Quantum Registers

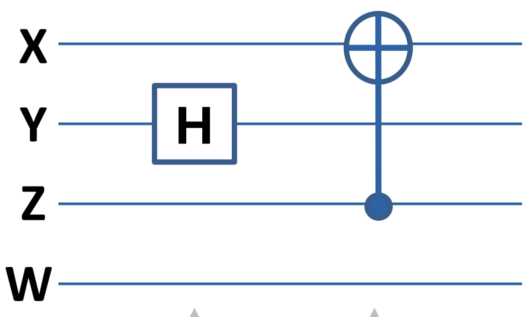
Dominique Unruh

University of Tartu, Estonia

Overview

- Why quantum registers make me sad
- How I got happy again

A tiny quantum program



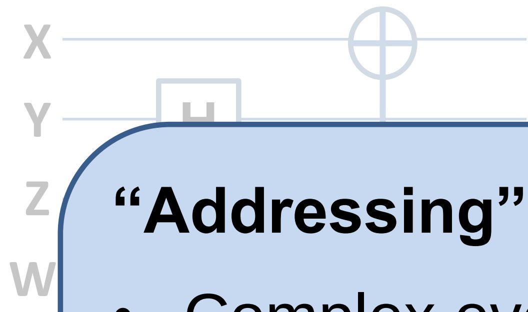

```
apply H on Y;
apply CNOT on Z,X;
```

$$I_1 \otimes H \otimes I_2$$

$$(I_1 \otimes U_\alpha) \cdot U_\alpha \cdot \big((I_1 \otimes U_\sigma) \otimes I_1\big) \cdot (U_\alpha \otimes I_1) \cdot \big((U_\sigma \otimes I_1) \otimes I_1\big) \cdot \big((CNOT \otimes I_1) \otimes I_1\big) \cdot \big((U_\sigma \otimes I_1) \otimes I_1\big) \cdot \big(U_\alpha^\dagger \otimes I_1\big) \cdot \big((I_1 \otimes U_\sigma) \otimes I_1\big) \cdot U_\alpha^\dagger \cdot \big(I_1 \otimes U_\alpha^\dagger\big)$$

- $U_\sigma|x, y\rangle = |y, x\rangle$
- $\mathcal{H} \otimes (\mathcal{K} \otimes \mathcal{L}) \neq (\mathcal{H} \otimes \mathcal{K}) \otimes \mathcal{L}$
- $U_\alpha|(x, y), z\rangle = |x, (y, z)\rangle$

A tiny quantum program



apply H on Y;
apply $CNOT$ on Z,X;

“Addressing” registers:

- Complex even in tiniest programs
- More like compiler output than abstract semantics
- Could be generated by computer, but: Need semantics of “*on register X*” first

- $U_\sigma|x,y\rangle = |y,x\rangle$
- $\mathcal{H} \otimes (\mathcal{K} \otimes \mathcal{L}) \neq (\mathcal{H} \otimes \mathcal{K}) \otimes \mathcal{L}$
- $U_\alpha|(x,y),z\rangle = |x,(y,z)\rangle$

More “easy” things

In pseudo-code, informal explanations:

- “we apply U to X in the diagonal basis”
  → “X in the diagonal basis” treated as register
- “let X_i be the i-th qubit of X”
  → “i-th qubit” is treated as a (sub-)register
- “initialize X, Y with an EPR pair”
  → X, Y is treated like a register (composition)
- “measuring position disturbs momentum”
  → position & momentum are “registers”

Semantics of quantum registers?

Existing approaches:

- Lists of qubits / dimension counting
- Swaps / associators
- Tensor product of family of spaces

Limited and/or complicated

~~Semantics of quantum registers?!?~~ Our results

Want:

- “What is a regis
- Captures th ples
- No fiddling on
- Independen
- Easy to formal e/HOL)

Also for classical registers!

[Image: Wikipedia, Mr. Smiley Face, Otakuma, 0xF8E8]

Classical registers

- Register with domain A in a memory M
- Getter + setter
- Or: updater $(A \Rightarrow A) \Rightarrow (M \Rightarrow M)$

Register F:

- Function F from $A \xrightarrow{part} A$ to $M \xrightarrow{part} M$
- Some properties:
  - $F(f) \circ F(g) = F\,(f \circ g)$
  - etc.

What is a quantum register?

- Register on $\mathcal{H}_A$ in a memory $\mathcal{H}_M$
- Given unitary/projector/... U on $\mathcal{H}_A$, register must explain "U on $\mathcal{H}_M$"

Register F:

- Function F from $\mathcal{H}_A \xrightarrow{lin} \mathcal{H}_A$ to $\mathcal{H}_M \xrightarrow{lin} \mathcal{H}_M$
- Linear
- Multiplicative $(F(ab) = F(a)F(b))$
- $\dagger$-preserving $(F(a^\dagger) = F(a)^\dagger)$

Properties of registers

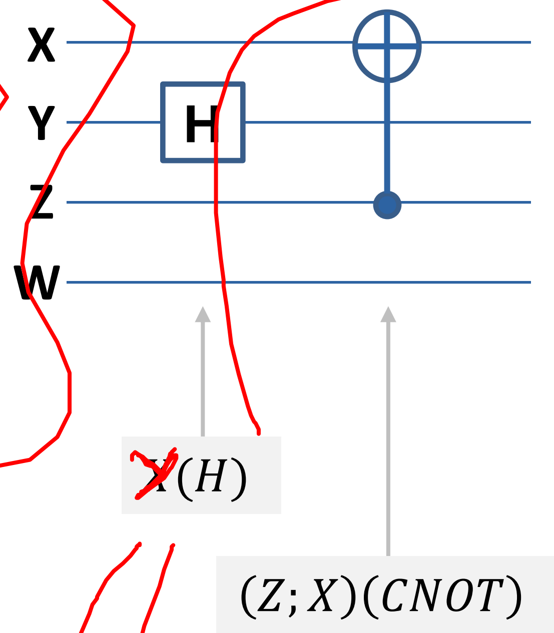
- Immediately gives semantics:
$$[\![apply\ U\ on\ X]\!] := X(U)$$
- **Pairing:** For "compatible" registers X, Y, pair $(X; Y)$ well-defined
- **Chaining:** Register X inside register Y. E.g., i-th qubit of Y.
- **Basis trafo:** Register X, basis-transformed via unitary U

+ combinations

Pairing

- **Def:** F, G compatible iff $F(a)G(b) = G(b)F(a)$
- Then exists unique $H(a \otimes b) = F(a) \overset{\circ}{\cancel{\otimes}} G(b)$
- We call this the pair $(F; G) := H$

Tiny program, revisited



$(X;Y;Z)(Z_3(H))$

- Let X, Y, Z, W be arbitrary compatible registers

$((Z;X);Y)(c$

$\neq (Z;(X;Y))(c)$

~~$= (X;Y;Z)(\text{huge} \ldots)$~~

$= (Z;X;Y)(CNOT \otimes I)$

~~$(X;Y;Z)(I \otimes H \otimes I)$~~

$\to (Z;X;Y)(H) = (I \otimes H)$

Register category

General axioms:

- (i) The objects (a.k.a. *update monoids*) $\mathbf{A}, \mathbf{B}, \dots$ of $\mathcal{L}$ are monoids. (Which monoids are objects depends on the specific register category.)
- (ii) The pre-registers are functions $\mathbf{A} \to \mathbf{B}$. (Which functions are pre-registers depends on the specific register category.) They satisfy the axioms for categories, i.e., they are closed under composition (if F, G are pre-registers, $F \circ G$ is a pre-register) and the identity is a pre-register.
- (iii) For any $a \in \mathbf{A}$, the functions $x \mapsto a \cdot x$ and $x \mapsto x \cdot a$ are pre-registers $\mathbf{A} \to \mathbf{A}$.
- $\mathcal{L}$ has a tensor product $\otimes$ such that:
  - (iv) For all $\mathbf{A}, \mathbf{B}$, $\mathbf{A} \otimes \mathbf{B}$ is an object of $\mathcal{L}$, and for $a \in \mathbf{A}, b \in \mathbf{B}$, $a \otimes b \in \mathbf{A} \otimes \mathbf{B}$.
  - (v) For pre-registers $F, G : \mathbf{A} \otimes \mathbf{B} \to \mathbf{C}$, if $\forall a, b : F(a \otimes b) = G(a \otimes b)$, then $F = G$.
  - (vi) The tensor product is distributive with respect to the monoid multiplication $\cdot$, i.e., $(a \otimes b) \cdot (c \otimes d) = (a \cdot c) \otimes (b \cdot d)$.

Registers:

- (vii) Registers are pre-registers. (Which pre-registers are also registers depends on the specific register category.)
- (viii) Registers satisfy the axioms for morphisms in categories, i.e., they are closed under composition (if F, G are registers, $F \circ G$ is a register) and the identity is a register.
- (ix) Registers are monoid homomorphisms. ($F(1) = 1$ and $F(a \cdot b) = F(a) \cdot F(b)$.)
- (x) $x \mapsto x \otimes 1$ and $x \mapsto 1 \otimes x$ are registers.
- (xi) If registers $F : \mathbf{A} \to \mathbf{C}$ and $G : \mathbf{B} \to \mathbf{C}$ have commuting ranges (i.e., $F(a), G(b)$ commute for all a, b), there exists a register $(F; G) : \mathbf{A} \otimes \mathbf{B} \to \mathbf{C}$ such that $\forall a\, b.\ (F;G)(a \otimes b) = F(a) \cdot G(b)$.

Our contribution

- Definition of “register category”
- Instantiated quantum/classical
- Infinite-dimensional case
- Isabelle/HOL formalization
- Teleportation example

Teleporting Isabelle

```
locale teleport_locale = qhoare "TYPE('mem::finite)" +
  fixes X :: "bit update ⇒ 'mem::finite update"
    and Φ :: "(bit*bit) update ⇒ 'mem update"
    and A :: "'atype::finite update ⇒ 'mem update"
    and B :: "'btype::finite update ⇒ 'mem update"
  assumes compat[compatible]: "mutually compatible (X,Φ,A,B)"
begin
```

Declaring variables

Declaring the program

```
definition "teleport a b = [
    apply CNOT (X;Φ1),
    apply hadamard X,
    ifthen Φ1 a,
    ifthen X b,
    apply (if a=1 then pauliX else idOp) Φ2,
    apply (if b=1 then pauliZ else idOp) Φ2
  ]"
```

```
lemma teleport:
  assumes [simp]: "norm ψ = 1"
  shows "hoare (XAB =q ψ ⊓ Φ =q β00) (teleport a b) (Φ2AB =q ψ)"
```

Hoare logic analysis

Postdoc/phd at University of Tartu:

- **Quantum logic/programs?**
- **Thm proving?**
- **Q info-theo/crypto?**

http://tinyurl.com/postdoc-vqc

European Research Council

Established by the European Commission