The quest for duplicability

Benoît Valiron

Setting: Quantum computation with classical control

* two units of data: bit (classical)
qubit (quantum).

* one qubit: has a state, represented by a normalized vector in $\mathcal{H} = \mathbb{C}^2$.

n qubits: have a state in $\mathcal{H}^{\otimes n}$.

Corollary: non-locality of data:

$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ is the state of 2 qubits,

living in $\mathcal{H} \otimes \mathcal{H}$

qubit 1 is here

qubit 2 is here.

but one cannot separate them → entangled states.

* Operations on qubits:
    - unitary maps on the states → reversible
    - measurements:

$$\begin{tikzcd} 0 & \alpha|0\rangle + \beta|1\rangle \arrow[dl, "\text{prob. } |\alpha|^2"'] \arrow[dr, "\text{prob. } |\beta|^2"] & 1 \\ |0\rangle & & |1\rangle \end{tikzcd}$$

→ destructive operation

→ probabilistic

* Thus : non-duplicability: no map

$$\alpha|0\rangle + \beta|1\rangle \longmapsto (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle)$$

* Want classical control: tests, abstraction, etc

Basic Framework:

* want to be in a sym mon. category:
  $(\mathcal{E}, \otimes, I)$.
* want to be able to assert "this obj is duplicable"
  → want a Functor $!: \mathcal{E} \to \mathcal{E}$.

a duplicable object does not need to be duplicated:

→ want a map $!A \xrightarrow{\varepsilon_A} A$

a duplicable object <u>is</u> duplicable:

→ want an iso $!A \xrightarrow{\delta_A} !!A$

⇝ we get a idempotent comonad $(!, \varepsilon, \delta)$

Data:

$$\left[\begin{array}{l} (\mathcal{E}, \otimes, I) \text{ sym. mon. cat} \\ (!, \delta, \varepsilon) \text{ idemp. comonad} \end{array}\right] \quad \text{i.e.} \quad \left[\begin{array}{l} \text{Reflexive adjunction:} \\ \mathcal{E}^c \underset{\perp}{\rightleftarrows} (\mathcal{E}, \otimes, I) \end{array}\right]$$

Need some more:

A pair of duplicable obj is duplicable:

$$!A \otimes !B \xrightarrow{d^!_{A,B}} !(A \otimes B)$$

$$I \xrightarrow{d^!_I} !I \qquad \text{(For symmetry)}.$$

These are not isos: if you can have as many pairs of socks as you want, you can't just throw the right ones away: we don't have weakening.

Updated Data:

$$\left[\begin{array}{l} (\mathcal{C}, \otimes, I) \text{ sym. mon. cat} \\ (!, \delta, \varepsilon, d^!, d^!) \text{ monoidal} \\ \qquad \text{idempotent comonad} \end{array}\right] \text{ i.e. } \left[\begin{array}{l} (\mathcal{C}^L, \otimes, I) \text{ sym. mon.} \\ (\mathcal{C}^L, \otimes, I) \underset{\perp}{\rightleftarrows} (\mathcal{C}, \otimes, I) \\ \text{sym. mon. refl. adjunction} \end{array}\right]$$

It's not enough:

we said "duplication", we need maps of duplication:

$$\Delta_A : !A \longrightarrow !A \otimes !A$$

$$\Diamond_A : !A \longrightarrow T \qquad \text{(for symmetry)}.$$

together with nice enough structure:

* $(!A, \Delta_A, \Diamond_A)$ commutative comonoid
* Δ_A and $\Diamond_A$ need to be monoidal NT's
* they need to be !-coalgebra morphisms
* δ_A need to be comonoid morphism.

Updated data:

$(\mathcal{C}, \otimes, I)$ sym. mon. cat
$(!, \delta, \varepsilon, d, d', \Diamond, \Delta)$ idemp. linear exponential comonad

i.e.

$(\mathcal{M}, \times, T)$ cart. cat.
$(\mathcal{C}, \otimes, I)$ sym. mon. cat
$(\mathcal{M}, \times, T) \underset{\perp}{\rightleftarrows} (\mathcal{C}, \otimes, I)$ refl. monoidal adjunction

Modulo the idempotency, we have a model for intuitionistic linear logic (see Benton, Bierman, Schalk, ...).

But it is not enough: the measurement introducing side effects, we need more and we add a computational monad (T, μ, η), following Moggi's framework for computations.

Final categorical setting:

cart. cat — sym. mon. cat — Kleisli category coming from T.

$$\begin{tikzcd} (\mathcal{C}, \times, \top) \arrow[r, bend left] \arrow[r, phantom, "\perp"] & (\mathcal{E}, \otimes, I) \arrow[l, bend left] \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathcal{E}_T \arrow[l, bend left] \end{tikzcd}$$

here we find computations

here we find values.

here we find duplicable (classical) values.

Conclusion:

* There's more to this: we want higher-order, recursion and additive types.
* The commutative diagrams need to be matched with axiomatic rules of terms manipulation.
* This dry set of rules need a decent model.