

A vectorial type system

<work-in-progress>

Alejandro Díaz-Caro

joint work with Pablo Arrighi and Benoît Valiron

Université de Grenoble
Laboratoire d'Informatique de Grenoble

Types 2010 • October 13th, 2010 • Warsaw

Context

- ▶ Oddity of Quantum theory \implies Quantum Logic?¹ (developed *ad hoc* before quantum computing, no clear relation with quantum programs).

¹[Birkhoff, von Neumann 1936]

Context

- ▶ Oddity of Quantum theory \implies Quantum Logic?¹ (developed *ad hoc* before quantum computing, no clear relation with quantum programs).
- ▶ Curry-Howard : (programs,types) \implies (proofs,logics).
Quantum Computation : (quantum programs, quantum types).
CH \wedge QC : (quantum th. proofs, quantum th. logics)?
Quantum logics : isolating the reasoning behind quantum algorithms?

¹[Birkhoff, von Neumann 1936]

Context

- ▶ Oddity of Quantum theory \implies Quantum Logic?¹ (developed *ad hoc* before quantum computing, no clear relation with quantum programs).
- ▶ Curry-Howard : (programs,types) \implies (proofs,logics).
Quantum Computation : (quantum programs, quantum types).
CH \wedge QC : (quantum th. proofs, quantum th. logics)?
Quantum logics : isolating the reasoning behind quantum algorithms?

What are quantum types?

¹[Birkhoff, von Neumann 1936]

Linear [Arrighi, Dowek 2008]

Higher-order computation

$\mathbf{t, r, s} ::= x \mid \lambda x. \mathbf{t} \mid (\mathbf{t}) \mathbf{r}$ |

Linear [Arrighi, Dowek 2008]

Higher-order computation

$\mathbf{t}, \mathbf{r}, \mathbf{s} ::= x \mid \lambda x. \mathbf{t} \mid (\mathbf{t}) \mathbf{r}$

Linear algebra

$\mathbf{t} + \mathbf{r} \mid \alpha. \mathbf{t} \mid \mathbf{0}$

Linear [Arrighi, Dowek 2008]

Higher-order computation

$\mathbf{t}, \mathbf{r}, \mathbf{s} ::= x \mid \lambda x. \mathbf{t} \mid (\mathbf{t}) \mathbf{r}$

▶ $(\lambda x. \mathbf{t}) \mathbf{b} \rightarrow \mathbf{t}[\mathbf{b}/x]$

\mathbf{b} an abstraction or a variable.

Linear algebra

$\mathbf{t} + \mathbf{r} \mid \alpha. \mathbf{t} \mid \mathbf{0}$

Higher-order computation

$\mathbf{t}, \mathbf{r}, \mathbf{s} ::= x \mid \lambda x. \mathbf{t} \mid (\mathbf{t}) \mathbf{r}$

▶ $(\lambda x. \mathbf{t}) \mathbf{b} \rightarrow \mathbf{t}[\mathbf{b}/x]$

\mathbf{b} an abstraction or a variable.

Linear algebra

$\mathbf{t} + \mathbf{r} \mid \alpha. \mathbf{t} \mid \mathbf{0}$

- ▶ Elementary rules such as $\mathbf{u} + \mathbf{0} \rightarrow \mathbf{u}$ and $\alpha.(\mathbf{u} + \mathbf{v}) \rightarrow \alpha. \mathbf{u} + \alpha. \mathbf{v}$.
- ▶ Factorisation rules such as $\alpha. \mathbf{u} + \beta. \mathbf{u} \rightarrow (\alpha + \beta). \mathbf{u}$.
- ▶ Application rules such as $\mathbf{u} (\mathbf{v} + \mathbf{w}) \rightarrow (\mathbf{u} \mathbf{v}) + (\mathbf{u} \mathbf{w})$.

Vectorial type system

Grammar

Types grammar:

$$T, R, S := U \mid T + R \mid \alpha.T \quad (\text{no zero}) \quad \text{general types}$$

$$U, V, W := X \mid U \rightarrow T \mid \forall X.U \quad \text{unit types}$$

where $\alpha \in \mathcal{S}$ and $(\mathcal{S}, +, \times)$ is commutative ring.

Equivalences:

$$1. T \equiv T$$

$$\alpha.(\beta.T) \equiv (\alpha \times \beta).T$$

$$\alpha.T + \alpha.R \equiv \alpha.(T + R)$$

$$\alpha.T + \beta.T \equiv (\alpha + \beta).T$$

$$T + R \equiv R + T$$

$$T + (R + S) \equiv (T + R) + S$$

Vectorial type system

Not **one** zero... a lot of them

$$\frac{\Gamma \vdash \lambda x.x : U \rightarrow U \quad \frac{\Gamma \vdash \mathbf{r} : R \quad \Gamma \vdash -\mathbf{r} : -R}{\Gamma \vdash \mathbf{r} - \mathbf{r} : \bar{0}}}{\Gamma \vdash (\lambda x.x + \mathbf{r} - \mathbf{r}) : U \rightarrow U} \quad \frac{}{\Gamma \vdash \mathbf{t} : U}}{\Gamma \vdash (\lambda x.x + \mathbf{r} - \mathbf{r}) \mathbf{t} : U}$$

Vectorial type system

Not **one** zero... a lot of them

$$\frac{\Gamma \vdash \lambda x.x : U \rightarrow U \quad \frac{\Gamma \vdash \mathbf{r} : R \quad \Gamma \vdash -\mathbf{r} : -R}{\Gamma \vdash \mathbf{r} - \mathbf{r} : \bar{0}}}{\Gamma \vdash (\lambda x.x + \mathbf{r} - \mathbf{r}) : U \rightarrow U} \quad \frac{}{\Gamma \vdash \mathbf{t} : U}}{\Gamma \vdash (\lambda x.x + \mathbf{r} - \mathbf{r}) \mathbf{t} : U}$$

However, $(\lambda x.x + \mathbf{r} - \mathbf{r}) \mathbf{t} \rightarrow (\lambda x.x) \mathbf{t} + (\mathbf{r} \mathbf{t}) - (\mathbf{r} \mathbf{t})$.

Vectorial type system

Not **one** zero... a lot of them

$$\frac{\Gamma \vdash \lambda x.x : U \rightarrow U \quad \frac{\Gamma \vdash \mathbf{r} : R \quad \Gamma \vdash -\mathbf{r} : -R}{\Gamma \vdash \mathbf{r} - \mathbf{r} : \bar{0}}}{\Gamma \vdash (\lambda x.x + \mathbf{r} - \mathbf{r}) : U \rightarrow U} \quad \frac{}{\Gamma \vdash \mathbf{t} : U}}{\Gamma \vdash (\lambda x.x + \mathbf{r} - \mathbf{r}) \mathbf{t} : U}$$

However, $(\lambda x.x + \mathbf{r} - \mathbf{r}) \mathbf{t} \rightarrow (\lambda x.x) \mathbf{t} + (\mathbf{r} \mathbf{t}) - (\mathbf{r} \mathbf{t})$.

We need to keep track of the zeros... where they came from!

Vectorial type system

Not **one** zero... a lot of them

$$\frac{\Gamma \vdash \lambda x.x : U \rightarrow U \quad \frac{\Gamma \vdash \mathbf{r} : R \quad \Gamma \vdash -\mathbf{r} : -R}{\Gamma \vdash \mathbf{r} - \mathbf{r} : \bar{0}}}{\Gamma \vdash (\lambda x.x + \mathbf{r} - \mathbf{r}) : U \rightarrow U} \quad \frac{}{\Gamma \vdash \mathbf{t} : U}}{\Gamma \vdash (\lambda x.x + \mathbf{r} - \mathbf{r}) \mathbf{t} : U}$$

However, $(\lambda x.x + \mathbf{r} - \mathbf{r}) \mathbf{t} \rightarrow (\lambda x.x) \mathbf{t} + (\mathbf{r} \mathbf{t}) - (\mathbf{r} \mathbf{t})$.

We need to keep track of the zeros... where they came from!

Instead of $\bar{0}$, we have $0.R$ and $T + 0.R \neq T$

Vectorial type system

The factorisation rule problem

$$\frac{\Gamma \vdash \mathbf{t}: T \quad \Gamma \vdash \mathbf{t}: T'}{\Gamma \vdash \alpha.\mathbf{t} + \beta.\mathbf{t}: \alpha.T + \beta.T'}$$

- ▶ However, $\alpha.\mathbf{t} + \beta.\mathbf{t} \rightarrow (\alpha + \beta).\mathbf{t}$
- ▶ In general $\alpha.T + \beta.T' \neq (\alpha + \beta).T \neq (\alpha + \beta).T'$

Vectorial type system

The factorisation rule problem

Several possible solutions:

- ▶ Remove factorisation rule (Done. SR and SN both work)
 - ▶ $+$ in scalars not used anymore. Scalars \Rightarrow Monoid
 - ▶ It works!... but it is no so expressive, no so useful

Vectorial type system

The factorisation rule problem

Several possible solutions:

- ▶ Remove factorisation rule (Done. SR and SN both work)
 - ▶ $+$ in scalars not used anymore. Scalars \Rightarrow Monoid
 - ▶ It works!... but it is not so expressive, not so useful
- ▶ Add several typing rules to allow typing $(\alpha + \beta).u$ with $\alpha.A + \beta.B$
 - ▶ As soon as we add one, we have to add many to make it work
 - ▶ Too complex and inelegant

Vectorial type system

The factorisation rule problem

Several possible solutions:

- ▶ Remove factorisation rule (Done. SR and SN both work)
 - ▶ $+$ in scalars not used anymore. Scalars \Rightarrow Monoid
 - ▶ It works!... but it is not so expressive, not so useful
- ▶ Add several typing rules to allow typing $(\alpha + \beta).u$ with $\alpha.A + \beta.B$
 - ▶ As soon as we add one, we have to add many to make it work
 - ▶ Too complex and inelegant
- ▶ Church style (work-in-progress)
 - ▶ Seems to be the natural solution
 - ▶ Big complexity with polymorphism and distributivity

Vectorial type system

The factorisation rule problem

Several possible solutions:

- ▶ Remove factorisation rule (Done. SR and SN both work)
 - ▶ $+$ in scalars not used anymore. Scalars \Rightarrow Monoid
 - ▶ It works!... but it is no so expressive, no so useful
- ▶ Add several typing rules to allow typing $(\alpha + \beta).\mathbf{u}$ with $\alpha.A + \beta.B$
 - ▶ As soon as we add one, we have to add many to make it work
 - ▶ Too complex and inelegant
- ▶ **Church style** (work-in-progress)
 - ▶ Seems to be the natural solution
 - ▶ Big complexity with polymorphism and distributivity
- ▶ **Weaker subject reduction** (this presentation)

Vectorial type system

Typing rules

$$\frac{}{\Gamma, x: U \vdash x: U} \text{ax} \quad \frac{\Gamma \vdash \mathbf{t}: T}{\Gamma \vdash \mathbf{0}: \mathbf{0}.T} \text{0}_I \quad \frac{\Gamma \vdash \mathbf{t}: T}{\Gamma \vdash \alpha.\mathbf{t}: \alpha.T} \alpha_I$$
$$\frac{\Gamma \vdash \mathbf{t}: \sum_{i=1}^n \alpha_i. \forall \vec{X}. (U \rightarrow T_i) \quad \Gamma \vdash \mathbf{r}: \sum_{j=1}^m \beta_j. V_j \quad \forall V_j, \exists \vec{W} / U[\vec{W}/\vec{X}] = V_j}{\Gamma \vdash (\mathbf{t}) \mathbf{r}: \sum_{i=1}^n \sum_{j=1}^m \alpha_i \times \beta_j. T_i[\vec{W}/\vec{X}]} \rightarrow_E$$

$$\frac{\Gamma, x: U \vdash \mathbf{t}: T}{\Gamma \vdash \lambda x. \mathbf{t}: U \rightarrow T} \rightarrow_I \quad \frac{\Gamma \vdash \mathbf{t}: T \quad \Gamma \vdash \mathbf{r}: R}{\Gamma \vdash \mathbf{t} + \mathbf{r}: T + R} +_I$$

$$\frac{\Gamma \vdash \mathbf{t}: U \quad x \notin FV(\Gamma)}{\Gamma \vdash \mathbf{t}: \forall X. U} \forall_I \quad \frac{\Gamma \vdash \mathbf{t}: \forall X. U}{\Gamma \vdash \mathbf{t}: U[V/X]} \forall_E$$

Vectorial type system

A weaker subject reduction

$$(\alpha + \beta).T \leq \alpha.T + \beta.T'$$

(and its contextual closure)

Vectorial type system

A weaker subject reduction

$$(\alpha + \beta).T \leq \alpha.T + \beta.T'$$

(and its contextual closure)

Theorem (Weak subject reduction)

$$\left. \begin{array}{l} \mathbf{t} \rightarrow_R \mathbf{s} \\ \Gamma \vdash \mathbf{t} : T \end{array} \right\} \exists S \leq T / \Gamma \vdash \mathbf{s} : S$$

Moreover, if $R \notin \text{Group } F \Rightarrow S \equiv T$.

▶ Extra: orthogonality

▶ Conclusion

Orthogonality

Why

Now, say we would like normalised terms

$$\alpha \cdot \mathbf{t} + \beta \cdot \mathbf{s}$$

with $|\alpha|^2 + |\beta|^2 = 1$.

Orthogonality

Why

Now, say we would like normalised terms

$$\alpha.\mathbf{t} + \beta.\mathbf{s}$$

with $|\alpha|^2 + |\beta|^2 = 1$.

So, we want to allow $\frac{1}{\sqrt{2}}.\mathbf{t} + \frac{1}{\sqrt{2}}.\mathbf{s}$ and disallow $\frac{1}{2}.\mathbf{t} + \frac{1}{2}.\mathbf{s}$

Orthogonality

Why

Now, say we would like normalised terms

$$\alpha.\mathbf{t} + \beta.\mathbf{s}$$

with $|\alpha|^2 + |\beta|^2 = 1$.

So, we want to allow $\frac{1}{\sqrt{2}}.\mathbf{t} + \frac{1}{\sqrt{2}}.\mathbf{s}$ and disallow $\frac{1}{2}.\mathbf{t} + \frac{1}{2}.\mathbf{s}$

Or do we? What happens if $\mathbf{t} \rightarrow^* \mathbf{s}$?

Orthogonality

Why

Now, say we would like normalised terms

$$\alpha \cdot \mathbf{t} + \beta \cdot \mathbf{s}$$

with $|\alpha|^2 + |\beta|^2 = 1$.

So, we want to allow $\frac{1}{\sqrt{2}} \cdot \mathbf{t} + \frac{1}{\sqrt{2}} \cdot \mathbf{s}$ and disallow $\frac{1}{2} \cdot \mathbf{t} + \frac{1}{2} \cdot \mathbf{s}$

Or do we? What happens if $\mathbf{t} \rightarrow^* \mathbf{s}$?

We need a notion of orthogonality between terms (and also a way to check it).

Orthogonality

Inner product

Definition (Terms inner product)

Let $\mathbf{t}, \mathbf{s}, \mathbf{r}$ be any term, and let \mathbf{u}, \mathbf{v} be terms in normal form which are not sum of terms nor a scalar times a term. We define the function 'delta' as follows:

$$\delta_{\mathbf{t}, \mathbf{s}} = \begin{cases} 0 & \text{if } \mathbf{t} \neq \mathbf{s} \vee \mathbf{t} = \mathbf{0} \vee \mathbf{s} = \mathbf{0} \\ 1 & \text{in any other case} \end{cases}$$

Then, we define the inner product between terms recursively by

$$\langle \mathbf{u} | \mathbf{v} \rangle = \overline{\langle \mathbf{v} | \mathbf{u} \rangle} = \delta_{\mathbf{u}, \mathbf{v}}$$

$$\langle \mathbf{t} | \mathbf{s} \rangle = \overline{\langle \mathbf{s} | \mathbf{t} \rangle} = \langle \mathbf{t} \downarrow | \mathbf{s} \downarrow \rangle$$

$$\langle \alpha \cdot \mathbf{t} + \beta \cdot \mathbf{s} | \mathbf{r} \rangle = \alpha \times \langle \mathbf{t} | \mathbf{r} \rangle + \beta \times \langle \mathbf{s} | \mathbf{r} \rangle$$

Orthogonality

Inner product

Definition (Terms inner product)

Let $\mathbf{t}, \mathbf{s}, \mathbf{r}$ be any term, and let \mathbf{u}, \mathbf{v} be terms in normal form which are not sum of terms nor a scalar times a term. We define the function 'delta' as follows:

$$\delta_{\mathbf{t}, \mathbf{s}} = \begin{cases} 0 & \text{if } \mathbf{t} \neq \mathbf{s} \vee \mathbf{t} = \mathbf{0} \vee \mathbf{s} = \mathbf{0} \\ 1 & \text{in any other case} \end{cases}$$

Then, we define the inner product between terms recursively by

$$\langle \mathbf{u} | \mathbf{v} \rangle = \langle \mathbf{v} | \mathbf{u} \rangle = \delta_{\mathbf{u}, \mathbf{v}}$$

$$\langle \mathbf{t} | \mathbf{s} \rangle = \langle \mathbf{s} | \mathbf{t} \rangle = \langle \mathbf{t} \downarrow | \mathbf{s} \downarrow \rangle$$

$$\langle \alpha \cdot \mathbf{t} + \beta \cdot \mathbf{s} | \mathbf{r} \rangle = \alpha \times \langle \mathbf{t} | \mathbf{r} \rangle + \beta \times \langle \mathbf{s} | \mathbf{r} \rangle$$

Examples

$$\langle (\lambda x \ x) \ \mathbf{v} | \mathbf{v} \rangle = 1$$

Orthogonality

Inner product

Definition (Terms inner product)

Let $\mathbf{t}, \mathbf{s}, \mathbf{r}$ be any term, and let \mathbf{u}, \mathbf{v} be terms in normal form which are not sum of terms nor a scalar times a term. We define the function 'delta' as follows:

$$\delta_{\mathbf{t}, \mathbf{s}} = \begin{cases} 0 & \text{if } \mathbf{t} \neq \mathbf{s} \vee \mathbf{t} = \mathbf{0} \vee \mathbf{s} = \mathbf{0} \\ 1 & \text{in any other case} \end{cases}$$

Then, we define the inner product between terms recursively by

$$\langle \mathbf{u} | \mathbf{v} \rangle = \langle \mathbf{v} | \mathbf{u} \rangle = \delta_{\mathbf{u}, \mathbf{v}}$$

$$\langle \mathbf{t} | \mathbf{s} \rangle = \langle \mathbf{s} | \mathbf{t} \rangle = \langle \mathbf{t} \downarrow | \mathbf{s} \downarrow \rangle$$

$$\langle \alpha \cdot \mathbf{t} + \beta \cdot \mathbf{s} | \mathbf{r} \rangle = \alpha \times \langle \mathbf{t} | \mathbf{r} \rangle + \beta \times \langle \mathbf{s} | \mathbf{r} \rangle$$

Examples

$$\langle (\lambda x \ x) \ \mathbf{v} | \mathbf{v} \rangle = 1$$

$$\langle \mathbf{true} | \mathbf{false} \rangle = 0$$

Orthogonality

Inner product

Definition (Terms inner product)

Let $\mathbf{t}, \mathbf{s}, \mathbf{r}$ be any term, and let \mathbf{u}, \mathbf{v} be terms in normal form which are not sum of terms nor a scalar times a term. We define the function 'delta' as follows:

$$\delta_{\mathbf{t}, \mathbf{s}} = \begin{cases} 0 & \text{if } \mathbf{t} \neq \mathbf{s} \vee \mathbf{t} = \mathbf{0} \vee \mathbf{s} = \mathbf{0} \\ 1 & \text{in any other case} \end{cases}$$

Then, we define the inner product between terms recursively by

$$\langle \mathbf{u} | \mathbf{v} \rangle = \langle \mathbf{v} | \mathbf{u} \rangle = \delta_{\mathbf{u}, \mathbf{v}}$$

$$\langle \mathbf{t} | \mathbf{s} \rangle = \langle \mathbf{s} | \mathbf{t} \rangle = \langle \mathbf{t} \downarrow | \mathbf{s} \downarrow \rangle$$

$$\langle \alpha \cdot \mathbf{t} + \beta \cdot \mathbf{s} | \mathbf{r} \rangle = \alpha \times \langle \mathbf{t} | \mathbf{r} \rangle + \beta \times \langle \mathbf{s} | \mathbf{r} \rangle$$

Examples

$$\langle (\lambda x x) \mathbf{v} | \mathbf{v} \rangle = 1$$

$$\langle \mathbf{true} | \mathbf{false} \rangle = 0$$

$$\left| \begin{aligned} & \langle \frac{1}{\sqrt{2}} \cdot \mathbf{true} + \frac{1}{\sqrt{2}} \cdot \mathbf{false} | \mathbf{true} \rangle = \\ & \frac{1}{\sqrt{2}} \times \langle \mathbf{true} | \mathbf{true} \rangle + \frac{1}{\sqrt{2}} \times \langle \mathbf{false} | \mathbf{true} \rangle = \frac{1}{\sqrt{2}} \end{aligned} \right.$$

Orthogonality

How to check it

Let $\Gamma \vdash \mathbf{u} : T$ and $\Gamma \vdash \mathbf{v} : S$.

Does $\mathbf{u} \perp \mathbf{v} \Leftrightarrow T \perp S$?

(for some definition of orthogonality between types)

Orthogonality

How to check it

Let $\Gamma \vdash \mathbf{u} : T$ and $\Gamma \vdash \mathbf{v} : S$.

Does $\mathbf{u} \perp \mathbf{v} \Leftrightarrow T \perp S$?

(for some definition of orthogonality between types)

► $\mathbf{u} \perp \mathbf{v} \not\Rightarrow T \perp S$

Orthogonality

How to check it

Let $\Gamma \vdash \mathbf{u} : T$ and $\Gamma \vdash \mathbf{v} : S$.

Does $\mathbf{u} \perp \mathbf{v} \Leftrightarrow T \perp S$?

(for some definition of orthogonality between types)

► $\mathbf{u} \perp \mathbf{v} \not\Leftrightarrow T \perp S$

Counterexample

$\vdash \text{true} : \text{Bool}$

$\vdash \text{false} : \text{Bool}$

Orthogonality

How to check it

Let $\Gamma \vdash \mathbf{u} : T$ and $\Gamma \vdash \mathbf{v} : S$.

Does $\mathbf{u} \perp \mathbf{v} \Leftrightarrow T \perp S$?

(for some definition of orthogonality between types)

▶ $\mathbf{u} \perp \mathbf{v} \not\Leftrightarrow T \perp S$

▶ $T \perp S \not\Leftrightarrow \mathbf{u} \perp \mathbf{v}$

Counterexample

$\vdash \text{true} : \text{Bool}$

$\vdash \text{false} : \text{Bool}$

Orthogonality

How to check it

Let $\Gamma \vdash \mathbf{u} : T$ and $\Gamma \vdash \mathbf{v} : S$.

Does $\mathbf{u} \perp \mathbf{v} \Leftrightarrow T \perp S$?

(for some definition of orthogonality between types)

▶ $\mathbf{u} \perp \mathbf{v} \not\Leftrightarrow T \perp S$

▶ $T \perp S \not\Leftrightarrow \mathbf{u} \perp \mathbf{v}$

Counterexample

$\vdash \text{true} : \text{Bool}$

$\vdash \text{false} : \text{Bool}$

Counterexample

$y : W \vdash \lambda x. y : U \rightarrow W$

$y : W \vdash \lambda x. y : V \rightarrow W$

Orthogonality

How to check it

Let $\Gamma \vdash \mathbf{u} : T$ and $\Gamma \vdash \mathbf{v} : S$.

Does $\mathbf{u} \perp \mathbf{v} \Leftrightarrow T \perp S$?

(for some definition of orthogonality between types)

▶ $\mathbf{u} \perp \mathbf{v} \not\Rightarrow T \perp S$

▶ $T \perp S \not\Rightarrow \mathbf{u} \perp \mathbf{v}$

Counterexample

$\vdash \text{true} : \text{Bool}$

$\vdash \text{false} : \text{Bool}$

Counterexample

$y : W \vdash \lambda x. y : U \rightarrow W$

$y : W \vdash \lambda x. y : V \rightarrow W$

Workaround

▶ Church-style $\Rightarrow T \perp S \Rightarrow \mathbf{u} \perp \mathbf{v}$

Work-in-progress

Aim: to find a quantum logic from a quantum type system

- ▶ *Vectorial* type system, with a weaker SR ✓
- ▶ Church-version (work-in-progress)
- ▶ May lead to a *quantum computational logic*