## 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

## Programming languages

$$
\begin{aligned}
\text { def fact }(\mathrm{n}): & \\
\text { if } n= & 0: \\
& \text { return }(1) \\
\text { else : } & \\
& \text { return }(n * \text { fact }(n-1))
\end{aligned}
$$

## Programming languages

$$
\begin{aligned}
\text { let rec } & \text { fact }(\mathrm{n}: \text { int })= \\
& \text { if } \mathrm{n}=0 \text { then } 1 \\
& \text { else } \mathrm{n} * \text { fact }(\mathrm{n}-1)
\end{aligned}
$$

## Programming languages

fact:
movq $\$ 0, \% r a x$
je .case_zero
movq \%rdi, \%rax
pushq \%rax
subq \$1, \%rdi
call fact
popq \%rbx
multq \%rbx, \%rax
jmp .else
.case_zero:
movq $\$ 1, \% r a x$
.else:
ret

## Programming languages

$$
\begin{aligned}
& \lambda m \cdot \lambda g \cdot(\lambda x \cdot g(x x))(\lambda x . g(x x))(\lambda p . \lambda a \cdot \lambda b . p a b) \\
& (\lambda n \cdot n(\lambda x \cdot(\lambda a \cdot \lambda b . b))(\lambda a \cdot \lambda b . a))(\lambda f . \lambda x . f x) \\
& (m(\lambda n \cdot \lambda f \cdot \lambda x \cdot n(\lambda g \cdot \lambda h . h(g f))(\lambda u \cdot x)(\lambda x \cdot x)))
\end{aligned}
$$

## Semantics

Syntax: term.
Interpreted as a mathematical object: 【term】.

Full abstraction:

$$
\text { term } \simeq \text { term }^{\prime} \text { iff } \llbracket \text { term } \rrbracket=\llbracket \text { term }^{\prime} \rrbracket .
$$

## What is recursion? What is 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:

## What is recursion? What is 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.


## Domain theoretic point of view

A small shift of point of view.

## Domain theoretic point of view

A small shift of point of view.
$\mathbb{N}_{\perp}$ is a partial-ordered set with nice properties. $\perp$ means undefined.


## Domain theoretic point of view

A small shift of point of view.
$\mathbb{N}_{\perp}$ is a partial-ordered set with nice properties. $\perp$ means undefined.


The set $\left[D \rightarrow D^{\prime}\right]$ of monotone functions also has nice properties.

## Domain theoretic point of view

A small shift of point of view.
$\mathbb{N}_{\perp}$ is a partial-ordered set with nice properties. $\perp$ means undefined.


The set $\left[D \rightarrow D^{\prime}\right]$ of monotone functions also has nice properties.
How do we use it?

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\perp \perp \perp \perp \perp \perp \perp
$$

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\perp \perp \perp \perp \perp \perp \perp
$$

$F(\perp)$ is:

$$
1 \perp \perp \perp \perp \perp \perp \perp
$$

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\perp \perp \perp \perp \perp \perp \perp
$$

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

$$
\begin{array}{lllllll}
1 & 1 & \perp & \perp & \perp
\end{array}
$$

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\perp \perp \perp \perp \perp \perp \perp
$$

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

$$
\begin{array}{lllllllll}
1 & 1 & 2 & \perp & \perp & \perp & \cdots
\end{array}
$$

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\perp \perp \perp \perp \perp \perp \perp
$$

$$
F(F(F(F(\perp)))) \text { is: }
$$

$$
\begin{array}{llllllll}
1 & 1 & 2 & 6 & \perp & \perp & \perp & \cdots
\end{array}
$$

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\perp \perp \perp \perp \perp \perp \perp
$$

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

$$
\begin{array}{lllllllll}
1 & 1 & 2 & 6 & 24 & \perp & \perp & \cdots
\end{array}
$$

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\perp \perp \perp \perp \perp \perp \perp
$$

$$
F(F(F(F(F(F(\perp)))))) \text { is: }
$$

$$
\begin{array}{llllllll}
1 & 1 & 2 & 6 & 24 & 120 & \perp & \cdots
\end{array}
$$

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\begin{array}{rllllllll} 
& \perp & \perp & \perp & \perp & \perp & \perp & \perp & \cdots \\
F(F(F(F(F(F(\perp)))))) \text { is: } & & & & & \\
1 & 1 & 2 & 6 & 24 & 120 & \perp & \cdots
\end{array}
$$

The interpretation of the factorial is then the limit of $\left(F^{n}(\perp)\right)_{n}$.

## Interpretation of recursion

Define $F:\left[\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right] \rightarrow\left[\mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}\right]\right]$ as:

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

Write $\perp: \mathbb{N}_{\perp} \rightarrow \mathbb{N}_{\perp}$ the constant function with output $\perp$.

$$
\left.\begin{array}{rlllllllll} 
& \perp & \perp & \perp & \perp & \perp & \perp & \perp & \cdots
\end{array}\right)
$$

The interpretation of the factorial is then the limit of $\left(F^{n}(\perp)\right)_{n}$.
A notion of limit linked to the order is necessary.

## The Quantum Case

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

$$
\begin{aligned}
\text { id } & =|0\rangle\langle 0|+|1\rangle\langle 1| \\
H & =|0\rangle\langle+|+|1\rangle\langle-| \\
X & =|0\rangle\langle 1|+|1\rangle\langle 0|
\end{aligned}
$$

Arguably, undefined $(\perp)$ in a Hilbert space is the zero vector.
0 is less defined than $|0\rangle\langle 0|$ which is less defined than id.

## The Quantum Case

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

$$
\begin{aligned}
\mathrm{id} & =|0\rangle\langle 0|+|1\rangle\langle 1| \\
H & =|0\rangle\langle+|+|1\rangle\langle-| \\
X & =|0\rangle\langle 1|+|1\rangle\langle 0|
\end{aligned}
$$

Arguably, undefined $(\perp)$ in a Hilbert space is the zero vector.
0 is less defined than $|0\rangle\langle 0|$ which is less defined than id.
However, this order does not interact well with composition:


## The Quantum Case

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

$$
\begin{aligned}
\mathrm{id} & =|0\rangle\langle 0|+|1\rangle\langle 1| \\
H & =|0\rangle\langle+|+|1\rangle\langle-| \\
X & =|0\rangle\langle 1|+|1\rangle\langle 0|
\end{aligned}
$$

Arguably, undefined $(\perp)$ in a Hilbert space is the zero vector.
0 is less defined than $|0\rangle\langle 0|$ which is less defined than id.
However, this order does not interact well with composition:


Theorem (No go theorem)
There is no such order on contractive maps between Hilbert spaces.

## Recursion through a Trace



## Recursion through a Trace



The recurring example: $g: \begin{cases}\mathbb{N} \uplus \mathbb{N} \times \mathbb{N} & \rightarrow \mathbb{N} \uplus \mathbb{N} \times \mathbb{N} \\ \operatorname{left} 0 & \mapsto \text { left } 1 \\ \operatorname{right}(n+1, m) & \mapsto \text { right }(n, n \times m) \\ \text { right }(0, m) & \mapsto \text { left } m\end{cases}$

## Recursion through a Trace



The recurring example: $g: \begin{cases}\mathbb{N} \uplus \mathbb{N} \times \mathbb{N} & \rightarrow \mathbb{N} \uplus \mathbb{N} \times \mathbb{N} \\ \text { left } 0 & \mapsto \text { left } 1 \\ \text { right }(n+1, m) & \mapsto \text { right }(n, n \times m) \\ \text { right }(0, m) & \mapsto \text { left } m\end{cases}$

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


## Recursion through a Trace



The recurring example: $g: \begin{cases}\mathbb{N} \uplus \mathbb{N} \times \mathbb{N} & \rightarrow \mathbb{N} \uplus \mathbb{N} \times \mathbb{N} \\ \text { left } 0 & \mapsto \text { left } 1 \\ \text { right }(n+1, m) & \mapsto \text { right }(n, n \times m) \\ \text { right }(0, m) & \mapsto \text { left } m\end{cases}$

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


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

## The Quantum Case



The canonical trace formula is:

$$
\operatorname{Tr}(f)=f_{A B}+\sum_{n} f_{U B} f_{U U}^{n} f_{A U}
$$

## The Quantum Case



The canonical trace formula is:

$$
\operatorname{Tr}(f)=f_{A B}+\sum_{n} f_{U B} f_{U U}^{n} f_{A U}
$$

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

## The Quantum Case



The canonical trace formula is:

$$
\operatorname{Tr}(f)=f_{A B}+\sum_{n} f_{U B} f_{U U}^{n} f_{A U}
$$

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 Quantum Case



The canonical trace formula is:

$$
\operatorname{Tr}(f)=f_{A B}+\sum_{n} f_{U B} f_{U U}^{n} f_{A U}
$$

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.

## Until we meet again

Is quantum recursion not possible?

## Until we meet again

Is quantum recursion not possible?
Philosophically, it makes sense:

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


## Until we meet again

Is quantum recursion not possible?
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...

## Until we meet again

Is quantum recursion not possible?
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!

