Sequent Calculus Representations for Quantum Circuits

Cameron Beebe
MCMP

October 31, 2014

Abstract

When considering a sequent-style proof system for quantum programs, there are certain elements of quantum mechanics that we may wish to capture, such as phase, dynamics of unitary transformations, and measurement probabilities. Traditional quantum logics which focus primarily on the abstract orthomodular lattice theory and structures of Hilbert spaces have not satisfactorily captured some of these elements, and neither have recent attempts at constructing a system for quantum programs. Rather than focusing on these previous attempts, we can start from 'scratch' in an attempt to conceptually characterize the types of proof rules which should be in a system that represents elements necessary for quantum algorithms. This present work attempts to do this from the perspective of the quantum circuit model of quantum computation. A sequent calculus based on single quantum circuits is suggested, and its ability to incorporate important conceptual and dynamic aspects of quantum computing is discussed. In particular, representing measurement probabilities and phase preserves the representation of interference as a resource in quantum computation. Additionally, denying left weakening on the intuitive basis of interference leads to a non-monotonic calculus. Instead of abstracting away the conceptually important phase and measurement probabilities we can abstract away from (some) matrix and linear algebra calculations—relegating them to sub-routines not explicitly handled in the higher-level language. This approach helps preserve the logical representation of essential ingredients and resources in quantum algorithms.

Contents

1 Introduction

In Gentzen-style sequents for structural proof theory, for example the system **G1cp**, elements in the antecedent (represented by the multi-set Γ) are interpreted as having a conjuctive relationship to one another, whereas elements in the succedent (the multi-set Δ) are interpreted as having a disjunctive relationship between one another. In other words, on the 'meta-level' we read the sequent $\Gamma \Rightarrow \Delta$[1] classically as

$$\bigwedge(\Gamma) \rightarrow \bigvee(\Delta) \tag{1}$$

When writing out some particular proof of an argument, the elements of Γ and Δ are separated by commas (i.e. $\Sigma, A, A \rightarrow B \Rightarrow B, \Theta$). These 'structural' commas between the terms are read in the metalanguage as '$\wedge$' on the left side, and as '$\vee$' on the right side. From the conjunction of the 'premises' in the antecedent, we can obtain proofs of the 'conclusions' in the succedent. Then, a truth-functional interpretation of the sequent corresponds classically in that the sequent holds iff at least one element in Γ is false or one of Δ is true. (see e.g. [Paoli, 2002, §3.1])

In the sequent calculus **G1cp**, in addition to the operational rules for the various connectives $\{\wedge, \vee, \rightarrow, \neg\}$ we include the structural rules of (left and right) weakening and contraction, and can show the admissibility of *Cut*. Various so-called "sub-structural logics" take aim at these structural rules, for example we could have reasons for dropping weakening, contraction, or *Cut* from a logical system. An intuitionistic system such as **G1ip**, for example, will restrict proofs to one term (or no term) in Δ in the succedent—denying the general ability to introduce formula by right weakening[2] and therefore also we will never be able to eliminate a duplicate occurrence of a formula by right contraction.

Other examples are found in the literature concerning a quantum logic, or a logic that is based on theoretical or mathematical considerations from quantum physics. Unfortunately, there are many problems to be found in the traditional motivations and attempt at defining a satisfactory quantum logic—not to mention problems in explicating why exactly these logical systems are important, or how they are related to the physical world.[3] It is not the main purpose of this present paper to criticize what can be called the traditional attempts in this area, however a few comments can be mentioned.

First, the original attempt from Birkhoff and von Neumann (BvN) [Birkhoff and Neumann, 1936] (which seems to be much of the basis for subsequent

[1] For now, the sequent arrow $\Rightarrow$ is equivalent with $\vdash$, although later there will be a distinction between the two symbols.

[2] Unless, of course, Δ is empty.

[3] For example, consider that the non-distributive lattices are 'read off' from Hilbert spaces—complex vector spaces with an inner product defined. This doesn't explain the relationship Hilbert spaces have to the quantum world, or why the quantum world can be formalized through Hilbert spaces. A logic based on such a formalism does not get any closer to a physical *meaning*, and in fact may obscure certain physical notions rather than clarify them.

literature, e.g. in [Putnam, 1979, §10]) appears to be a discussion over *static* properties, such as posterior outcomes of measurements. For quantum programs, it makes little sense to base our logical calculus on the posterior (and static) aspects of event structures. Rather, we will be using coherent unitary transformations describing temporal dynamics in our quantum algorithms—and thus they should appear as the primary focus in the proof system. A few recent attempts have shown that incorporating a time index or dynamic modalities can begin to reconcile this issue. While I find it relieving that these recent attempts have breathed fresh life into the subject, I have not been particularly convinced.

Briefly, I can mention what aspects of quantum computation these approaches have not captured but which *should* be captured in a proof system for quantum programs. Importantly, as [Baltag and Smets, 2004] readily admit,[4] treatments of phase relations and measurement probabilities are lacking of representation in their system. This is critical, since many quantum protocols involve *interference*—which is perhaps most apparent in "sandwiched" applications of Hadamard gates **H** (a particularly useful unitary transformation).

The quantum lambda calculus approach developed in [van Tonder, 2004] seems like a very fruitful contribution to quantum computation. It is unclear to me at this time whether such lambda expressions could also be applied to the sequent representation suggested here—although this would be ideal and should be explored (i.e. are parts of the formulation here equivalent to the more advanced exposition in [van Tonder, 2004]).[5] Again, though, my focus is on the conceptual representation of quantum computational procedures in a sequent-like formulation. Intuitively, the deductions in proofs will be seen as roughly equivalent expressions to the circuit model depictions of quantum algorithms.

In my view, a logic for quantum programs should abstract away from some operations but still preserve the intuitive nature of quantum algorithms and the resources involved. In particular it should represent (i) the *phase* of a superposition; and (ii) it should represent measurements, amplitudes, and measurement probabilities.

[4]One can glean the quantum circuit model motivation for the present approach from the admission: "Our notion of 'state' in this paper is closely connected to the way quantum logicians approach quantum systems; i.e., contrary to identifying states with unitary vectors (as customary in quantum computation), we took them to be *one dimensional subspaces* generated by these vectors. This imposes some limits to our approach, mainly that we will not be able to express *phase*-related properties." [Baltag and Smets, 2004, p. 41] The authors then rightly admit that a more complicated logic would result from a treatment of these factors, and that such a logic would require a tensor operator. Presumably what is meant by this latter comment is that there must be a connective corresponding to the tensor product, which we will incorporate here into a proof system.

[5]For example, I will take unitary transformations such as the Hadamard gate to be of critical logical importance in the sequent formulation, they are the basic logical operations. In the lambda calculus developed by van Tonder, these appear to be constants in the quantum lambda calculus.

1.1 Motivation for a Logic of Quantum Programs

Using the terms algorithm and program as loosely equivalent, a quantum algorithm is an explicit step by step procedure using the formalism of quantum mechanics[6] to achieve some computational task. Some famous examples of quantum algorithms are the Deutsch-Josza and Shor algorithms—the purpose of the latter is to factor products of two large prime numbers efficiently through period finding, enabling us to break for example the widely-used RSA cryptosystem. The possible implementation of these type of powerful tasks on some physical architecture creates a demand for deeper philosophical analysis and practical development of quantum programs.

Much of the philosophical literature on quantum logic has unfortunately been on why its underlying partial Boolean algebra (PBA) is of the structure of non-distributive orthomodular lattices, and drawing consequences from this non-classicality. However, in the opinion of the present author, such attempts are mislead based on the aforementioned lack of dynamics in the logics. It is simply not very interesting to discuss quantum logics which are static (as many recent authors, some of them already mentioned, have noted). Girard [Girard, 2003, §1] would perhaps even trace the problem to a more general one in the history of logic, in which it seems that the static denotational or semantic aspects have been much more developed than the dynamic, constructive, and proof-theoretic aspects of logic.

Rather, we should focus on finding a *constructive* approach to a quantum logic, enabling us to establish a correspondence between proofs in this logic and algorithms or programs. Also, the so-called Curry-Howard correspondence extends to a correspondence between propositions and types in type theory.

> "A specification [of a problem to be solved in a program/what a program should do] is in type theory expressed as a set, the set of all correct programs satisfying the specification. The programming process is the activity of finding and formulating a program which satisfies the specification. In type theory, this means that the programmer constructs an element in the set which is expressed by the specification." [Nordström et al., 1990, p. 4]

We can formulate this relation something like:

$$\frac{proof}{proposition} :: \frac{term}{type} :: \frac{program}{specification} \tag{2}$$

While it is not my intention to prove this correspondence for the rules to be discussed, or to give typed lambda calculus representations of the proposed sequent calculus, it is with this correspondence in mind that I approach the field of quantum logic through proof theory rather than the traditional analysis of lattices and Hilbert space structures.

[6]Dirac notation and the Hilbert space representation is presumed, rather than the somewhat cumbersome wave mechanical formulation.

Other approaches towards developing quantum programming languages, and related formal aspects of theoretical quantum computer science, are varied and fragmented yet developing quickly. See for example the survey in [Ying et al., 2012]. Some particularly interesting developments in this area with respect to logic are dynamic modal logics put forward by Sonja Smets and Alexandru Baltag in order to capture important aspects of quantum information theory and quantum programming. These are inspired by what they call the "dynamic turn" in logic. [Baltag and Smets, 2012]

It should be mentioned that the project discussed here is not completely original, nor is it directly in line with other similar approaches. For example the project in [Selesnick, 2003] also explores the proof-theoretic and type-theoretic foundations for quantum computing, although the conceptual and semantic discussion is less present than what I hope to achieve here. It is my hope that the familiar reader will admit that certain treatments found in this present work are more conceptually economical than the aforementioned work for example with regard to measurement, but also it is not clear yet at this stage whether the proof rules discussed are capable to be incorporated into previous work in this area. It may also be the case that similar rules have been proposed elsewhere, but as of the time of writing I have not encountered them.

So, while I am looking to develop the proof-theoretic side of quantum programs, I am motivated by intuitive semantics and formalisms from quantum computation and the circuit model more than I am by formal quantum logics (which in my experience are *not* used in the typical exposition of quantum computation and quantum algorithms).

1.2 What are we talking about?

Before entering into a discussion, where formal sequent calculus will primarily be used, we need to first establish what kind of notation to use—and determine what exactly the notation should be representing. It seems that the former is in some ways more difficult than the latter. It would be very nice to be able to write quantum algorithms[7] down without calculus, linear algebra, or vector notation (even though Dirac notation is quite nice). However even then we would still like to have representations of phase and measurement probabilities, since these may be important aspects of the procedure—either for practical implementation or for theoretical understanding. It seems that some mix of sequent calculus, Dirac notation, and linear algebra may end up being the most clear—and this is the route pursued here.

While I cannot possibly hope to resolve the entire situation in this present paper, the motivation is clear: if we are to have quantum algorithms, and implement them and discuss them in philosophy and theoretical computer science, then a 'higher-order language' involving symbolic logic is preferable to working with differential equations and vector spaces. In a logic for quantum programs, then, we should try to find a balance between

[7]Or the quantum aspects of a procedure, if there are also classical tasks involved in an algorithm.

compact formal logical representations of states, unitary operations, and measurements—without compromising the mathematical expressiveness that is necessary to understand *how* quantum algorithms work. For example, it seems that in our logic for quantum programs we should require that the Born Rule and its associated probabilities are represented—but I have not seen such distributions in the sequent-style representations put forward elsewhere. (e.g. see [Selesnick, 2003])

As far as the language which the specification/propositions/types will range over, I consider it generally to consist of physical statements about the quantum world. Typically these would be formulated in wave mechanics or equivalently in matrix mechanics (however the wave mechanics is more 'ontic' with respect to the imagined token operation performing some type of task). These statements should include statements about what happens during unitary transformations and the dynamics of the physical world in-between and independent of our observations or measurements—and importantly shall not be limited to what Birkhoff and von Neumann originally called *experimental* propositions (which would appear to be a subset of the possible *physical* propositions).

On a superficial level, the background language here can just be considered linear algebra. Logical operations will be abbreviations of, for example, the tensor products of vectors and matrices. With this in mind, we begin in the next section with a sequent analysis of representing and manipulating such statements as the above.

2 Reading Sequents as Quantum Computations

In the field of theoretical quantum computation, the following expression represents the quantum analogue of a binary digit or *bit*, the *qubit*:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{3}$$

Here, $|\psi\rangle$ (read as "ket psi" in what is known as Dirac notation) is a superposition, i.e. a linear combination represented by the addition sign, of the two basis states $|0\rangle$ and $|1\rangle$. Kets are taken to be column vectors whose entries are complex numbers, whereas 'bras' $\langle\cdot|$ are row vectors whose entries are the complex conjugates of the corresponding ket. For example, our basis states are the following column vectors:

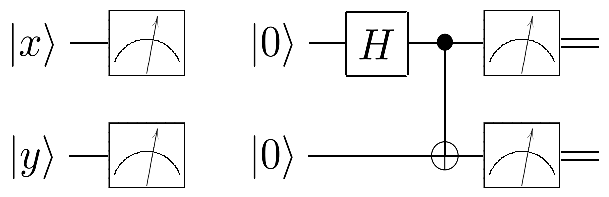
$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \qquad\qquad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \tag{4}$$

The coefficients α, β in (3) are complex numbers called *amplitudes*, and this title is best understood through the wave mechanical formalism of quantum mechanics. We require of these amplitudes that $|\alpha|^2 + |\beta|^2 = 1$, where $|\cdot|$ denotes the modulus of the complex number. We call the results of such a calculation the Born Rule, to be discussed further in due time.

From combinations of these building blocks, including unitary linear operators on state vectors and vector spaces containing them, linear algebra forms a formal language which is capable of representing procedures useful in theoretical computer science. A sequent calculus based on these formal aspects will informally be read with this background in mind. Furthermore, intuitive considerations from the field of quantum computation will help determine the types of rules to be used in the system, as well as how to properly read and interpret the applications of the rules.

2.1 Quantum Circuits

The primary motivation for the sequent system to be discussed in the following sections is found in the quantum circuit model of quantum computation. In this model, procedures and manipulations of states are represented diagrammatically in what are called 'circuits'. As an example, consider the following 'circuit':[8]



(5)

Where $|x\rangle$ and $|y\rangle$ are unknown states of single qubits, put through a measurement (represented by the box with the arrow). Say both of these detectors display 0. Then, as discussed in more detail shortly, we can assign the state $|0\rangle$ to both of these qubits. This is called *state preparation*, and will be used to define axioms in our sequent system—allowing us to cut the left part of this diagram off (which one could also do in the circuit diagram). Then, in the subsequent part of the diagram, two transformations are performed on these states—the first is one of the most common gates in quantum computation called a Hadamard gate **H**, represented by the 2×2 matrix

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{6}$$

Subsequently, the operation represented by the vertical link represents a two-qubit gate **CNOT**, or controlled-not. Depending on the state of the control qubit (where the black dot is), this 4×4 matrix will xor (addition modulo 2) the second qubit.

[8]The double lines after measurement are supposed to represent that we have a classical state, not a quantum one. One might think that there should also be double lines connecting to the $|0\rangle$s, and I agree, but whether they should be double or single lines is not the point right now. Unfortunately, for practical reasons, the programming behind this diagram, called qasm2circ (which was produced by Isaac Chuang), is not developed enough to be as expressive as we might wish for the present nuanced discussion. Alas, this is a regrettable practical limitation at this moment.

$$\mathbf{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{7}$$

This is remembered since the normal notation for addition modulo 2 in this context is represented by $\oplus$, but this should not be confused with the later use of the same symbol to denote exclusive disjunction. Whereas the first set of measurement gates were for state preparation, the second set on the right side of the circuit will be upon the two-qubit entangled (non-factorizable) state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$, which the reader can check with the above statement of Born's rule will result in measuring either $|00\rangle$ or $|11\rangle$ with the equal probability of one-half.

With this preliminary crash course in the formalism used in quantum computation and the circuit model of representing quantum protocols, we can now begin to outline a sequent calculus built to represent similar functions in quantum computer science. We will see how this same circuit is represented in a sequent calculus for single quantum circuits.

3 Building a Sequent Calculus for Single Quantum Circuits

3.1 Axiom and State Preparation

Given an unknown qubit state we can subject it to a measurement, and by noting whether or not our measurement device shows a 0 or 1, we can assign the post-measurement state to either the basis state $|0\rangle$ or $|1\rangle$. [Mermin, 2007, §1.10] We can apply the $\mathbf{X}$ gate to the qubits in state $|1\rangle$ and thus prepare all of our states uniformly into $|0\rangle$ states. This initialization procedure can be seen as giving us the 'premises' of our sequent proofs, i.e. the leaf nodes will be initialized into the $|0\rangle$ state. The following pure state can be considered effectively as an *axiom* in our system:

$$|0\rangle \Rightarrow$$

Although, given the above statements, such a state may be the conclusion of a measurement in a sub-derivation higher up in our proof tree. This will be returned to later. It will be useful to note that we will be able to initialize n of these basis states by $n-1$ applications of the conjunction rule $\otimes$, discussed in the next section. The reason for keeping the succedent blank will also be explained in due course, but for those who cannot wait, the way we will read this axiom is actually

$$|0\rangle \Rightarrow P(|0\rangle)$$

Where P represents a probability distribution calculated through the Born Rule (BR). For this pure state, $P(|0\rangle) = 1$. We will want to 'normalize' the introduction of this probability-calculation (represented by the predicate P), by putting it off until the penultimate step of the proof (i.e. the step which is the premise of a concluding measurement rule application). From now on, my choice of notation of keeping the succedent empty will reflect this 'normal' form. One can see, though, that this isn't too much of a stretch from the usual axiom $A \Rightarrow A$, and one will see that it is quite natural given the measurement conditional reading of the sequent arrow, to be discussed in section 4.3. These initialized states are the primary inputs of our unitary transformations, which will be considered our quantum logic gates—or, in other words, the logical operations that introduce or eliminate other formal symbols.

While the premise of the following rule has not yet been discussed (and its notation may be slightly worrying—no doubt adding to the discontent) I list it here for its relevance to the axiom above. In general, after we make a measurement we read out some classical value. As a general rule, if we were to measure again we would obtain the same result, leading us to ascribe the outcome of the measurement as the current state of the system:

$$(Prep)\,\frac{\Sigma \vdash_p |x\rangle_n}{|x\rangle_n \Rightarrow}$$

Σ is used instead of Γ, as a reminder that it can be regarded generally as the superposition of n qubits:[9]

$$\sum_{0\leq x\leq 2^n} \alpha_x \, |x\rangle_n \tag{8}$$

Which would mean that the corresponding circuit being represented in sequent form was an n qubit circuit, and that there were n measurements leading subsequently to n prepared qubits.

3.2 Terms in the Antecedent

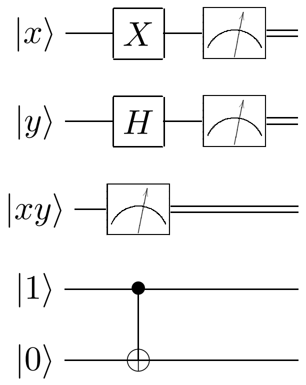
If Γ (i.e. the entire antecedent) were non-empty with more than one Σ, how would we 'read' the antecedent of the sequent? Suppose we wrote the following:

$$|0\rangle\,,|0\rangle \Rightarrow$$

How would we interpret the structural comma for a logic whose purpose is to describe or represent computational procedures in a quantum protocol—specifically a *single* circuit? In other words, what *state* is the 'computer' in? For a constructivist account of the antecedent

[9]Superpositions will be denoted by the symbol '+'. In the system considered here, it is introduced via use of a Hadamard gate, as a byproduct rather than a 'true' connective with an introduction and elimination rule (it is 'eliminated' when total destructive interference occurs, see section 6 for more). It will be important to note later, however, that it is precisely the '+' in a superposition which will give us the conjunctive-then-disjunctive sense on the meta level of reading our quantum sequent calculus.

above, how did we arrive at $|0\rangle, |0\rangle$? The commas will return later to denote the separation between multiple circuits. It may be helpful to keep in mind the difference in circuit notation between two qubits in one circuit, and two qubits in separate circuits, such as the following:



(9)

Here, $|x\rangle$ and $|y\rangle$ are single qubits each in their own separate circuits—the gates **X** and **H** are single qubit gates. These circuits could be performed separately, at different times, or in different procedures. The circuit measuring $|xy\rangle$ could also be represented as two separate wires, but only because the two-qubit measurement gate can just be constructed by scaling up a single measurement gate. The third example with a **CNOT** gate is a case where we *could* represent the situation with multiple single qubit circuits (i.e. in this case where $|1\rangle$ has an identity operator applied, and $|0\rangle$ is flipped by a not operator **X**), however for convenience we might like the multiple qubit circuit notation.

In other words, it becomes useful to consider some multiple qubit circuits as independent from their single qubit circuit representations. Similar reasoning later on will show it is useful to discuss multiple circuits as being practically independent (and non-interfering) from a larger single circuit in which these are embedded as 'sub'-circuits. For now, though, we are concerned with a single circuit sequent calculus, which can be thought of as a way to incorporate multiple circuits into a larger single circuit. Let us first take a look at some typical left conjunction rules, since both states must be in the same circuit.

3.2.1 Multiplicative vs. Additive AND

The following is the intuitionistic rule (in **G1ip**, the same rule as **G1cp** but restricted in the succedent) for introducing $\wedge$ in the antecedent:

$$\frac{\Gamma, A_i \Rightarrow C}{\Gamma, A_0 \wedge A_1 \Rightarrow C}\,(L\wedge)(i \in \{0,1\})$$

In a system like **G1ip** with left weakening (and left contraction), the above 'additive' rule can be shown equivalent to its 'multiplicative' counterpart:

$$\frac{\Gamma, A_0, A_1 \Rightarrow C}{\Gamma, A_0 \wedge A_1 \Rightarrow C}\,(L\wedge')$$

This multiplicative conjunction is obviously the type of conjunction that can help us create a single circuit from $|0\rangle, |0\rangle$ simply because both circuits must be present in order to combine them. However, it will be noted in further detail later on that we should *not* generally admit left weakening into a proof system for quantum programs because of interference—and so there is no such equivalence between additive and multiplicative conjunctions. There is no additive conjunction. In this case, it may be better not to use the $\wedge$ symbol to represent our multiplicative conjunction. From Girard's linear logic, we could also use the conjunctive symbol '$\otimes$' for our conjunction:

$$\frac{\Gamma, A_0, A_1 \Rightarrow C}{\Gamma, A_0 \otimes A_1 \Rightarrow C}\,(L\otimes)$$

This symbol will be preferred, since it will be mimicking the tensor product in linear algebra. While no tensor products will actually be calculated in the logic, such operations must be carried out in the background or by hand. There will be no contexts Γ in the single circuit calculus, but they will return when multiple circuits are discussed in the appendix.

We now have a proposal for a multiplicative conjunction to include in our system—call it **QMC**(Quantum sequent calculus for Measurements and Circuits). The system will also be intuitionistic, since there will only ever be one formula in the succedent—the subformulas of which will be candidate formulas for succedents of conclusions in a quantum proof.[10] More on this later.

Unfortunately there are still some unsatisfactory issues, which can perhaps be alleviated by the following constructive account of the terms in the antecedent. Take for example the notation mentioned earlier of the terms denoted by $|0\rangle, |0\rangle$. Let the following 'rule' *LComma* construct such a term from two instances of our axiom—but notice that the 'conversion' on the right shows the equivalence between $L\otimes$ above and $L\otimes'$ in the presence of the comma introduction rule:

$$(L\otimes)\,\frac{(LComma)\,\dfrac{|0\rangle \Rightarrow \qquad |0\rangle \Rightarrow}{|0\rangle, |0\rangle \Rightarrow}}{|0\rangle \otimes |0\rangle \Rightarrow} \quad \rightsquigarrow \quad (L\otimes')\,\frac{|0\rangle \Rightarrow \qquad |0\rangle \Rightarrow}{|0\rangle \otimes |0\rangle \Rightarrow}$$

The rule $L\otimes'$ is considered more primitive since it not only captures the constructive, multiplicative, and conjunctive aspect of our resources—it also avoids the *LComma* rule application. Since we will not have a $R\otimes$ rule, I will refer to $L\otimes'$ as $\otimes$ in **QMC**. In the general case it takes the form

$$(\otimes)\,\frac{\Sigma_n \Rightarrow \qquad \Sigma_m \Rightarrow}{\Sigma_n \otimes \Sigma_m \Rightarrow}$$

[10] In other words, measurement will pick out some components from the superposition.

where n, m denote the number of qubits in a general superposition. Using this rule, we can initialize (or represent the initialization of) n $|0\rangle$ states (written by convention $|0\rangle^{\otimes n}$)[11] by the application of $n-1$ gates, or rule applications, of $\otimes$.

3.3 Measurement as the Sequent Arrow

In [Hughes, 1989, p. 303], the author discusses the notion of a "measurement conditional", formulated in terms of the probability of a quantum observable A. This is to be read "If an ideal measurement M of A is made, then the result will lie within some range Δ."

$$MA \rightarrow (A, \Delta) \tag{10}$$

However, for the purposes here it will be helpful to instead formulate measurement as a proof rule rather than a predicate associated with a connective. In other words, there will be not be an arrow connective in **QMC**. Rather, the sequent arrow will be read in the way that the above measurement conditional arrow is read. While the counterfactual reading of the measurement conditional will be retained in the sequent calculus used here, a proof ending in an actual measurement cannot continue in this manner since a measurement *has been performed*. Thus, the reading of the sequent arrow $\Rightarrow$ will switch from a counterfactual one to a statement of what has been proved (measured).

The result of a quantum program will depend on measurements of the state, in which we 'call' the result of a sequence of unitary transformations (the program). It is at this point that the 'conclusion' of the proof, a.k.a. the outcome of the program, is reported. Very generally, and again this is part of a conceptual approach being developed here, we can regard the sequent arrow '$\Rightarrow$' as indicating a potential quantum measurement. Antecedent terms discussed in the previous section are taken to be coherent states of computations in a quantum computer, whereas the terms in the succedent will represent possible outcomes after evaluating the result of the computations.

The principle of deferred measurement in the circuit model of quantum computation states that we can always put off our measurements to the end of whatever computational procedure we wish to perform. That is, in cases where we do not use classical information extracted from the measurement (as in the teleportation protocol), we can defer the measurements to a later stage while preserving the essential elements of the algorithm.

When we view the sequent arrow in this manner, we 'bake' into the syntax a dynamic aspect. This can be interpreted as representing in one sense *time*, and in another sense the non-unitary character of a measurement process (either epistemically or literally depending on the interpretation). The temporal aspect is that the potential outcomes of measurements, catalogued in the succedent, are presumed to occur at some time after a measurement. In other words, the succedent would be later in time than the antecedent if

[11]Convention in the quantum computational literature for tensor products (here mimicked by the same symbol $\otimes$) of n states is $|0\rangle^{\otimes n} = |0\rangle_1 \otimes \cdots \otimes |0\rangle_n = |0\rangle_1 |0\rangle_2 \ldots |0\rangle_n = |0_1 \ldots 0_n\rangle$.

the proof ended and a measurement was made. Additionally, we may want to characterize the conclusion of each unitary logic gate as being at a later time than the line before—i.e. each line is at a later time than the previous line. This latter temporal sense may be problematic, however, and should be done carefully with the acknowledgement that it loses relevance in the multiple circuit calculus discussed in the appendix.

A further dynamic aspect is the essentialy dynamic calculation of measurement probabilities—which changes every time we apply unitary transformations to part or all of the coherent state in the antecedent.

3.4 Updating the Born Rule and Disjunction in the Succedent

To make sense of quantum measurements, Max Born famously proposed we take the complex-valued *amplitudes* α, β of wave components in a state and calculate the modulus squared of each such that $|\alpha|^2 + |\beta|^2 = 1$, interpreting the result as a probability density of observing a quanta or particle at that location. Since in the measurement of, for example, the state $\alpha |0\rangle + \beta |1\rangle$, we will only observe *either* the state represented by $|0\rangle$ *or* the state represented by $|1\rangle$—but with corresponding probabilities determined by the Born Rule process.

In other words, we will only observe one of the possible components in the superposition—an *exclusive* sense of the disjunction.

> "Measurement gates therefore play two roles in a quantum computation. They get the Qbits ready for the subsequent action of the computer, and they extract from the Qbits a digital output after the computer has acted. The initial action of the measurement gates is called state preparation, since the Qbits emerging from the process can be characterized by a definite state. The association of unitary operators with the gates that subsequently act on the Qbits permits one to update that initial state assignment into the corresponding unitary transformation of the initial state, thereby making it possible to calculate, using the Born rule, the probabilities of the outcomes of the final measurement gates."[Mermin, 2007, p. 31]

In the project here, two types of calculations in particular are relegated to background sub-routines (i.e. the formalism here abstracts away from them). These are the basic linear algebraic manipulations such as evaluating tensor products of matrices, but also the calculation through Born's rule of the measurement probabilities. The introduction of a predicate P into the succedent represents this on a conceptual level, but of course the distribution must be calculated for specific examples (i.e. specific Σ):

$$(BR)\,\frac{\Sigma \Rightarrow}{\Sigma \Rightarrow P(\Sigma)}$$

In general, the probability distribution introduced by BR would need to be updated after every application of a unitary gate. However, we aren't necessarily concerned with such calculations until the time when we *actually* want to make a measurement. So perhaps it is time to propose a sequent rule for measurement:

$$\frac{\Sigma \Rightarrow P(\Sigma)}{\Sigma \Rightarrow |x\rangle_n}$$

Where again Σ can be regarded as containing the general superposition in (8). However, as anticipated earlier, in this rule the sequent arrow has completely changed from a potential indicator of possibilities to an actual statement of what the state Σ has ended up being *actually* measured as. An alternative formulation in which the conclusion is represented instead by the turnstile is warranted:

$$(M)\ \frac{\Sigma \Rightarrow P(\Sigma)}{\Sigma \vdash_p |x\rangle_n}$$

Then we can still read the penultimate sequent arrow in the counterfactual sense found in Hughes' measurement conditional, while reading the $\vdash$ as 'proves' or 'measures', with probability p as a subscript. If part of the definition of an intuitionistic system is that there is only one term in the succedent, then this system must be intuitionistic in this sense. There is not an M application which results in more than one term in the succedent.

As a final note in this section, weakening on the *right* is disallowed on the intuitive basis that either (i) before measurement we cannot extend the Born rule (i.e. our probability distribution) over more states in the succedent than are present in the antecedent (i.e. in the coherent quantum state); or (ii) after measurement our derivation (or sub-derivation) is finished, and the only rule allowing us to continue from the turnstile form is $Prep$ which moves the measured state into the antecedent.

4 Interference and Non-monotonicity

The addition of a new state during our computation will *interfere* with the other states in a superposition. This can be viewed as a desirable representational feature of the calculus. One of the primary features of a quantum computer, acknowledged in the literature as a potential mechanism or resource enabling the so-called quantum speed up, is quantum interference. (see e.g. [Fortnow, 2003]) Interference allows certain 'computational paths' to destructively interfere (i.e. two equal, oppositely signed amplitudes of the same state will make a term disappear).

An important feature which this proof system should then exhibit is that adding 'premises' (or states in the antecedent) does not automatically guarantee that the same terms in the succedent are available to be measured or proved. This is an important logical feature, called non-monotonicity, and for many modern logicians such a property is seen as a beneficial representational aspect of a logical system. For example, it can reflect that upon adding new information to our knowledge or set of beliefs, the conclusions we might draw from reasoning with this set might change. Monotony would imply that adding elements to a premise set would still entail the same conclusions. In sequent notation, we can see that left weakening closely represents the monotonic notion (here presented with the turnstile, although this is not necessarily the particular post-measurement turnstile used previously):

$$(LWeak/Mon)\,\frac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta}$$

Thus, we can intuitively explain why our system will not in general admit of left weakening—interference implies that the sequent consequence arrow (interpreted in terms of measurement as discussed above) is *non-monotonic*. That a quantum proof system should not generally admit weakening because of interference has also been noted in [Selesnick, 2003, p. 406]. For a single-circuit sequent calculus, we have already discussed that the comma here must be interpreted as (and replaced by) the conjunctive connective $\otimes$.[12] There are, however, several formulations of monotonicity. One could call the following *general* monotonicity (again using the turnstile sequent notation here):

$$(GMon)\,\frac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$$

This rule could be admissible (i.e. shown that it does not lead to new derivations) in a system with both left and right weakening. For the present purposes, this is obviously not allowed in the quantum sequent system being considered. Weakening is not allowed on either side. The reader may wish to take a look at the appendix concerning a sequent calculus for multiple circuits, where general monotonicity is included in the specialized sense assuming non-interfering separate circuits. However, the more fundamental discussion of sequent representations of single circuits here is needed first to understand the underlying properties, such as the relationship of interference to non-monotonicity within single circuits. Another form of monotonicity can be called *restricted* [Gabbay, 1993, p. 110] and takes the following form:

$$(RMon)\,\frac{\Gamma \vdash A \qquad \Gamma \vdash \Delta}{\Gamma, A \vdash \Delta}$$

Unfortunately this formulation also fails to hold in **QMC**, if only for the use of the comma. However, we could imagine a rule in the spirit of the discussion earlier regarding the conversion from $|0\rangle, |0\rangle \rightsquigarrow |0\rangle \otimes |0\rangle$ with the $\otimes$ connective:

[12] Following this note, one can see that unlike in the classical case the set $\{a, b\}$ is not equivalent as a premise to the conjunction $a \otimes b$. Makinson [Makinson, 2005, p. 6] calls it 'conjunctive' monotony, when the set does behave the same as the conjunction.

$$(RMon/\otimes)\ \frac{\Gamma \vdash A \qquad \Gamma \vdash \Delta}{\Gamma \otimes A \vdash \Delta}$$

This can now just be seen as an application of $\otimes$, which would only hold if the turnstile is interpreted as the measurement conditional sequent arrow (there is no $\otimes$ rule for the post-measurement turnstile). This would mean that two probability distributions were calculated for the same state Γ, $A = P(\Gamma)$ and $\Delta = P(\Gamma)$, meaning that $A = \Delta$. However, the antecedent in the conclusion would then be $\Gamma \otimes P(\Gamma)$, and not only is there no rule which allows a BR application in the antecedent—it will not be that $\Gamma \otimes P(\Gamma) \Rightarrow P(\Gamma)$ since there would be double the amount of qubits in the antecedent.[13]

So, if we were to add a state in the antecedent according to 'conjunctive weakening' or 'conjunctive monotony' (i.e. an application of $\otimes$), we know intuitively that our probability distribution (our bookkeeping) over the succedent will change accordingly—and this is the primary motivation for postponing the calculation of such a probability distribution until the penultimate step before measurement. If A is added on the left, it will interfere with whatever Σ is also there, and so will the calculated probabilities on the right change. Really, then, adding some A in the antecedent can only be characterized by also adding $P(A)$ in the succedent—but in a single circuit calculus there can only be one probability distribution.

5 Proof Rules for Quantum Programs

In addition to considering unitary transformations (such as a universal set of such operators) as inference rules in the proof system, we add the proof-theoretic rules that have so far been discussed in this paper. The following rules which are defined are not to be considered exhaustive.

5.1 Definition of QMC

Earlier in the discussion, we were a bit more relaxed about the number of qubits involved in the rules. Here, subscripts n, m are added to denote the general rules for finite numbers of qubits. First there are what can be considered the 'identity' and 'structural' rules:

$$(Axiom)\ |0\rangle \Rightarrow \qquad (Prep)\ \frac{\Sigma_n \vdash_p |x\rangle_n}{|x\rangle_n \Rightarrow}$$

[13]One might object and say that $P(\Sigma)$ does not actually contain qubits. In that case, one can also easily show that if Σ is tensored with probabilities calculated through the Born Rule the resulting amplitudes will differ, and thus the distribution in the succedent will no longer match an updated Σ. As an example say $\Sigma = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. The resulting $P(\Sigma) = \frac{1}{2}(|0\rangle + |1\rangle)$. Even if we disregard the basis vectors in $P(\Sigma)$, and just multiply Σ by the probability $\frac{1}{2}$ (without the vectors the tensor product is ill-defined) then we obtain amplitudes in a new Σ' of $\frac{1}{2\sqrt{2}}$ which, when calculated by the Born Rule result in $P(\Sigma') \neq P(\Sigma)$.

Then perhaps the 'operational' rules:[14]

$$(BR)\,\frac{\Sigma_n \Rightarrow}{\Sigma_n \Rightarrow P(\Sigma_n)}$$

$$(\otimes)\,\frac{\Sigma_n \Rightarrow \qquad \Sigma_m \Rightarrow}{\Sigma_n \otimes \Sigma_m \Rightarrow}$$

$$(M)\,\frac{\Sigma_n \Rightarrow P(\Sigma_n)}{\Sigma_n \vdash_p |x\rangle_n}$$

Measurements on n qubit systems can be simulated by n singular qubit measurement gates. There are many unitary transformations that perform logical operations on antecedents. That is, there is a group of left rules U, where U is ideally some universal set of unitary operators. These rules are more difficult to represent in a general form, and in multi-partite operations their applications to specific qubits requires care. The form of the Hadamard gate for example, for x in the basis states $\{0, 1\}$, is:

$$(\mathbf{H})\,\frac{|x\rangle \Rightarrow}{\frac{(-1)^x}{\sqrt{2}}\,|x\rangle + \frac{1}{\sqrt{2}}\,|1-x\rangle \Rightarrow}$$

More work should be done on representing other unitary gates in the sequent formalism. Unitary transformations are the types of rule applications that are relevant for quantum computation, and instead of abstracting away the phase relations and probabilities (as I noted some previous attempts in this field do) we abstract away from the linear algebra calculations (which should be computed in the background via a subroutine).

6 Examples and Theoretical Results

6.1 Generating Entanglement

Applying unitary matrices to *states* in quantum computation are the type of operations we want to represent in a quantum proof system. In effect, the sequent representation proposed here is strikingly similar to quantum circuit representations by construction. For example, to generate or initiate an entangled state we can construct the following sub-proof (or sub-program):

[14]The traditional distinction between these types of rules, however, seems a bit unwarranted (or at least unclear) for M and *Prep*.

$$
\dfrac{\dfrac{\dfrac{|0\rangle \Rightarrow}{\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \Rightarrow}(\mathbf{H}) \qquad |0\rangle \Rightarrow}{(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \otimes |0\rangle \Rightarrow}(\otimes)}{\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \Rightarrow}(\mathbf{CNOT}) \tag{11}
$$

This is an unfinished program since we have not calculated measurement probabilities in the succedent, and have not applied a measurement rule M yet. There are many different ways to construct this particular entangled state. However, such a coherent state will be useful as an input for other computational steps. In the background, the state written as the conclusion of $\otimes$ actually is an example of distributivity. In other words, $(|x\rangle + |y\rangle) \otimes |z\rangle = |x\rangle \otimes |z\rangle + |y\rangle \otimes |z\rangle$. (The **CNOT** application above flipped the $|0\rangle$ above in the second $|z\rangle$ position to a $|1\rangle$.) This illustrates that on the surface of this proof system, we regard $\otimes$ as a more or less normal connective satisfying distributivity (but not commutativity), since in the background calculations performed it is distributive (linear) over the superposition '+' of vector components.

Continuing the above sub-derivation we can conclude the proof with an application of BR and a measurement. Remembering that $|00\rangle = |0\rangle\,|0\rangle$, we can use letters as subscripts to denote Alice's and Bob's registers (useful for example in discussing cryptographic protocols):

$$
\dfrac{\dfrac{\frac{1}{\sqrt{2}}|00\rangle_{AB} + \frac{1}{\sqrt{2}}|11\rangle_{AB} \Rightarrow}{\frac{1}{\sqrt{2}}|00\rangle_{AB} + \frac{1}{\sqrt{2}}|11\rangle_{AB} \Rightarrow \frac{1}{2}|00\rangle_{AB} + \frac{1}{2}|11\rangle_{AB}}(BR)}{\frac{1}{\sqrt{2}}|00\rangle_{AB} + \frac{1}{\sqrt{2}}|11\rangle_{AB} \vdash_{\frac{1}{2}} |00\rangle_{AB}}(M_2) \tag{12}
$$

The other proof available with equal probability would end in $|11\rangle_{AB}$ in the succedent of the conclusion of the measurement application. In either case, there is perfect correlation between Alice's and Bob's registers for this entangled state.

6.2 Interference as a Resource in Quantum Computation

In the constructive approach to quantum algorithms, here represented by step-by-step proof rules and background sub-routines evaluating linear algebra, we find an interesting result concerning the potential physical mechanism or resource used in the theoretical quantum computer. Not only are entangled states constructed (i.e. not primitive), we get an insight into perhaps the primary resource involved in a quantum protocol. *Interference*, while acknowledged in the literature, seems to be seen as less important of a resource than entanglement.

Another example will help to show a case in which interference is prominent. Two applications of the Hadamard gate **H** creates two oppositely phased components of the same state in the superposition.

$$\cfrac{(\mathbf{H})\ \cfrac{|0\rangle \Rightarrow}{\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \Rightarrow}}{\frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|0\rangle - \frac{1}{2}|1\rangle \Rightarrow}\ (\mathbf{H}) \tag{13}$$

Again, background calculations have been performed by a sub-routine (in this case, by hand). One can see, though, that the two $|1\rangle$ components when added together cancel out, leaving us with an amplitude (and subsequent probability) of 1 for $|0\rangle$. In other words, on the last line the register is simply in state $|0\rangle$. We could represent this by an interference proof rule, for example by explicitly canceling the two states out as a rule, but it seems that any sub-routine doing the calculations should already have done this work for us. I have written out the interfering state explicitly for the reader to see interference in action—to see why phase is an important aspect of our representation.

This "Hadamard-sandwiching" technique is used in a variety of quantum computational protocols, including some of the most powerful protocols such as Grover's and Shor's algorithms, as well as in ancillary qubits for error correction (see e.g. [Mermin, 2007]). Hadamard gates are applied to qubits initially, preparing them into uniform superpositions where subsequent transformations are applied, and afterwards a second Hadamard is applied to these qubits. Optical Hadamard gates may be physically realized through a beam-splitter, and such a gate was even used in the previous example to generate an entangled state. The conceptual semantics of the proof system given here emphasizes the role interference seems to play in the basic interactions between states in quantum computation.

7 Conclusion

The approach advocated for in this paper is contextualized in the history of logic through what can be called the proof theoretic semantic tradition, in which the meaning of logical connectives is just identified with the introduction rules for the connectives (and the elimination rules should follow). Their use in a proof (or *program*) determines the semantic meaning of the connective rather than a model or boolean function.

In the introduction, it is noted that the traditional or 'classical' reading of sequents is that the terms in the antecedent stand in a conjunctive relationship, and that terms in the succedent stand in a disjunctive relation to each other:

$$\bigwedge(\Gamma) \to \bigvee(\Delta) \tag{14}$$

However, after the above discussion, sequents for quantum programs (on single circuits) should rather be interpreted with alternative conjunctive and disjunctive relations:

$$\bigotimes(\Sigma) \to \bigoplus(P(\Sigma)) \tag{15}$$

Importantly, however, with the constructive interpretation of the multiplicative conjunction discussed earlier, we aren't interpreting structural commas. Since we have also restricted the formulas possible to introduce in the succedent, we also give up structural commas there. Rather, this meta-level reading is over the '+' involved in the summation (superposition) of a coherent state in the antecedent, and of the probability distribution in the succedent. The multiplicative conjunction seems to be the correct way to read the basic '+' in a superposition—i.e. to construct a coherent superposition $A+B$ we need *both* components simultaneously present in the same space (i.e. a superposition term is only well formed if the components share a medium). The exclusive disjunction also seems to be the correct way to interpret the '+' between elements in our calculation from the Born rule. However, interpreting $\Sigma, P(\Sigma)$ as multi-sets (or even sets) in the single circuit calculus is unnecessary—there is only ever one term involved in the antecedent and succedent.

An understandable objection to this project is that *this doesn't feel like logic anymore. This feels like cramming irregularly-shaped objects into regularly shaped holes.* To which I say: *exactly!* A quantum logic, particularly an applied one for algorithms and programs, *should* feel different. Quantum algorithms are not about the abstract ortho-logics and lattices read off of subspaces in a Hilbert space. Rather, they are about manipulating the quantum formalism through coherent unitary transformations and performing measurements on constructed states.

As in [Girard, 1995, §1.1.4-1.1.5], we should view terms and operations in the language as representing and manipulating computational resources—specifically quantum resources. These sort of "resource-based semantics" provide a rich way of approaching logic, particularly sub-structural logic, that deserve more attention in the philosophical community.

A final note should remind the reader that the formulation used here is partly inspired by other somewhat similar attempts, but also partly motivated as a response to limitations in other approaches. A proof system for quantum programs *should* represent phase and probabilities, since these may reflect essential resources or manipulations of resources that are important to explicitly represent in quantum algorithms.

8 Appendix A: A 'Multiple Circuit' Calculus

In the main text above, the proof system was developed directly from the consideration of the circuit model of quantum computation. Proofs represented *single* circuits. If we allow for the representation of multiple circuits in a sequent, say, by comma separation and indices, some structural rules begin reappearing. For example, since there is presumably no interference between terms in the separate circuits we can have left weakening—and this is interpreted as adding another circuit or circuit 'track'.[15] These separate circuits can be identified by an index, which will be denoted by a superscript to distinguish it from the subscript used to denote multiple qubits. Furthermore, the dynamic notion of time as we progress downwards in the proof will be distorted since we can perform all of the gate applications on one of the circuits and then write the first gate application for a second circuit. Consider the following, where we weaken in a basis state $|0\rangle$:

$$(BR^1, BR^2)\,\dfrac{(H^2)\,\dfrac{(\mathbf{CNOT})\,\dfrac{(LWeak)\,\dfrac{(\otimes)\,\dfrac{(\mathbf{X})\,\dfrac{|0\rangle \Rightarrow}{|1\rangle \Rightarrow} \qquad |0\rangle \Rightarrow}{|10\rangle \Rightarrow}}{|10\rangle^1, |0\rangle^2 \Rightarrow}}{|11\rangle^1, |0\rangle^2 \Rightarrow}}{|11\rangle^1, \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)^2 \Rightarrow}}{|11\rangle^1, \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)^2 \Rightarrow P^1, P^2}$$

Notice that in the succedent we must have two probability distributions, and therefore two applications of BR are required. (For ease of exposition I have substituted P^1 and P^2 in for the expanded calculations.) In this case, the question of which circuit the application of **CNOT** is unambiguous since it is represented by a 4×4 matrix and requires two qubits (a control and target qubit). Contraction is also still prohibited on the left, since this would remove an entire circuit. There may be some refined or carefully controlled situation where one could imagine admitting left contraction, but I leave that for future work.

We must presume, and this is a strong presumption, that the states separated by commas are not spatially overlapping. If they were, we would have to worry about interference among the terms in the respective circuits—essentially bringing us back to the single circuit calculus. Also, the available formulas in the antecedent and succedent can now be represented by $\Gamma \Rightarrow \Delta$, where Γ and Δ are now sets containing Σ's and $P(\Sigma)$'s respectively. Contexts return in the schematic statements of the proof rules of $\mathbf{QMC^M}$:

$$(\otimes)\,\dfrac{\Gamma, \Sigma_n \Rightarrow \qquad \Theta, \Sigma_m \Rightarrow}{\Gamma, \Theta, \Sigma_n \otimes \Sigma_m \Rightarrow}$$

$$(BR)\,\dfrac{\Gamma, \Sigma_n \Rightarrow \Delta}{\Gamma, \Sigma_n \Rightarrow P(\Sigma_n), \Delta}$$

$$(M)\,\dfrac{\Gamma, \Sigma_n \Rightarrow P(\Sigma_n), \Delta}{\Gamma, \Sigma_n \vdash_p |x\rangle_n, \Delta}$$

[15]We still cannot weaken on the right—it makes no sense to add an arbitrary probability distribution.

$$(Prep)\,\frac{\Gamma, \Sigma_n \vdash_p |x\rangle_n\,, \Delta}{\Gamma, |x\rangle_n \Rightarrow \Delta}$$

Δ in the succedent will be empty before the first application of BR. Indices could be added to help keep track of which circuits or states have been affected, as in the example above. The biggest issue with this calculus, with respect to the earlier calculus, seems to be the treatment of the sequent arrow as a measurement conditional, and the use of the post-measurement turnstile. Prolonging *Prep* after a measurement M makes the notation a bit confusing—for example the probability p will be an unwanted artifact in subsequent lines of the proof. More work is warranted on this issue, and the potential of this multi-circuit calculus to represent various quantum algorithms and protocols should be investigated further.

References

[Baltag and Smets, 2004] Baltag, A. and Smets, S. (2004). The logic of quantum programs.

[Baltag and Smets, 2012] Baltag, A. and Smets, S. (2012). The dynamic turn in quantum logic. *Synthese*, 186:753–773.

[Birkhoff and Neumann, 1936] Birkhoff, G. and Neumann, J. V. (1936). The Logic of Quantum Mechanics. *The Annals of Mathematics*, Vol. 37(No. 4):p. 823–843.

[Fortnow, 2003] Fortnow, L. (2003). One complexity theorist's view of quantum computing. *Theoretical Computer Science*, 292(3):597–610.

[Gabbay, 1993] Gabbay, D. M. (1993). A General Theory of Structured Consequence Relations. In Schroeder-Heister, P. and sen, K. D., editors, *Substructural Logics*, volume 2 of *Studies in Logic and Computation*. Oxford University Press.

[Girard, 1995] Girard, J.-Y. (1995). *Linear Logic: its syntax and semantics*, pages 1–42. Cambridge University Press. Cambridge Books Online.

[Girard, 2003] Girard, J.-Y. (2003). *Proofs and Types*. Cambridge University Press.

[Hughes, 1989] Hughes, R. I. G. (1989). *The Structure and Interpretation of Quantum Mechanics*. Harvard University Press.

[Makinson, 2005] Makinson, D. (2005). *Bridges from Classical to Nonmonotonic Logic*. King's College Publications.

[Mermin, 2007] Mermin, N. D. (2007). *Quantum Computer Science: An Introduction*. Cambridge University Press.

[Nordström et al., 1990] Nordström, B., Petersson, K., and Smith, J. (1990). *Programming in Martin-Löf's Type Theory*. Oxford University Press.

[Paoli, 2002] Paoli, F. (2002). *Substructural logics: a primer*. Kluwer Academic Publishers.

[Putnam, 1979] Putnam, H. (1979). *Mathematics, Matter, and Method*, volume 1 of *Philosophical Papers*. Cambridge University Press.

[Selesnick, 2003] Selesnick, S. A. (2003). Foundation for Quantum Computing. *International Journal of Theoretical Physics*, 42(3):383–426.

[van Tonder, 2004] van Tonder, A. (2004). A Lambda Calculus for Quantum Computation. *ArXiv*.

[Ying et al., 2012] Ying, M. S., Feng, Y., Duan, R. Y., Li, Y. J., and Yu, N. K. (2012). Quantum programming: from theories to implementations. *Chinese Science Bulletin*, 57:1903–1909.