Semantics of Recursion in Quantum Control

Louis Lemonnier

Titouan Carette, Kostia Chardonnet, Robin Kaarsgaard, JS Lemay, Benoît Valiron, Vladimir Zamdzhiev



GT IQ. September, 21th 2023

let rec fact
$$(n:int) =$$

if n=0 then 1
else n * fact $(n-1)$

Programming languages

fact:

movq \$0, %rax je .case zero movq %rdi, %rax pushq %rax subq \$1, %rdi call fact popq %rbx multq %rbx, %rax jmp .else .case zero: movq \$1, %rax .else: ret

$$\begin{split} \lambda m.\lambda g.(\lambda x.g(xx))(\lambda x.g(xx))(\lambda p.\lambda a.\lambda b.pab)\\ (\lambda n.n(\lambda x.(\lambda a.\lambda b.b))(\lambda a.\lambda b.a))(\lambda f.\lambda x.fx)\\ (m(\lambda n.\lambda f.\lambda x.n(\lambda g.\lambda h.h(gf))(\lambda u.x)(\lambda x.x))) \end{split}$$

Syntax: term.

Interpreted as a mathematical object: [term].

Full abstraction:

$$\mathtt{term} \simeq \mathtt{term}' ext{ iff } \llbracket \mathtt{term}
rbracket = \llbracket \mathtt{term}'
rbracket.$$

A recurring example:

$$f(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times f(n-1) & \text{else.} \end{cases}$$

In general, when the definition of a function f involves itself.

Quantum control:

A recurring example:

$$f(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times f(n-1) & \text{else.} \end{cases}$$

In general, when the definition of a function f involves itself.

Quantum control:

- Not only superposition of states,
- Superposition of *processes*.

 \mathbb{N}_{\perp} is a partial-ordered set with *nice properties.* \perp means *undefined.*



 \mathbb{N}_{\perp} is a partial-ordered set with *nice properties.* \perp means *undefined.*



The set $[D \rightarrow D']$ of monotone functions also has nice properties.

 \mathbb{N}_{\perp} is a partial-ordered set with *nice properties.* \perp means *undefined.*



The set $[D \rightarrow D']$ of monotone functions also has nice properties. How do we use it? Define $F : [[\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}] \to [\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\bot : \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output \bot .

 $\bot \ \bot \ \bot \ \bot \ \bot \ \bot \ \bot \ \cdots$

 $F(\perp)$ is:

Define $F: [[\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}] \to [\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\bot \colon \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output \bot .

Define $F : [[\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}] \to [\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}]]$ as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\bot : \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output \bot .

Interpretation of recursion

Define
$$F : [[\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}] \to [\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}]]$$
 as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\bot \colon \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output $\bot.$

 $\bot \ \bot \ \bot \ \bot \ \bot \ \bot \ \bot \ \cdots$

 $F(F(F(\perp)))$ is:

$$1 \quad 1 \quad 2 \quad \bot \quad \bot \quad \bot \quad \bot \quad \cdots$$

Define
$$F : [[\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}] \to [\mathbb{N}_{\perp} \to \mathbb{N}_{\perp}]]$$
 as:

$$F(g)(n) = \begin{cases} 1 & \text{if } n = 0, \\ n \times g(n-1) & \text{else.} \end{cases}$$

Write $\bot \colon \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output $\bot.$

 $\bot \ \bot \ \bot \ \bot \ \bot \ \bot \ \bot \ \cdots$

 $F(F(F(F(\perp))))$ is:

1 1 2 6
$$\perp$$
 \perp \perp \cdots

Write $\bot \colon \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output $\bot.$

Write $\bot \colon \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output $\bot.$

Write $\bot \colon \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output $\bot.$

The interpretation of the factorial is then the *limit* of $(F^n(\perp))_n$.

Write $\bot \colon \mathbb{N}_{\bot} \to \mathbb{N}_{\bot}$ the constant function with output $\bot.$

 $\bot \ \bot \ \bot \ \bot \ \bot \ \bot \ \bot \ \cdots$

 $F(F(F(F(F(F(\bot))))))$ is:

1 1 2 6 24 120 \perp \cdots

The interpretation of the factorial is then the *limit* of $(F^n(\perp))_n$.

A notion of *limit* linked to the order is necessary.

Unitaries are written as sum and composition of *contractive* linear maps:

$$\begin{array}{lll} \mathrm{id} & = & \left| 0 \right\rangle \left\langle 0 \right| + \left| 1 \right\rangle \left\langle 1 \right| \\ H & = & \left| 0 \right\rangle \left\langle + \right| + \left| 1 \right\rangle \left\langle - \right| \\ X & = & \left| 0 \right\rangle \left\langle 1 \right| + \left| 1 \right\rangle \left\langle 0 \right| \\ \end{array}$$

Arguably, *undefined* (\perp) in a Hilbert space is the *zero* vector.

0 is less defined than $\left|0\right\rangle\left\langle 0\right|$ which is less defined than $\mathrm{id}.$

Unitaries are written as sum and composition of *contractive* linear maps:

$$\begin{array}{lll} \mathrm{id} & = & \left| 0 \right\rangle \left\langle 0 \right| + \left| 1 \right\rangle \left\langle 1 \right| \\ H & = & \left| 0 \right\rangle \left\langle + \right| + \left| 1 \right\rangle \left\langle - \right| \\ X & = & \left| 0 \right\rangle \left\langle 1 \right| + \left| 1 \right\rangle \left\langle 0 \right| \end{array}$$

Arguably, *undefined* (\perp) in a Hilbert space is the *zero* vector.

0 is less defined than $\left|0\right\rangle\left\langle 0\right|$ which is less defined than $\mathrm{id}.$

However, this order does not interact well with composition:

$$\begin{array}{rrr} \left|+\right\rangle \left\langle+\right| &\leq & \mathrm{id}, \\ \mathsf{thus} & \left\langle0\right|\circ\left|+\right\rangle \left\langle+\right|\circ\left|1\right\rangle &\leq & \left\langle0\right|1\right\rangle \\ \mathsf{i.e.} & & 1/2 &\leq & 0. \end{array}$$

Unitaries are written as sum and composition of *contractive* linear maps:

Arguably, *undefined* (\perp) in a Hilbert space is the *zero* vector.

0 is less defined than $\left|0\right\rangle\left\langle 0\right|$ which is less defined than $\mathrm{id.}$

However, this order does not interact well with composition:

$$\begin{array}{rrrr} |+\rangle \left\langle +| &\leq & \mathrm{id}, \\ \text{thus} & \left\langle 0 \right| \circ |+\rangle \left\langle +| \circ |1\rangle &\leq & \left\langle 0 |1\rangle, \\ \text{i.e.} & 1/2 &\leq & 0. \end{array} \right.$$

Theorem (No go theorem)

There is no such order on contractive maps between Hilbert spaces.







The trace is:
$$\operatorname{Tr}_{U}^{A,B}(f) = \underbrace{\begin{array}{c} A \\ U \\ U \end{array}}_{U} B$$



The trace is:
$$\operatorname{Tr}_{U}^{A,B}(f) = \bigcup_{U} \bigcup_{U} \bigcup_{U} \bigcup_{U}$$

And $\operatorname{Tr}_{\mathbb{N} \oplus \mathbb{N}}^{\mathbb{N},\mathbb{N}}(g)(n) = n!$.



The canonical trace formula is:

$$\mathrm{Tr}(f) = f_{AB} + \sum_{n} f_{UB} f_{UU}^{n} f_{AU}$$



The canonical trace formula is:

$$\mathrm{Tr}(f) = f_{AB} + \sum_{n} f_{UB} f_{UU}^{n} f_{AU}$$

This is a trace for unitaries between **finite dimension** Hilbert space. However:



The canonical trace formula is:

$$\mathrm{Tr}(f) = f_{AB} + \sum_{n} f_{UB} f_{UU}^{n} f_{AU}$$

This is a trace for unitaries between finite dimension Hilbert space.

However:

Lemma (Towards no go)

The hypotheses to have a canonical trace for unitaries between infinite dimension Hilbert spaces is not verified.



The canonical trace formula is:

$$\mathrm{Tr}(f) = f_{AB} + \sum_{n} f_{UB} f_{UU}^{n} f_{AU}$$

This is a trace for unitaries between finite dimension Hilbert space.

However:

Lemma (Towards no go)

The hypotheses to have a canonical trace for unitaries between infinite dimension Hilbert spaces is not verified.

My take:

Conjecture: No go

There is no trace for unitaries between infinite dimension Hilbert spaces.

Philosophically, it makes sense:

- Classical recursion allows non terminaison.
- A non terminating function **cannot** be a unitary.

Philosophically, it makes sense:

- Classical recursion allows non terminaison.
- A non terminating function **cannot** be a unitary.

Terminaison needs to be ensured!

A technique called **guarded** recursion. To be continued...

Philosophically, it makes sense:

- Classical recursion allows non terminaison.
- A non terminating function **cannot** be a unitary.

Terminaison needs to be ensured!

A technique called guarded recursion. To be continued...

Thank you!