES</u>	<u>1-72</u>
<u>(75-80) SCANNING POL, COVER, DOD, TRETS (3)</u>	

Scanning Agent Signoff:

Date Received: 10/19/95    Date Scanned: 10/19/95    Date Returned: 10/26/95

Scanning Agent Signature: Michael N. Cook

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AI-TR-925	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  A Computational Model for Observation in Quantum Mechanics		5. TYPE OF REPORT & PERIOD COVERED  Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)  Guillermo Juan Rozas		8. CONTRACT OR GRANT NUMBER(s)  N00014-86-K-0180
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE 3/16/87
		13. NUMBER OF PAGES 73
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report)  UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Distribution is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES  None		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Quantum Mechanics Computational Models Scheme Search		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  A computational model of observation in quantum mechanics is presented. The model provides a clean and simple computational paradigm which can be used to illustrate and possibly explain some of the unintuitive and unexpected behavior of some quantum mechanical systems. As examples, the model is used to simulate three seminal quantum mechanical experiments. The results obtained agree with the predictions of quantum mechanics (and physical measurements), yet the model is perfectly deterministic and maintains a notion of locality.		

# Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives**, using funds from the **Advanced Research Projects Agency** of the **United States Government** under Grant: **MDA972-92-J1029**.

The scanning agent for this project was the **Document Services** department of the **M.I.T. Libraries**. Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences**.

