Toward a Curry-Howard Correspondence for Linear, Reversible Computation

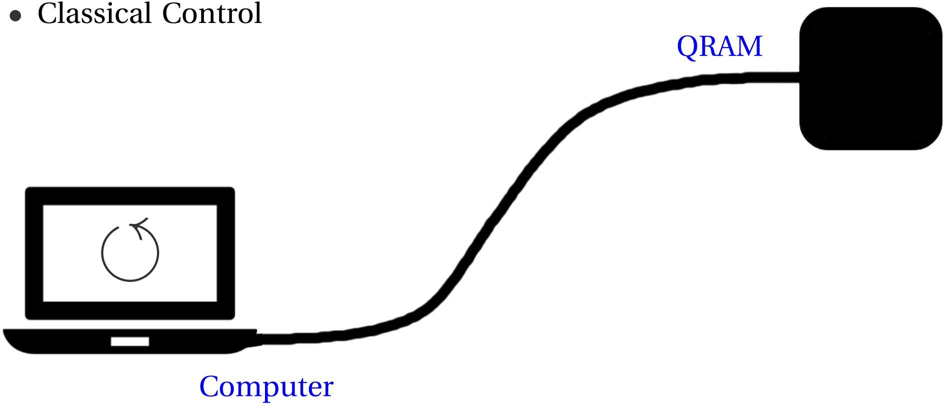
Reversible Computation 2020

Kostia Chardonnet[1,2] Alexis Saurin[2] Benoît Valiron[1]

[1]Université Paris Saclay [2]Université de Paris

Known :

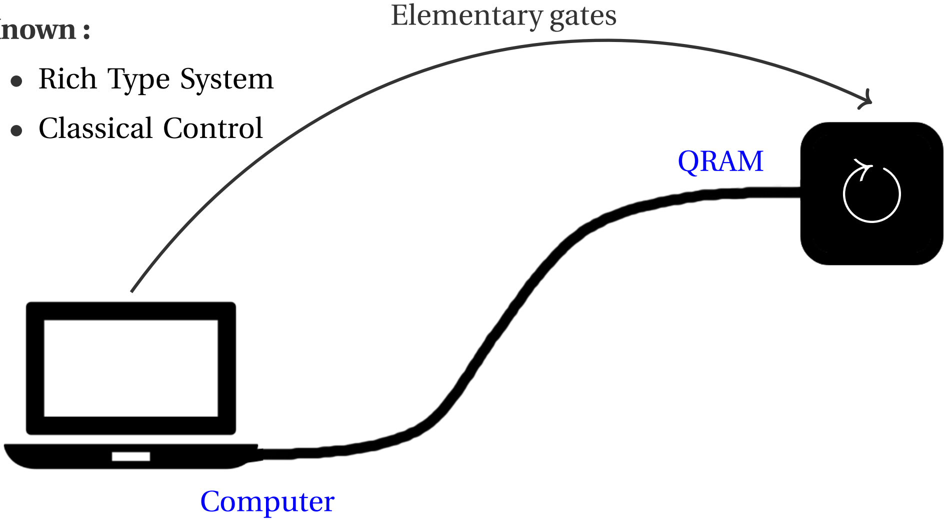
- Rich Type System
- Classical Control



Classical vs Quantum Control

Known :

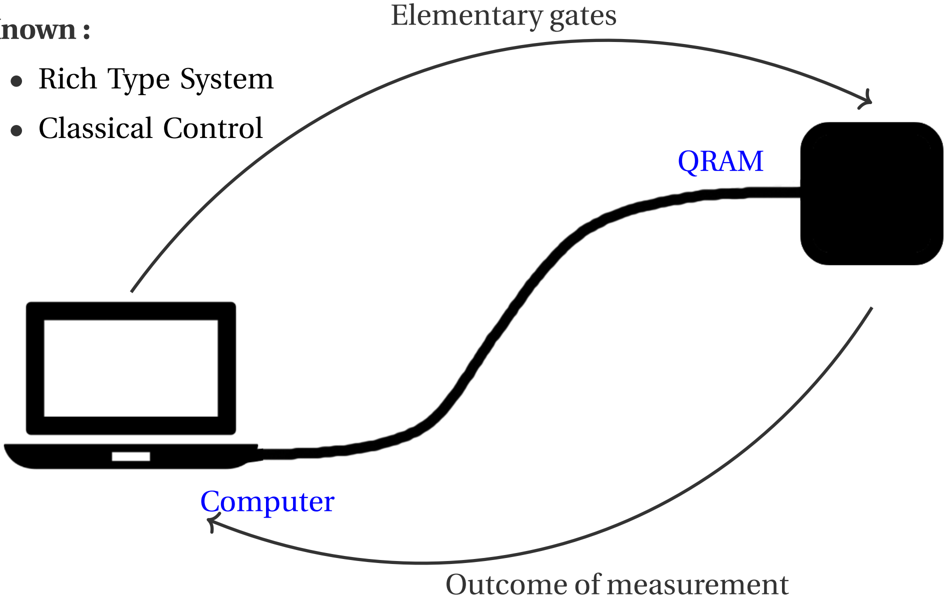
- Rich Type System
- Classical Control



Classical vs Quantum Control

Known :

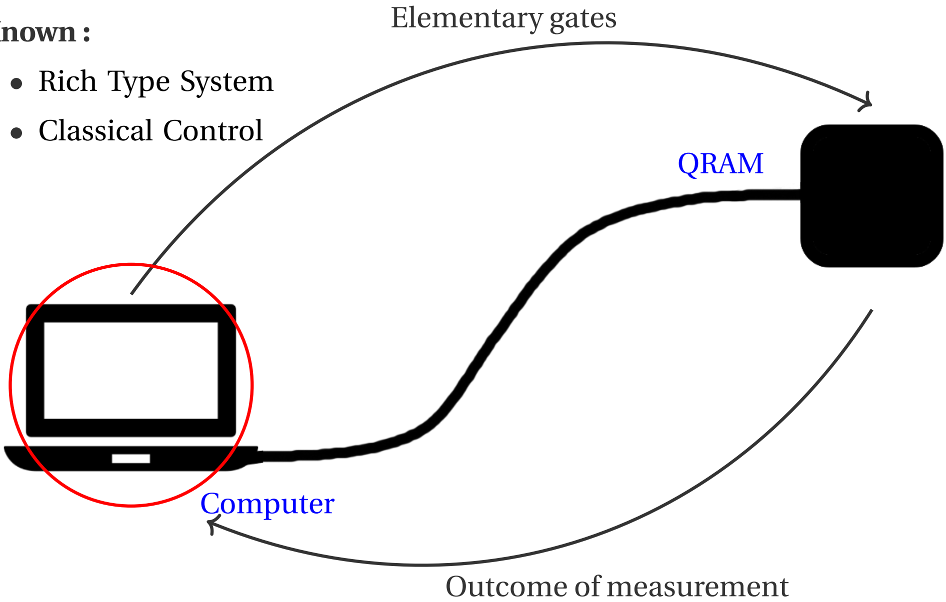
- Rich Type System
- Classical Control



Classical vs Quantum Control

Known :

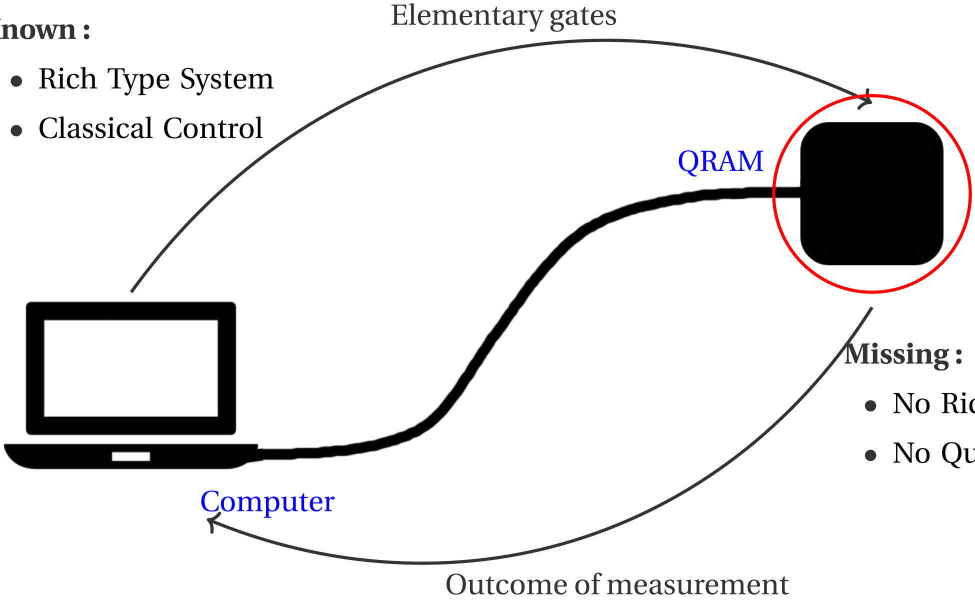
- Rich Type System
- Classical Control



Classical vs Quantum Control

Known :

- Rich Type System
- Classical Control



Missing :

- No Rich Type System
- No Quantum Control

Types

- Types describe data, structure programs.
- “Well-typed Programs Cannot Go Wrong” - Robin Milner

Types

- Types describe data, structure programs.
- “Well-typed Programs Cannot Go Wrong” - Robin Milner

$$\textbf{Example}: \mathit{toString} : \mathit{nat} \to \mathit{string} \qquad \mathit{toString}(5) = \text{"five"}.$$

Types

- Types describe data, structure programs.
- "Well-typed Programs Cannot Go Wrong" - Robin Milner

Example : $\mathit{toString} : \mathit{nat} \to \mathit{string}$ $\qquad \mathit{toString}(5) = \text{"five"}.$

$\mathit{toString}(5)$ $\qquad\qquad$ $\mathit{toString}(\text{"toto"})$

Well typed ! $\qquad\qquad$ Ill typed !

The Curry-Howard Correspondence

$$\frac{\mathit{toString} : \mathit{nat} \to \mathit{string} \qquad 5 : \mathit{nat}}{\mathit{toString}(5) : \mathit{string}}$$

$\mathit{toString}(5)$

Well typed !

$\mathit{toString}(\text{"toto"})$

Ill typed !

$$\frac{\mathit{toString} : \mathit{nat} \to \mathit{string} \qquad 5 : \mathit{nat}}{\mathit{toString}(5) : \mathit{string}}$$

$$\frac{A \to B \qquad A}{B}$$

$\mathit{toString}(5)$

Well typed !

$\mathit{toString}(\text{"toto"})$

Ill typed !

$$\frac{\mathit{nat} \to \mathit{string} \qquad \mathit{nat}}{\mathit{string}} \qquad\qquad \frac{A \to B \qquad A}{B}$$

$\mathit{toString}(5)$

Well typed !

$\mathit{toString}(\text{"toto"})$

Ill typed !

$$\frac{\mathit{nat} \to \mathit{string} \qquad \mathit{nat}}{\mathit{string}}$$

$$\frac{A \to B \qquad A}{B}$$

Curry-Howard Correspondence !

$\mathit{toString}(5)$

Well typed !

$\mathit{toString}(\text{"toto"})$

Ill typed !

Formal Program Verification

$\boxed{\text{Curry-Howard Correspondence !}}$

$\Downarrow$

λ-calculus	**Logic & Proofs**
Types	Formulas
Typed terms	Proofs
Evaluation	Cut Elimination

Our Work

Based on [Sabry, Valiron, Vizzotto] and [Baelde, Doumane, Saurin]

	Sabry et al.	Baelde et al.	This Work
Linear	✓	✓	✓
Reversible	✓	✗	✓
(Co)-Inductive	✗	✓	✓
Curry-Howard	✗	✗	✓
Quantum Case	✓	✗	WIP

Syntax

(Base types)	$A, B ::=$	$\mathbb{1} \mid X \mid A \oplus B \mid A \otimes B \mid \mu X.A \mid \nu X.A$
(Isos, first-order)	$\alpha ::=$	$A \leftrightarrow B$
(Isos, higher-order)	$T ::=$	$\alpha_1 \to \cdots \to \alpha_n \to \alpha$

Syntax

(Base types)	$A, B ::= \mathbb{1} \mid X \mid A \oplus B \mid A \otimes B \mid \mu X.A \mid \nu X.A$
(Isos, first-order)	$\alpha ::= A \leftrightarrow B$
(Isos, higher-order)	$T ::= \alpha_1 \to \cdots \to \alpha_n \to \alpha$

- $nat = \mu X.\mathbb{1} \oplus X$
- $lists(A) = [A] = \mu X.\ \mathbb{1} \oplus (A \otimes X)$
- $streams(A) = \nu X.\ A \otimes X$

Syntax

$$
\begin{array}{lrl}
\text{(Base types)} & A, B ::= & \mathbb{1} \mid X \mid A \oplus B \mid A \otimes B \mid \mu X.A \mid \nu X.A \\
\text{(Isos, first-order)} & \alpha ::= & A \leftrightarrow B \\
\text{(Isos, higher-order)} & T ::= & \alpha_1 \to \cdots \to \alpha_n \to \alpha \\
\text{(Isos)} & \omega ::= & \{e_1 \leftrightarrow e'_1 \mid \ldots \mid e_n \leftrightarrow e'_n\} \mid \lambda f.\omega \mid \\
 & & \mu f.\omega \mid f \mid \omega_1\,\omega_2 \mid \mathtt{inv}\,\omega
\end{array}
$$

$$
\lambda g.\mu f.\left\{\begin{array}{lcl}
[\,] & \leftrightarrow & [\,] \\
h :: t & \leftrightarrow & \mathtt{let}\,x = g\ h\ \mathtt{in} \\
 & & \mathtt{let}\,y = f\ t\ \mathtt{in}\ x :: y
\end{array}\right\} : A \leftrightarrow B \to [A] \leftrightarrow [B]
$$

Syntax

$$
\begin{array}{lrcl}
\text{(Base types)} & A, B & ::= & \mathbb{1} \mid X \mid A \oplus B \mid A \otimes B \mid \mu X.A \mid \nu X.A \\
\text{(Isos, first-order)} & \alpha & ::= & A \leftrightarrow B \\
\text{(Isos, higher-order)} & T & ::= & \alpha_1 \to \cdots \to \alpha_n \to \alpha \\
\text{(Isos)} & \omega & ::= & \{e_1 \leftrightarrow e'_1 \mid \ldots \mid e_n \leftrightarrow e'_n\} \mid \lambda f.\omega \mid \\
 & & & \mu f.\omega \mid f \mid \omega_1\,\omega_2 \mid \mathtt{inv}\ \omega
\end{array}
$$

$$
\lambda g.\mu f.\left\{\begin{array}{lcl}
[\,] & \leftrightarrow & [\,] \\
h :: t & \leftrightarrow & \mathtt{let}\, x = g\ h\ \mathtt{in} \\
 & & \mathtt{let}\, y = f\ t\ \mathtt{in}\ x :: y
\end{array}\right\} : A \leftrightarrow B \to [A] \leftrightarrow [B]
$$

Syntax

$$
\begin{array}{lrl}
\text{(Base types)} & A, B ::= & \mathbb{1} \mid X \mid A \oplus B \mid A \otimes B \mid \mu X.A \mid \nu X.A \\
\text{(Isos, first-order)} & \alpha ::= & A \leftrightarrow B \\
\text{(Isos, higher-order)} & T ::= & \alpha_1 \to \cdots \to \alpha_n \to \alpha \\
\text{(Isos)} & \omega ::= & \{e_1 \leftrightarrow e_1' \mid \ldots \mid e_n \leftrightarrow e_n'\} \mid \lambda f.\omega \mid \\
 & & \mu f.\omega \mid f \mid \omega_1\, \omega_2 \mid \mathtt{inv}\; \omega
\end{array}
$$

$$
\lambda g.\mu f.\underbrace{\left\{\begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ h :: t & \leftrightarrow & \mathtt{let}\, x = g\ h\ \mathtt{in} \\ & & \mathtt{let}\, y = f\ t\ \mathtt{in}\ x :: y \end{array}\right\}} : A \leftrightarrow B \to [A] \leftrightarrow [B]
$$

Properties

Syntax

- Language comes with a rewriting system and a type system.
- Ensuring exhaustivity and non-overlapping of clauses.
- Ensuring productivity.

Semantic

- Isos denote computations from $A \to B$ and $B \to A$.

Syntax - Example 2

$$\mathtt{map} = \lambda \mathtt{g}.\mu \mathtt{f}. \left\{ \begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ h :: t & \leftrightarrow & \mathtt{let}\, x = g\ h\ \mathtt{in} \\ & & \mathtt{let}\, y = f\ t\ \mathtt{in}\ x :: y \end{array} \right\} : \mathtt{A} \leftrightarrow \mathtt{B} \rightarrow [\mathtt{A}] \leftrightarrow [\mathtt{B}]$$

Syntax - Example 2

$$\mathtt{map} = \lambda \mathrm{g}.\mu \mathtt{f}. \left\{ \begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ h :: t & \leftrightarrow & \mathtt{let}\, x = g\ h\ \mathtt{in} \\ & & \mathtt{let}\, y = f\ t\ \mathtt{in}\ x :: y \end{array} \right\} : \mathtt{A} \leftrightarrow \mathtt{B} \rightarrow [\mathtt{A}] \leftrightarrow [\mathtt{B}]$$

$$\mathtt{map}^{\perp} = \lambda \mathrm{g}.\mu \mathtt{f}. \left\{ \begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ \mathtt{let}\, x = g\ h\ \mathtt{in} & \leftrightarrow & h :: t \\ \mathtt{let}\, y = f\ t\ \mathtt{in}\ x :: y & & \end{array} \right\} : \mathtt{A} \leftrightarrow \mathtt{B} \rightarrow [\mathtt{B}] \leftrightarrow [\mathtt{A}]$$

$$\mathtt{map} = \lambda \mathtt{g}.\mu \mathtt{f}. \left\{ \begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ h :: t & \leftrightarrow & \mathtt{let}\, x = g\ h\ \mathtt{in} \\ & & \mathtt{let}\, y = f\ t\ \mathtt{in}\ x :: y \end{array} \right\} : \mathtt{A} \leftrightarrow \mathtt{B} \rightarrow [\mathtt{A}] \leftrightarrow [\mathtt{B}]$$

$$\mathtt{map}^{\perp} = \lambda \mathtt{g}.\mu \mathtt{f}. \left\{ \begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ \mathtt{let}\, x = g\ h\ \mathtt{in} & \leftrightarrow & h :: t \\ \mathtt{let}\, y = f\ t\ \mathtt{in}\ x :: y & & \end{array} \right\} : \mathtt{A} \leftrightarrow \mathtt{B} \rightarrow [\mathtt{B}] \leftrightarrow [\mathtt{A}]$$

$$\rightsquigarrow$$

$$\mathtt{map}^{\perp} = \lambda \mathtt{g}.\mu \mathtt{f}. \left\{ \begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ x :: y & \leftrightarrow & \mathtt{let}\, h = (\mathtt{inv}\ (g))\ x\ \mathtt{in} \\ & & \mathtt{let}\, t = f\ y\ \mathtt{in}\ h :: t \end{array} \right\} : \mathtt{A} \leftrightarrow \mathtt{B} \rightarrow [\mathtt{B}] \leftrightarrow [\mathtt{A}]$$

Confluence

$$\begin{tikzcd} t_1 \arrow[r, "*"] \arrow[d, "*"] & t_2 \arrow[d, dotted, "*"] \\ t_3 \arrow[r, dotted, "*"'] & t_4 \end{tikzcd}$$

Type Preservation

If $\vdash t : A$ and $t \to t'$ then $\vdash t' : A$.

Progress

If $\vdash t : A$ either $t \to t'$ or t is a value.

Isos

For each ω we have $\omega \circ (\mathtt{inv}\ \omega) = id = (\mathtt{inv}\ \omega) \circ \omega$.

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$\frac{A \vdash B[X \leftarrow \mu X.B]}{A \vdash \mu X.B}\ \mu_R$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![0]\!] = \quad \underbrace{\vdash \mu X.\mathbb{1} \oplus X}_{nat}$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![0]\!] = \cfrac{\vdash \mathbb{1} \oplus \mu X.\mathbb{1} \oplus X}{\underbrace{\vdash \mu X.\mathbb{1} \oplus X}_{nat}}\ \mu_R$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![0]\!] = \dfrac{\dfrac{\vdash \mathbb{1}}{\vdash \mathbb{1} \oplus \mu X.\mathbb{1} \oplus X}\ \oplus_1}{\vdash \underbrace{\mu X.\mathbb{1} \oplus X}_{nat}}\ \mu_R$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![0]\!] = \dfrac{\dfrac{\dfrac{}{\vdash \mathbb{1}}\ \mathbb{1}_R}{\vdash \mathbb{1} \oplus \mu X.\mathbb{1} \oplus X}\ \oplus_1}{\underbrace{\vdash \mu X.\mathbb{1} \oplus X}_{nat}}\ \mu_R$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![0]\!] = \dfrac{\dfrac{\dfrac{}{\vdash \mathbb{1}}\ \mathbb{1}_R}{\vdash \mathbb{1} \oplus \mu X.\mathbb{1} \oplus X}\ \oplus_1}{\vdash \underbrace{\mu X.\mathbb{1} \oplus X}_{nat}}\ \mu_R \qquad [\![n+1]\!] = \quad \vdash nat$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$
[\![0]\!] = \dfrac{\dfrac{\dfrac{}{\vdash \mathbb{1}}\ \mathbb{1}_R}{\vdash \mathbb{1} \oplus \mu X.\mathbb{1} \oplus X}\ \oplus_1}{\vdash \underbrace{\mu X.\mathbb{1} \oplus X}_{nat}}\ \mu_R
\qquad
[\![n+1]\!] = \dfrac{\vdash \mathbb{1} \oplus nat}{\vdash nat}\ \mu_R
$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$
[\![0]\!] = \dfrac{\dfrac{\dfrac{}{\vdash \mathbb{1}}\ \mathbb{1}_R}{\vdash \mathbb{1} \oplus \mu X.\mathbb{1} \oplus X}\ \oplus_1}{\vdash \underbrace{\mu X.\mathbb{1} \oplus X}_{nat}}\ \mu_R
\qquad
[\![n+1]\!] = \dfrac{\dfrac{\vdash nat}{\vdash \mathbb{1} \oplus nat}\ \oplus_2}{\vdash nat}\ \mu_R
$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![0]\!] = \cfrac{\cfrac{\cfrac{}{\vdash \mathbb{1}}\ \mathbb{1}_R}{\vdash \mathbb{1} \oplus \mu X.\mathbb{1} \oplus X}\ \oplus_1}{\vdash \underbrace{\mu X.\mathbb{1} \oplus X}_{nat}}\ \mu_R \qquad [\![n+1]\!] = \cfrac{\cfrac{\cfrac{[\![n]\!]}{\vdash nat}}{\vdash \mathbb{1} \oplus nat}\ \oplus_2}{\vdash nat}\ \mu_R$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![0]\!] = \cfrac{\cfrac{\cfrac{}{\vdash \mathbb{1}}\ \mathbb{1}_R}{\vdash \mathbb{1} \oplus \mu X.\mathbb{1} \oplus X}\ \oplus_1}{\vdash \underbrace{\mu X.\mathbb{1} \oplus X}_{nat}}\ \mu_R \qquad [\![n+1]\!] = \cfrac{\cfrac{\cfrac{[\![n]\!]}{\vdash nat}}{\vdash \mathbb{1} \oplus nat}\ \oplus_2}{\vdash nat}\ \mu_R$$

$$[\![1]\!] = \cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\vdash \mathbb{1}}\ \mathbb{1}_R}{\vdash \mathbb{1} \oplus nat}\ \oplus_1}{\vdash nat}\ \mu_R}{\vdash \mathbb{1} \oplus nat}\ \oplus_2}{\vdash nat}\ \mu_R$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![Stream_0]\!] = \cfrac{\cfrac{\cfrac{[\![0]\!]}{\vdash nat} \quad \cfrac{}{\vdash \nu X.nat \otimes X}}{\vdash nat \otimes (\nu X.nat \otimes X)} \otimes}{\vdash \nu X.nat \otimes X} \nu_R$$

Linear Logic with Induction ($\mu X.A$) and Co-Induction ($\nu X.A$)

$$[\![Stream_0]\!] = \dfrac{\dfrac{\dfrac{[\![0]\!]}{\vdash nat} \qquad \dfrac{}{\vdash \nu X.nat \otimes X}}{\vdash nat \otimes (\nu X.nat \otimes X)}\ \otimes}{\vdash \nu X.nat \otimes X}\ \nu_R$$

Words on μMALL

Linear Logic with Induction and Co-Induction

$$\dfrac{\dfrac{\vdots}{\vdash \mu X.X}\,\mu_R}{\vdash \mu X.X}\,\mu_R \qquad\qquad \dfrac{\vdash \mu X.X}{\vdash \mu X.X}\,\mu_R$$

Infinite derivations represented as graphs

A derivation is valid if in every infinite branch :

- Infinity of rules μ_L
- Infinity of rules ν_R

From Type Derivation To Proofs

$$\text{Typed Terms} \rightsquigarrow \text{Proofs}$$

$$\omega : A \leftrightarrow B \rightsquigarrow \pi : A \vdash B$$

$$\omega^{\perp} : B \leftrightarrow A \rightsquigarrow \pi^{\perp} : B \vdash A$$

From Derivations To Proofs - Example

Let us take the recursive *identity* on lists

$$\mu f.\ \left\{\begin{array}{ll} [\,] & \leftrightarrow [\,] \\ h :: t & \leftrightarrow \texttt{let}\ t' = f\ t\ \texttt{in} \\ & \quad\ \ h :: t' \end{array}\right\} : [A] \leftrightarrow [A]$$

From Derivations To Proofs - Example

$$
\dfrac{\dfrac{\vdash [A]}{\mathbb{1} \vdash [A]}\ \mathbb{1}_L \qquad \dfrac{A, [A] \vdash [A]}{A \otimes [A] \vdash [A]}\ \otimes_L}{\dfrac{\mathbb{1} \oplus (A \otimes [A]) \vdash [A]}{[A] \vdash [A]}\ \mu_L}\ \oplus_L
$$

$$
\mu f.\ \left\{ \begin{array}{lll} [\,] & \leftrightarrow & [\,] \\ h :: t & \leftrightarrow & \texttt{let}\ t' = f\ t\ \texttt{in} \\ & & h :: t' \end{array} \right\}
$$

From Derivations To Proofs - Example

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$\mathbb{1}_R$}
\UnaryInfC{$\vdash \mathbb{1}$}
\RightLabel{$\oplus^1_R$}
\UnaryInfC{$\vdash \mathbb{1} \oplus (A \otimes [a])$}
\RightLabel{μ_R}
\UnaryInfC{$\vdash [A]$}
\RightLabel{$\mathbb{1}_L$}
\UnaryInfC{$\mathbb{1} \vdash [A]$}
\AxiomC{$A, [A] \vdash [A]$}
\RightLabel{$\otimes_L$}
\UnaryInfC{$A \otimes [A] \vdash [A]$}
\RightLabel{$\oplus_L$}
\BinaryInfC{$\mathbb{1} \oplus (A \otimes [A]) \vdash [A]$}
\RightLabel{μ_L}
\UnaryInfC{$[A] \vdash [A]$}
\end{prooftree}
$$

$$
\mu f.\ \left\{
\begin{array}{lcl}
[\,] & \leftrightarrow & [\,] \\
h :: t & \leftrightarrow & \texttt{let}\ t' = f\ t\ \texttt{in} \\
 & & \quad h :: t'
\end{array}
\right\}
$$

From Derivations To Proofs - Example

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{\vdash \mathbb{1}}\ \mathbb{1}_R
      }{\vdash \mathbb{1} \oplus (A \otimes [a])}\ \oplus^1_R
    }{\vdash [A]}\ \mu_R
  }{\mathbb{1} \vdash [A]}\ \mathbb{1}_L
  \qquad
  \dfrac{
    \dfrac{
      [A] \vdash [A] \qquad A, [A] \vdash [A]
    }{A, [A] \vdash [A]}\ \text{cut}
  }{A \otimes [A] \vdash [A]}\ \otimes_L
}{
  \dfrac{\mathbb{1} \oplus (A \otimes [A]) \vdash [A]}{[A] \vdash [A]}\ \mu_L
}\ \oplus_L
$$

$$
\mu f.\ \left\{ \begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ h :: t & \leftrightarrow & \texttt{let}\ t' = f\ t\ \texttt{in} \\ & & h :: t' \end{array} \right\}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \mathbb{1}}\,\mathbb{1}_R}{\vdash \mathbb{1} \oplus (A \otimes [a])}\,\oplus^1_R}{\vdash [A]}\,\mu_R}{\mathbb{1} \vdash [A]}\,\mathbb{1}_L \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{[A] \vdash [A]}\,\text{id} \qquad [A] \vdash [A]}{[A] \vdash [A]}\,\text{cut} \qquad A, [A] \vdash [A]}{A, [A] \vdash [A]}\,\text{cut}}{A \otimes [A] \vdash [A]}\,\otimes_L}{\dfrac{\mathbb{1} \oplus (A \otimes [A]) \vdash [A]}{[A] \vdash [A]}\,\mu_L}\,\oplus_L
$$

$$
\mu f.\ \left\{ \begin{array}{lcl} [\,] & \leftrightarrow & [\,] \\ h :: t & \leftrightarrow & \texttt{let}\ t' = f\ t\ \texttt{in} \\ & & \quad h :: t' \end{array} \right\}
$$

From Derivations To Proofs - Example

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{}{\vdash \mathbb{1}}\;\mathbb{1}_R
    }{\vdash \mathbb{1} \oplus (A \otimes [a])}\;\oplus^1_R
  }{
    \dfrac{\vdash [A]}{\mathbb{1} \vdash [A]}\;\mathbb{1}_L
  }\;\mu_R
  \qquad
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{[A] \vdash [A]}\;\mathrm{id}
        \qquad
        [A] \vdash [A]
      }{[A] \vdash [A]}\;\mathrm{cut}
      \qquad
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{}{a \vdash a}\;\mathrm{id}
            \qquad
            \dfrac{}{[a] \vdash [a]}\;\mathrm{id}
          }{A, [A] \vdash A \otimes [A]}\;\otimes_R
        }{A, [A] \vdash \mathbb{1} \oplus A \otimes [A]}\;\oplus^2_R
      }{A, [A] \vdash [A]}\;\mu_R
    }{A, [A] \vdash [A]}\;\mathrm{cut}
  }{A \otimes [A] \vdash [A]}\;\otimes_L
}{
  \dfrac{\mathbb{1} \oplus (A \otimes [A]) \vdash [A]}{[A] \vdash [A]}\;\mu_L
}\;\oplus_L
$$

$$
\mu f.\ \left\{
\begin{array}{lcl}
[\,] & \leftrightarrow & [\,] \\
h :: t & \leftrightarrow & \texttt{let}\ t' = f\ t\ \texttt{in} \\
 & & \color{blue}{h :: t'}
\end{array}
\right\}
$$

Results

- Confluence.
- Type Preservation.
- Progress.
- Isos are isomorphisms.

In Progress

- Show that derivations built isos are valid.
- Show that our reduction simulates cut-elimination.
- Show that $\pi, \pi^{\perp}$ are isomorphisms.
- Consider the quantum case.