Agregando Medición al Cálculo de van Tonder

Tesina de Grado de Licenciatura en Ciencias de la Computación

Alejandro Díaz-Caro

(Janus)

Director

Dr. Manuel Gadella (Manolo)

Co-Director

Dr. Pablo Martínez López (Fidel)

Departamento de Ciencias de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario

21 de Diciembre de 2007

Motivación

- λ-Cálculo (*clásico*):
  - Modelo equivalente a las Máquinas de Turing
  - De gran utilidad en la teoría de la computación y el estudio de lenguajes y sus semánticas

Motivación

- λ-Cálculo (*clásico*):
  - Modelo equivalente a las Máquinas de Turing
  - De gran utilidad en la teoría de la computación y el estudio de lenguajes y sus semánticas
- André van Tonder desarrolló un λ-Cálculo Cuántico[vT04] equivalente a la Máquina de Turing Cuántica

Motivación

- λ-Cálculo (*clásico*):
  - Modelo equivalente a las Máquinas de Turing
  - De gran utilidad en la teoría de la computación y el estudio de lenguajes y sus semánticas
- André van Tonder desarrolló un λ-Cálculo Cuántico[vT04] equivalente a la Máquina de Turing Cuántica

Posibles problemas

- El Cálculo de van Tonder no tiene medición

Motivación

- λ-Cálculo (*clásico*):
  - Modelo equivalente a las Máquinas de Turing
  - De gran utilidad en la teoría de la computación y el estudio de lenguajes y sus semánticas
- André van Tonder desarrolló un λ-Cálculo Cuántico[vT04] equivalente a la Máquina de Turing Cuántica

Posibles problemas

- El Cálculo de van Tonder no tiene medición
- Él afirma que cualquier algoritmo se puede rediseñar para diferir la medición

Motivación

- λ-Cálculo (*clásico*):
  - Modelo equivalente a las Máquinas de Turing
  - De gran utilidad en la teoría de la computación y el estudio de lenguajes y sus semánticas
- André van Tonder desarrolló un λ-Cálculo Cuántico[vT04] equivalente a la Máquina de Turing Cuántica

Posibles problemas

- El Cálculo de van Tonder no tiene medición
- Él afirma que cualquier algoritmo se puede rediseñar para diferir la medición
- Pero esto puede generar dificultad para entender los algoritmos

Contenido de la presentación

1. Introducción
   - Motivación
   - Contenido de la presentación
   - Computación Cuántica
2. λ_q: El Cálculo de van Tonder
   - Reversibilidad
   - El lenguaje
   - Ejemplo: Teleportación con medición diferida
3. Agregando Medición a λ_q
   - Cálculo Probabilístico
   - El lenguaje
   - Ejemplo: Teleportación
4. Conclusiones y trabajo futuro

Computación Cuántica: modelo de computación basado en la Mecánica Cuántica

Computación Cuántica: modelo de computación basado en la Mecánica Cuántica

Unidad básica: "Qubit" (o quantum bit)

Definición

qubit: vector de dos dimensiones:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

con $\alpha, \beta \in \mathbb{C}$ y $|\alpha|^2 + |\beta|^2 = 1$. (Forma un Espacio Vectorial)

Una base del espacio vectorial de un qubit es

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Una base del espacio vectorial de un qubit es

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Notación:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Una base del espacio vectorial de un qubit es

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Notación:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Entonces, un qubit queda definido por

$$\alpha\,|0\rangle + \beta\,|1\rangle$$

Una base del espacio vectorial de un qubit es

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Notación:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Entonces, un qubit queda definido por

$$\alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Una base del espacio vectorial de un qubit es

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

Notación:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Entonces, un qubit queda definido por

$$\alpha\,|0\rangle + \beta\,|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Sistema de n-qubits: espacio vectorial de dimensión 2^n
Una base de dicho espacio:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \right\}$$

Sistema de n-qubits: espacio vectorial de dimensión 2^n
Una base de dicho espacio:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \right\}$$

Notación:
$|i_1, \ldots, i_n\rangle$ con $i_k \in \{0, 1\}$

Sistema de n-qubits: espacio vectorial de dimensión 2^n

Una base de dicho espacio:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \right\}$$

Notación:

$|i_1, \ldots, i_n\rangle \text{ con } i_k \in \{0, 1\}$

ó

$|i_1\rangle \otimes \cdots \otimes |i_n\rangle \text{ con } i_k \in \{0, 1\}$

Sistema de n-qubits: espacio vectorial de dimensión 2^n
Una base de dicho espacio:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \right\}$$

Notación:

$|i_1, \ldots, i_n\rangle \text{ con } i_k \in \{0, 1\}$

ó

$|i_1\rangle \otimes \cdots \otimes |i_n\rangle \text{ con } i_k \in \{0, 1\}$

ó

$|i\rangle \text{ con } i \in \{0, \ldots, 2^{n-1}\}$

Sistema de n-qubits: espacio vectorial de dimensión 2^n
Una base de dicho espacio:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \dots, \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \right\}$$

Notación:
$|i_1, \dots, i_n\rangle \text{ con } i_k \in \{0, 1\}$
ó
$|i_1\rangle \otimes \cdots \otimes |i_n\rangle \text{ con } i_k \in \{0, 1\}$
ó
$|i\rangle \text{ con } i \in \{0, \dots, 2^{n-1}\}$

Entonces un n-qubit queda definido por

$$\sum_{k=0}^{2^{n-1}} \alpha_k |k\rangle \text{ tal que } \sum_{k=0}^{2^{n-1}} |\alpha_k|^2 = 1$$

Compuertas cuánticas: matrices

(satisfacen propiedades de manera que multiplicadas por qubits dan qubits)

Ejemplo

Compuerta NOT

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$X|0\rangle = X\binom{1}{0} = \binom{0}{1} = |1\rangle$$
$$X|1\rangle = X\binom{0}{1} = \binom{1}{0} = |0\rangle$$

Matriz x qubit: aplicación lineal

$$U\sum_{i=0}^{2^{n-1}}\alpha_i\left|i\right\rangle=\sum_{i=0}^{2^{n-1}}\alpha_i U\left|i\right\rangle$$

Matriz x qubit: aplicación lineal

$$U \sum_{i=0}^{2^{n-1}} \alpha_i \left|i\right\rangle = \sum_{i=0}^{2^{n-1}} \alpha_i U \left|i\right\rangle$$

Especificando cómo actúa la compuerta con la base del espacio de qubits, ya se ha especificado todo lo necesario

Matriz x qubit: aplicación lineal

$$U \sum_{i=0}^{2^{n-1}} \alpha_i |i\rangle = \sum_{i=0}^{2^{n-1}} \alpha_i U |i\rangle$$

Especificando cómo actúa la compuerta con la base del espacio de qubits, ya se ha especificado todo lo necesario

Ejemplo

$$Z|0\rangle = |0\rangle$$
$$Z|1\rangle = -|1\rangle$$

Entonces
$Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle$

Matriz x qubit: aplicación lineal

$$U \sum_{i=0}^{2^{n-1}} \alpha_i \left|i\right\rangle = \sum_{i=0}^{2^{n-1}} \alpha_i U \left|i\right\rangle$$

Especificando cómo actúa la compuerta con la base del espacio de qubits, ya se ha especificado todo lo necesario

Ejemplo

$$Z\left|0\right\rangle = \left|0\right\rangle$$
$$Z\left|1\right\rangle = -\left|1\right\rangle$$

Entonces
$Z(\alpha\left|0\right\rangle + \beta\left|1\right\rangle) = \alpha\left|0\right\rangle - \beta\left|1\right\rangle$

Obs Todas las compuertas cuánticas son reversibles y coinciden con su inversa (aplicando dos veces una compuerta, se vuelve al estado original)

Medir un n-qubit $\sum_{i=0}^{2^{n-1}} \alpha_i \left|i\right\rangle$ *respecto a la base* $\{\left|i\right\rangle\}_{i=0}^{2^{n-1}}$ del espacio de n-qubits, devuelve un vector $\left|k\right\rangle$ de dicha base con probabilidad $|\alpha_k|^2$

Medir un n-qubit $\sum_{i=0}^{2^n-1} \alpha_i \left|i\right\rangle$ *respecto a la base* $\{\left|i\right\rangle\}_{i=0}^{2^n-1}$ del espacio de n-qubits, devuelve un vector $\left|k\right\rangle$ de dicha base con probabilidad $|\alpha_k|^2$

Ejemplo

Qubit a medir: $\frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle)$

Resultado: $\left|0\right\rangle$ ó $\left|1\right\rangle$ (con probabilidad $\frac{1}{2}$ cada uno)

Medir un n-qubit $\sum_{i=0}^{2^{n-1}} \alpha_i \left| i \right\rangle$ *respecto a la base* $\{\left| i \right\rangle\}_{i=0}^{2^{n-1}}$ del espacio de n-qubits, devuelve un vector $\left| k \right\rangle$ de dicha base con probabilidad $|\alpha_k|^2$

Ejemplo

Qubit a medir: $\frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle)$

Resultado: $\left|0\right\rangle$ ó $\left|1\right\rangle$ (con probabilidad $\frac{1}{2}$ cada uno)

- Algoritmo cuántico: Hacer evolucionar un sistema de n-qubits mediante la aplicación de compuertas y mediciones

Medir un n-qubit $\sum_{i=0}^{2^n-1} \alpha_i \left|i\right\rangle$ *respecto a la base* $\{\left|i\right\rangle\}_{i=0}^{2^n-1}$ del espacio de n-qubits, devuelve un vector $\left|k\right\rangle$ de dicha base con probabilidad $|\alpha_k|^2$

Ejemplo

Qubit a medir: $\frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle)$

Resultado: $\left|0\right\rangle$ ó $\left|1\right\rangle$ (con probabilidad $\frac{1}{2}$ cada uno)

- Algoritmo cuántico: Hacer evolucionar un sistema de n-qubits mediante la aplicación de compuertas y mediciones
- Debido a la reversibilidad de las compuertas, los algoritmos son reversibles (excepto en la medición)

Circuito cuántico: Representación gráfica de un algoritmo cuántico

Circuito cuántico: Representación gráfica de un algoritmo cuántico

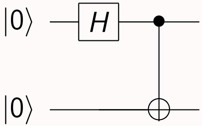
Ejemplo

Dado $|00\rangle$, aplicar la compuerta Hadamard (H) al primer qubit y luego una compuerta $CNOT$ entre el primero y el segundo

Circuito cuántico: Representación gráfica de un algoritmo cuántico

Ejemplo

Dado $|00\rangle$, aplicar la compuerta Hadamard (H) al primer qubit y luego una compuerta $CNOT$ entre el primero y el segundo



Un estado enredado es un estado en el que no puedo “factorizar” los qubits

Un estado enredado es un estado en el que no puedo "factorizar" los qubits

Ejemplo

No enredado

$$\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$$

$$= \underbrace{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)}_{\text{1er qubit}} \underbrace{|0\rangle}_{\text{2do qubit}}$$

Ejemplo

Enredado

$$\beta_{00} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

No los puedo separar!

El algoritmo de Teleportación[BBCJPW93] nos permitirá mostrar cómo se pueden escribir algoritmos cuánticos en los diversos formalismos

Ejemplo

El objetivo de esta técnica es transmitir un qubit de Alice a Bob mediante el envío de dos bits clásicos

Pasos a seguir por Alice y Bob:

Paso 1. Alice y Bob preparan un estado β_{00}

Pasos a seguir por Alice y Bob:

Paso 1. Alice y Bob preparan un estado β_{00}

Paso 2. Alice se queda con el primer qubit del par y Bob se lleva el segundo

Pasos a seguir por Alice y Bob:

Paso 1. Alice y Bob preparan un estado β_{00}

Paso 2. Alice se queda con el primer qubit del par y Bob se lleva el segundo

Paso 3. Alice aplica CNOT entre el qubit a transmitir y el primero del par β_{00}

Pasos a seguir por Alice y Bob:

Paso 1. Alice y Bob preparan un estado β_{00}

Paso 2. Alice se queda con el primer qubit del par y Bob se lleva el segundo

Paso 3. Alice aplica CNOT entre el qubit a transmitir y el primero del par β_{00}

Paso 4. Alice aplica Hadamard al primero de sus dos qubits, luego realiza una medición sobre ambos y envía el resultado de la medición (2 bits clásicos) a Bob

Pasos a seguir por Alice y Bob:

Paso 1. Alice y Bob preparan un estado β_{00}

Paso 2. Alice se queda con el primer qubit del par y Bob se lleva el segundo

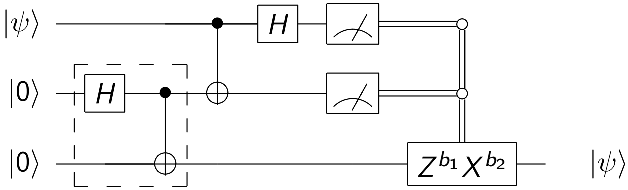
Paso 3. Alice aplica CNOT entre el qubit a transmitir y el primero del par β_{00}

Paso 4. Alice aplica Hadamard al primero de sus dos qubits, luego realiza una medición sobre ambos y envía el resultado de la medición (2 bits clásicos) a Bob

Paso 5. Bob aplica una transformación sobre su qubit, de acuerdo a los bits recibidos, basándose en la siguiente tabla

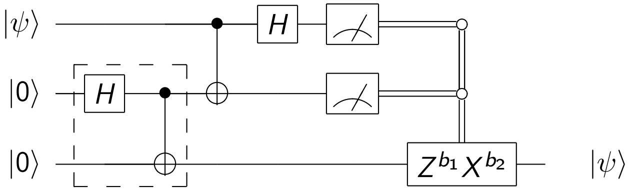
Bits recibidos	00	01	10	11
Compuerta a aplicar	I	X	Z	ZX

El circuito completo queda de la siguiente manera:



donde $|\psi\rangle$ es el qubit a teleportar

El circuito completo queda de la siguiente manera:



donde $|\psi\rangle$ es el qubit a teleportar

Más adelante mostraremos cómo se puede escribir este algoritmo en los diversos formalismos que se harán en esta presentación

- En λ-Cálculo *clásico*, las β-reducciones descartan información en cada paso haciendo el proceso irreversible

- En λ-Cálculo *clásico*, las β-reducciones descartan información en cada paso haciendo el proceso irreversible
- Cualquier cómputo clásico se puede transformar en un cómputo reversible[Ben73]

Cómo tener reversibilidad (una forma naïve):

- Sea x un término y $\beta : x \to \beta(x)$ una β-reducción
- Consideremos la función $x \to (x, \beta(x))$, la cual es invertible

El cómputo podría proceder de la siguiente manera:

$$x \to (x, \beta(x)) \to (x, \beta(x), \beta^2(x)) \to (x, \beta(x), \beta^2(x), \beta^3(x)) \to \dots$$

En cada paso guardaremos qué subtérmino se ha reducido y qué operación se ha aplicado en un historial

En cada paso guardaremos qué subtérmino se ha reducido y qué operación se ha aplicado en un historial

Definición

El estado computacional se toma como una superposición cuántica de secuencias de la forma

$$h_1; \dots ; h_n; t$$

donde a $h_1; \dots ; h_n$ se le llama Historial y a t Registro computacional

";" denota la concatenación de strings

Notación: Al historial en general lo notamos con $\mathcal{H}$

Sintaxis del λ_q

$t ::=$		término
	x	variable
	$(\lambda x.t)$	abstracción
	$(t\ t)$	aplicación
	c	constante
	$!t$	término no lineal
	$(\lambda !x.t)$	abstracción no lineal
$c ::=$		
	$0 \| 1 \| H \| cnot \| X \| \ldots$	constantes

0 y 1 son primitivas. El resto de las constantes denotan compuertas elementales entre qubits

- Esta sintaxis, que distingue términos lineales y no lineales, es una extensión una la sintaxis introducida por Philip Wadler[Wad94]

- Esta sintaxis, que distingue términos lineales y no lineales, es una extensión una la sintaxis introducida por Philip Wadler[Wad94]
- Para entender el por qué de incluirla aquí, veamos cómo se la utiliza en este contexto

Definición

Decimos que una subexpresión es definida con respecto a la base computacional si es textualmente la misma en todos los branches de una superposición

Definición

Decimos que una subexpresión es definida con respecto a la base computacional si es textualmente la misma en todos los branches de una superposición

Ejemplo

$$\frac{1}{\sqrt{2}}(|(\lambda x.0)\ 0\rangle + |(\lambda x.0)\ 1\rangle)$$

- La subexpresión $(\lambda x.0)$ es definida
- El argumento $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ no lo es

"Reglas de uso"

"Reglas de uso"

- Los términos de la forma $!t$ son llamados no lineales
- Los términos no lineales serán términos definidos con respecto a la base computacional

"Reglas de uso"

- Los términos de la forma $!t$ son llamados no lineales
- Los términos no lineales serán términos definidos con respecto a la base computacional
- $(\lambda !x.t)$: denota funciones con argumentos no lineales
- $(\lambda x.t)$: el argumento es lineal

"Reglas de uso"

- Los términos de la forma $!t$ son llamados no lineales
- Los términos no lineales serán términos definidos con respecto a la base computacional
- $(\lambda !x.t)$: denota funciones con argumentos no lineales
- $(\lambda x.t)$: el argumento es lineal
- Una abstracción lineal contiene todos los términos no lineales que se quiera (o ninguno), pero debe haber un término lineal (y sólo uno) en el cuerpo de la función

"Reglas de uso"

- Los términos de la forma $!t$ son llamados no lineales
- Los términos no lineales serán términos definidos con respecto a la base computacional
- $(\lambda!x.t)$: denota funciones con argumentos no lineales
- $(\lambda x.t)$: el argumento es lineal
- Una abstracción lineal contiene todos los términos no lineales que se quiera (o ninguno), pero debe haber un término lineal (y sólo uno) en el cuerpo de la función

Estas "reglas" evitan que el historial quede en *enredo cuántico* con el registro computacional

Para evitar que se evalúen términos de la forma $!t$, seguimos a Abramsky[Abr93] y extendemos nuestra definición de valores de la siguiente manera:

Para evitar que se evalúen términos de la forma $!t$, seguimos a Abramsky[Abr93] y extendemos nuestra definición de valores de la siguiente manera:

Valores en el λ_q

$v ::=$		valores:
	x	variable
	c	constante
	$(\lambda x.t)$	abstracción lineal
	$(\lambda !x.t)$	abstracción no lineal
	$!t$	!-suspensión

La semántica se da de manera operacional mediante reglas

- reglas de reducción tradicionales (beta, etc.) con algunos cambios para manejar el historial

Ejemplo

$$\frac{}{\mathcal{H}; ((\lambda x.t)\ v) \to \mathcal{H}; ((\lambda x.\overline{t}_x)\ \_); t[v/x]}\ (\beta)$$

donde $\overline{t}_x$ se obtiene de t reemplazando recursivamente todos los subtérminos que no contienen x con el símbolo _ y manteniendo x

La semántica se da de manera operacional mediante reglas

- reglas de reducción tradicionales (beta, etc.) con algunos cambios para manejar el historial

Ejemplo

$$\frac{}{\mathcal{H}; ((\lambda x.t)\ v) \rightarrow \mathcal{H}; ((\lambda x.\bar{t}_x)\ \_); t[v/x]}\ (\beta)$$

donde $\bar{t}_x$ se obtiene de t reemplazando recursivamente todos los subtérminos que no contienen x con el símbolo _ y manteniendo x

- reglas específicas para las compuertas

Ejemplo

$H\ 0$ evaluará a $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

Teorema

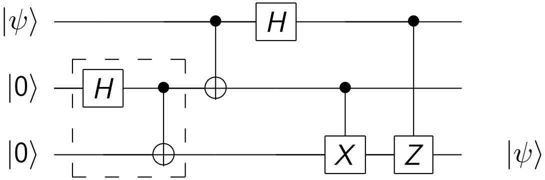
En el cálculo cuántico λ_q, la evolución del registro computacional está gobernada por las reglas de reducción, eliminando toda mención al historial □

Esto provee un conjunto de reglas que nos permite pensar los algoritmos *sin tener que acarrear el historial*

- El λ_q no puede representar la operación medición
- Por lo tanto se modifica el algoritmo para diferir la medición

- El λ_q no puede representar la operación medición
- Por lo tanto se modifica el algoritmo para diferir la medición

El circuito queda de la siguiente manera:



Este circuito es menos claro, pues no es evidente que Alice y Bob quedan separados físicamente durante el proceso

En λ_q este algoritmo queda definido por:

Teleportación con medición diferida

$$\begin{aligned}\text{teleport } x \longrightarrow\ & \textbf{let } (e_1, e_2) = \text{epr } \textbf{in}\\ & \textbf{let } (x', y') = \text{alice } (x, e_1) \textbf{ in}\\ & \quad \text{bob } (x', y', e_2)\end{aligned}$$

donde

$$\text{alice } (x, e_1) \longrightarrow\ \textbf{let } (x', y') = cnot\ (x, e_1) \textbf{ in } ((H\ x'), y')$$

$$\begin{aligned}\text{bob } (x', y', e_2) \longrightarrow\ & \textbf{let } (y'', e_2') = cX\ (y', e_2) \textbf{ in}\\ & \textbf{let } (x'', e_2'') = cZ\ (x', e_2') \textbf{ in}\\ & \quad (x'', y'', e_2'')\end{aligned}$$

$$\text{epr} \equiv cnot\ ((H\ 0), 0)$$

- La medición cuántica es intrínsecamente probabilística

- La medición cuántica es intrínsecamente probabilística
- Las reglas de transición deberán ser probabilísticas

- La medición cuántica es intrínsecamente probabilística
- Las reglas de transición deberán ser probabilísticas
- Siguiendo a Di Pierro y otros[DHW05], usaremos un modelo simple en el cual cada regla de transición puede tener varias conclusiones, cada una con una probabilidad asociada

- La medición cuántica es intrínsecamente probabilística
- Las reglas de transición deberán ser probabilísticas
- Siguiendo a Di Pierro y otros[DHW05], usaremos un modelo simple en el cual cada regla de transición puede tener varias conclusiones, cada una con una probabilidad asociada

Ejemplo

$$\frac{\textit{Premisas sobre } P}{\begin{array}{c} P \rightarrow_p Q_1 \\ P \rightarrow_q Q_2 \end{array}}$$

P tiene probabilidad p de transicionar a Q_1 y probabilidad q de transicionar a Q_2

La operación medición tiene en cuenta la "forma" del qubit, por lo tanto necesitamos dar la sintaxis de las constantes y reglas que determinen cuándo éstas están bien formadas

La operación medición tiene en cuenta la "forma" del qubit, por lo tanto necesitamos dar la sintaxis de las constantes y reglas que determinen cuándo éstas están bien formadas

Sintaxis del λ_q^M

$t ::=$		*términos:*
	x	*variable*
	$(\lambda x.t)$	*abstracción*
	$(t\ t)$	*aplicación*
	$!t$	*término no lineal*
	$(\lambda !x.t)$	*abstracción no lineal*
	c_U	*constante para compuertas*
	q	*qubit*
	M_n	*constante para medición*

Sintaxis del λ_q^M (cont.)

$q ::=$		*qubits:*
	$\|0\rangle \mid \|1\rangle$	*qubits definidos*
	$(q \otimes q)$	*producto tensorial*
	$(q + q)$	*superposición*
	$\alpha(q)$	*producto escalar*
$c_U ::=$		*constantes para compuertas:*
	$cnot \mid H \mid q(q)^T \mid etc\ \ldots$	

Además de esta sintaxis, se han definido reglas que determinan cuándo un término está bien formado

Además de esta sintaxis, se han definido reglas que determinan cuándo un término está bien formado

Ejemplo

$$\frac{\Gamma \vdash q_1 \qquad \Delta \vdash q_2}{\Gamma, \Delta \vdash q_1 \otimes q_2}$$

$$\frac{\sum_{i=1}^{2^n} |\alpha_i|^2 = 1 \qquad \alpha_i \in \mathbb{C}, i = 1, \ldots, 2^n}{\vdash \alpha_0(!\,|0\rangle \otimes \cdots \otimes !\,|0\rangle) + \cdots + \alpha_{2^n}(!\,|1\rangle \otimes \cdots \otimes !\,|1\rangle)}$$

La definición de valores se extiende trivialmente:

Definición de valores del λ_q^M

$v ::=$		valores
	x	variable
	$(\lambda x.t)$	abstracción
	$(\lambda !x.t)$	abstracción no lineal
	$!t$	!-suspensión
	c_U	compuerta
	M_n	medición
	q	qubit

- El modelo operacional sufrió algunos cambios para adaptarse a la nueva sintaxis de las constantes

- El modelo operacional sufrió algunos cambios para adaptarse a la nueva sintaxis de las constantes
- Todas las transiciones que no involucren la medición tendrán probabilidad 1 de ocurrir

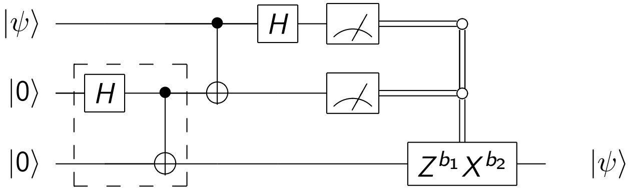
- El modelo operacional sufrió algunos cambios para adaptarse a la nueva sintaxis de las constantes
- Todas las transiciones que no involucren la medición tendrán probabilidad 1 de ocurrir
- Además, se agrega la regla que determina cómo se comporta la medición:

Regla de transición para la medición en λ_q^M

$$\text{M: } \frac{q = \alpha_0(!q_{00} \otimes \cdots \otimes !q_{10}) + \cdots + \alpha_{2^n}(!q_{02^n} \otimes \cdots \otimes !q_{n2^n})}{\mathcal{H}; (M_{2^n}\ q) \rightarrow_{|\alpha_i|^2} (!q_{0i} \otimes \cdots \otimes !q_{1i})}$$

Esta regla descarta el historial, ya que la medición es irreversible

Recordemos el algoritmo de teleportación



Teleportación en λ_q^M

$$\begin{aligned} \text{teleport } q \rightarrow_1 \ & \textbf{let } x \otimes y = \text{epr } \textbf{in} \\ & \textbf{let } b_1 \otimes b_2 = \text{alice } (q, x) \textbf{ in} \\ & \quad \text{bob } (b_1, b_2, y) \end{aligned}$$

donde

$\text{epr} \equiv cnot \ \ ((H \ ! \, |0\rangle \otimes |0\rangle)$

$\text{alice } (q, x) \rightarrow_1 \ \textbf{let } r \otimes w = cnot \ \ q \otimes x \textbf{ in}$

$\qquad M_2 \ ((H \ \ r) \otimes w)$

$\text{bob } (b_1, b_2, y) \rightarrow_1 (\text{zed } \ b_1) \ (\text{ex } \ b_2) \ y$

$\text{ex } b_2 \rightarrow_1 ! \, |0\rangle \ {b_2}^T + ! \, |1\rangle \ (X \ \ b_2)^T$

$\text{zed } b_1 \ \rightarrow_1$

$Z \ (! \, |0\rangle \ (! \, |0\rangle)^T + b_1 (! \, |1\rangle)^T) - ! \, |0\rangle \ (! \, |1\rangle)^T + (X \ \ b_1)(! \, |1\rangle)^T$

$(\text{ex } \ b_2)$ es simplemente X^{b_2} y $(\text{zed } \ b_1)$ es Z^{b_1}

Conclusiones y trabajo futuro I

Algunas conclusiones

- El λ_q^M permite razonar los algoritmos cuánticos de una manera más familiar que los circuitos cuánticos

Conclusiones y trabajo futuro I

Algunas conclusiones

- El λ_q^M permite razonar los algoritmos cuánticos de una manera más familiar que los circuitos cuánticos
- Modelo de van Tonder: equivalente al de la máquina de Turing cuántica

  Agregar Medición:
  - no amplia el modelo computacional,
  - pero provee una manera más intuitiva de razonar con el lenguaje

Conclusiones y trabajo futuro II

Trabajo futuro

- Trabajos recientes que se han hecho a partir del λ_q
  - Desarrollo de una teoría de tipos y su semántica denotacional
  - Semántica categórica
  - etc.

Como trabajo futuro queda extender dichos resultados al λ_q^M

Conclusiones y trabajo futuro II

Trabajo futuro

- Trabajos recientes que se han hecho a partir del λ_q
  - Desarrollo de una teoría de tipos y su semántica denotacional
  - Semántica categórica
  - etc.

  Como trabajo futuro queda extender dichos resultados al λ_q^M

- El paper original de van Tonder incluye una teoría ecuacional
  - La medición no podría ser parte de una teoría ecuacional,
  - pero queda por estudiar las implicaciones de esta afirmación

Bibliografía I

André van Tonder.
A lambda calculus for quantum computation.
SIAM Journal on Computing, 33(5):1109–1135, 2004.

Charles H. Bennett, Gilles Brassard, Claude Crepeau, Richard Jozsa, Asher Peres, and William K. Wootters.
Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels.
Phys. Rev. Lett. 70(1895), 1993.

Charles H. Bennett.
Logical reversibility of computation.
IBM Journal of Research and Development, 17(525532), 1973

Bibliografía II

Philip Wadler.
A syntax for linear logic.
In *Proceedings of the 9th International Conference on Mathematical Foundations of Programming Semantics*, pages 513–529, London, UK, 1994. Springer-Verlag.

Samson Abramsky.
Computational interpretations of linear logic.
Theoretical Computer Science, 111(1–2):3–57, 1993.

Alessandra Di Pierro, Chris Hankin, and Herbert Wiklicky
Probabilistic λ-Calculus and Quantitative Program Analysis.
Journal of Logic and Computation 15(2), 159–179, 2005