Measurements in quantum programming language QML

Nely Plata-Cesar[1], J. Raymundo Marcial-Romero[1], and J. Antonio Hernández-Servín[1]

Facultad de Ingeniería
Universidad Autónoma del Estado de México
nplatac@uaemex.mx, jrmarcialr@uaemex.mx, xoseahernandez@uaemex.mx

Abstract. We present a proposal to add measurement operators to the quantum programming language QML, this strategy modifying the semantics on projections of products ($\otimes$), by adding some rules to allow measurements.

Keywords: Functional quantum programming · QML · Measurement

1 Introduction

Programming languages are an important field in computer science and quantum physics. In the last field it allows to represent algorithms applying quantum properties. Among the quantum programming languages, quantum lambda calculus λ_q and QML can be mentioned, being functional programming languages and the foundations.

The properties that quantum languages have are that programs operate with quantum data, that is, superpositions of the form $\alpha_t * t + \alpha_u * u$; Operations are represented with matrix and can be applied to superpositions, also having mixed states and finally to allow quantum measurements.

Quantum measurements on a system determine the probability that an experiment will happen. These occur with some interference (observer) or some phenomenon of nature that happens in these system, these measurements generate irreversibility once they occur.

The measurements in a programming language make it possible to move from the quantum to the classical environment, emitting the output of a program. The contribution of measurements to some languages, has been gradual, for example, the incorporation of measurements in quantum lambda happens after its definition [10].

Lambda calculus is a representative and useful language to theoretically study the foundations of mathematics and recursion [3, 4]. Van Tonder proposes quantum lambda calculus λ_q, with the mentioned above properties [10]. Subsequently, Díaz-Caro, Arrighi, et al. incorporate a family of measurement operators to λ_q calculus [5].

λ_q calculus in its initial definition does not incorporate measurements, similar to the QML language, which has a semantics with quantum data and control, but without measurements. In this paper the measurements are added to QML, taking as a guide the work carried out by Díaz-Caro, et al. [5].

Organization: This article is structured as follows: In chapter 2, you will find the definition of the QML language and general concepts of quantum measurements. In chapter 3, the description of the quantum lambda calculus language. Chapter 4, presents the contribution of adding measurement operators to QML. This proposal will be found as an example. And finally in part 5, the conclusions and future work are summarized.

2 Preliminaries

2.1 Quantum programming language QML

The QML language will be approached from the perspective of Altenkirch, Grattage, Vizzotto and Sabry, they work on the pure fragment of language and define your semantic model, they also develop a sound and complete equational theory, omitting recursive types and measurements [1]. This latest research will be retaken, the syntax of terms is:

(Variables)	$x, y, \ldots \in$	$Vars$
(Prob. amplitudes)	$\kappa, \iota, \ldots \in$	$\mathbb{C}$
(Patterns)	p, q	$::= x \mid (x, y)$
(Terms)	t, u	$::= x \mid () \mid (t, u) \mid$
		$\mathbf{let}\ p = t\ \mathbf{in}\ u \mid \mathbf{0} \mid$
		$false \mid true \mid \kappa * t \mid t + u \mid$
		$\mathbf{if}^{\circ}\ t\ \mathbf{then}\ u\ \mathbf{else}\ u'$

Table 1: Syntax for QML.

From the syntax you can form conventional programs such as tuples, let, if, among others, and in turn append terms with a probability amplitude κ, that is, have a probability value that they happen, and also define superposition $t + u$, it means than a term can be in t and u at the same time.

The types are given by the grammar: $\sigma = \mathcal{Q}_1 | \mathcal{Q}_2 | \sigma \otimes \tau$, where $\mathcal{Q}_1$ is the type (), which carry no information and $\mathcal{Q}_2$ corresponds to qubits (0 and 1). Semantically $[\![\sigma \otimes \tau]\!] = [\![\sigma]\!] \times [\![\tau]\!]$, where $\otimes$ is the standard product type and returns a tuple.
Typing contexts (Γ, Δ) are given by: $\Gamma, x : \sigma = \bullet | \Gamma, x : \sigma$, where $\bullet$ stands for the empty context. The context correspond to functions from a finite set of variables to types. For this case, we assume that every variable appears at most once. To maps pairs of contexts to contexts, the $\otimes$ operator is incorporated, with the following operation:

$$\begin{aligned} \Gamma, x : \sigma \otimes \Delta, x : \sigma &= (\Gamma \otimes \Delta), x : \sigma \\ \Gamma, x : \sigma \otimes \Delta &= (\Gamma \otimes \Delta), x : \sigma \text{ si } x \notin \mathrm{dom}\Delta \\ \bullet \otimes \Delta &= \Delta \end{aligned}$$

Interpret a judgement $\Gamma \vdash t : \sigma$ as a function is $[\![\Gamma \vdash t : \sigma]\!] \in [\![\Gamma]\!] \to [\![\sigma]\!]$, where, given a typing contexts, it returns a type of the collection of qubits.

To define well-formed programs $\Gamma \vdash t : \sigma$, the rules can be observed in Table 2. For example: $\Gamma \otimes \Delta \vdash \mathbf{if}^{\circ}\ c\ \mathbf{then}\ t\ \mathbf{else}\ u : \sigma$, requires that $\Gamma \vdash c : Q_2$ and $\Delta \vdash t, u : \sigma$, that is, c has a value of qubit, and t and u are the same type.

$$\frac{}{x : \sigma \vdash x : \sigma}\mathrm{var} \qquad \frac{\Gamma \vdash t : \sigma \qquad \Delta, x : \sigma \vdash t : \tau}{\Gamma \otimes \Delta \vdash \mathbf{let}\ x = t\ \mathbf{in}\ u : \tau}\mathrm{let}$$

$$\frac{\Gamma \vdash c : Q_2 \qquad \Delta \vdash t, u : \sigma}{\Gamma \otimes \Delta \vdash \mathbf{if}^{\circ}\ c\ \mathbf{then}\ t\ \mathbf{else}\ u : \sigma}\mathbf{if}^{\circ} \qquad \frac{\Gamma \vdash t : \sigma \qquad \Delta \vdash u : \tau}{\Gamma \otimes \Delta \vdash (t, u) : \sigma \otimes \tau} \otimes \mathrm{intro}$$

$$\frac{}{\bullet \vdash false : Q_2}\text{f-intro} \qquad \frac{}{\bullet \vdash true : Q_2}\text{t-intro}$$

$$\frac{\Gamma \vdash t : \sigma \otimes \tau \qquad \Delta, x : \sigma, y : \tau \vdash u : \rho}{\Gamma \otimes \Delta \vdash \mathbf{let}\ (x, y) = t\ \mathbf{in}\ u : \rho} \otimes -\mathrm{elim}$$

$$\frac{\Gamma \vdash t : \sigma}{\Gamma, x : Q_1 \vdash t : \sigma}\text{wk-unit} \qquad \frac{}{\bullet \vdash () : Q_1}\mathrm{unit}$$

Table 2: Typing classical terms (Source: [1], p. 29).

The definition of this language is the basis for understanding the process of adding measurements, the next section presents a brief introduction about the quantum measurements.

2.2 Quantum measurements

The quantum measurements are the third postulate of quantum computing, interpreted as:

In a quantum system the measurements disturb a system, collapsing from a quantum to classic state, implying loss of information from the initial state. Theoretically, the measurements determine how probable a state can collapse, for example, take the initial state $\alpha_0 |0\rangle + \alpha_1 |1\rangle$, when applying a measurement, it can collapse to $|0\rangle$ with probability $|\alpha_0|^2$ or $|1\rangle$ with probability $|\alpha_1|^2$ [2].

The measurement of state ψ consists of being exposed to an observer, with the following conditions [7–9]. Let quantum state $|\varphi\rangle = \sum_i \alpha |\psi_i\rangle$:

- The index m_i stands for the measurements outcomes that may occur in the experiment:
$$m_1, m_2, \ldots, m_i \tag{1}$$
- Each m_i has an associated matrix M_{m_i}, where M_{m_i} are called measurement operators:
$$M_{m_1}, M_{m_2}, \ldots, M_{m_i} \tag{2}$$

- The operators M_{m_i} must satisfy:

$$M^*_{m_1}M_{m_1} + M^*_{m_2}M_{m_2} + \cdots + M^*_{m_n}M_{m_n} = I$$

called *completeness* or *completeness* equation. This equation guarantees that the sum of the probabilities of the state is 1.
- Let $|\varphi\rangle$ be the current state of the system, if it is observed then it collapses to: $\frac{M_{m_i}|\varphi\rangle}{||M_{m_i}|\varphi\rangle||}$, with probability $P(m_i)$, where: $P(m_i) = \langle\varphi| M^*_{m_i}M_{m_i}|\varphi\rangle$.

Definition 1 (Operator). *An operator of $\mathbb{C}^2$ is a square matrix of dimension n with complex coefficients.*

Definition 2 (Projection operator). *A projection or measurement matrix, are operators of the form $P = |\psi\rangle\langle\psi|$.*

If $P = |\psi\rangle\langle\psi|$ is a projection operator and is applied to the $|\varphi\rangle$ state, then:

$$\begin{aligned} P|\varphi\rangle &= (|\psi\rangle\langle\psi|)|\varphi\rangle \\ &= |\psi\rangle\langle\psi|\varphi\rangle \\ &= \alpha|\psi\rangle \quad \text{where } \alpha \in \langle\psi|\varphi\rangle \end{aligned}$$

These measurement operators will be considered to incorporate them into the QML language.

Example 1. The base is $\{|0\rangle, |1\rangle\}$, with $|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$, $|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$, the operators are:
$P_0 = |0\rangle\langle 0| = \begin{pmatrix}1\\0\end{pmatrix}(1\ 0) = \begin{pmatrix}1&0\\0&0\end{pmatrix}$, $\quad P_1 = |1\rangle\langle 1| = \begin{pmatrix}0\\1\end{pmatrix}(0\ 1) = \begin{pmatrix}0&0\\0&1\end{pmatrix}$. Suppose $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, if we apply P_0, then:

$$P_0|\psi\rangle = \begin{pmatrix}1&0\\0&0\end{pmatrix}\left(\alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix}\right) = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\0\end{pmatrix} = \alpha|0\rangle$$

Also developed as: $P_0|\psi\rangle = |0\rangle\langle 0|\big(\alpha|0\rangle + \beta|1\rangle\big) = |0\rangle\big(\alpha\langle 0|0\rangle + \beta\langle 0|1\rangle\big) = \alpha|0\rangle$

Therefore, the application of a measurement operator is conceived as:$|\psi\rangle \stackrel{M_i}{\to} |\psi'\rangle$.

3 Measurements in Quantum Lambda Calculi

Quantum lambda calculus is a functional language that is not typed and without measurements, which allows operations and analysis of properties of this and other languages. After this, a way of adding quantum measurements is incorporated, allowing us to guide the implementation of the same technique to the QML language, this is possible since QML is a quantum programming language, with a functional paradigm, typed and without measurements.

Next we will describe important information and the procedure for adding measurement operators to the quantum lambda calculi language λ_q [5, 10].

λ_q calculus carries a history track to preserve the information needed to reverse reductions, ensuring that the computing process satisfies quantum properties. A state of this language, after the measurement is irreversibly changed, even retaining the history track.

Lambda calculus λ was developed with classical data [4], subsequently, quantum data is added resulting in λ_q [10]. Of λ_q its syntax is reconsidered, which extends with a family of measurement operators.

Of the most significant modifications, qubits are defined explicitly, the constants (of the original syntax) are divided into: qubits, measurement operators and gates and the measurement operators M_I are added. As a result of the above, the syntax is (Table 3):

t::=	pre-terms:		
x	variable	(λx.t)	abstraction
(t t)	application	!t	nonlinear term
(λ!x.t)	nonlinear abstraction	(c_U)	gate-constant
q	qubit-constant	M_I	measurement-constant
q::=	Qubit-constants:		
$(\vert 0\rangle,\ \vert 1\rangle)$	base-qubit	$(q \otimes q)$	tensorial product
$(q+q)$	superposition	$\alpha(q)$	scalar product
C_U::=	Gate-constants:		
$H \mid \mid X \mid cnot \mid \ldots$			

Table 3: Modified syntax of λ_q.

The rules for well-formed terms are found in Fig. 3 of the following article [5]. With the previous content, we proceed with the description of quantum measurements in λ_q. The corresponding rule is initially mentioned, and then its components and functionality will be explained.

The measurement rule is defined as:

$$\frac{q = \sum_{u=0}^{2^m-1} \alpha_u q^{(u)}}{\mathcal{H}; (M_I \quad q) \to_{p_w} \sum_{u \in C(w,m,I)} \frac{\alpha_u}{\sqrt{p_w}} q^{(u)}} \quad \forall i \in I, 1 \le i \le m$$

where:

- $w = 0, \ldots, 2^{|I|} - 1$ corresponds to the expected value in a measurement, that is, w has associated the possible values m_i that were mentioned in the equation 1. The w value must be converted to its binary number, for example, $w = 2$, corresponds to 010.

- I indicates the indices or positions that are observed in a measurement. For example, if $I = \{2, 3, 5\}$, the positions 2, 3 and 5 will be observed.
- $q^{(u)} = !q_1^{(u)} \otimes !q_2^{(u)} \otimes \cdots \otimes !q_m^{(u)}$, with $!q_k^{(u)} = ! \, |0\rangle$ or $! \, |1\rangle$, for $k = 1, \ldots, m$. Such states (in binary) conform to q with their corresponding probability, these represent all possible values between 0 and $2^m - 1$, for example, the $\alpha_0 \, |101\rangle$ state in terms of this rule is written as: $q^{(5)} = \big(! \, |1\rangle \otimes ! \, |0\rangle \otimes ! \, |1\rangle\big)$.
- $C(w, m, I)$,is the set of binary strings of length m, such that they coincide with w over the letters of index I. Assume $w = 010$ and the set $I = \{2, 3, 5\}$, in each qubit you will look for the first 0 (of w) in index 2, 1 in position 3 and 0 in index 5: $\boxed{q_1}\boxed{0}\boxed{1}\boxed{q_4}\boxed{0}$.
- $p_w = \displaystyle\sum_{u \in C(w,m,I)} |\alpha_u|^2$, is the probability of collapsing to the state w.
- The notation $t \rightarrow_{p_w} t'$, means that t goes to t' with probability p_w.

To conclude this section, a complete example the application of the rule is shown.

Example 2. Let $m = 3, \; I = \{1, 3\}$ and $w = 01$.

$$\begin{aligned} q \; &= \frac{1}{\sqrt{8}}\Big(! \, |0\rangle \otimes ! \, |0\rangle \otimes ! \, |0\rangle\Big) + \frac{1}{\sqrt{8}}\Big(! \, |0\rangle \otimes ! \, |0\rangle \otimes ! \, |1\rangle\Big) \\ &+ \frac{1}{\sqrt{8}}\Big(! \, |0\rangle \otimes ! \, |1\rangle \otimes ! \, |0\rangle\Big) + \frac{1}{\sqrt{8}}\Big(! \, |0\rangle \otimes ! \, |1\rangle \otimes ! \, |1\rangle\Big) \\ &+ \cdots + \frac{1}{\sqrt{8}}\Big(! \, |1\rangle \otimes ! \, |0\rangle \otimes ! \, |1\rangle\Big) + \frac{1}{\sqrt{8}}\Big(! \, |1\rangle \otimes ! \, |1\rangle \otimes ! \, |1\rangle\Big) \end{aligned}$$

Considering $C(w, m, I) = C(01, 3, \{1, 3\})$, of the states in q (0 is in index 1, and 1 in the index 3) you get the following:

$$\frac{1}{\sqrt{8}}\Big(!|\mathbf{0}\rangle \otimes ! \, |0\rangle \otimes !|\mathbf{1}\rangle\Big) + \frac{1}{\sqrt{8}}\Big(!|\mathbf{0}\rangle \otimes ! \, |1\rangle \otimes !|\mathbf{1}\rangle\Big)$$

The probability of collapsing with $w = 01$ in q is given by: $p_w = |\frac{1}{\sqrt{8}}|^2 + |\frac{1}{\sqrt{8}}|^2 = \frac{1}{4}$. With this, when measuring we get:

$$\begin{aligned} \mathcal{H}; (M_{\{1,3\}} \;\; q) &= \frac{\frac{1}{\sqrt{8}}}{\sqrt{\frac{1}{4}}}\Big(! \, |0\rangle \otimes ! \, |0\rangle \otimes ! \, |1\rangle\Big) + \frac{\frac{1}{\sqrt{8}}}{\sqrt{\frac{1}{4}}}\Big(! \, |0\rangle \otimes ! \, |1\rangle \otimes ! \, |1\rangle\Big) \\ &= \frac{1}{\sqrt{2}}\Big(! \, |0\rangle \otimes ! \, |0\rangle \otimes ! \, |1\rangle\Big) + \frac{1}{\sqrt{2}}\Big(! \, |0\rangle \otimes ! \, |1\rangle \otimes ! \, |1\rangle\Big) \end{aligned}$$

With probability $\frac{1}{4}$.

4 Measurements in QML

When implementing measurements according to λ_q, it is required to access the elements of each state, and considering that the types in QML are given by $[\![\sigma \otimes \tau]\!] = [\![\sigma]\!] \times [\![\tau]\!]$, the projections are used which will allow to access to the first or second element of the projection.

Definition 3 (Projections).

- $fst : A \times B \rightarrow A$, *where* $fst(a, b) \mapsto a$.
- $snd : A \times B \rightarrow B$, *where* $snd(a, b) \mapsto b$.

For every $x \in A \times B$ you have $x = (fst(x), snd(x))$. In a category with finite products, the object $\times(A_1, \ldots, A_n)$ is inductively defined as:

$$\begin{aligned}
&\times() \\
&\times(A) = 1 \times A \\
&\times(A_1 \ldots, A_n) = (\times(A_1 \ldots, A_{n-1})) \times A_n
\end{aligned}$$

With this, the projection $\pi_i^n : \times(A_1 \ldots, A_n) \rightarrow A_i$ is generalized, establishing: $\pi_n^n = snd$ and $\pi_i^n = \pi_i^{n-1} \circ fst$, when $i < n$ [6].

This definition applies in QML to the rules of well-formed terms:

$$\frac{\Gamma \otimes \Delta \vdash t : \sigma \otimes \tau}{\Gamma \vdash first\ t : \sigma}\,first \qquad\qquad \frac{\Gamma \otimes \Delta \vdash t : \sigma \otimes \tau}{\Delta \vdash snd\ t : \tau}\,snd$$

Table 4: Typing quantum data.

It is also required to define a grouping when you have a $\sigma \otimes \tau$ type, inductively applying the following (for associative purposes $\otimes$):

$$\begin{aligned}
&\sigma = Q_1 \otimes \sigma \\
&\sigma \otimes (\tau_1 \otimes \tau_2) = (\sigma \otimes \tau_1) \otimes \tau_2 \\
&(\sigma_1 \otimes \sigma_2) \otimes (\tau_1 \otimes \tau_2) = \big((\sigma_1 \otimes \sigma_2) \otimes \tau_1\big) \otimes \tau_2
\end{aligned}$$

For example, be type $\sigma = \Big(\big((Q_1 \otimes \sigma_1) \otimes \sigma_2\big) \otimes \sigma_3\Big) \otimes \sigma_4$. The projection π_1^4 is:

$$\begin{aligned}
\pi_1^4 &= (((snd \circ fst) \circ fst) \circ fst) \\
&= ((snd \circ fst) \circ fst) \circ fst\Big(\big((Q_1 \otimes \sigma_1) \otimes \sigma_2\big) \otimes \sigma_3\Big) \otimes \sigma_4 \\
&= (snd \circ fst) \circ fst\big((Q_1 \otimes \sigma_1) \otimes \sigma_2\big) \otimes \sigma_3 = snd \circ fst\big((Q_1 \otimes \sigma_1) \otimes \sigma_2\big) \\
&= snd\ (Q_1 \otimes \sigma_1) = \sigma_1
\end{aligned}$$

With the above rules, we proceed with the incorporation of quantum measurements.

4.1 Quantum Measurement Rule

The components that will integrate the rule are listed below.

- m, is the number of sub-terms that compose the qubit q_1. For example, in the following state q_1, $m = 2$:

$$\begin{aligned} q_1 =& \frac{1}{\sqrt{2}} * (\frac{1}{\sqrt{2}} * (false, false) + \frac{1}{\sqrt{2}} * (false, true)) \\ &+ \frac{1}{\sqrt{2}} * (\frac{1}{\sqrt{2}} * (true, false) + \frac{1}{\sqrt{2}} * (true, true)) \end{aligned} \tag{3}$$

- The quantum states in superposition should be expressed in conventional manner as: $\sum_{i=0}^{2^m-1} \alpha_i * q_i$, however, the rule (by definition) to form superpositions in QML is $q = \lambda_1 * t_1 + \lambda_2 * t_2$. Where every t' in q can be formed with $true$, $false$ or tuples $(true, false)$, $(true, (false, false)), \ldots$ and superpositions.
  What it means is that the sub-terms t' can be defined in terms of others and successively; once these t' have the desired type, then they will have the form $\sum_{i=0}^{2^m-1} \alpha_i * q_i$ to perform a quantum measurement, for which this should be apply the **red°** rule in the Table 5.
- The definition of I is still conceived as the set of indexes or positions to consider when measuring q. Defined as:

$$I = \{\ i \mid i \in \mathbb{N},\ 1 \leq i \leq m\}$$

  In practical terms this means that if $I = \{2\}$ in a q state, the indexes to be observed are identified as: $\frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} * (false, \underbrace{false}_{I=\{2\}}) + \frac{1}{\sqrt{2}} * (false, \underbrace{true}_{I=\{2\}}) + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} * (true, \underbrace{false}_{I=\{2\}}) + \frac{1}{\sqrt{2}} * (true, \underbrace{true}_{I=\{2\}})$. The elements of I will determine the projections π_i^m (Definition 3) that will be accessed in each state of a qubit.
- With respect to w, it is necessary that $w = 0, \ldots, 2^{|I|} - 1$, being the expected value when measuring. The chosen value w, must be coded in binary ($1 = true$, $0 = false$), as the following example: $w = 5$, would be $101 \equiv true, false, true$.
  For measurement purposes, the w string is broken down as follows:

$$w = w_1, w_2, \cdots, w_n, \text{ where } n \leq m$$

  The subscripts of w must match I. For example: If $I = \{1, 3\}$, $w = true, false$; so, $w = w_1, w_3$ where $w_1 = true$ and $w_3 = false$.
- $C(w, m, I)$, it is the function that returns the set of strings of length m, such that they coincide with the expected values w in the indexes I. This function can be defined as:

$$\forall i \in I,\ \forall \alpha_{t'} * t' \text{ in } q, \quad \text{if} \quad \alpha_{t'} * \pi_i^m(t') = \alpha_{t'} * w_i \ \Rightarrow\ \alpha_{t'} * t' \in C(w, m, I)$$

  where t' is an irreducible term. For example, taking the equation 3, with $I = \{1, 3\}$, $w = \underbrace{false}_{w_1}, \underbrace{true}_{w_3}$, $m = 3$, if it is verified that: $\forall i \in \{1, 3\}$, $\forall \alpha_{t'} * t'$ in q, satisfy $\alpha_{t'} * \pi_i^3(t') = \alpha_{t'} * w_i$, then t' belongs to that function:

- $\frac{1}{\sqrt{8}} * \pi_1^3(false, (false, true)) = \frac{1}{\sqrt{8}} * false = \frac{1}{\sqrt{8}} * w_1$, and $\frac{1}{\sqrt{8}} * \pi_3^3(false, (false, true)) = \frac{1}{\sqrt{8}} * true = \frac{1}{\sqrt{8}} * w_3$
- $\frac{1}{\sqrt{8}} * \pi_1^3(false, (true, true)) = \frac{1}{\sqrt{8}} * false = \frac{1}{\sqrt{8}} * w_1$, and $\frac{1}{\sqrt{8}} * \pi_3^3(false, (true, true)) = \frac{1}{\sqrt{8}} * true = \frac{1}{\sqrt{8}} * w_3$

$\therefore \quad C(w, m, I) = \{\frac{1}{\sqrt{8}} * (false, (false, true)) + \frac{1}{\sqrt{8}} * (false, (true, true))\}$

With the above, the rules for reducing terms described in superpositions and the measurement rule in QML are deduced.

$$\frac{q = \alpha_1 * t_1 + \alpha_2 * t_2 \qquad t_1 = \alpha_{11} * t_{11} + \alpha_{12} * t_{12} \qquad t_2 = \alpha_{21} * t_{21} + \alpha_{22} * t_{22}}{q' = \alpha_1 * \alpha_{11} * t_{11} \ \underset{(M_I \ \ q) \longrightarrow (M_I \ \ q')}{+} \ \alpha_1 * \alpha_{12} * t_{12} \ + \ \alpha_2 * \alpha_{21} * t_{21} \ + \ \alpha_2 * \alpha_{22} * t_{22}} \ \mathbf{red}^\circ$$

$$\frac{q = \lambda_1 * t_1 + \lambda_2 * t_2}{(M_I \ \ q) \to_{p_w} \sum_{\alpha_{t'} * t' \in C(w,m,I)} \frac{1}{\sqrt{p_w}} * t'} \ \mathbf{M}^\circ \ \forall i \in I, 1 \leq i \leq m$$

Table 5: Quantum measure rule for quantum data.

4.2 Example

Let $I = \{2\}$, $m = 2$, $w = true$ and the state q:

$$q = \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} * (false, false) + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} * (false, true) + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} * (true, false) + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} * (true, true)$$

where $C(w, m, I) = \{(false, true),\ (true, true)\}$, $p_{true} = \sum_{u \in C(w,m,I)} |\lambda_u|^2 = \frac{1}{2}$.

Performing the measurement with respect to $w = false$:

$$\begin{aligned} M_I \ \ q \ &\underset{1/2}{\to} \sum_{u \in C(w,m,I)} \frac{\alpha_u}{\sqrt{1/2}} * t_u \\ &\underset{1/2}{\to} \frac{1/2}{\sqrt{1/2}} * (false, true) + \frac{1/2}{\sqrt{1/2}} * (true, true) \\ &\underset{1/2}{\to} \frac{1}{\sqrt{2}} * (false, true) + \frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} * (true, true) \end{aligned}$$

5 Conclusions

The addition of a measurement operator in QML, is that given a quantum state (consisting of sub-terms), it is initially determined with respect to which position of each sub-state will be measured and the expected value, collapsing to the state or states that coincide with the above, determining with what probability and normalizing to the final state.

References

1. Thorsten Altenkirch, Jonathan Grattage, Juliana K. Vizzotto, and Amr Sabry. An algebra of pure quantum programming. *Electronic Notes in Theoretical Computer Science*, 170:23 – 47, 2007. Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005).
2. Guido Bacciagaluppi. Is logic empirical? In Kurt Engesser, Dov M. Gabbay, and Daniel Lehmann, editors, *Handbook of Quantum Logic and Quantum Structures*, pages 49 – 78. Elsevier, Amsterdam, 2009.
3. Paul Bernays. Alonzo church. an unsolvable problem of elementary number theory. american journal of mathematics, vol. 58 (1936), pp. 345,Äì363. *Journal of Symbolic Logic*, 1(2):73,Äì74, 1936.
4. A. Church. *The Calculi of Lambda-conversion*. Annals of Mathematics Studies. Princeton University Press, 1941.
5. Alejandro Díaz-Caro, Pablo Arrighi, Manuel Gadella, and Jonathan Grattage. Measurements and confluence in quantum lambda calculi with explicit qubits. *Electronic Notes in Theoretical Computer Science*, 270(1):59 – 74, 2011. Proceedings of the Joint 5th International Workshop on Quantum Physics and Logic and 4th Workshop on Developments in Computational Models (QPL/DCM 2008).
6. C.A. Gunter. *Semantics of Programming Languages: Structures and Techniques*. Foundations of Computing. MIT Press, 1992.
7. S. Imre and F. Balazs. *Quantum Computing and Communications: An Engineering Approach*. Wiley, 2005.
8. N. David Mermin. From cbits to qbits: Teaching computer scientists quantum mechanics. *American Journal of Physics*, 71(1):23–30, 2003.
9. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
10. André van Tonder. A lambda calculus for quantum computation. *SIAM J. Comput.*, 33(5):1109–1135, 2004.