Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 170 (2007) 125–138
www.elsevier.com/locate/entcs

Quantum Patterns and Types for Entanglement and Separability

Simon Perdrix[1]

Leibniz Laboratory
IMAG-INPG
Grenoble, France

Abstract

As a first step toward a notion of quantum data structures, we introduce a typing system for reflecting entanglement and separability. This is presented in the context of classically controlled quantum computation where a classical program controls a sequence of quantum operations, *i.e.* unitary transformations and measurements acting on a quantum memory. Abstract models for such quantum computations are the Quantum Random Access Machine (*QRAM* [5]) and the Classically-Controlled Quantum Turing Machine (*CQTM* [9]). Several quantum programming languages follow this model [1,3,6,12,13]. Among them, the functional language defined by Valiron [15] is the basis for the language developed in this paper. This is work in progress.

Keywords: Quantum programming languages, quantum types, entanglement and separability.

1 Basic Notions: Separability and Entanglement

The state of a qubit is a normalized vector in the 2-dimensional Hilbert space $\mathbb{C}^2$. The state of a set of n qubits is generally described by a normalized vector in the 2^n-dimensional Hilbert space $\otimes_{i=1}^{n}\mathbb{C}^2$.

Among all possible states of a set of qubits, some of them are separable:

Definition 1.1 For a state $|\varphi\rangle$ of a set S of qubits, $|\varphi\rangle$ is **separable** *iff* there exists a partition $\{A, B\}$ of S (where both A and B are non empty sets) and two states $|\varphi_A\rangle$ and $|\varphi_B\rangle$ of the respective parts A and B such that $|\varphi\rangle = |\varphi_A\rangle \otimes |\varphi_B\rangle$.

[1] Email: simon.perdrix@imag.fr

doi:10.1016/j.entcs.2006.12.015

Definition 1.2 A quantum state $|\varphi\rangle$ is **entangled** *iff* $|\varphi\rangle$ is not separable.

For instance, a *Bell* state $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$ and the *GHZ* state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ are entangled. Entanglement is the basis for non-local operations (quantum teleportation [8]) and constitutes a fundamental resource for measurement based quantum computation [10,2]. Moreover, significant speedups (e.g. Shor algorithm [14]) are made possible thanks to entanglement.

Notions of entanglement and separability can be extended to qubits, and can be represented by a relation over the qubits of a set of qubits:

Definition 1.3 For a given state $|\varphi\rangle$ of a set S of qubits, two qubits x, y of S are **separable** *iff* there exists a partition $\{X, Y\}$ of the qubits of S such that $x \in X$ and $y \in Y$, and a state $|\varphi_X\rangle$ ($|\varphi_Y\rangle$) of the qubits in X (Y), such that $|\varphi\rangle = |\varphi_X\rangle \otimes |\varphi_Y\rangle$.

Definition 1.4 Two qubits x, y are **entangled** *iff* they are not separable.

Definition 1.5 For a given state $|\varphi\rangle$ of a set S of qubits, $R_E(|\varphi\rangle)$ is the **entanglement relation** over the qubits of S: $(x, y) \in R_E$ *iff* x and y are entangled.

Lemma 1.6 *For any* $|\varphi\rangle$, $R_E(|\varphi\rangle)$ *is an equivalence relation.*

Proof.

- For any qubit x, according to the previous definitions, x is entangled with itself, so $R_E(|\varphi\rangle)$ is *reflexive*;
- $R_E(|\varphi\rangle)$ is trivially *symmetric*;
- given x, y, z such that $(x, y) \in R_E(|\varphi\rangle)$ and $(y, z) \in R_E(|\varphi\rangle)$. If x and z are separable, there exist a partition $\{X, Z\}$ of the qubits of $|\varphi\rangle$ and two states $|\varphi_X\rangle, |\varphi_Z\rangle$ such that $|\varphi\rangle = |\varphi_X\rangle \otimes |\varphi_Z\rangle$. Since y is either in X or Z, either $(x, y) \notin R_E(|\varphi\rangle)$ or $(y, z) \notin R_E(|\varphi\rangle)$. So, by contradiction, $R_E(|\varphi\rangle)$ is *transitive*.

Thus $R_E(|\varphi\rangle)$ is an *equivalence relation.* □

2 Types for Teleportation

In order to point out the importance of handling entanglement and separability in quantum programming languages, we analyze the specifications of a program for teleportation.

Teleportation can be represented as a function taking three qubits and outputting three qubits. The type of this function could be:

$$teleportation : qbit \otimes qbit \otimes qbit \multimap qbit \otimes qbit \otimes qbit$$

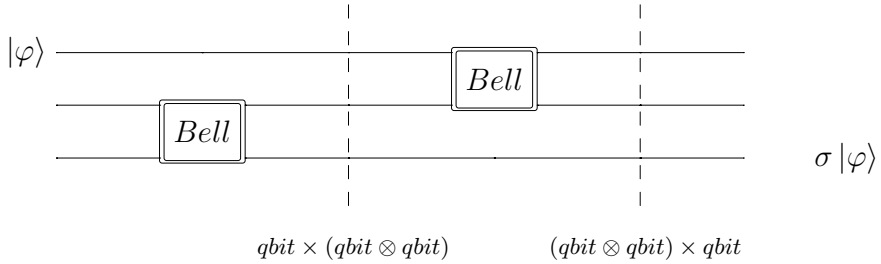

Fig. 1. Input and output types for teleportation

where $qbit \otimes qbit$ is a type for any pair of two qubits, either entangled or separable.

After teleportation, the third qubit (Bob's qubit) is not entangled with the other two because of the Bell measurement performed on them by Alice. Thus, this qubit can be manipulated independently without affecting the others. In order to represent this separability, a cartesian product, instead of a tensor product, may be used in the typing:

$$teleportation : qbit \otimes qbit \otimes qbit \multimap (qbit \otimes qbit) \times qbit$$

Furthermore, the input to this function must be separable, since teleportation has no meaning if the input state is for instance a 3-qubit GHZ state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle$. This constraint of input separability could be stated as follows:

$$teleportation : qbit \times (qbit \otimes qbit) \multimap (qbit \otimes qbit) \times qbit$$

In order to show how such specifications could be conveyed through a typing system, we introduce a typed quantum functional language, using linear logic with two products: a tensor product for the general case and a cartesian product to represent separability constraints. $A \times B$ is then a subtype of $A \otimes B$ since separable states are also general states. Thus the cartesian product provides additional information on the state of a system, while specifying a constraint for an input to be separable.

3 Terms

The language introduced here is largely inspired from Valiron's quantum functional language [15]. A quantum program state is a triple $[Q, f, M]$, where:

- M is a term:

$$
\begin{array}{rcl}
M & ::= & \mathbf{fun}\ P \to M \mid (MM) \mid \mathbf{if}\ (M;M;M) \mid \langle M,M\rangle \mid U_1\ M \mid U_2\ M \mid \\
 & & \mathbf{meas}\ M \mid \mathbf{let}\ x = \mathbf{new}\ M\ \mathbf{in}\ M \mid x \mid 0 \mid 1 \\
P & ::= & T \mid \langle P \diamond P\rangle \\
T & ::= & x \mid \langle T,T\rangle
\end{array}
$$

with x ranging over $\mathcal{V}$ a countable set of variables, and c ranging over $\mathcal{C}$ a set of constants;

- f is a function from $\mathcal{V}$ to $\{1,\dots,n\}$, the index set of qubits in the memory;
- Q is a vector in a vector space $\mathbb{C}^{2^n}$ of dimension 2^n.

In order to simplify the writing of the reduction rules, p_i denotes a variable x such that $f(x) = i$. The quantum program state then becomes a pair $[Q, M']$ with $M' = M[p_{f(x_1)}/x_1]\dots[p_{f(x_n)}/x_n]$ if the domain of f is $\{x_1,\dots x_n\}$.

The syntax of terms allows: abstraction and application of functions; conditional expressions and formation of pairs of terms; application of quantum operations like measurements and unitary transformations; and creation and initialization of qubits. Patterns are specified as single variables or as pairs $\langle P_1, P_2\rangle$ or $\langle T_1, T_2\rangle$. The diamond ($\diamond$) is a syntactic way of expressing the constraint that an argument must be separable: for instance if the teleportation function is defined as $\mathbf{fun}\ \langle x \diamond \langle y,z\rangle\rangle \to M$, its type is $qbit \times (qbit \otimes qbit) \multimap A$, where A is the type of M.

The entanglement relation of a new qubit relates this qubit with itself. Since the information about entanglement is stated in the typing rules (5) by a relation over variable names, the new qubit must be immediately named. That is why the syntax always gives names to newly created qubits $\mathbf{let}\ x = \mathbf{new}\ M\ \mathbf{in}\ M$.

4 Reduction

Following the approach developed in [15] by Valiron, a call-by-value reduction system is required to tackle the consequences of no-cloning in the language: the argument must be evaluated before actually applying a function.

A value is a term V of the following form:

$$V \quad ::= \quad x \mid \mathbf{fun}\ P \to M \mid 0 \mid 1 \mid \langle V,V\rangle$$

Due to the probabilistic nature of measurement, the reduction system is probabilistic: state $\to_p$ state can be applied with probability p.

- Function application:

$$\overline{[Q, (\,\mathbf{fun}\; x \to M)\; V] \to_1 [Q, M[V/x]]}$$

If the argument is a pair:

$$\overline{[Q, (\,\mathbf{fun}\; P \to M)\; V] \to_1 [Q, M[V/P]]}$$

where the substitution $M[V/P]$ is recursively defined as:

$$\begin{aligned} M[\langle V_1, V_2\rangle/\langle P_1 \diamond P_2\rangle] &= M[V_1/P_1][V_2/P_2] \\ M[\langle V_1, V_2\rangle/\langle T_1, T_2\rangle] &= M[V_1/T_1][V_2/T_2] \end{aligned}$$

- In any case, the argument is evaluated first, before function application, then the function is evaluated:

$$\frac{[Q, N] \to_p [Q', N']}{[Q, MN] \to_p [Q', MN']}$$

$$\frac{[Q, M] \to_p [Q', M']}{[Q, MV] \to_p [Q', M'V]}$$

- The cases for **if** :

$$\overline{[Q,\ \mathbf{if}\ (0; N_1; N_0)\ ] \to_1 [Q, N_0]}$$

$$\overline{[Q,\ \mathbf{if}\ (1; N_1; N_0)\ ] \to_1 [Q, N_0]}$$

$$\frac{[Q, M] \to_p [Q', M']}{[Q,\ \mathbf{if}\ (M; N_1; N_0)\ ] \to_p [Q',\ \mathbf{if}\ (M'; N_1; N_0)\ ]}$$

- Measurement of a qubit indexed by i:

For a given qubit i, Q is a superposition of two states, where $\{|0_i\rangle, |1_i\rangle\}$ is the standard basis for qubit i:

$$Q = \alpha\,|0_i\rangle \otimes Q_0 + \beta\,|1_i\rangle \otimes Q_1$$

Thus:

$$\overline{[Q, \mathbf{meas}\; p_i] \to_{|\alpha|^2} [|0_i\rangle \otimes Q_0, p_i]}$$

$$\overline{[Q, \mathbf{meas}\; p_i] \to_{|\beta|^2} [|1_i\rangle \otimes Q_1, p_i]}$$

- Creation and initialisation of a qubit:

Let i be a fresh qubit index ($i \notin range(f)$):

$$\overline{[Q, \mathbf{let}\; x = \mathbf{new}\; 0\, \mathbf{in}\; M\ ] \to_1 [Q \otimes |0_i\rangle, M[p_i/x]]}$$

$$\overline{[Q,\ \mathbf{let}\ x = \mathbf{new}\ 1\ \mathbf{in}\ M\ ] \to_1 [Q \otimes |1_i\rangle\ , M[p_i/x]]}$$

$$\frac{[Q, N] \to_p [Q', N']}{[Q,\ \mathbf{let}\ x = \mathbf{new}\ N\ \mathbf{in}\ M\ ] \to_p [Q',\ \ \mathbf{let}\ x = \mathbf{new}\ N'\ \mathbf{in}\ M]}$$

- The case for one-qubit unitary transformations:
  For a given qubit i :

$$\overline{[Q, U_1\ p_i] \to_1 [U_1^{(i)} Q, p_i]}$$

  where $U_1^{(i)}$ is the unitary transformation which applies U_1 to qubit i.
- The case for two-qubit unitary transformations:
  Given two qubits i, j:

$$\overline{[Q, U_2\langle p_i, p_j\rangle] \to_1 [U_2^{(i,j)} Q, \langle p_i, p_j\rangle]}$$

5 Types

There is a basic type for classical data (*bit*) and a basic type for quantum data (*qbit*). These types can be combined by operators of linear logic: $!A$ means that A is duplicable; $A \multimap B$ is the type of a function with an argument of type A outputting a result of type B. Additionally, two products of types are allowed, $\otimes$ and $\times$: the type of a pair composed of a term of type A and a term of type B is $A \otimes B$ or $A \times B$. On duplicable data the two products are equivalent; on quantum data a type $A \times B$ means that the two terms of the pair are separable.

$$\begin{aligned} A &::= \mathit{bit} \mid\ !A \mid A \multimap A \mid A \times A \mid B \\ B &::= \mathit{qbit} \mid B \otimes B \end{aligned}$$

5.1 Subtyping

Let $\prec$ be an order relation over types. This ordering relation expresses the property of linear logic that a duplicable type $!A$ is a subtype of the non-duplicable type A. Moreover subtyping is also induced by the two different products: $A \times B$ is a subtype of $A \otimes B$ because a separable state is a special case of a general state.

$$\frac{}{A \prec A} \qquad \frac{A \prec A' \quad B \prec B'}{A \otimes B \prec A' \otimes B'}$$

$$\frac{A \prec B}{!A \prec B}$$

$$\frac{!A \prec B}{!A \prec !B} \qquad \frac{A \prec A' \quad B \prec B'}{A \times B \prec A' \times B'}$$

$$\frac{A \prec A' \quad B \prec B'}{A' \multimap B \prec A \multimap B'} \qquad \frac{}{A \times B \prec A \otimes B}$$

5.2 Typing Rules

A typing judgment is $[p; R^{\uparrow}; R^{\downarrow}; \Delta] \vdash M : A$, where $[p; R^{\uparrow}; R^{\downarrow}; \Delta]$ is a context, M a term, and A a type.

The context $[p; R^{\uparrow}; R^{\downarrow}; \Delta]$ is composed of:

- a function p which associates with each syntactic position in a term the set of variables below that position. For instance, consider the following term: $\langle\langle a, U_1\ b\rangle, U_2\ \langle c, d\rangle\rangle$. Then $p(\epsilon) = \{a, b, c, d\}$, $p(1) = \{c, d\}, p(01) = \{a\}$.
- an equivalence relation $R^{\uparrow}$ over the quantum variables of $\mathcal{V}$. For the free variables of M, $R^{\uparrow}$ denotes a superset of the entanglement relation of these variables before the reduction of M: for any $x, y \in FV(M)$, if $(x, y) \notin R^{\uparrow}$ then x and y are separable before the reduction of M. For bound variables, $R^{\uparrow}$ depends on the patterns (see abstraction rule).
- an equivalence relation $R^{\downarrow}$ over the quantum variables. For the free variables of M, $R^{\downarrow}$ is an superset of the entanglement relation of these variables after the reduction of M. For any $x, y \in FV(M)$, if $(x, y) \notin R^{\downarrow}$ then x and y are separable after the reduction of M.
- and a set Δ denoted by $\{x_1 : A_1, \ldots, x_n : A_n\}$ where x_i's are variables and A_i's are types.

The context $[p; R^{\uparrow}; R^{\downarrow}; \Delta_1, \Delta_2]$ is a context $[p; R^{\uparrow}; R^{\downarrow}; \Delta]$, where $\Delta = \Delta_1 \cup \Delta_2$ and $\Delta_1 \cap \Delta_2 = \emptyset$. $[p; R^{\uparrow}; R^{\downarrow}; \Delta, x : A]$ means $[p; R^{\uparrow}; R^{\downarrow}; \Delta, \{x : A\}]$. $!\Delta$ contains only variables of duplicable types $(x :!A)$.

In order to illustrate the respective roles of $R^{\uparrow}$ and $R^{\downarrow}$ consider the following examples:

- $[p; \emptyset; \emptyset; \emptyset] \vdash$ **fun** $\langle x \diamond y\rangle \to \langle x, y\rangle : !(A \times B \multimap A \times B)$. Here the pattern $\langle x \diamond y\rangle$ impose that (x, y) is not in $R^{\uparrow}$. Since there is no free variables and no other pattern, $R^{\uparrow} = \emptyset$ is valid.

- $[p;\emptyset;\{(x,y)\}^*;x : qbit, y : qbit] \vdash U_2\ \langle x,y\rangle :\ qbit \otimes qbit$, where R^* means the reflexive and transitive closure of R. Here, since $(x,y) \notin R^{\uparrow}$, the free variables x and y are assumed to be separable, but after the reduction the unitary transformation can create entanglement between x and y, thus $(x,y) \in R^{\downarrow}$.

The typing rules are:

- Axiom :
  If $R^{\uparrow} \subseteq R^{\downarrow}$:
$$\frac{A \prec B}{[\epsilon \to x; R^{\uparrow}; R^{\downarrow}; \Delta, x : A] \vdash x : B}\ ax$$
- Product terms :
  $FV(M)$ is the set of free variables in term M.
  If $\forall x \in FV(M), \forall y \in FV(N), (x,y) \notin R_1^{\downarrow} \cap R_2^{\downarrow}$:
$$\frac{[p_1; R^{\uparrow}; R_1^{\downarrow}; \Gamma_1, !\Delta] \vdash M : A \quad [p_2; R^{\uparrow}; R_2^{\downarrow}; \Gamma_2, !\Delta] \vdash N : B}{[p; R^{\uparrow}; R_1^{\downarrow} \cap R_2^{\downarrow}; \Gamma_1, \Gamma_2, !\Delta] \vdash \langle M, N\rangle : A \times B}\ \times term$$
  where $p(\epsilon) = p_1(\epsilon) \cup p_2(\epsilon)$, $p(0.c) = p_1(c)$, and $p(1.c) = p_2(c)$.
  Otherwise:
$$\frac{[p_1; R^{\uparrow}; R_1^{\downarrow}; \Gamma_1, !\Delta] \vdash M : A \quad [p_2; R^{\uparrow}; R_2^{\downarrow}; \Gamma_2, !\Delta] \vdash N : B}{[p; R^{\uparrow}; (R_1^{\downarrow} \cap R_2^{\downarrow}); \Gamma_1, \Gamma_2, !\Delta] \vdash \langle M, N\rangle : A \otimes B}\ \otimes term$$
  where $p(\epsilon) = p_1(\epsilon) \cup p_2(\epsilon)$, $p(0.c) = p_1(c)$, and $p(1.c) = p_2(c)$.

Remark 5.1 According to the rules $\times term$ and $\otimes term$, the relation "$R^{\downarrow}$" of a pair is obtained by intersecting the relations of both elements of the pair. This construction is illustrated with the following example:

$$\frac{\dfrac{}{[p_1; \{(x,y)\}^*; \{(x,y)\}^*; x : qbit] \vdash x : qbit}\ ax \qquad [p_2; \{(x,y)\}^*; \emptyset; y : qbit] \vdash \textbf{meas}\ y : qbit \times !bit}{[p; \{(x,y)\}^*; \emptyset; x : qbit, y : qbit] \vdash \langle x, \textbf{meas}\ y\rangle : qbit \times (qbit \times !bit)}$$

Even if x and y are initially entangled, the reduction of $\langle x, \textbf{meas}\ y\rangle$ leads to a state where x and y are separable since y is measured. This information of separability is in the right hand side of the tree, thus in order to transmit the information to the pair, intersection of the two relations must be done. In other words, a measurement is a "non local" way of consuming entanglement: acting on only one variable may consume entanglement, whereas creation of entanglement between two variables is "local" because it requires an operation on both variables.

- **if** term:

$$\frac{[p_1; R^{\uparrow}; R_1^{\downarrow}; \Gamma_1, !\Delta] \vdash M : bit \quad [p_2; R_1^{\downarrow}; R_2^{\downarrow}; \Gamma_2, !\Delta] \vdash N_1 : A \quad [p_3; R_1^{\downarrow}; R_3^{\downarrow}; \Gamma_2, !\Delta] \vdash N_0 : A}{[p; R^{\uparrow}; (R_2^{\downarrow} \cup R_3^{\downarrow})^*; \Gamma_1, \Gamma_2, !\Delta] \vdash \textbf{if } (M; N_1; N_0) : A} \; if$$

where $p(c) = p_2(c) \cup p_3(c)$.

Remark 5.2 Here, contrary to the case of product terms, only one of N_1 and N_0 will be reduced. Thus a superset of the entanglement relation of both N_1 and N_0 must be considered, since it is unknown which of them will actually be reduced: $(R_2^{\downarrow} \cup R_3^{\downarrow})^*$ is considered.

- Application:

$$\frac{[p_1; R_1^{\uparrow}; R_1^{\downarrow}; \Gamma_1, !\Delta] \vdash N : A \quad [p_2; R_2^{\uparrow}; R_2^{\downarrow}; \Gamma_1, !\Delta] \vdash \textbf{fun } P \to M : A \multimap B}{[p; R_1^{\uparrow}; subs(R_2^{\downarrow}, p_1, P); \Gamma_1, \Gamma_2, !\Delta] \vdash (\textbf{fun } P \to M)N : B} \multimap app$$

where $subs(R_2^{\downarrow}, p_1, P)$ uses the position function p_1 to replace in $R_2^{\downarrow}$ the variables of the pattern P by the variables of N. The new position function p is also obtained by replacing in p_2 the variables of the pattern P according to p_1.

Moreover, this rule is applicable only if $R_1^{\downarrow} \subseteq subs(R_2^{\uparrow}, p_1, P)$, i.e. entanglement and separability of N are maintained in M.

- Abstraction:

If $FV(M) \cap dom(\Gamma) = \emptyset$:

$$\frac{(P : A)\Box[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta] \vdash M : B}{[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta] \vdash \textbf{fun } P \to M : !(A \multimap B)} \multimap abs$$

Otherwise:

$$\frac{(P : A)\Box[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta] \vdash M : B}{[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta] \vdash \textbf{fun } P \to M : \; A \multimap B} \multimap abs$$

where the operation $\Box$ introduces the variables of a pattern into the context, while verifying that the relation $R^{\uparrow}$ agrees with the structure of the pattern.

$\Box$ is defined as follows:

$$\frac{T\Box[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta, x : A] \vdash M : B}{T\Box(x : A)\Box[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta] \vdash M : B} \Box var$$

where T is recursively defined as $T = \epsilon \mid T\Box(P : A)$.

If $\forall x \in Var(P_1), \forall y \in Var(P_2), (x,y) \in R^{\uparrow}$:

$$\frac{T\Box(P_1 : A)\Box(P_2 : B)\Box[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta,] \vdash M : C}{T\Box(\langle P_1, P_2\rangle : A \otimes B)\Box[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta] \vdash M : C}\ \Box ent$$

If $\forall x \in Var(P_1), \forall y \in Var(P_2), (x,y) \notin R^{\uparrow}$:

$$\frac{T\Box(P_1 : A)\Box(P_2 : B)\Box[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta,] \vdash M : C}{T\Box(\langle P_1 \diamond P_2\rangle : A \times B)\Box[p; R^{\uparrow}; R^{\downarrow}; \Gamma, !\Delta] \vdash M : C}\ \Box sep$$

- Measurement:

$$\frac{[p; R^{\uparrow}; R^{\downarrow}; \Delta] \vdash M : qbit}{[p; R^{\uparrow}; \tilde{R^{\downarrow}}; \Delta] \vdash \mathbf{meas}\ M : qbit \times !bit}\ meas$$

  where $\tilde{R^{\downarrow}} = R^{\downarrow} \setminus \{(x,y) \mid x \in FV(M)\ or\ y \in FV(M)\}$.
- Unitary transformations:

$$\frac{[p; R^{\uparrow}; R^{\downarrow}; \Delta] \vdash M : qbit}{[p; R^{\uparrow}; R^{\downarrow}; \Delta] \vdash U_1\ M : qbit}\ unit1$$

$$\frac{[p; R^{\uparrow}; R^{\downarrow}; \Delta] \vdash M : qbit \otimes qbit}{[p; R^{\uparrow}; \tilde{R^{\downarrow}}; \Delta] \vdash U_2\ M : qbit \otimes qbit}\ unit2$$

  where $\tilde{R^{\downarrow}} = (R^{\downarrow} \cup \{(x,y) \mid x,y \in FV(M)\})^*$
- Initialization:
  If $x \notin dom(!\Delta, \Gamma_1, \Gamma_2)$, and $\forall y \neq x, (x,y) \notin R_2^{\downarrow}$ and $(y,x) \notin R_2^{\downarrow}$:

$$\frac{[p_1; R_1^{\uparrow}; R_1^{\downarrow}; \Gamma_1, !\Delta] \vdash M : bit \quad [p_2; R_2^{\uparrow}; R_2^{\downarrow}; \Gamma_2, !\Delta, x : qbit] \vdash N : A}{[p_2; R_1^{\uparrow}; R_2^{\downarrow}; \Gamma_1, \Gamma_2, !\Delta] \vdash \mathbf{let}\ x = \mathbf{new}\ M\ \mathbf{in}\ N : A}\ new$$

  This rule is applicable only if $R_1^{\downarrow} \subseteq R_2^{\uparrow}$.

Definition 5.3 A program $[Q, M]$ is well-typed of type A iff there exist two equivalence relations $R^{\uparrow}$ and $R^{\downarrow}$ and a position function p such that $[p; R^{\uparrow}; R^{\downarrow}; \Delta] \vdash M : A$, where $\Delta = \{x : qbit | x \in FV(M)\}$. In this case, we write $[Q, M] : A$.

Example 5.4 Consider

$$\mathbf{P} = \mathbf{fun}\ \langle x \diamond y\rangle \rightarrow \langle x, y\rangle,$$

we prove that $[Q,\ \mathbf{P}\ ]$ is well-typed and $[Q,\ \mathbf{P}\ ] :!(A \times B \multimap A \times B)$:

Here, $\Delta = \emptyset$, let $R^{\uparrow} = \emptyset$, the description of p is omitted:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[p; R^{\uparrow}; \emptyset; x : A] \vdash x : A}\, ax \quad \dfrac{}{[p; R^{\uparrow}; \emptyset; y : B] \vdash y : B}\, ax}{[p; R^{\uparrow}; \emptyset; y : B, x : A] \vdash \langle x, y\rangle : A \times B}\, \times term}{(x : A)\Box[p; R^{\uparrow}; \emptyset; y : B] \vdash \langle x, y\rangle : A \times B}\, \Box var}{(x : A)\Box(y : B)\Box[p; R^{\uparrow}; \emptyset; \emptyset] \vdash \langle x, y\rangle : A \times B}\, \Box var}{(\langle x \diamond y\rangle : A \times B)\Box[p; R^{\uparrow}; \emptyset; \emptyset] \vdash \langle x, y\rangle : A \times B}\, \Box sep}{[p; R^{\uparrow}; \emptyset; \emptyset] \vdash \mathbf{fun}\ \langle x \diamond y\rangle \to \langle x, y\rangle : !(A \times B \multimap A \times B)}\, \multimap abs$$

Example 5.5 Consider

$$\mathbf{P}\ =\ \mathbf{fun}\ \langle x, y\rangle \to \langle x, y\rangle,$$

we prove that $[Q,\ \mathbf{P}\ ]$ is well-typed and $[Q,\ \mathbf{P}\ ] :!(A \otimes B \multimap A \otimes B)$:

Here, $\Delta = \emptyset$, let $R^{\uparrow} = \{(x, y)\}^*$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[p; R^{\uparrow}; \{(x,y)\}^*; x : A] \vdash x : A}\, ax \quad \dfrac{}{[p; R^{\uparrow}; \{(x,y)\}^*; y : B] \vdash y : B}\, ax}{[p; R^{\uparrow}; \{(x,y)\}^*; y : B, x : A] \vdash \langle x, y\rangle : A \otimes B}\, \otimes term}{(x : A)\Box[p; R^{\uparrow}; \{(x,y)\}^*; y : B] \vdash \langle x, y\rangle : A \otimes B}\, \Box var}{(x : A)\Box(y : B)\Box[p; R^{\uparrow}; \{(x,y)\}^*; \emptyset] \vdash \langle x, y\rangle : A \otimes B}\, \Box var}{(\langle x, y\rangle : A \otimes B)\Box[p; R^{\uparrow}; \{(x,y)\}^*; \emptyset] \vdash \langle x, y\rangle : A \otimes B}\, \Box ent}{[p; R^{\uparrow}; \{(x,y)\}^*; \emptyset] \vdash \mathbf{fun}\ \langle x, y\rangle \to \langle x, y\rangle : !(A \otimes B \multimap A \otimes B)}\, \multimap abs$$

Example 5.6 Consider

$$\mathbf{P}\ =\ \mathbf{fun}\ \langle x \diamond \langle y, z\rangle\rangle \to \langle U_2\ \langle x, y\rangle, (\ \mathbf{fun}\ \langle a, b\rangle \to a)(\ \mathbf{meas}\ z)\rangle,$$

we prove that $[Q,\ \mathbf{P}\ ]$ is well-typed and $[Q,\ \mathbf{P}\ ] :!(qbit \times qbit \otimes qbit \multimap qbit \otimes qbit \times qbit)$:

Let $\mathbf{Q}\ =\ \mathbf{fun}\ \langle a, b\rangle \to a$ be the first projection.

Here, $\Delta = \emptyset$, let $R^{\uparrow} = \{(y, z)\}^*$:

T_1 :

$$\dfrac{\dfrac{\dfrac{}{[p;R^{\uparrow};\{(y,z)\}^*;x:qbit]\vdash x:qbit}\,ax \quad \dfrac{}{[p;R^{\uparrow};\{(y,z)\}^*;y:qbit]\vdash y:qbit}\,ax}{[p;R^{\uparrow};\{(y,z)\}^*;x:qbit,y:qbit]\vdash \langle x,y\rangle:qbit\otimes qbit}\,\otimes term}{[p;R^{\uparrow};\{(y,z),(x,y)\}^*;x:qbit,y:qbit]\vdash U_2\ \langle x,y\rangle:qbit\otimes qbit}\,unit2$$

T_2 :

$$\dfrac{\dfrac{\dfrac{}{[p;\emptyset;\emptyset;a:qbit,b:!bit]\vdash a:qbit}\,ax}{[p;\emptyset;\emptyset;\emptyset]\vdash \mathbf{Q}\ :qbit\times!bit\multimap qbit}\,abs \quad \dfrac{\dfrac{}{[p;R^{\uparrow};\{(y,z)\}^*;z:qbit]\vdash z:qbit}\,ax}{[p;R^{\uparrow};\emptyset;z:qbit]\vdash \mathbf{meas}\ z:qbit\times!bit}\,meas}{[p;R^{\uparrow};\emptyset;z:qbit]\vdash \mathbf{Q}\ (\mathbf{meas}\ z):qbit}\,app$$

$$\dfrac{\dfrac{T_1 \qquad T_2}{[p;R^{\uparrow};\emptyset;x:qbit,y:qbit,z:qbit]\vdash \langle U_2\ \langle x,y\rangle,\ \mathbf{Q}\ (\mathbf{meas}\ z)\rangle:qbit\otimes qbit\times qbit}\,\times term}{[p;R^{\uparrow};\emptyset;\emptyset]\vdash \mathbf{P}\ :!(qubit\times qbit\otimes qbit\multimap qbit\otimes qbit\times qbit)}\,\multimap abs$$

6 Properties of Quantum Typing

Proofs of the following lemmas are done by structural induction:

Lemma 6.1 *If* $[p;R^{\uparrow},R^{\downarrow},\Delta]\vdash M:A$ *and* $A\prec B$, *then* $[p;R^{\uparrow},R^{\downarrow},\Delta]\vdash M:B$;

Lemma 6.2 *If* $[p;R^{\uparrow},R^{\downarrow},\Delta]\vdash M:A$ *and* $\Gamma\prec\Delta$ *(where* $\prec$ *is naturally extended), then* $[p;R^{\uparrow},R^{\downarrow},\Gamma]\vdash M:A$;

Lemma 6.3 *If* $[p;R^{\uparrow},R_1^{\downarrow},\Delta]\vdash M:A$ *and* $R_1^{\downarrow}\subseteq R_2^{\downarrow}$, *then there exists* $A\prec B$ *such that* $[p;R^{\uparrow},R_2^{\downarrow},\Delta]\vdash M:B$;

Since $R_1^{\downarrow}$ denotes the separability of the variables after the reduction, with a weaker relation $R_2^{\downarrow}$ ($R_1^{\downarrow}\subseteq R_2^{\downarrow}$) the term is still typable but with a weaker type, as it is illustrated in Lemma 6.3.

Lemma 6.4 *If* $[p;R_1^{\uparrow},R^{\downarrow},\Delta]\vdash M:A$ *and* $R_2^{\uparrow}\subseteq R_1^{\uparrow}$ *where* $R_1^{\uparrow}$ *and* $R_2^{\uparrow}$ *differs only on free variables of* M, *then* $[p;R_2^{\uparrow},R^{\downarrow},\Delta]\vdash M:A$;

For the free variables of M, $R_1^{\uparrow}$ denotes the separability before the reduction, thus with a stronger relation $R_2^{\uparrow}$ ($R_2^{\uparrow}\subseteq R_1^{\uparrow}$), M is still typable.

Conjecture 6.5 (Substitution) *If* $[p;R^{\uparrow};R_1^{\downarrow};\Gamma_1,!\Delta,x:A]\vdash M:B$ *and* $[p;R^{\uparrow};R_2^{\downarrow};\Gamma_2,!\Delta]\vdash V:A$, *then* $[p;R^{\uparrow};R_1^{\downarrow}\cap R_2^{\downarrow};\Gamma_1,\Gamma_2,!\Delta]\vdash M[V/x]:B$.

The substitution property is not proved, as a consequence *subject reduction* and *progress* are also conjectured but not proved.

7 Toward quantum types for quantum data structures

The fact that quantum data can be entangled or separable requires additional specification that is not present in the usual notion of data structures as proposed in [13,15]. An abstracted specification of the entanglement structure which is present or allowed within quantum data is clearly part of the information that has to be conveyed with notions of quantum data structures and quantum data types. An example given in this paper is the quantum data structure used by teleportation. This is achieved in this paper by the distinction between the two kinds of patterns $\langle P_1, P_1 \rangle$ and $\langle P_1 \diamond P_2 \rangle$ and between the types $A \otimes B$ and $A \times B$.

A more elaborate example would be the data used during the execution of the One Way Quantum Computer [11]: a rectangular array of qubits, where all qubits are initially entangled with their neighbours, and where the entanglement within this quantum data structure is consumed in a stepwise manner by successive 1-qubit measurements. An interesting special case, which would at least fit the One Way Quantum Computer, is the family of quantum data structures corresponding to graph states [4,7]. This will be studied further in the scope of this work in progress, with the associated notion of types.

Acknowledgements

I would like to thank Philippe Jorrand and Pablo Arrighi for discussions and comments.

References

[1] S. Bettelli, T. Calarco and L. Serafini, *Toward an architecture for quantum programming*, Eur. Phys. J. D, Vol. 25, No. 2, pp. 181-200 (2003)

[2] V. Danos, E. Kashefi, P. Panangaden *The Measurement Calculus*, e-print arXiv:quant-ph/0412135, 2004.

[3] S. J. Gay and R. Nagarajan *Communicating quantum processes*, Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Jens Palsberg and Martin Abadi (Eds.), POPL 2005, January 12–14, 2005, pp. 145–157.

[4] M. Hein, J. Eisert and J. H. Briegel, *Multi-party entanglement in graph states*, Phys. Rev. A 69, 062311, 2004.

[5] E. Knill, *Conventions for Quantum Pseudocode*, LANL report LAUR-96-2724, 1996.

[6] M. Lalire and Ph. Jorrand, *A process algebraic approach to concurrent and distributed quantum computation: operational semantics*, Proceedings of the 2nd International Workshop on Quantum Programming Languages, 2004, pp. 109–126.

[7] M. Mhalla and S. Perdrix, *Complexity of Graph State Preparation*, arXiv:quant-ph/0412071, 2004.

[8] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

[9] S. Perdrix and Ph. Jorrand, *Classically-Controlled Quantum Computation*, arXiv:quant-ph/0407008, to appear in Mathematical Structures in Computer Science.

[10] S. Perdrix. *State Transfer instead of Teleportation in Measurement-based Quantum Computation*, International Journal of Quantum Information, 3(1):219-224, 2005.

[11] R. Raussendorf and H. J. Briegel, *A One-Way Quantum Computer*, Phys. Rev. Lett. 86, 5188–5191, 2001.

[12] J.W. Sanders and P. Zuliani: *Quantum Programming, Mathematics of Program Construction*, Springer LNCS 1837, 80–99, 2000.

[13] P. Selinger, *Towards a Quantum Programming Language*, Mathematical Structures in Computer Science, 14(4):527–586, 2004.

[14] P. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM Journal of Computing 26, pp. 1484–1509, 1997.

[15] B. Valiron, *Quantum Typing*, Proceedings of the 2nd International Workshop on Quantum Programming Languages, 2004 pp. 163–178.