Quantum Recursion and Second Quantisation

Basic Ideas and Examples

Mingsheng Ying

University of Technology, Sydney, Australia
and
Tsinghua University, China

Happy Birthday Prakash!

Happy Birthday Prakash!

I'm very grateful to Prakash for teaching me second quantisation method during his visit to UTS in 2013

Outline

Outline

Classical Recursion of Quantum Programs

- Recursive procedure in quantum programming language QPL [Selinger, *Mathematical Structures in Computer Science*'2004].

Classical Recursion of Quantum Programs

- Recursive procedure in quantum programming language QPL [Selinger, *Mathematical Structures in Computer Science*'2004].
- Termination of quantum while-loops in finite-dimensional state spaces [Ying, Feng, *Acta Informatica*'2010].

Classical Recursion of Quantum Programs

- Recursive procedure in quantum programming language QPL [Selinger, *Mathematical Structures in Computer Science*'2004].
- Termination of quantum while-loops in finite-dimensional state spaces [Ying, Feng, *Acta Informatica*'2010].

- Selinger's slogan: ***Quantum data, classical control*** - control flow of the quantum recursions is classical because branchings are determined by the outcomes of certain quantum measurements.

Quantum Control Flow

Quantum programming language QML [Altenkirch and Grattage, *LICS'*2005]:

Two case constructs in the quantum setting:

- **case**, measure a qubit in the data it analyses - The control flow is determined by the outcome of a measurement and thus is classical.

Quantum Control Flow

Quantum programming language QML [Altenkirch and Grattage, *LICS'*2005]:

Two case constructs in the quantum setting:

- **case**, measure a qubit in the data it analyses - The control flow is determined by the outcome of a measurement and thus is classical.
- **case**$^{\circ}$, analyse quantum data without measuring - **if**$^{\circ}$ $-$ **then** $-$ **else** statement.

Quantum Control Flow

Hadamard gate:

$$\begin{aligned} had\ x = \ & \mathbf{if}^{\circ}\ x \\ & \quad \mathbf{then}\ \{\frac{1}{\sqrt{2}}(\mathtt{qfalse} - \mathtt{qtrue})\} \\ & \quad \mathbf{else}\ \ \{\frac{1}{\sqrt{2}}(\mathtt{qfalse} + \mathtt{qtrue})\} \end{aligned}$$

Quantum Control Flow

NOT gate:

$$\begin{aligned} \mathit{not}\ x = \ & \mathbf{if}^{\circ}\ x \\ & \quad \mathbf{then}\ \texttt{qfalse} \\ & \quad \mathbf{else}\ \ \texttt{qtrue} \end{aligned}$$

CNOT gate:

$$\begin{aligned} \mathit{cnot}\ c\ x = \ & \mathbf{if}^{\circ}\ c \\ & \quad \mathbf{then}\ (\texttt{qtrue}, \mathit{not}\ x) \\ & \quad \mathbf{else}\ \ (\texttt{qfalse}, x) \end{aligned}$$

Outline

"Coined" Quantum Case Statement

- Introduce an external "quantum coin" c:
  The state Hilbert space $\mathcal{H}_c = \text{span}\{|0\rangle, |1\rangle\}$

"Coined" Quantum Case Statement

- Introduce an <u>external</u> "quantum coin" c:
  The state Hilbert space $\mathcal{H}_c = \text{span}\{|0\rangle, |1\rangle\}$
- U_0 and U_1 two unitary transformations on a quantum system q - the state Hilbert space $\mathcal{H}_q$.

"Coined" Quantum Case Statement

- Introduce an <u>external</u> "quantum coin" c:
  The state Hilbert space $\mathcal{H}_c = \text{span}\{|0\rangle, |1\rangle\}$
- U_0 and U_1 two unitary transformations on a quantum system q - the state Hilbert space $\mathcal{H}_q$.
- A *quantum case statement* employing "quantum coin" c:

$$\begin{aligned}&\mathbf{qif}\ [c]\ |0\rangle \rightarrow U_0[q]\\ &\qquad\ \ \square\ |1\rangle \rightarrow U_1[q]\\ &\mathbf{fiq}\end{aligned}$$

“coined” Quantum Case Statement

- The semantics is an unitary operator U in $\mathcal{H}_c \otimes \mathcal{H}_q$ - the state Hilbert space of the composed system of “coin” c and principal system q:

$$U|0,\psi\rangle = |0\rangle U_0|\psi\rangle, \quad U|1,\psi\rangle = |1\rangle U_1|\psi\rangle$$

“coined” Quantum Case Statement

- The semantics is an unitary operator U in $\mathcal{H}_c \otimes \mathcal{H}_q$ - the state Hilbert space of the composed system of “coin” c and principal system q:

$$U|0,\psi\rangle = |0\rangle U_0|\psi\rangle, \quad U|1,\psi\rangle = |1\rangle U_1|\psi\rangle$$

- Matrix representation:

$$U = |0\rangle\langle 0| \otimes U_0 + |1\rangle\langle 1| \otimes U_1 = \begin{pmatrix} U_0 & 0 \\ 0 & U_1 \end{pmatrix}.$$

Quantum Choice

- V a unitary operator in the state Hilbert space $\mathcal{H}_c$ of the "coin".

Quantum Choice

- V a unitary operator in the state Hilbert space $\mathcal{H}_c$ of the "coin".
- The quantum choice of $U_0[q]$ and $U_1[q]$ with "coin-tossing" $V[c]$:

$$\begin{aligned}
& U_0[q] \oplus_{V[c]} U_1[q] \\
& \stackrel{\text{def}}{=} \\
& V[c];\ \textbf{qif}\ [c]\ |0\rangle \rightarrow U_0[q] \\
& \qquad\qquad\quad \square\ |1\rangle \rightarrow U_1[q] \\
& \qquad\quad \textbf{fiq}
\end{aligned}$$

Quantum Choice

- V a unitary operator in the state Hilbert space $\mathcal{H}_c$ of the "coin".
- The quantum choice of $U_0[q]$ and $U_1[q]$ with "coin-tossing" $V[c]$:

$$
\begin{aligned}
& U_0[q] \oplus_{V[c]} U_1[q] \\
& \stackrel{\text{def}}{=} \\
& V[c];\ \mathbf{qif}\ [c]\ |0\rangle \rightarrow U_0[q] \\
& \qquad\qquad\quad \square\ |1\rangle \rightarrow U_1[q] \\
& \qquad\quad \mathbf{fiq}
\end{aligned}
$$

- Compare with probabilistic choice [McIver and Morgan, *Abstraction, Refinement and Proof for Probabilistic Systems*, 2005]

External “Quantum Coin”

- Superpositions of time evolutions of a quantum system [Aharonov, Anandan, Popescu, Vaidman, *Plysical Review Letters* 1990].

External “Quantum Coin”

- Superpositions of time evolutions of a quantum system [Aharonov, Anandan, Popescu, Vaidman, *Plysical Review Letters* 1990].
- Quantum walks [Ambainis, Bach, Nayak, Vishwanath, Watrous, *STOC*’2001; Aharonov, Ambainis, Kempe, Vazirani, *STOC*’2001].

External "Quantum Coin"

- Superpositions of time evolutions of a quantum system [Aharonov, Anandan, Popescu, Vaidman, *Plysical Review Letters* 1990].
- Quantum walks [Ambainis, Bach, Nayak, Vishwanath, Watrous, *STOC*'2001; Aharonov, Ambainis, Kempe, Vazirani, *STOC*'2001].
- Unitary transformations $U_0[q], U_1[q]$ are replaced by general quantum programs that may contain quantum measurements? [Ying, Yu, Feng, arXiv:1209.4379]

Outline

A new notion of quantum recursion can be defined based on quantum case statement and quantum choice

A new notion of quantum recursion can be defined based on quantum case statement and quantum choice

Example - One-dimensional quantum walk

- One-dimensional random walk - a particle moves on a line marked by integers Z; at each step it moves one position left or right, depending on the flip of a fair coin.
- Hadamard walk - a quantum variant of one-dimensional random walk.

A new notion of quantum recursion can be defined based on quantum case statement and quantum choice

Example - One-dimensional quantum walk

- One-dimensional random walk - a particle moves on a line marked by integers Z; at each step it moves one position left or right, depending on the flip of a fair coin.
- Hadamard walk - a quantum variant of one-dimensional random walk.
- Its state Hilbert space $\mathcal{H}_d \otimes \mathcal{H}_p$:

A new notion of quantum recursion can be defined based on quantum case statement and quantum choice

Example - One-dimensional quantum walk

- One-dimensional random walk - a particle moves on a line marked by integers Z; at each step it moves one position left or right, depending on the flip of a fair coin.
- Hadamard walk - a quantum variant of one-dimensional random walk.
- Its state Hilbert space $\mathcal{H}_d \otimes \mathcal{H}_p$:
  - $\mathcal{H}_d = \text{span}\{|L\rangle, |R\rangle\}$, L, R indicate the direction Left and Right.

A new notion of quantum recursion can be defined based on quantum case statement and quantum choice

Example - One-dimensional quantum walk

- One-dimensional random walk - a particle moves on a line marked by integers Z; at each step it moves one position left or right, depending on the flip of a fair coin.
- Hadamard walk - a quantum variant of one-dimensional random walk.
- Its state Hilbert space $\mathcal{H}_d \otimes \mathcal{H}_p$:
  - $\mathcal{H}_d = \text{span}\{|L\rangle, |R\rangle\}$, L, R indicate the direction Left and Right.
  - $\mathcal{H}_p = \text{span}\{|n\rangle : n \in Z\}$, n indicates the position marked by integer n.

Example - One-dimensional quantum walk

- One step of Hadamard walk — $W = T(H \otimes I)$:

Example - One-dimensional quantum walk

- One step of Hadamard walk — $W = T(H \otimes I)$:
  - Translation T:

$$T|L,n\rangle = |L,n-1\rangle, \quad T|R,n\rangle = |R,n+1\rangle$$

  is unitary operator in $\mathcal{H}_d \otimes \mathcal{H}_p$.

Example - One-dimensional quantum walk

- One step of Hadamard walk — $W = T(H \otimes I)$:
  - Translation T:

    $$T|L,n\rangle = |L,n-1\rangle, \quad T|R,n\rangle = |R,n+1\rangle$$

    is unitary operator in $\mathcal{H}_d \otimes \mathcal{H}_p$.
  - $$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

    is Hadamard transform in the direction space $\mathcal{H}_d$

Example - One-dimensional quantum walk

- Define the left and right translation operators T_L and T_R in the position space $\mathcal{H}_p$:

$$T_L|n\rangle = |n-1\rangle, \quad T_R|n\rangle = |n+1\rangle$$

Example - One-dimensional quantum walk

- Define the left and right translation operators T_L and T_R in the position space $\mathcal{H}_p$:

$$T_L|n\rangle = |n-1\rangle, \quad T_R|n\rangle = |n+1\rangle$$

- Then the translation operator T is a quantum case statement:

$$\begin{aligned} T = \mathbf{qif}\ [d]\ & |L\rangle \rightarrow T_L[p] \\ & \square\ |R\rangle \rightarrow T_R[p] \\ \mathbf{fiq} & \end{aligned}$$

Example - One-dimensional quantum walk

- Define the left and right translation operators T_L and T_R in the position space $\mathcal{H}_p$:

$$T_L|n\rangle = |n-1\rangle, \quad T_R|n\rangle = |n+1\rangle$$

- Then the translation operator T is a quantum case statement:

$$\begin{aligned} T = \mathbf{qif}\ [d]\ |L\rangle &\rightarrow T_L[p] \\ \square\ |R\rangle &\rightarrow T_R[p] \\ \mathbf{fiq} & \end{aligned}$$

- The single-step walk operator W is a quantum choice:

$$T_L[p] \oplus_{H[d]} T_R[p]$$

Example - One-dimensional quantum walk

- Define the left and right translation operators T_L and T_R in the position space $\mathcal{H}_p$:

$$T_L|n\rangle = |n-1\rangle, \quad T_R|n\rangle = |n+1\rangle$$

- Then the translation operator T is a quantum case statement:

$$\begin{aligned} T = \mathbf{qif}\ [d]\ & |L\rangle \rightarrow T_L[p] \\ & \square\ |R\rangle \rightarrow T_R[p] \\ \mathbf{fiq} & \end{aligned}$$

- The single-step walk operator W is a quantum choice:

$$T_L[p] \oplus_{H[d]} T_R[p]$$

- Hadamard walk — ***repeated applications*** of operator W.

(Unidirectional) Recursive Hadamard Walk

- The walk first runs the "coin-tossing" Hadamard operator $H[d]$, and then a quantum case statement:

(Unidirectional) Recursive Hadamard Walk

- The walk first runs the "coin-tossing" Hadamard operator $H[d]$, and then a quantum case statement:
  - if the "direction coin" d is in state $|L\rangle$ then the walker moves one position left;

(Unidirectional) Recursive Hadamard Walk

- The walk first runs the "coin-tossing" Hadamard operator $H[d]$, and then a quantum case statement:
  - if the "direction coin" d is in state $|L\rangle$ then the walker moves one position left;
  - if d is in state $|R\rangle$ then it moves one position right, followed by a procedure ***behaving as the recursive walk itself***.

(Unidirectional) Recursive Hadamard Walk

- The walk first runs the "coin-tossing" Hadamard operator $H[d]$, and then a quantum case statement:
  - if the "direction coin" d is in state $|L\rangle$ then the walker moves one position left;
  - if d is in state $|R\rangle$ then it moves one position right, followed by a procedure ***behaving as the recursive walk itself***.
- Recursive Hadamard walk — program X declared by the recursive equation:

$$X \Leftarrow T_L[p] \oplus_{H[d]} (T_R[p]; X)$$

Bidirectional Recursive Hadamard Walk

- The walk first runs the “coin-tossing” Hadamard operator $H[d]$ and then a quantum case statement:

Bidirectional Recursive Hadamard Walk

- The walk first runs the "coin-tossing" Hadamard operator $H[d]$ and then a quantum case statement:
  - if the "direction coin" d is in state $|L\rangle$ then the walker moves one position left, followed by **a procedure behaving as the recursive walk itself**;

Bidirectional Recursive Hadamard Walk

- The walk first runs the "coin-tossing" Hadamard operator $H[d]$ and then a quantum case statement:
  - if the "direction coin" d is in state $|L\rangle$ then the walker moves one position left, followed by **a procedure behaving as the recursive walk itself**;
  - if d is in state $|R\rangle$ then it moves one position right, also followed by **a procedure behaving as the recursive walk itself**.

Bidirectional Recursive Hadamard Walk

- The walk first runs the "coin-tossing" Hadamard operator $H[d]$ and then a quantum case statement:
  - if the "direction coin" d is in state $|L\rangle$ then the walker moves one position left, followed by **a procedure behaving as the recursive walk itself**;
  - if d is in state $|R\rangle$ then it moves one position right, also followed by **a procedure behaving as the recursive walk itself**.
- The walk — Program X (or program Y) declared by the recursive equation:

$$X \Leftarrow (T_L[p]; X) \oplus_{H[d]} (T_R[p]; X)$$

Bidirectional Recursive Hadamard Walk

- The walk first runs the "coin-tossing" Hadamard operator $H[d]$ and then a quantum case statement:
  - if the "direction coin" d is in state $|L\rangle$ then the walker moves one position left, followed by **a procedure behaving as the recursive walk itself**;
  - if d is in state $|R\rangle$ then it moves one position right, also followed by **a procedure behaving as the recursive walk itself**.
- The walk — Program X (or program Y) declared by the recursive equation:

$$X \Leftarrow (T_L[p]; X) \oplus_{H[d]} (T_R[p]; X)$$

- A variant of the bidirectional recursive Hadamard walk is declared by the following system of recursive equations:

$$\begin{cases} X \Leftarrow T_L[p] \oplus_{H[d]} (T_R[p]; Y), \\ Y \Leftarrow (T_L[p]; X) \oplus_{H[d]} T_R[p] \end{cases}$$

A More Interesting Recursive Quantum Walk

- Let $n \geq 2$. Another variant of unidirectional recursive quantum walk is defined as the program declared by the following recursive equation:

$$X \Leftarrow ((T_L[p]; X) \oplus_{H[d]} (T_R[p]; X)); (T_L[p] \oplus_{H[d]} T_R[p])^n$$

A More Interesting Recursive Quantum Walk

- Let $n \geq 2$. Another variant of unidirectional recursive quantum walk is defined as the program declared by the following recursive equation:

$$X \Leftarrow ((T_L[p]; X) \oplus_{H[d]} (T_R[p]; X)); (T_L[p] \oplus_{H[d]} T_R[p])^n$$

How to solve these quantum recursive equations?

Syntactic Approximation

- A recursive program X declared by equation

$$X \Leftarrow F(X)$$

Syntactic Approximation

- A recursive program X declared by equation

$$X \Leftarrow F(X)$$

- Syntactic approximations:

$$\begin{cases} X^{(0)} = \textbf{Abort}, \\ X^{(n+1)} = F[X^{(n)}/X] \text{ for } n \geq 0. \end{cases}$$

Program $X^{(n)}$ is the nth syntactic approximation of X.

Syntactic Approximation

- A recursive program X declared by equation

$$X \Leftarrow F(X)$$

- Syntactic approximations:

$$\begin{cases} X^{(0)} = \mathbf{Abort}, \\ X^{(n+1)} = F[X^{(n)}/X] \text{ for } n \geq 0. \end{cases}$$

  Program $X^{(n)}$ is the nth syntactic approximation of X.

- Semantics $[\![X]\!]$ of X is the limit

$$[\![X]\!] = \lim_{n\to\infty} [\![X^{(n)}]\!]$$

Example - (Unidirectional) Recursive Hadamard Walk

$$X^{(0)} = \textbf{abort},$$
$$X^{(1)} = T_L[p] \oplus_{H[d]} (T_R[p]; \textbf{abort}),$$
$$X^{(2)} = T_L[p] \oplus_{H[d]} (T_R[p]; T_L[p] \oplus_{H[d_1]} (T_R[p]; \textbf{abort})),$$
$$X^{(3)} = T_L[p] \oplus_{H[d]} (T_R[p]; T_L[p] \oplus_{H[d_1]} (T_R[p]; T_L[p] \oplus_{H[d_2]} (T_R[p]; \textbf{abort}))),$$

....

Example - (Unidirectional) Recursive Hadamard Walk

$$X^{(0)} = \textbf{abort},$$
$$X^{(1)} = T_L[p] \oplus_{H[d]} (T_R[p]; \textbf{abort}),$$
$$X^{(2)} = T_L[p] \oplus_{H[d]} (T_R[p]; T_L[p] \oplus_{H[d_1]} (T_R[p]; \textbf{abort})),$$
$$X^{(3)} = T_L[p] \oplus_{H[d]} (T_R[p]; T_L[p] \oplus_{H[d_1]} (T_R[p]; T_L[p] \oplus_{H[d_2]} (T_R[p]; \textbf{abort}))),$$

....

Observation

- Continuously introduce new "coin" to avoid variable conflict.
- Variables $d, d_1, d_2, ...$ denote identical particles.

Example - (Unidirectional) Recursive Hadamard Walk

$$
\begin{aligned}
X^{(0)} &= \mathbf{abort}, \\
X^{(1)} &= T_L[p] \oplus_{H[d]} (T_R[p]; \mathbf{abort}), \\
X^{(2)} &= T_L[p] \oplus_{H[d]} (T_R[p]; T_L[p] \oplus_{H[d_1]} (T_R[p]; \mathbf{abort})), \\
X^{(3)} &= T_L[p] \oplus_{H[d]} (T_R[p]; T_L[p] \oplus_{H[d_1]} (T_R[p]; T_L[p] \oplus_{H[d_2]} (T_R[p]; \mathbf{abort}))),
\end{aligned}
$$

....

Observation

- Continuously introduce new "coin" to avoid variable conflict.
- Variables $d, d_1, d_2, \ldots$ denote identical particles.
- The number of the "coin" particles that are needed in running the recursive walk is unknown beforehand.

Example - (Unidirectional) Recursive Hadamard Walk

$$X^{(0)} = \mathbf{abort},$$
$$X^{(1)} = T_L[p] \oplus_{H[d]} (T_R[p]; \mathbf{abort}),$$
$$X^{(2)} = T_L[p] \oplus_{H[d]} (T_R[p]; T_L[p] \oplus_{H[d_1]} (T_R[p]; \mathbf{abort})),$$
$$X^{(3)} = T_L[p] \oplus_{H[d]} (T_R[p]; T_L[p] \oplus_{H[d_1]} (T_R[p]; T_L[p] \oplus_{H[d_2]} (T_R[p]; \mathbf{abort}))),$$

....

Observation

- Continuously introduce new "coin" to avoid variable conflict.
- Variables $d, d_1, d_2, ...$ denote identical particles.
- The number of the "coin" particles that are needed in running the recursive walk is unknown beforehand.
- We need to deal with ***quantum systems where the number of particles of the same type may vary***.

Outline

Fock Spaces

- ***The principle of symmetrisation***: the states of n identical particles are either completely symmetric or completely antisymmetric with respect to the permutations of the particles. [bosons - symmetric; fermions - antisymmetric]

Fock Spaces

- ***The principle of symmetrisation***: the states of n identical particles are either completely symmetric or completely antisymmetric with respect to the permutations of the particles. [bosons - symmetric; fermions - antisymmetric]
- Let $\mathcal{H}$ be the state Hilbert space of one particle.

Fock Spaces

- ***The principle of symmetrisation***: the states of n identical particles are either completely symmetric or completely antisymmetric with respect to the permutations of the particles. [bosons - symmetric; fermions - antisymmetric]
- Let $\mathcal{H}$ be the state Hilbert space of one particle.
- For each permutation π of $1, ..., n$, define the permutation operator P_π in $\mathcal{H}^{\otimes n}$ by

$$P_\pi |\psi_1 \otimes ... \otimes \psi_n\rangle = |\psi_{\pi(1)} \otimes ... \otimes \psi_{\pi(n)}\rangle$$

Fock Spaces

- ***The principle of symmetrisation***: the states of n identical particles are either completely symmetric or completely antisymmetric with respect to the permutations of the particles. [bosons - symmetric; fermions - antisymmetric]
- Let $\mathcal{H}$ be the state Hilbert space of one particle.
- For each permutation π of $1, ..., n$, define the permutation operator P_π in $\mathcal{H}^{\otimes n}$ by

$$P_\pi |\psi_1 \otimes ... \otimes \psi_n\rangle = |\psi_{\pi(1)} \otimes ... \otimes \psi_{\pi(n)}\rangle$$

- Define the symmetrisation and antisymmetrisation operators in $\mathcal{H}^{\otimes n}$:

$$S_+ = \frac{1}{n!} \sum_\pi P_\pi, \qquad S_- = \frac{1}{n!} \sum_\pi (-1)^\pi P_\pi$$

Fock Spaces

$v = +$ for bosons, $v = -$ for fermions.

- Write

$$|\psi_1, ..., \psi_n\rangle_v = S_v |\psi_1 \otimes ... \otimes \psi_n\rangle.$$

Fock Spaces

$v = +$ for bosons, $v = -$ for fermions.

- Write
$$|\psi_1, ..., \psi_n\rangle_v = S_v |\psi_1 \otimes ... \otimes \psi_n\rangle.$$
- The state space of n bosons and that of fermions are

$$\mathcal{H}_v^{\otimes n} = S_v \mathcal{H}^{\otimes n} = \mathrm{span}\{|\psi_1, ..., \psi_n\rangle_v : |\psi_1\rangle, ..., |\psi_n\rangle \text{ are in } \mathcal{H}\}$$

Fock Spaces

$v = +$ for bosons, $v = -$ for fermions.

- Write
$$|\psi_1, ..., \psi_n\rangle_v = S_v |\psi_1 \otimes ... \otimes \psi_n\rangle.$$
- The state space of n bosons and that of fermions are
$$\mathcal{H}_v^{\otimes n} = S_v \mathcal{H}^{\otimes n} = \text{span}\{|\psi_1, ..., \psi_n\rangle_v : |\psi_1\rangle, ..., |\psi_n\rangle \text{ are in } \mathcal{H}\}$$
- Introduce the vacuum state $|\mathbf{0}\rangle$ and the one-dimensional space $\mathcal{H}_v^{\otimes 0} = \mathcal{H}^{\otimes 0} = \text{span}\{|\mathbf{0}\rangle\}$.

Fock Spaces

$v = +$ for bosons, $v = -$ for fermions.

- Write
$$|\psi_1, ..., \psi_n\rangle_v = S_v|\psi_1 \otimes ... \otimes \psi_n\rangle.$$
- The state space of n bosons and that of fermions are
$$\mathcal{H}_v^{\otimes n} = S_v \mathcal{H}^{\otimes n} = \text{span}\{|\psi_1, ..., \psi_n\rangle_v : |\psi_1\rangle, ..., |\psi_n\rangle \text{ are in } \mathcal{H}\}$$
- Introduce the vacuum state $|\mathbf{0}\rangle$ and the one-dimensional space $\mathcal{H}_v^{\otimes 0} = \mathcal{H}^{\otimes 0} = \text{span}\{|\mathbf{0}\rangle\}$.
- The space of the states of variable particle number is the Fock space:
$$\mathcal{F}_v(\mathcal{H}) = \sum_{n=0}^{\infty} \mathcal{H}_v^{\otimes n}$$

Fock Spaces

$v = +$ for bosons, $v = -$ for fermions.

- Write
$$|\psi_1, ..., \psi_n\rangle_v = S_v|\psi_1 \otimes ... \otimes \psi_n\rangle.$$
- The state space of n bosons and that of fermions are
$$\mathcal{H}_v^{\otimes n} = S_v\mathcal{H}^{\otimes n} = \text{span}\{|\psi_1, ..., \psi_n\rangle_v : |\psi_1\rangle, ..., |\psi_n\rangle \text{ are in } \mathcal{H}\}$$
- Introduce the vacuum state $|\mathbf{0}\rangle$ and the one-dimensional space $\mathcal{H}_v^{\otimes 0} = \mathcal{H}^{\otimes 0} = \text{span}\{|\mathbf{0}\rangle\}$.
- The space of the states of variable particle number is the Fock space:
$$\mathcal{F}_v(\mathcal{H}) = \sum_{n=0}^{\infty} \mathcal{H}_v^{\otimes n}$$
- The free Fock space:
$$\mathcal{F}(\mathcal{H}) = \sum_{n=0}^{\infty} \mathcal{H}^{\otimes n}$$

Evolution in the Fock Spaces

- Let the (discrete-time) evolution of one particle be unitary operator U.

Evolution in the Fock Spaces

- Let the (discrete-time) evolution of one particle be unitary operator U.
- The evolution of n particles without mutual interactions is operator $\mathbf{U}$ in $\mathcal{H}^{\otimes n}$:

$$\mathbf{U}|\psi_1 \otimes ... \otimes \psi_n\rangle = |U\psi_1 \otimes ... \otimes U\psi_n\rangle$$

Evolution in the Fock Spaces

- Let the (discrete-time) evolution of one particle be unitary operator U.
- The evolution of n particles without mutual interactions is operator $\mathbf{U}$ in $\mathcal{H}^{\otimes n}$:

$$\mathbf{U}|\psi_1 \otimes ... \otimes \psi_n\rangle = |U\psi_1 \otimes ... \otimes U\psi_n\rangle$$

- $$\mathbf{U}|\psi_1, ..., \psi_n\rangle_v = |U\psi_1, ...U\psi_n\rangle_v.$$

Evolution in the Fock Spaces

- Let the (discrete-time) evolution of one particle be unitary operator U.
- The evolution of n particles without mutual interactions is operator $\mathbf{U}$ in $\mathcal{H}^{\otimes n}$:

$$\mathbf{U}|\psi_1 \otimes ... \otimes \psi_n\rangle = |U\psi_1 \otimes ... \otimes U\psi_n\rangle$$

- $$\mathbf{U}|\psi_1, ..., \psi_n\rangle_v = |U\psi_1, ...U\psi_n\rangle_v.$$
- Extend to the Fock spaces $\mathcal{F}_v(\mathcal{H})$ and $\mathcal{F}(\mathcal{H})$:

$$\mathbf{U}\left(\sum_{n=0}^{\infty} |\Psi(n)\rangle\right) = \sum_{n=0}^{\infty} \mathbf{U}|\Psi(n)\rangle$$

Creation and Annihilation of Particles

- The transitions between states of different particle numbers.

Creation and Annihilation of Particles

- The transitions between states of different particle numbers.
- Creation operator $a^*(\psi)$ in $\mathcal{F}_v(\mathcal{H})$:

$$a^*(\psi)|\psi_1, ..., \psi_n\rangle_v = \sqrt{n+1}|\psi, \psi_1, ..., \psi_n\rangle_v$$

Add a particle in the individual state $|\psi\rangle$ to the system of n particles without modifying their respective states.

Creation and Annihilation of Particles

- The transitions between states of different particle numbers.
- Creation operator $a^*(\psi)$ in $\mathcal{F}_v(\mathcal{H})$:

$$a^*(\psi)|\psi_1, ..., \psi_n\rangle_v = \sqrt{n+1}|\psi, \psi_1, ..., \psi_n\rangle_v$$

  Add a particle in the individual state $|\psi\rangle$ to the system of n particles without modifying their respective states.
- Annihilation operator $a(\psi)$ — the Hermitian conjugate of $a^*(\psi)$:

$$a(\psi)|\mathbf{0}\rangle = 0,$$

$$a(\psi)|\psi_1, ..., \psi_n\rangle_v = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}(v)^{i-1}\langle\psi|\psi_i\rangle|\psi_1, ..., \psi_{i-1}, \psi_{i+1}, ..., \psi_n\rangle_v$$

  Decrease the number of particles by one unit, while preserving the symmetry of the state.

Outline

Second quantisation provides us with the necessary tool for defining the semantics of quantum recursion!

Second quantisation provides us with the necessary tool for defining the semantics of quantum recursion!

Example - (Unidirectional) Recursive Hadamard Walk

Semantics of the recursive Hadamard walk:

$$[\![X]\!] = \left[\sum_{i=0}^{\infty}\left(\bigotimes_{j=0}^{i-1}|R\rangle_{d_j}\langle R| \otimes |L\rangle_{d_i}\langle L|\right) \otimes T_L T_R^i\right](\mathbf{H} \otimes I)$$

- An operator in

$$\mathcal{F}_v(\mathcal{H}_d) \otimes \mathcal{H}_p \to \mathcal{F}(\mathcal{H}_d) \otimes \mathcal{H}_p.$$

- The sign v is $+$ or $-$, depending on using bosons or fermions to implement the "direction coins" $d, d_1, d_2, ...$

Principal System Semantics

- Each state $|\Psi\rangle$ in Fock space $\mathcal{F}_v(\mathcal{H}_d)$ induces mapping:

$$
\begin{aligned}
&[\![X, \Psi]\!]_p : \text{pure states} \rightarrow \text{partial density operators in } \mathcal{H}_p \\
&[\![X, \Psi]\!]_p(|\psi\rangle) = tr_{\mathcal{F}(\mathcal{H}_d)}(|\Phi\rangle\langle\Phi|)
\end{aligned}
$$

  where $|\Phi\rangle = [\![X]\!](|\Psi\rangle \otimes |\psi\rangle)$

Principal System Semantics

- Each state $|\Psi\rangle$ in Fock space $\mathcal{F}_v(\mathcal{H}_d)$ induces mapping:

$$[\![X,\Psi]\!]_p : \text{pure states} \to \text{partial density operators in } \mathcal{H}_p$$
$$[\![X,\Psi]\!]_p(|\psi\rangle) = tr_{\mathcal{F}(\mathcal{H}_d)}(|\Phi\rangle\langle\Phi|)$$

where $|\Phi\rangle = [\![X]\!](|\Psi\rangle \otimes |\psi\rangle)$

- Mapping $[\![X,\Psi]\!]_p$ is called the principal system semantics of X with “coin” initialisation $|\Psi\rangle$.

Bidirectional Recursive Quantum Walk

$$\begin{cases} X \Leftarrow T_L[p] \oplus_{H[d]} (T_R[p]; Y), \\ Y \Leftarrow (T_L[p]; X) \oplus_{H[d]} T_R[p] \end{cases}$$

- Coherent state of bosons in the symmetric Fock space $\mathcal{F}_+(\mathcal{H})$ over $\mathcal{H}$:

$$|\psi\rangle_{\text{coh}} = \exp\left(-\frac{1}{2}\langle\psi|\psi\rangle\right) \sum_{n=0}^{\infty} \frac{[a^*(\psi)]^n}{n!} |\mathbf{0}\rangle$$

Bidirectional Recursive Quantum Walk

$$\begin{cases} X \Leftarrow T_L[p] \oplus_{H[d]} (T_R[p]; Y), \\ Y \Leftarrow (T_L[p]; X) \oplus_{H[d]} T_R[p] \end{cases}$$

- Coherent state of bosons in the symmetric Fock space $\mathcal{F}_+(\mathcal{H})$ over $\mathcal{H}$:

$$|\psi\rangle_{\mathrm{coh}} = \exp\left(-\frac{1}{2}\langle\psi|\psi\rangle\right) \sum_{n=0}^{\infty} \frac{[a^*(\psi)]^n}{n!} |\mathbf{0}\rangle$$

- The walk starts from position 0 and the coins are initialised in the coherent states of bosons corresponding to $|L\rangle$:

$$\begin{aligned} [\![X, L_{\mathrm{coh}}]\!]_p(|0\rangle) &= \frac{1}{\sqrt{e}} \left(\sum_{k=0}^{\infty} \frac{1}{2^{2k+1}} |-1\rangle\langle -1| + \sum_{k=0}^{\infty} \frac{1}{2^{2k+2}} |2\rangle\langle 2| \right) \\ &= \frac{1}{\sqrt{e}} \left(\frac{2}{3} |-1\rangle\langle -1| + \frac{1}{3} |2\rangle\langle 2| \right). \end{aligned}$$

Quantum while-loop

- Program X declared by the recursive equation

$$\begin{aligned} X \Leftarrow\ & W[c,q]; \mathbf{qif}[c]\ |0\rangle \rightarrow \mathbf{skip} \\ & \qquad\qquad\quad \square\ |1\rangle \rightarrow U[q]; X \\ & \qquad\quad \mathbf{fiq} \end{aligned}$$

  where W a unitary operator in $\mathcal{H}_c \otimes \mathcal{H}_q$ — the interaction between the "coin" c and the principal system q.

Quantum while-loop

- Program X declared by the recursive equation

$$X \Leftarrow \; W[c,q]; \mathbf{qif}[c] \; |0\rangle \rightarrow \mathbf{skip} \\ \square \; |1\rangle \rightarrow U[q]; X \\ \mathbf{fiq}$$

where W a unitary operator in $\mathcal{H}_c \otimes \mathcal{H}_q$ — the interaction between the "coin" c and the principal system q.

- Semantics of X:

$$[\![X]\!] = \sum_{k=1}^{\infty} \prod_{j=0}^{k-1} W[c_j, q] \left(\bigotimes_{j=0}^{k-2} |1\rangle_{c_j}\langle 1| \otimes |0\rangle_{c_{k-1}}\langle 0| \otimes U^{k-1}[q] \right)$$

from the space $\mathcal{F}_v(\mathcal{H}_2) \otimes \mathcal{H}_q$ into $\mathcal{F}(\mathcal{H}_2) \otimes \mathcal{H}_q$.

Outline

Problems:

- What kind of problems can be solved more conveniently by using quantum recursion? Sorting? [Høyer, Neerbek, Shi, *ICALP*'2001]

Problems:

- What kind of problems can be solved more conveniently by using quantum recursion? Sorting? [Høyer, Neerbek, Shi, *ICALP*'2001]
- Hoare logic for quantum while-loops defined using quantum "coins"?

Problems:

- What kind of problems can be solved more conveniently by using quantum recursion? Sorting? [Høyer, Neerbek, Shi, *ICALP*'2001]
- Hoare logic for quantum while-loops defined using quantum "coins"?
  - Fock space can serve as a model of linear logic with exponential types [Blute, Panangaden, Seely, *MFPS*'1994].

Problems:

- What kind of problems can be solved more conveniently by using quantum recursion? Sorting? [Høyer, Neerbek, Shi, *ICALP*'2001]
- Hoare logic for quantum while-loops defined using quantum "coins"?
  - Fock space can serve as a model of linear logic with exponential types [Blute, Panangaden, Seely, *MFPS*'1994].
  - Combine linear logic with Hoare logic for quantum programs [Ying, *TOPLAS*'2011]?

Problems:

- What kind of problems can be solved more conveniently by using quantum recursion? Sorting? [Høyer, Neerbek, Shi, *ICALP*'2001]
- Hoare logic for quantum while-loops defined using quantum "coins"?
  - Fock space can serve as a model of linear logic with exponential types [Blute, Panangaden, Seely, *MFPS*'1994].
  - Combine linear logic with Hoare logic for quantum programs [Ying, *TOPLAS*'2011]?
- What kind of physical systems can be used to implement quantum recursion where new "coins" must be continuously created?

Thank You!