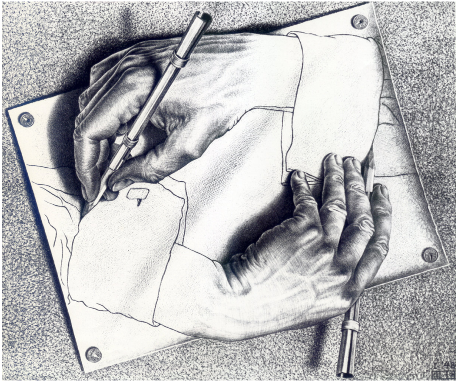
Jérôme Fortier, with Luigi Santocanale

Cuts in circular proofs



Delft – February 17, 2014

Natural numbers

(Circular) Definition

A natural number is either 0, or of the form $\mathtt{suc}(n)$ where n is a natural number.

Natural numbers

(Circular) Definition

A natural number is either 0, or of the form $\mathtt{suc}(n)$ where n is a natural number.

$$\begin{tikzcd} 1 + \mathbb{N} \arrow[d, "{\{0, \mathrm{suc}\}}"'] \\ \mathbb{N} \end{tikzcd}$$

Natural numbers

(Circular) Definition

A natural number is either 0, or of the form $\mathtt{suc}(n)$ where n is a natural number. $\mathbb{N}$ is the least fixpoint of this definition!

Initial algebra!

$$
\begin{tikzcd}
1+\mathbb{N} \arrow[r, dashed, "1+f"] \arrow[d, "\{0,\mathrm{suc}\}"'] & 1+X \arrow[d, "\{a,g\}"] \\
\mathbb{N} \arrow[r, dashed, "\exists! f"'] & X
\end{tikzcd}
$$

$$
\begin{aligned}
f(0) &= a \\
f(\mathtt{suc}(x)) &= g(f(x))
\end{aligned}
$$

(Circular) Definition

A natural number is either 0, or of the form $\mathtt{suc}(n)$ where n is a natural number. $\mathbb{N}$ is the least fixpoint of this definition!

Initial algebra!

$$
\begin{tikzcd}
1+\mathbb{N} \arrow[r, dashed, "1+f"] \arrow[d, bend right, "\{0,\mathtt{suc}\}"'] & 1+X \arrow[d, "\{a,g\}"] \\
\mathbb{N} \arrow[u, bend right, "\mathtt{pre}"'] \arrow[r, dashed, "\exists! f"'] & X
\end{tikzcd}
$$

$$
\begin{aligned}
f(0) &= a \\
f(\mathtt{suc}(x)) &= g(f(x))
\end{aligned}
$$

Natural numbers

(Circular) Definition

A natural number is either 0, or of the form $\mathtt{suc}(n)$ where n is a natural number. $\mathbb{N}$ is the least fixpoint of this definition!

Initial algebra!

$$\begin{tikzcd}
1+\mathbb{N} \arrow[r, dashed, "1+f"] \arrow[d, bend right, "\{0,\mathtt{suc}\}"'] & 1+X \arrow[d, "\{a,g\}"] \\
\mathbb{N} \arrow[u, bend right, "\mathtt{pre}"'] \arrow[r, dashed, "\exists! f"'] & X
\end{tikzcd}$$

$$\begin{aligned} f(0) &= a \\ f(\mathtt{suc}(x)) &= g(f(x)) \end{aligned}$$

$$\mathbb{N} = \mu X.(1+X)$$

Other inductive types...

Other inductive types...

- $\mu X.(1 + A \times X) = A^* =$ Finite words over A

$$\begin{array}{rcl} * & \mapsto & \varepsilon \\ \langle a, w \rangle & \mapsto & a \cdot w \end{array}$$

Other inductive types...

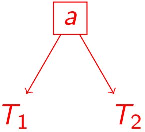
- $\mu X.(1 + A \times X) = A^* =$ Finite words over A

$$\begin{aligned} * &\mapsto \varepsilon \\ \langle a, w \rangle &\mapsto a \cdot w \end{aligned}$$

- $\mu X.(1 + A \times X \times X) =$ Finite binary labelled trees

$$* \mapsto \text{Empty tree}$$

$$\langle a, T_1, T_2 \rangle \mapsto$$



Streams

(Circular) Definition

A stream over an alphabet A is made of a head $a \in A$, and another stream called the tail.

(Circular) Definition

A stream over an alphabet A is made of a head $a \in A$, and another stream called the tail.

$$\begin{tikzcd} A^\omega \arrow[d, "\langle \mathtt{head}, \mathtt{tail} \rangle"] \\ A \times A^\omega \end{tikzcd}$$

Streams

(Circular) Definition

A stream over an alphabet A is made of a head $a \in A$, and another stream called the tail. A^ω is the greatest fixpoint of this definition!

Final coalgebra!

$$
\begin{tikzcd}
X \arrow[r, dashed, "\exists ! f"] \arrow[d, "\langle a, g \rangle"'] & A^\omega \arrow[d, "\langle \mathtt{head}, \mathtt{tail} \rangle"] \\
A \times X \arrow[r, dashed, "\mathtt{id} \times f"'] & A \times A^\omega
\end{tikzcd}
$$

$$
\begin{aligned}
\mathtt{head}(f(x)) &= a(x) \\
\mathtt{tail}(f(x)) &= f(g(x))
\end{aligned}
$$

Streams

(Circular) Definition

A stream over an alphabet A is made of a head $a \in A$, and another stream called the tail. A^ω is the greatest fixpoint of this definition!

Final coalgebra!

$$\begin{tikzcd}
X \arrow[r, dashed, "\exists! f"] \arrow[d, "\langle a, g\rangle"'] & A^\omega \arrow[d, bend left, "\langle \mathtt{head}, \mathtt{tail}\rangle"] \\
A \times X \arrow[r, dashed, "\mathtt{id} \times f"'] & A \times A^\omega \arrow[u, bend left, "\mathtt{cons}"]
\end{tikzcd}$$

$$\begin{aligned}
\mathtt{head}(f(x)) &= a(x) \\
\mathtt{tail}(f(x)) &= f(g(x))
\end{aligned}$$

Streams

(Circular) Definition

A stream over an alphabet A is made of a head $a \in A$, and another stream called the tail. A^ω is the greatest fixpoint of this definition!

Final coalgebra!

$$\begin{tikzcd}[column sep=large, row sep=large]
X \arrow[r, dashed, "\exists ! f"] \arrow[d, "\langle a, g\rangle"'] & A^\omega \arrow[d, bend left, "\langle \mathtt{head}, \mathtt{tail}\rangle"] \\
A \times X \arrow[r, dashed, "\mathrm{id} \times f"'] & A \times A^\omega \arrow[u, bend left, "\mathtt{cons}"]
\end{tikzcd}$$

$$\begin{aligned}
\mathtt{head}(f(x)) &= a(x) \\
\mathtt{tail}(f(x)) &= f(g(x))
\end{aligned}$$

$$A^\omega = \nu X.(A \times X)$$

Other coinductive types...

Other coinductive types...

- $\nu X.(A \times X \times X) =$ Infinite binary labelled trees

$$\begin{tikzcd}[column sep=small] & \boxed{a} \arrow[dl] \arrow[dr] & \\ T_1 & & T_2 \end{tikzcd} \qquad \mapsto \qquad \langle a, T_1, T_2 \rangle$$

Lattice μ-calculus

Lattice μ-terms are generated by the following grammar:

$$t := X \mid 1 \mid t \times t \mid 0 \mid t + t \mid \color{red}{\mu X.t} \mid \color{red}{\nu X.t}$$

Lattice μ-calculus

Lattice μ-terms are generated by the following grammar:

$$t := X \mid 1 \mid t \times t \mid 0 \mid t + t \mid \mu X.t \mid \nu X.t$$

Example

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

Lattice μ-calculus

Lattice μ-terms are generated by the following grammar:

$$t := X \mid 1 \mid t \times t \mid 0 \mid t + t \mid \mu X.t \mid \nu X.t$$

Example

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

Functorial interpretation

- $\times, +$ = Product / Coproduct;
- $0, 1$ = Initial / Final object;
- $\mu X.F(X), \nu X.F(X)$ = Initial F-algebra / Final F-coalgebra.

Lattice μ-calculus

Lattice μ-terms are generated by the following grammar:

$$t := X \mid 1 \mid t \times t \mid 0 \mid t + t \mid \mu X.t \mid \nu X.t$$

Example

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

Functorial interpretation

- $\times, +$ = Product / Coproduct;
- $0, 1$ = Initial / Final object;
- $\mu X.F(X), \nu X.F(X)$ = Initial F-algebra / Final F-coalgebra.

Definition

A category $\mathcal{C}$ is μ-bicomplete iff this interpretation makes sense in $\mathcal{C}$.

Game semantics (in $\mathcal{S}ets$)

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

Game semantics (in $\mathcal{S}ets$)

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

$\mathcal{S}(t)$:

Game semantics (in $\mathcal{Sets}$)

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

$$\mathcal{S}(t): \qquad X \quad =_\nu \quad A \times Y$$

Game semantics (in $\mathcal{Sets}$)

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

$$\mathcal{S}(t): \quad \begin{array}{lcl} X & =_\nu & A \times Y \\ A & =_\nu & \sum_{a \in A} 1 \end{array}$$

Game semantics (in $\mathcal{S}ets$)

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

$$\mathcal{S}(t): \quad \begin{array}{lll} X & =_\nu & A \times Y \\ A & =_\nu & \sum_{a \in A} 1 \\ Y & =_\mu & 1 + Z \end{array}$$

Game semantics (in $\mathcal{Sets}$)

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

$$\mathcal{S}(t): \quad \begin{array}{lll} X & =_\nu & A \times Y \\ A & =_\nu & \sum_{a \in A} 1 \\ Y & =_\mu & 1 + Z \\ Z & =_\mu & X \times Y \end{array}$$

Game semantics (in $\mathcal{S}ets$)

$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

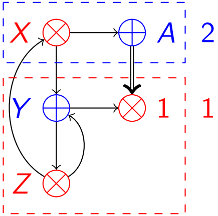
$$\mathcal{S}(t): \quad \begin{array}{lclc} X & =_\nu & A \times Y & (2) \\ A & =_\nu & \sum_{a \in A} 1 & (2) \\ Y & =_\mu & 1 + Z & (1) \\ Z & =_\mu & X \times Y & (1) \end{array}$$

$$\text{Priority}(V) \text{ is } \begin{cases} \text{even} & \text{, if } V =_\nu \ldots \\ \text{odd} & \text{, if } V =_\mu \ldots \end{cases}$$

Game semantics (in $\mathcal{S}ets$)

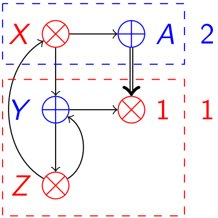
$$t = \nu X.(A \times \mu Y.(1 + X \times Y))$$

$$\mathcal{S}(t): \quad \begin{array}{lcll} X & =_\nu & A \times Y & (2) \\ A & =_\nu & \sum_{a \in A} 1 & (2) \\ Y & =_\mu & 1 + Z & (1) \\ Z & =_\mu & X \times Y & (1) \end{array} \quad \Rightarrow$$

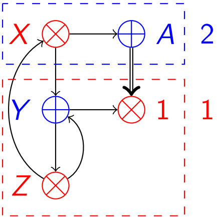


$$\text{Priority}(V) \text{ is } \begin{cases} \text{even} & \text{, if } V =_\nu \dots \\ \text{odd} & \text{, if } V =_\mu \dots \end{cases}$$

Parity games!



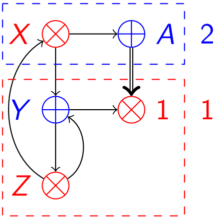
Parity games!



Player $\oplus$ wins if:

Game semantics (in $\mathcal{S}ets$)

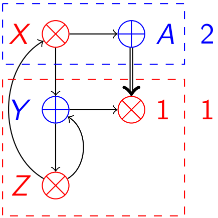
Parity games!



Player $\oplus$ wins if:

- At some point, Player $\otimes$ cannot play,

Parity games!

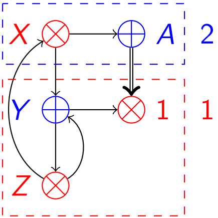


Player $\oplus$ wins if:

- At some point, Player $\otimes$ cannot play,
- Or the game is infinite, and the highest priority visited infinitely often is even.

Game semantics (in $\mathcal{S}ets$)

Parity games!



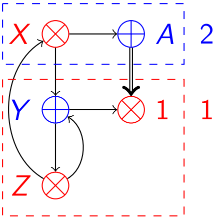
Player $\oplus$ wins if:

- At some point, Player $\otimes$ cannot play,
- Or the game is infinite, and the highest priority visited infinitely often is even.

Player $\otimes$ wins in the dual situation.

Game semantics (in *Sets*)

Parity games!



Player $\oplus$ wins if:

- At some point, Player $\otimes$ cannot play,
- Or the game is infinite, and the highest priority visited infinitely often is even.

Player $\otimes$ wins in the dual situation.

Theorem (Santocanale, 2002)

Solutions for variable V in $\mathcal{S}(t)$ $\simeq$ The set of deterministic winning strategies for $\oplus$ from position V.

Therefore, we have a combinatorial (dynamic) characterization of the μ-defined objects.

Curry–Howard correspondence

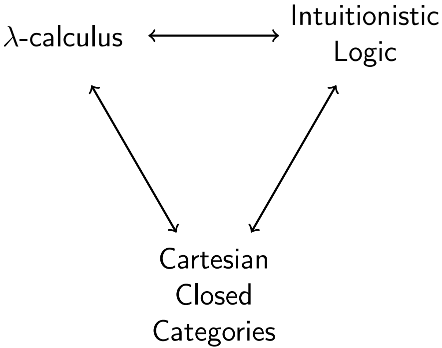
Goal

Find a "good" (dynamic) syntax for expressing (and computing) functions (arrows) between objects of this kind.

Curry–Howard correspondence

Goal

Find a "good" (dynamic) syntax for expressing (and computing) functions (arrows) between objects of this kind.



Curry–Howard correspondence

Goal

Find a "good" (dynamic) syntax for expressing (and computing) functions (arrows) between objects of this kind.

$$\begin{tikzcd} \mu\text{-calculus} \arrow[rr, leftrightarrow] \arrow[dr, leftrightarrow] & & \text{Intuitionistic Logic} \arrow[dl, leftrightarrow] \\ & \text{Cartesian Closed Categories} & \end{tikzcd}$$

Curry–Howard correspondence

Goal

Find a "good" (dynamic) syntax for expressing (and computing) functions (arrows) between objects of this kind.

$$\begin{tikzcd} \mu\text{-calculus} \arrow[rr, leftrightarrow] \arrow[dr, leftrightarrow] & & \text{Intuitionistic Logic} \arrow[dl, leftrightarrow] \\ & \mu\text{-bicomplete Categories} & \end{tikzcd}$$

Curry–Howard correspondence

Goal

Find a "good" (dynamic) syntax for expressing (and computing) functions (arrows) between objects of this kind.

$$\begin{tikzcd}
\mu\text{-calculus} \arrow[rr, leftrightarrow] \arrow[dr, leftrightarrow] & & \text{Circular Proofs} \arrow[dl, leftrightarrow] \\
& \mu\text{-bicomplete Categories} &
\end{tikzcd}$$

Inference rules (Gentzen style)

Axioms:	$\dfrac{}{0 \vdash t}\,\texttt{LAx}$	$\dfrac{}{t \vdash 1}\,\texttt{RAx}$	$\dfrac{}{t \vdash t}\,\texttt{Id}$
Product: (conjunction)	$\dfrac{s_i \vdash t}{s_0 \times s_1 \vdash t}\,\texttt{L}\times_i$	$\dfrac{s \vdash t_0 \quad s \vdash t_1}{s \vdash t_0 \times t_1}\,\texttt{R}\times$	
Coproduct: (disjunction)	$\dfrac{s_0 \vdash t \quad s_1 \vdash t}{s_0 + s_1 \vdash t}\,\texttt{L}+$	$\dfrac{s \vdash t_i}{s \vdash t_0 + t_1}\,\texttt{R}+_i$	
Fixpoint:	$\dfrac{F(X) \vdash t}{X \vdash t}\,\texttt{LFix}$	$\dfrac{s \vdash F(X)}{s \vdash X}\,\texttt{RFix}$	if $X = F(X)$
Cut:	$\dfrac{r \vdash s \quad s \vdash t}{r \vdash t}\,\texttt{Cut}$		

Categorical interpretation

Axions:

$$\frac{}{0 \xrightarrow{?_t} t}\,\texttt{LAx} \qquad \frac{}{t \xrightarrow{!_t} 1}\,\texttt{RAx} \qquad \frac{}{t \xrightarrow{\mathrm{id}_t} t}\,\texttt{Id}$$

Product: (conjunction)

$$\frac{s_i \xrightarrow{f} t}{s_0 \times s_1 \xrightarrow{\mathtt{pr}_i \cdot f} t}\,\mathtt{L}\times_i \qquad \frac{s \xrightarrow{f} t_0 \quad s \xrightarrow{g} t_1}{s \xrightarrow{\langle f, g\rangle} t_0 \times t_1}\,\mathtt{R}\times$$

Coproduct: (disjunction)

$$\frac{s_0 \xrightarrow{f} t \quad s_1 \xrightarrow{g} t}{s_0 + s_1 \xrightarrow{\{f, g\}} t}\,\mathtt{L}+ \qquad \frac{s \xrightarrow{f} t_i}{s \xrightarrow{f \cdot \mathtt{in}_i} t_0 + t_1}\,\mathtt{R}+_i$$

Fixpoint:

$$\frac{F(X) \xrightarrow{f} t}{X \xrightarrow{\zeta_X^{-1} \cdot f} t}\,\texttt{LFix} \qquad \frac{s \xrightarrow{f} F(X)}{s \xrightarrow{f \cdot \zeta_X} X}\,\texttt{RFix} \qquad \text{if } X =_\mu F(X)$$

$$\frac{F(X) \xrightarrow{f} t}{X \xrightarrow{\xi_X \cdot f} t}\,\texttt{LFix} \qquad \frac{s \xrightarrow{f} F(X)}{s \xrightarrow{f \cdot \xi_X^{-1}} X}\,\texttt{RFix} \qquad \text{if } X =_\nu F(X)$$

Cut:

$$\frac{r \xrightarrow{f} s \quad s \xrightarrow{g} t}{r \xrightarrow{f \cdot g} t}\,\texttt{Cut}$$

Primitive recursion

$N =_\mu 1 + N$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathrm{suc}\}} \mathbb{N}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

Let

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

Let

$$N \vdash N$$

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

Let

$$\frac{1 + N \vdash N}{N \vdash N}\ \mathtt{LFix}$$

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

Let

$$\dfrac{\dfrac{1 \vdash N \qquad\qquad N \vdash N}{1 + N \vdash N}\,\mathtt{L+}}{N \vdash N}\,\mathtt{LFix}$$

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

$$\dfrac{\dfrac{1 \vdash 1 + N}{1 \vdash N}\,\mathtt{RFix} \qquad N \vdash N}{\dfrac{1 + N \vdash N}{N \vdash N}\,\mathtt{LFix}}\,\mathtt{L}+$$

Let

$$\begin{aligned}\mathtt{double}(0) &= 0\\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n)))\end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{1 \vdash 1}{1 \vdash 1 + N}\ \mathtt{R}+_0}{1 \vdash N}\ \mathtt{RFix} \qquad N \vdash N}{1 + N \vdash N}\ \mathtt{L}+}{N \vdash N}\ \mathtt{LFix}$$

Let

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$\color{red}{1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{1 \vdash 1}\,\mathtt{RAx}}{1 \vdash 1 + N}\,\mathtt{R}+_0}{1 \vdash N}\,\mathtt{RFix} \qquad N \vdash N}{\dfrac{1 + N \vdash N}{N \vdash N}\,\mathtt{LFix}}\,\mathtt{L}+$$

Let

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{1 \vdash 1}\,\mathtt{RAx}}{1 \vdash 1 + N}\,\mathtt{R}+_0}{1 \vdash N}\,\mathtt{RFix} \qquad \dfrac{N \vdash 1 + N}{N \vdash N}\,\mathtt{RFix}}{\dfrac{1 + N \vdash N}{N \vdash N}\,\mathtt{LFix}}\,\mathtt{L}+$$

Let

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{1 \vdash 1}\,\mathtt{RAx}}{1 \vdash 1 + N}\,\mathtt{R}+_0}{1 \vdash N}\,\mathtt{RFix} \qquad \dfrac{\dfrac{N \vdash N}{N \vdash 1 + N}\,\mathtt{R}+_1}{N \vdash N}\,\mathtt{RFix}}{1 + N \vdash N}\,\mathtt{L}+}{N \vdash N}\,\mathtt{LFix}$$

Let

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$\color{red}{1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{1 \vdash 1}\,\mathtt{RAx}}{1 \vdash 1 + N}\,\mathtt{R}+_0}{1 \vdash N}\,\mathtt{RFix} \qquad \dfrac{\dfrac{\dfrac{\dfrac{N \vdash 1 + N}{N \vdash N}\,\mathtt{RFix}}{N \vdash 1 + N}\,\mathtt{R}+_1}{N \vdash N}\,\mathtt{RFix}}{\dfrac{1 + N \vdash N}{N \vdash N}\,\mathtt{LFix}}\,\mathtt{L}+$$

Let

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

$$\dfrac{\dfrac{\dfrac{}{1 \vdash 1}\,\mathtt{RAx}}{1 \vdash 1 + N}\,\mathtt{R}+_0}{1 \vdash N}\,\mathtt{RFix} \qquad \dfrac{\dfrac{\dfrac{\dfrac{N \vdash N}{N \vdash 1 + N}\,\mathtt{R}+_1}{N \vdash N}\,\mathtt{RFix}}{N \vdash 1 + N}\,\mathtt{R}+_1}{N \vdash N}\,\mathtt{RFix}$$

$$\dfrac{\dfrac{1 \vdash N \qquad N \vdash N}{1 + N \vdash N}\,\mathtt{L}+}{N \vdash N}\,\mathtt{LFix}$$

Let

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

Primitive recursion

$$N =_\mu 1 + N$$

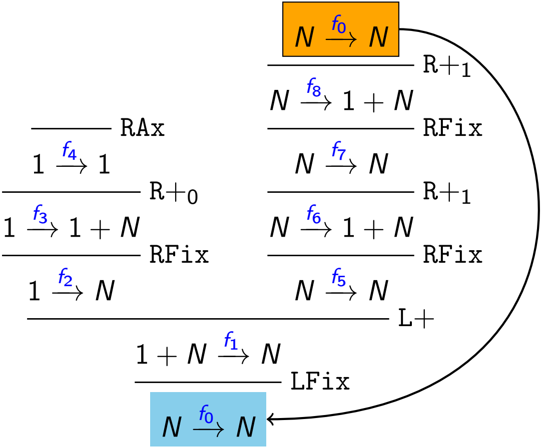
Solution:

$$1 + \mathbb{N} \xrightarrow{\{0,\mathtt{suc}\}} \mathbb{N}$$

Let

$$\begin{aligned} \mathtt{double}(0) &= 0 \\ \mathtt{double}(\mathtt{suc}(n)) &= \mathtt{suc}(\mathtt{suc}(\mathtt{double}(n))) \end{aligned}$$

$$\begin{prooftree}
\AxiomC{}
\RightLabel{$\mathtt{RAx}$}
\UnaryInfC{$1 \vdash 1$}
\RightLabel{$\mathrm{R}+_0$}
\UnaryInfC{$1 \vdash 1 + N$}
\RightLabel{$\mathtt{RFix}$}
\UnaryInfC{$1 \vdash N$}
\AxiomC{$\boxed{N \vdash N}$}
\RightLabel{$\mathrm{R}+_1$}
\UnaryInfC{$N \vdash 1 + N$}
\RightLabel{$\mathtt{RFix}$}
\UnaryInfC{$N \vdash N$}
\RightLabel{$\mathrm{R}+_1$}
\UnaryInfC{$N \vdash 1 + N$}
\RightLabel{$\mathtt{RFix}$}
\UnaryInfC{$N \vdash N$}
\RightLabel{$\mathrm{L}+$}
\BinaryInfC{$1 + N \vdash N$}
\RightLabel{$\mathtt{LFix}$}
\UnaryInfC{$\boxed{N \vdash N}$}
\end{prooftree}$$



$$
\left\{
\begin{array}{rcl}
f_8 & = & f_0 \cdot \mathtt{in}_1 \\
f_7 & = & f_8 \cdot \zeta_N \\
f_6 & = & f_7 \cdot \mathtt{in}_1 \\
f_5 & = & f_6 \cdot \zeta_N \\
\\
f_4 & = & !_1 \\
f_3 & = & f_4 \cdot \mathtt{in}_0 \\
f_2 & = & f_3 \cdot \zeta_N \\
\\
f_1 & = & \{f_2, f_5\} \\
f_0 & = & \zeta_N^{-1} \cdot f_1
\end{array}
\right\}
$$

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$\mathtt{RAx}$}
\UnaryInfC{$1 \xrightarrow{f_4} 1$}
\RightLabel{$\mathtt{R+}_0$}
\UnaryInfC{$1 \xrightarrow{f_3} 1+N$}
\RightLabel{$\mathtt{RFix}$}
\UnaryInfC{$1 \xrightarrow{f_2} N$}
\AxiomC{$\boxed{N \xrightarrow{f_0} N}$}
\RightLabel{$\mathtt{R+}_1$}
\UnaryInfC{$N \xrightarrow{f_8} 1+N$}
\RightLabel{$\mathtt{RFix}$}
\UnaryInfC{$N \xrightarrow{f_7} N$}
\RightLabel{$\mathtt{R+}_1$}
\UnaryInfC{$N \xrightarrow{f_6} 1+N$}
\RightLabel{$\mathtt{RFix}$}
\UnaryInfC{$N \xrightarrow{f_5} N$}
\RightLabel{$\mathtt{L+}$}
\BinaryInfC{$1+N \xrightarrow{f_1} N$}
\RightLabel{$\mathtt{LFix}$}
\UnaryInfC{$\boxed{N \xrightarrow{f_0} N}$}
\end{prooftree}
$$

$$
\left\{
\begin{array}{lcl}
f_8 & = & f_0 \cdot \mathtt{in}_1 \\
f_7 & = & f_8 \cdot \zeta_N \\
f_6 & = & f_7 \cdot \mathtt{in}_1 \\
f_5 & = & f_6 \cdot \zeta_N \\
\\
f_4 & = & !_1 \\
f_3 & = & f_4 \cdot \mathtt{in}_0 \\
f_2 & = & f_3 \cdot \zeta_N \\
\\
f_1 & = & \{f_2, f_5\} \\
f_0 & = & \zeta_N^{-1} \cdot f_1
\end{array}
\right\}
$$

Solution: $f_0 = \mathtt{double}$

Stream differential equations

$$\begin{aligned} X &=_\nu 2 \times X \times X \\ 2 &=_\nu 1 + 1 \end{aligned}$$

Stream differential equations

$$
\begin{aligned}
X &=_\nu 2 \times X \times X \\
2 &=_\nu 1 + 1
\end{aligned}
$$

Solution (Kupke–Rutten, 2012)

$$S \xrightarrow{\langle \mathtt{head}, \mathtt{even}, \mathtt{odd} \rangle} 2 \times S \times S$$

Stream differential equations

$$\begin{aligned} X &=_\nu 2 \times X \times X \\ 2 &=_\nu 1 + 1 \end{aligned}$$

Solution (Kupke–Rutten, 2012)

$$S \xrightarrow{\langle \mathtt{head}, \mathtt{even}, \mathtt{odd} \rangle} 2 \times S \times S$$

Thue-Morse stream

$$\begin{aligned} \sigma &= \langle 0, \sigma, \tau \rangle \\ \tau &= \langle 1, \tau, \sigma \rangle \end{aligned}$$

Stream differential equations

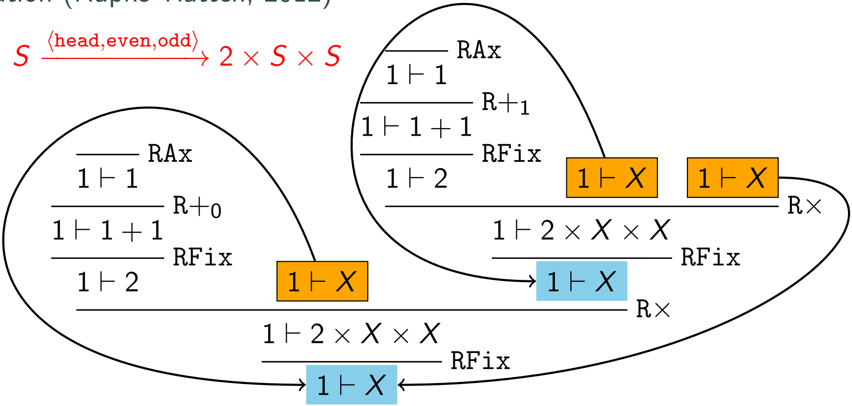
$$
\begin{aligned}
X &=_\nu 2 \times X \times X \\
2 &=_\nu 1 + 1
\end{aligned}
$$

Thue-Morse stream

$$
\begin{aligned}
\sigma &= \langle 0, \sigma, \tau \rangle \\
\tau &= \langle 1, \tau, \sigma \rangle
\end{aligned}
$$

Solution (Kupke–Rutten, 2012)

$$S \xrightarrow{\langle \texttt{head}, \texttt{even}, \texttt{odd} \rangle} 2 \times S \times S$$

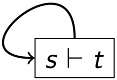


Non-valid circular proofs

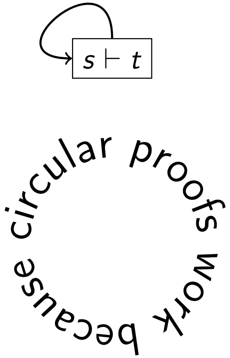
Non-valid circular proofs

$$\boxed{s \vdash t}$$

Non-valid circular proofs



circular proofs work because



$$
\begin{prooftree}
\AxiomC{}
\RightLabel{$\mathtt{RAx}$}
\UnaryInfC{$1 \vdash 1$}
\RightLabel{$\mathtt{R}{+}_0$}
\UnaryInfC{$1 \vdash 1 + 1$}
\RightLabel{$\mathtt{RFix}$}
\UnaryInfC{$1 \vdash 2$}
\AxiomC{$1 \vdash S$}
\AxiomC{}
\RightLabel{$\mathtt{Id}$}
\UnaryInfC{$S \vdash S$}
\RightLabel{$\mathtt{L}{\times}_1$}
\UnaryInfC{$2 \times S \vdash S$}
\RightLabel{$\mathtt{LFix}$}
\UnaryInfC{$S \vdash S$}
\RightLabel{$\mathtt{Cut}$}
\BinaryInfC{$1 \vdash S$}
\RightLabel{$\mathtt{R}{\times}$}
\BinaryInfC{$1 \vdash 2 \times S$}
\RightLabel{$\mathtt{RFix}$}
\UnaryInfC{$1 \vdash S$}
\end{prooftree}
$$

$$S \xrightarrow{\langle \mathtt{head},\mathtt{tail} \rangle} 2 \times S$$

$\boxed{s \vdash t}$

circular proofs work because

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{1 \vdash 1}\,\mathtt{RAx}}{1 \vdash 1 + 1}\,\mathtt{R}{+}_0}{1 \vdash 2}\,\mathtt{RFix} \qquad \dfrac{\boxed{1 \vdash S} \qquad \dfrac{\dfrac{\dfrac{}{S \vdash S}\,\mathtt{Id}}{2 \times S \vdash S}\,\mathtt{L}{\times}_1}{S \vdash S}\,\mathtt{LFix}}{1 \vdash S}\,\mathtt{Cut}}{1 \vdash 2 \times S}\,\mathtt{R}{\times}}{\boxed{1 \vdash S}}\,\mathtt{RFix}$$

$$S \xrightarrow{\langle \mathtt{head,tail} \rangle} 2 \times S$$

$$\dfrac{\dfrac{\boxed{\text{God exists}} \qquad \dfrac{\dfrac{}{\text{Bible says it is the word of God}}\,\mathtt{Fact} \qquad \boxed{\text{What Bible says is true}}}{\text{Bible is the word of God}}\,\mathtt{Cut}}{\boxed{\text{What Bible says is true}}}\,\mathtt{Cut} \qquad \dfrac{}{\text{Bible says that God exists}}\,\mathtt{Fact}}{\boxed{\text{God exists}}}\,\mathtt{Cut}$$

Guard condition

Idea: Each turn in a cycle must either read a part of an inductive input or write a part of a coinductive output.

Guard condition

Idea: Each turn in a cycle must either read a part of an inductive input or write a part of a coinductive output.

Definition

A path in Π has a left μ-trace if it

- *contains a left fixpoint rule, and the highest priority is odd;*
- *turns left at every cut.*

Guard condition

Idea: Each turn in a cycle must either read a part of an inductive input or write a part of a coinductive output.

Definition

A path in Π has a left μ-trace if it

- *contains a left fixpoint rule, and the highest priority is odd;*
- *turns left at every cut.*

Definition

A path in Π has a right ν-trace if it

- *contains a right fixpoint rule, and the highest priority is even;*
- *turns right at every cut.*

Guard conditions

The following are equivalent.

Guard conditions

The following are equivalent.

1. Every cycle in Π either has a left μ-trace or a right ν-trace.

Guard conditions

The following are equivalent.

1. Every cycle in Π either has a left μ-trace or a right ν-trace.
2. Every infinite path Γ in Π has a tail Γ' that has either a left μ-trace or a right ν-trace and every fixpoint rule in Γ' occurs infinitely often.

Guard conditions

The following are equivalent.

1. Every cycle in Π either has a left μ-trace or a right ν-trace.
2. Every infinite path Γ in Π has a tail Γ' that has either a left μ-trace or a right ν-trace and every fixpoint rule in Γ' occurs infinitely often.
3. Every strongly connected component of Π either has a left μ-trace or a right ν-trace.

Guard conditions

The following are equivalent.

1. Every cycle in Π either has a left μ-trace or a right ν-trace.
2. Every infinite path Γ in Π has a tail Γ' that has either a left μ-trace or a right ν-trace and every fixpoint rule in Γ' occurs infinitely often.
3. Every strongly connected component of Π either has a left μ-trace or a right ν-trace.

Definition

A circular proof is a finite pre-proof that satisfies the guard conditions.

Semantical results

Soundness Theorem (F.–S. 2013)

Every circular proof denotes a unique arrow of the free μ-bicomplete category $\mathcal{M}$.

Soundness Theorem (F.–S. 2013)

Every circular proof denotes a unique arrow of the free μ-bicomplete category $\mathcal{M}$.

Proof.

By induction on $\sharp(\Pi) = (\sharp_{\mathrm{L}}(\Pi) + \sharp_{\mathrm{R}}(\Pi), \mathrm{card}(\Pi))$.

Soundness Theorem (F.–S. 2013)

Every circular proof denotes a unique arrow of the free μ-bicomplete category $\mathcal{M}$.

Proof.

By induction on $\sharp(\Pi) = (\sharp_{\mathrm{L}}(\Pi) + \sharp_{\mathrm{R}}(\Pi), \mathrm{card}(\Pi))$.

- If Π is not strongly connected: we can split Π in two parts Π_1, Π_2 s.t. $\mathrm{card}(\Pi_1), \mathrm{card}(\Pi_2) < \mathrm{card}(\Pi)$.

Semantical results

Soundness Theorem (F.–S. 2013)

Every circular proof denotes a unique arrow of the free μ-bicomplete category $\mathcal{M}$.

Proof.

By induction on $\sharp(\Pi) = (\sharp_{\mathtt{L}}(\Pi) + \sharp_{\mathtt{R}}(\Pi), \mathsf{card}(\Pi))$.

- If Π is not strongly connected: we can split Π in two parts Π_1, Π_2 s.t. $\mathsf{card}(\Pi_1), \mathsf{card}(\Pi_2) < \mathsf{card}(\Pi)$.We then *glue* the two solutions together using the Bekič Lemma.

Soundness Theorem (F.–S. 2013)

Every circular proof denotes a unique arrow of the free μ-bicomplete category $\mathcal{M}$.

Proof.

By induction on $\sharp(\Pi) = (\sharp_{\mathrm{L}}(\Pi) + \sharp_{\mathrm{R}}(\Pi), \mathrm{card}(\Pi))$.

- If Π is not strongly connected: we can split Π in two parts Π_1, Π_2 s.t. $\mathrm{card}(\Pi_1), \mathrm{card}(\Pi_2) < \mathrm{card}(\Pi)$.We then *glue* the two solutions together using the Bekič Lemma.
- If Π is strongly connected: take a cycle Γ that covers Π.

Soundness Theorem (F.–S. 2013)

Every circular proof denotes a unique arrow of the free μ-bicomplete category $\mathcal{M}$.

Proof.

By induction on $\sharp(\Pi) = (\sharp_{\mathrm{L}}(\Pi) + \sharp_{\mathrm{R}}(\Pi), \mathrm{card}(\Pi))$.

- If Π is not strongly connected: we can split Π in two parts Π_1, Π_2 s.t. $\mathrm{card}(\Pi_1), \mathrm{card}(\Pi_2) < \mathrm{card}(\Pi)$.We then *glue* the two solutions together using the Bekič Lemma.
- If Π is strongly connected: take a cycle Γ that covers Π. If Γ has a left μ-trace, split Π in parts Π_i s.t. $\forall i, \sharp_{\mathrm{L}}(\Pi_i) < \sharp(\Pi)$.

Soundness Theorem (F.–S. 2013)

Every circular proof denotes a unique arrow of the free μ-bicomplete category $\mathcal{M}$.

Proof.

By induction on $\sharp(\Pi) = (\sharp_{\mathrm{L}}(\Pi) + \sharp_{\mathrm{R}}(\Pi), \mathrm{card}(\Pi))$.

- If Π is not strongly connected: we can split Π in two parts Π_1, Π_2 s.t. $\mathrm{card}(\Pi_1), \mathrm{card}(\Pi_2) < \mathrm{card}(\Pi)$.We then *glue* the two solutions together using the Bekič Lemma.
- If Π is strongly connected: take a cycle Γ that covers Π. If Γ has a left μ-trace, split Π in parts Π_i s.t. $\forall i, \sharp_{\mathrm{L}}(\Pi_i) < \sharp(\Pi)$. Then *glue* the parts together with the Yoneda Lemma.

Semantical results

Soundness Theorem (F.–S. 2013)

Every circular proof denotes a unique arrow of the free μ-bicomplete category $\mathcal{M}$.

Proof.

By induction on $\sharp(\Pi) = (\sharp_{\mathrm{L}}(\Pi) + \sharp_{\mathrm{R}}(\Pi), \mathrm{card}(\Pi))$.

- If Π is not strongly connected: we can split Π in two parts Π_1, Π_2 s.t. $\mathrm{card}(\Pi_1), \mathrm{card}(\Pi_2) < \mathrm{card}(\Pi)$.We then *glue* the two solutions together using the Bekič Lemma.
- If Π is strongly connected: take a cycle Γ that covers Π. If Γ has a left μ-trace, split Π in parts Π_i s.t. $\forall i, \sharp_{\mathrm{L}}(\Pi_i) < \sharp(\Pi)$. Then *glue* the parts together with the Yoneda Lemma. If Γ as a right ν-trace, same reasoning.

Semantical results

Fullness Theorem (F.–S. 2013)

Every arrow $f : s \to t$ of $\mathcal{M}$ is the solution of a circular proof.

Fullness Theorem (F.–S. 2013)

Every arrow $f : s \to t$ of $\mathcal{M}$ is the solution of a circular proof.

Proof.
Obvious for most diagrams (by contruction of the rules).

Semantical results

Fullness Theorem (F.–S. 2013)

Every arrow $f : s \to t$ of $\mathcal{M}$ is the solution of a circular proof.

Proof.

Obvious for most diagrams (by contruction of the rules).

Except for this one!

$$\begin{tikzcd}
F(x) \arrow[r, dashed, "F(f)"] \arrow[d, "\zeta_x"'] & F(a) \arrow[d, "\alpha = [\![\Pi]\!]"] \\
x \arrow[r, dashed, "f"'] & a
\end{tikzcd}$$

$$f = \zeta_x^{-1} \cdot \; F(f) \cdot \alpha$$

Fullness Theorem (F.–S. 2013)

Every arrow $f : s \to t$ of $\mathcal{M}$ is the solution of a circular proof.

Proof.

Obvious for most diagrams (by contruction of the rules).

Except for this one!

$$\begin{tikzcd} F(x) \arrow[r, dashed, "F(f)"] \arrow[d, "\zeta_x"'] & F(a) \arrow[d, "\alpha = [\![\Pi]\!]"] \\ x \arrow[r, dashed, "f"'] & a \end{tikzcd}$$

$$f = \zeta_x^{-1} \cdot F(f) \cdot \alpha$$

$$\dfrac{\dfrac{\begin{array}{c}\boxed{x \vdash a}\\ \vdots\ \texttt{Copycat}\\ F(x) \vdash F(a)\end{array} \qquad \begin{array}{c}\vdots\ \Pi\\ F(a) \vdash a\end{array}}{F(x) \vdash a}\,\texttt{Cut}}{\boxed{x \vdash a}}\,\texttt{LFix}$$

Semantical results

Fullness Theorem (F.–S. 2013)

Every arrow $f : s \to t$ of $\mathcal{M}$ is the solution of a circular proof.

Proof.

Obvious for most diagrams (by contruction of the rules).

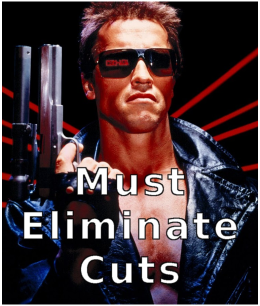
Except for this one!

$$\begin{tikzcd}
F(x) & & F(a) \arrow[ll, dashed, "F(f)"'] \\
x \arrow[u, "\xi_x"] & & a \arrow[u, "\alpha = [\![\Pi]\!]"'] \arrow[ll, dashed, "f"]
\end{tikzcd}$$

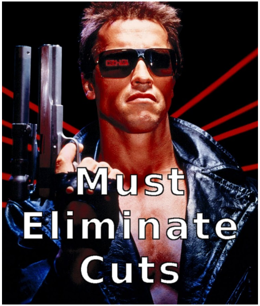
$$f = \alpha \cdot F(f) \cdot \xi_x^{-1}$$

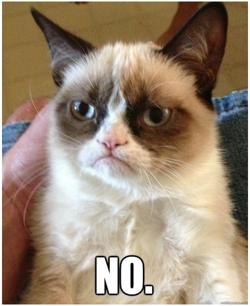
$$\dfrac{\dfrac{\begin{array}{c}\vdots\;\Pi\\ a \vdash F(a)\end{array} \qquad \begin{array}{c}\boxed{a \vdash x}\\ \vdots\;\texttt{Copycat}\\ F(a) \vdash F(x)\end{array}}{a \vdash F(x)}\,\texttt{Cut}}{\boxed{a \vdash x}}\,\texttt{RFix}$$

Cut-elimination



Cut-elimination





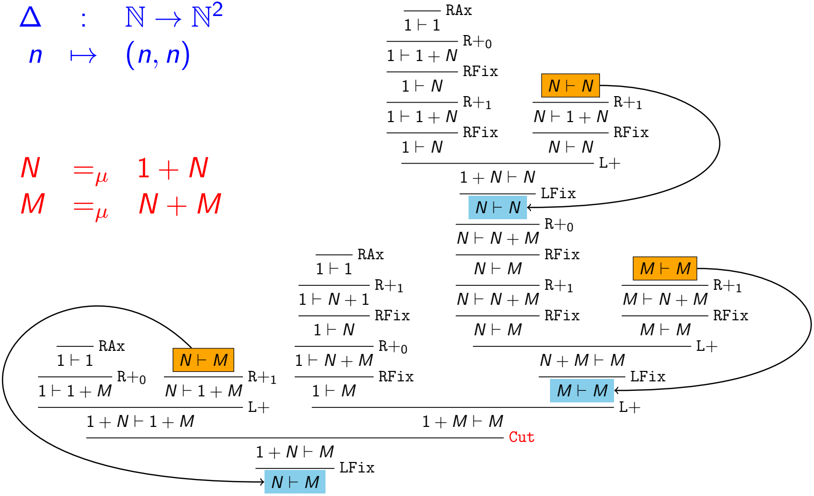
Theorem (Santocanale, 2001)

There is no cut-free circular proof whose interpretation in $\mathcal{S}ets$ is the diagonal $\Delta : \mathbb{N} \to \mathbb{N}^2$.

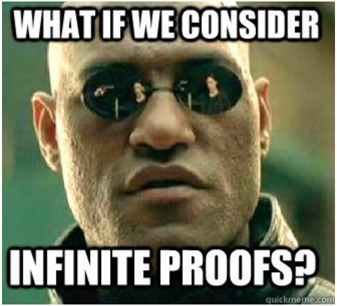
Diagonal map (with cuts)

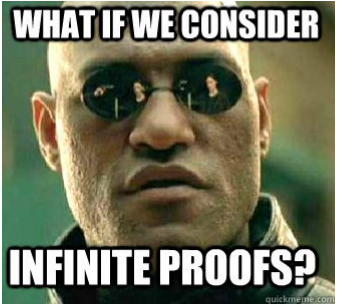
$$\begin{array}{rcl} \Delta & : & \mathbb{N} \to \mathbb{N}^2 \\ n & \mapsto & (n, n) \end{array}$$

$$\begin{array}{rcl} N & =_\mu & 1 + N \\ M & =_\mu & N + M \end{array}$$



Cut-elimination





$\Longrightarrow$

Tape automaton

Strategy: "Push" the cuts away from the root.

Tape automaton

Strategy: "Push" the cuts away from the root.

Main difficulties:

- We must use lazy evaluation (hence the output is infinite).

Tape automaton

Strategy: "Push" the cuts away from the root.

Main difficulties:

- We must use lazy evaluation (hence the output is infinite).
- Cut VS Cut

$$\dfrac{\dfrac{t_0 \vdash t_1 \quad t_1 \vdash t_2}{t_0 \vdash t_2}\,\texttt{Cut} \qquad t_2 \vdash t_3}{t_0 \vdash t_3}\,\texttt{Cut}$$

Tape automaton

Strategy: "Push" the cuts away from the root.

Main difficulties:

- We must use lazy evaluation (hence the output is infinite).
- Cut VS Cut

$$\dfrac{\dfrac{t_0 \vdash t_1 \quad t_1 \vdash t_2}{t_0 \vdash t_2}\,\texttt{Cut} \qquad t_2 \vdash t_3}{t_0 \vdash t_3}\,\texttt{Cut} \quad \begin{array}{c}\Rightarrow \\ \Leftarrow\end{array} \quad \dfrac{t_0 \vdash t_1 \qquad \dfrac{t_1 \vdash t_2 \quad t_2 \vdash t_3}{t_1 \vdash t_3}\,\texttt{Cut}}{t_0 \vdash t_3}\,\texttt{Cut}$$

Tape automaton

Strategy: "Push" the cuts away from the root.

Main difficulties:

- We must use lazy evaluation (hence the output is infinite).
- Cut VS Cut

$$\dfrac{\dfrac{t_0 \vdash t_1 \quad t_1 \vdash t_2}{t_0 \vdash t_2}\,\texttt{Cut} \qquad t_2 \vdash t_3}{t_0 \vdash t_3}\,\texttt{Cut} \quad \begin{array}{c}\Rightarrow \\ \Leftarrow\end{array} \quad \dfrac{t_0 \vdash t_1 \qquad \dfrac{t_1 \vdash t_2 \quad t_2 \vdash t_3}{t_1 \vdash t_3}\,\texttt{Cut}}{t_0 \vdash t_3}\,\texttt{Cut}$$

$\Downarrow$ Merge

$$\dfrac{t_0 \vdash t_1 \quad t_1 \vdash t_2 \quad t_2 \vdash t_3}{t_0 \vdash t_3}\,\texttt{Cut}$$

Tape automaton

Definition

A *tape* *is a finite list* $M := [u_1, \ldots, u_n]$ *of composable vertices of* Π.

Tape automaton

Definition

A *tape* *is a finite list* $M := [u_1, \ldots, u_n]$ *of composable vertices of* Π.

Cut Man - A tape automaton

Definition

A *tape* *is a finite list* $M := [u_1, \ldots, u_n]$ *of composable vertices of* Π.

Cut Man - A tape automaton

- Finite state machine (over a circular proof Π).

Tape automaton

Definition

A *tape* *is a finite list* $M := [u_1, \ldots, u_n]$ *of composable vertices of* Π.

Cut Man - A tape automaton

- Finite state machine (over a circular proof Π).
- Carries a tape (of states) in memory.

Tape automaton

Definition

A *tape* *is a finite list* $M := [u_1, \ldots, u_n]$ *of composable vertices of* Π.

Cut Man - A tape automaton

- Finite state machine (over a circular proof Π).
- Carries a tape (of states) in memory.
- Outputs a branch (chosen nondeterministically) of the cut-free infinite proof tree.

Commutative reductions (left)

$$\dfrac{\dfrac{}{0 \vdash t_1}\,\texttt{LAx} \qquad t_1 \vdash t_2 \qquad \cdots}{0 \vdash t_n}\,\texttt{Cut}$$

Commutative reductions (left)

$$\dfrac{\dfrac{}{0 \vdash t_1}\,\mathtt{LAx} \qquad t_1 \vdash t_2 \qquad \cdots}{0 \vdash t_n}\,\mathtt{Cut} \quad \xRightarrow{\text{LFlip}} \quad \dfrac{}{0 \vdash t_n}\,\mathtt{LAx}$$

Commutative reductions (left)

$$\dfrac{\dfrac{}{0 \vdash t_1}\,\texttt{LAx} \quad t_1 \vdash t_2 \quad \cdots}{0 \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{}{0 \vdash t_n}\,\texttt{LAx}$$

$$\dfrac{\dfrac{F(X) \vdash t_1}{X \vdash t_1}\,\texttt{LFix} \quad t_1 \vdash t_2 \quad \cdots}{X \vdash t_n}\,\texttt{Cut}$$

Commutative reductions (left)

$$\dfrac{\dfrac{}{0 \vdash t_1}\,\texttt{LAx} \qquad t_1 \vdash t_2 \qquad \cdots}{0 \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{}{0 \vdash t_n}\,\texttt{LAx}$$

$$\dfrac{\dfrac{F(X) \vdash t_1}{X \vdash t_1}\,\texttt{LFix} \qquad t_1 \vdash t_2 \qquad \cdots}{X \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{\dfrac{F(X) \vdash t_1 \qquad t_1 \vdash t_2 \qquad \cdots}{X \vdash t_n}\,\texttt{Cut}}{X \vdash t_n}\,\texttt{LFix}$$

Commutative reductions (left)

$$\dfrac{\dfrac{}{0 \vdash t_1}\,\mathtt{LAx} \qquad t_1 \vdash t_2 \qquad \cdots}{0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{}{0 \vdash t_n}\,\mathtt{LAx}$$

$$\dfrac{\dfrac{F(X) \vdash t_1}{X \vdash t_1}\,\mathtt{LFix} \qquad t_1 \vdash t_2 \qquad \cdots}{X \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{\dfrac{F(X) \vdash t_1 \qquad t_1 \vdash t_2 \qquad \cdots}{X \vdash t_n}\,\mathtt{Cut}}{X \vdash t_n}\,\mathtt{LFix}$$

$$\dfrac{\dfrac{s_k \vdash t_1}{s_0 \times s_1 \vdash t_1}\,\mathrm{L}\times_k \qquad t_1 \vdash t_2 \qquad \cdots}{s_0 \times s_1 \vdash t_n}\,\mathtt{Cut}$$

Commutative reductions (left)

$$\dfrac{\dfrac{}{0 \vdash t_1}\,\texttt{LAx} \quad t_1 \vdash t_2 \quad \cdots}{0 \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{}{0 \vdash t_n}\,\texttt{LAx}$$

$$\dfrac{\dfrac{F(X) \vdash t_1}{X \vdash t_1}\,\texttt{LFix} \quad t_1 \vdash t_2 \quad \cdots}{X \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{\dfrac{F(X) \vdash t_1 \quad t_1 \vdash t_2 \quad \cdots}{X \vdash t_n}\,\texttt{Cut}}{X \vdash t_n}\,\texttt{LFix}$$

$$\dfrac{\dfrac{s_k \vdash t_1}{s_0 \times s_1 \vdash t_1}\,\mathrm{L}{\times}_k \quad t_1 \vdash t_2 \quad \cdots}{s_0 \times s_1 \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{\dfrac{s_k \vdash t_1 \quad t_1 \vdash t_2 \quad \cdots}{s_k \vdash t_n}\,\texttt{Cut}}{s_0 \times s_1 \vdash t_n}\,\mathrm{L}{\times}_k$$

Commutative reductions (left)

$$\dfrac{\dfrac{}{0 \vdash t_1}\,\texttt{LAx} \qquad t_1 \vdash t_2 \qquad \cdots}{0 \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{}{0 \vdash t_n}\,\texttt{LAx}$$

$$\dfrac{\dfrac{F(X) \vdash t_1}{X \vdash t_1}\,\texttt{LFix} \qquad t_1 \vdash t_2 \qquad \cdots}{X \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{\dfrac{F(X) \vdash t_1 \qquad t_1 \vdash t_2 \qquad \cdots}{X \vdash t_n}\,\texttt{Cut}}{X \vdash t_n}\,\texttt{LFix}$$

$$\dfrac{\dfrac{s_k \vdash t_1}{s_0 \times s_1 \vdash t_1}\,\texttt{L}\times_k \qquad t_1 \vdash t_2 \qquad \cdots}{s_0 \times s_1 \vdash t_n}\,\texttt{Cut} \quad \overset{\text{LFlip}}{\Longrightarrow} \quad \dfrac{\dfrac{s_k \vdash t_1 \qquad t_1 \vdash t_2 \qquad \cdots}{s_k \vdash t_n}\,\texttt{Cut}}{s_0 \times s_1 \vdash t_n}\,\texttt{L}\times_k$$

$$\dfrac{\dfrac{s_0 \vdash t_1 \qquad s_1 \vdash t_1}{s_0 + s_1 \vdash t_1}\,\texttt{L+} \qquad t_1 \vdash t_2 \qquad \cdots}{s_0 + s_1 \vdash t_n}\,\texttt{Cut}$$

Commutative reductions (left)

$$\dfrac{\dfrac{}{0 \vdash t_1}\,\texttt{LAx} \qquad t_1 \vdash t_2 \qquad \cdots}{0 \vdash t_n}\,\texttt{Cut} \quad \xRightarrow{\text{LFlip}} \quad \dfrac{}{0 \vdash t_n}\,\texttt{LAx}$$

$$\dfrac{\dfrac{F(X) \vdash t_1}{X \vdash t_1}\,\texttt{LFix} \qquad t_1 \vdash t_2 \qquad \cdots}{X \vdash t_n}\,\texttt{Cut} \quad \xRightarrow{\text{LFlip}} \quad \dfrac{\dfrac{F(X) \vdash t_1 \qquad t_1 \vdash t_2 \qquad \cdots}{X \vdash t_n}\,\texttt{Cut}}{X \vdash t_n}\,\texttt{LFix}$$

$$\dfrac{\dfrac{s_k \vdash t_1}{s_0 \times s_1 \vdash t_1}\,\mathrm{L}\times_k \qquad t_1 \vdash t_2 \qquad \cdots}{s_0 \times s_1 \vdash t_n}\,\texttt{Cut} \quad \xRightarrow{\text{LFlip}} \quad \dfrac{\dfrac{s_k \vdash t_1 \qquad t_1 \vdash t_2 \qquad \cdots}{s_k \vdash t_n}\,\texttt{Cut}}{s_0 \times s_1 \vdash t_n}\,\mathrm{L}\times_k$$

$$\dfrac{\dfrac{s_0 \vdash t_1 \qquad s_1 \vdash t_1}{s_0 + s_1 \vdash t_1}\,\mathrm{L}+ \qquad t_1 \vdash t_2 \qquad \cdots}{s_0 + s_1 \vdash t_n}\,\texttt{Cut} \quad \xRightarrow{\text{LFlip}} \quad \dfrac{\dfrac{s_0 \vdash t_1 \qquad t_1 \vdash t_2 \qquad \cdots}{s_0 \vdash t_n}\,\texttt{Cut} \qquad \dfrac{s_1 \vdash t_1 \qquad t_1 \vdash t_2 \qquad \cdots}{s_1 \vdash t_n}\,\texttt{Cut}}{s_0 + s_1 \vdash t_n}\,\mathrm{L}+$$

Commutative reductions (right)

$$\dfrac{\cdots \quad t_{n-2} \vdash t_{n-1} \quad \dfrac{}{t_{n-1} \vdash 1}\,\mathtt{RAx}}{t_0 \vdash 1}\,\mathtt{Cut} \quad \overset{\text{RFlip}}{\Longrightarrow} \quad \dfrac{}{t_0 \vdash 1}\,\mathtt{RAx}$$

$$\dfrac{\cdots \quad t_{n-2} \vdash t_{n-1} \quad \dfrac{t_{n-1} \vdash F(X)}{t_{n-1} \vdash X}\,\mathtt{RFix}}{t_0 \vdash X}\,\mathtt{Cut} \quad \overset{\text{RFlip}}{\Longrightarrow} \quad \dfrac{\dfrac{\cdots \quad t_{n-2} \vdash t_{n-1} \quad t_{n-1} \vdash F(X)}{t_0 \vdash F(X)}\,\mathtt{Cut}}{t_0 \vdash X}\,\mathtt{RFix}$$

$$\dfrac{\cdots \quad t_{n-2} \vdash t_{n-1} \quad \dfrac{t_{n-1} \vdash s_k}{t_{n-1} \vdash s_0 + s_1}\,\mathtt{R}+_k}{t_0 \vdash s_0 + s_1}\,\mathtt{Cut} \quad \overset{\text{RFlip}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{n-2} \vdash t_{n-1} \quad \dfrac{t_{n-1} \vdash s_k}{t_{n-1} \vdash s_k}\,\mathtt{Cut}}{t_0 \vdash s_0 + s_1}\,\mathtt{R}+_k$$

$$\dfrac{\cdots \quad t_{n-2} \vdash t_{n-1} \quad \dfrac{t_{n-1} \vdash s_0 \quad t_{n-1} \vdash s_1}{t_{n-1} \vdash s_0 \times s_1}\,\mathtt{R}\times}{t_0 \vdash s_0 \times s_1}\,\mathtt{Cut} \quad \overset{\text{RFlip}}{\Longrightarrow} \quad \dfrac{\dfrac{\cdots \quad t_{n-2} \vdash t_{n-1} \quad t_{n-1} \vdash s_0}{t_0 \vdash s_0}\,\mathtt{Cut} \quad \dfrac{\cdots \quad t_{n-2} \vdash t_{n-1} \quad t_{n-1} \vdash s_1}{t_0 \vdash s_1}\,\mathtt{Cut}}{t_0 \vdash s_0 \times s_1}\,\mathtt{R}\times$$

Elimination of identities

$$\dfrac{\cdots \quad t_{i-1} \vdash s \quad \dfrac{}{s \vdash s}\,\mathtt{Id} \quad s \vdash t_{i+2} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

Elimination of identities

$$\dfrac{\cdots \quad t_{i-1} \vdash s \quad \dfrac{}{s \vdash s}\,\mathtt{Id} \quad s \vdash t_{i+2} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

$$\Downarrow \text{IdElim}$$

$$\dfrac{\cdots \quad t_{i-1} \vdash s \quad s \vdash t_{i+2} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

$\text{IdElim}(M, i) = \text{Remove } u_i \text{ from } M.$

Essential reductions

Otherwise, $M = [\mathtt{R}\ldots\mathtt{RL}\ldots\mathtt{L}]$.

Essential reductions

Otherwise, $M = [\mathtt{R} \ldots \mathtt{RL} \ldots \mathtt{L}]$.

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_0 \quad t_{i-1} \vdash s_1}{t_{i-1} \vdash s_0 \times s_1}\,\mathrm{R}\times \quad \dfrac{s_k \vdash t_{i+1}}{s_0 \times s_1 \vdash t_{i+1}}\,\mathrm{L}\times_k \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

Essential reductions

Otherwise, $M = [\mathtt{R}\ldots\mathtt{RL}\ldots\mathtt{L}]$.

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_0 \quad t_{i-1} \vdash s_1}{t_{i-1} \vdash s_0 \times s_1}\,\mathtt{R}\times \quad \dfrac{s_k \vdash t_{i+1}}{s_0 \times s_1 \vdash t_{i+1}}\,\mathtt{L}\times_k \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash s_k \quad s_k \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

Essential reductions

Otherwise, $M = [\mathtt{R}\ldots\mathtt{RL}\ldots\mathtt{L}]$.

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_0 \quad t_{i-1} \vdash s_1}{t_{i-1} \vdash s_0 \times s_1}\,\mathtt{R}\times \quad \dfrac{s_k \vdash t_{i+1}}{s_0 \times s_1 \vdash t_{i+1}}\,\mathtt{L}\times_k \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash s_k \quad s_k \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_k}{t_{i-1} \vdash s_0 + s_1}\,\mathtt{R}+_k \quad \dfrac{s_0 \vdash t_{i+1} \quad s_1 \vdash t_{i+1}}{s_0 + s_1 \vdash t_{i+1}}\,\mathtt{L}+ \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

Otherwise, $M = [\mathtt{R}\ldots\mathtt{RL}\ldots\mathtt{L}]$.

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_0 \quad t_{i-1} \vdash s_1}{t_{i-1} \vdash s_0 \times s_1}\,\mathtt{R}\times \quad \dfrac{s_k \vdash t_{i+1}}{s_0 \times s_1 \vdash t_{i+1}}\,\mathtt{L}\times_k \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash s_k \quad s_k \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_k}{t_{i-1} \vdash s_0 + s_1}\,\mathtt{R}+_k \quad \dfrac{s_0 \vdash t_{i+1} \quad s_1 \vdash t_{i+1}}{s_0 + s_1 \vdash t_{i+1}}\,\mathtt{L}+ \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash s_k \quad s_k \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

Otherwise, $M = [\mathtt{R} \ldots \mathtt{RL} \ldots \mathtt{L}]$.

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_0 \quad t_{i-1} \vdash s_1}{t_{i-1} \vdash s_0 \times s_1}\,\mathtt{R}\times \quad \dfrac{s_k \vdash t_{i+1}}{s_0 \times s_1 \vdash t_{i+1}}\,\mathtt{L}\times_k \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash s_k \quad s_k \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_k}{t_{i-1} \vdash s_0 + s_1}\,\mathtt{R}+_k \quad \dfrac{s_0 \vdash t_{i+1} \quad s_1 \vdash t_{i+1}}{s_0 + s_1 \vdash t_{i+1}}\,\mathtt{L}+ \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash s_k \quad s_k \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash F(X)}{t_{i-1} \vdash X}\,\mathtt{RFix} \quad \dfrac{F(X) \vdash t_{i+1}}{X \vdash t_{i+1}}\,\mathtt{LFix} \quad \cdots}{t_1 \vdash t_n}\,\mathtt{Cut}$$

Essential reductions

Otherwise, $M = [\mathtt{R} \ldots \mathtt{RL} \ldots \mathtt{L}]$.

$$\dfrac{\cdots \quad \dfrac{\dfrac{t_{i-1} \vdash s_0 \quad t_{i-1} \vdash s_1}{t_{i-1} \vdash s_0 \times s_1}\,\mathtt{R}\times \quad \dfrac{s_k \vdash t_{i+1}}{s_0 \times s_1 \vdash t_{i+1}}\,\mathtt{L}\times_k}{} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash s_k \quad s_k \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash s_k}{t_{i-1} \vdash s_0 + s_1}\,\mathtt{R}+_k \quad \dfrac{s_0 \vdash t_{i+1} \quad s_1 \vdash t_{i+1}}{s_0 + s_1 \vdash t_{i+1}}\,\mathtt{L}+ \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash s_k \quad s_k \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

$$\dfrac{\cdots \quad \dfrac{t_{i-1} \vdash F(X)}{t_{i-1} \vdash X}\,\mathtt{RFix} \quad \dfrac{F(X) \vdash t_{i+1}}{X \vdash t_{i+1}}\,\mathtt{LFix} \quad \cdots}{t_1 \vdash t_n}\,\mathtt{Cut} \quad \overset{\text{Reduce}}{\Longrightarrow} \quad \dfrac{\cdots \quad t_{i-1} \vdash F(X) \quad F(X) \vdash t_{i+1} \quad \cdots}{t_0 \vdash t_n}\,\mathtt{Cut}$$

Cut-elimination algorithm

Cut-elimination algorithm

- Internal phase: Perform internal transformations (Merge, IdElim, Reduce) while you can't do anything else.

Cut-elimination algorithm

- Internal phase: Perform internal transformations (Merge, IdElim, Reduce) while you can't do anything else.
- Production phase: Build a part of the output tree (LFlip, RFlip, IdOut) whenever you can!

Cut-elimination algorithm

- Internal phase: Perform internal transformations (Merge, IdElim, Reduce) while you can't do anything else.
- Production phase: Build a part of the output tree (LFlip, RFlip, IdOut) whenever you can!
- Repeat forever...

Cut-elimination algorithm

- Internal phase: Perform internal transformations (Merge, IdElim, Reduce) while you can't do anything else.
- Production phase: Build a part of the output tree (LFlip, RFlip, IdOut) whenever you can!
- Repeat forever...

Theorem (F.–S., 2013)

For every input tape M, the internal phase halts!

Proof. Suppose it does not halt...

$$
\begin{array}{rcl}
M_1 & = & [\; u_{11} \quad u_{12} \quad u_{13} \;] \\
\text{Merge} \Downarrow & & \\
M_2 & = & [\; u_{21} \quad u_{22} \quad u_{23} \quad u_{24} \;] \\
\text{Merge} \Downarrow & & \\
M_3 & = & [\; u_{31} \quad u_{32} \quad u_{33} \quad u_{34} \quad u_{35} \;] \\
\text{Reduce} \Downarrow & & \\
M_4 & = & [\; u_{41} \quad u_{42} \quad u_{43} \quad u_{44} \quad u_{45} \;] \\
\text{IdElim} \Downarrow & & \\
M_5 & = & [\; u_{51} \quad u_{52} \quad u_{53} \quad u_{54} \;] \\
\text{Reduce} \Downarrow & & \\
M_6 & = & [\; u_{61} \quad u_{62} \quad u_{63} \quad u_{64} \;] \\
 & \vdots &
\end{array}
$$

Proof. Suppose it does not halt...

$$
\begin{tikzcd}
M_1 \arrow[d, Rightarrow, "\text{Merge}"'] & = & {[\; u_{11}} & u_{12} \arrow[d, "1"'] \arrow[dr, "2"] & {u_{13} \;]} & & \\
M_2 \arrow[d, Rightarrow, "\text{Merge}"'] & = & {[\; u_{21}} \arrow[d, "1"'] \arrow[dr, "2"] & u_{22} & u_{23} & {u_{24} \;]} & \\
M_3 \arrow[d, Rightarrow, "\text{Reduce}"'] & = & {[\; u_{31}} & u_{32} \arrow[d, "0"] & u_{33} \arrow[d, "0"] & u_{34} & {u_{35} \;]} \\
M_4 \arrow[d, Rightarrow, "\text{IdElim}"'] & = & {[\; u_{41}} & u_{42} & u_{43} & u_{44} & {u_{45} \;]} \\
M_5 \arrow[d, Rightarrow, "\text{Reduce}"'] & = & {[\; u_{51}} & u_{52} & u_{53} \arrow[d, "0"] & {u_{54} \;]} \arrow[d, "0"] & \\
M_6 & = & {[\; u_{61}} & u_{62} & u_{63} & {u_{64} \;]} & \\
 & \vdots & & & & &
\end{tikzcd}
$$

Proof. Suppose it does not halt...

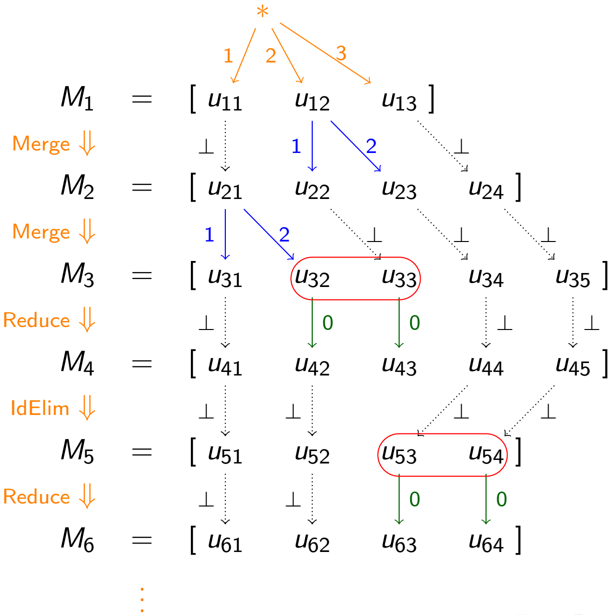
$$
\begin{tikzcd}
M_1 \arrow[d, Rightarrow, "\text{Merge}"'] & = & {[} & u_{11} \arrow[d, dotted, "\perp"'] & u_{12} \arrow[d, "1"'] \arrow[dr, "2"] & u_{13} \arrow[dr, dotted, "\perp"] & {]} & & \\
M_2 \arrow[d, Rightarrow, "\text{Merge}"'] & = & {[} & u_{21} \arrow[d, "1"'] \arrow[dr, "2"] & u_{22} \arrow[dr, dotted, "\perp"] & u_{23} \arrow[dr, dotted, "\perp"] & u_{24} \arrow[dr, dotted, "\perp"] & {]} & \\
M_3 \arrow[d, Rightarrow, "\text{Reduce}"'] & = & {[} & u_{31} \arrow[d, dotted, "\perp"'] & u_{32} \arrow[d, "0"] & u_{33} \arrow[d, "0"] & u_{34} \arrow[d, dotted, "\perp"] & u_{35} \arrow[d, dotted, "\perp"] & {]} \\
M_4 \arrow[d, Rightarrow, "\text{IdElim}"'] & = & {[} & u_{41} \arrow[d, dotted, "\perp"'] & u_{42} \arrow[d, dotted, "\perp"'] & u_{43} & u_{44} \arrow[dl, dotted, "\perp"] & u_{45} \arrow[dl, dotted, "\perp"] & {]} \\
M_5 \arrow[d, Rightarrow, "\text{Reduce}"'] & = & {[} & u_{51} \arrow[d, dotted, "\perp"'] & u_{52} \arrow[d, dotted, "\perp"'] & u_{53} \arrow[d, "0"] & u_{54} \arrow[d, "0"] & {]} & \\
M_6 & = & {[} & u_{61} & u_{62} & u_{63} & u_{64} & {]} &
\end{tikzcd}
$$

$\vdots$

Proof. Suspose it does not halt...

$$
\begin{tikzcd}
 & & & * \arrow[dl, "1"'] \arrow[d, "2"] \arrow[dr, "3"] & & & \\
M_1 \arrow[d, Rightarrow, "\text{Merge}"'] & = & {[}\ u_{11} \arrow[d, dotted, "\perp"'] & u_{12} \arrow[d, "1"'] \arrow[dr, "2"] & u_{13}\ {]} \arrow[dr, dotted, "\perp"] & & \\
M_2 \arrow[d, Rightarrow, "\text{Merge}"'] & = & {[}\ u_{21} \arrow[d, "1"'] \arrow[dr, "2"] & u_{22} \arrow[dr, dotted, "\perp"] & u_{23} \arrow[dr, dotted, "\perp"] & u_{24}\ {]} \arrow[dr, dotted, "\perp"] & \\
M_3 \arrow[d, Rightarrow, "\text{Reduce}"'] & = & {[}\ u_{31} \arrow[d, dotted, "\perp"'] & u_{32} \arrow[d, "0"] & u_{33} \arrow[d, "0"] & u_{34} \arrow[d, dotted, "\perp"] & u_{35}\ {]} \arrow[d, dotted, "\perp"] \\
M_4 \arrow[d, Rightarrow, "\text{IdElim}"'] & = & {[}\ u_{41} \arrow[d, dotted, "\perp"'] & u_{42} \arrow[d, dotted, "\perp"'] & u_{43} & u_{44} \arrow[dl, dotted, "\perp"] & u_{45}\ {]} \arrow[dl, dotted, "\perp"] \\
M_5 \arrow[d, Rightarrow, "\text{Reduce}"'] & = & {[}\ u_{51} \arrow[d, dotted, "\perp"'] & u_{52} \arrow[d, dotted, "\perp"'] & u_{53} \arrow[d, "0"] & u_{54}\ {]} \arrow[d, "0"] & \\
M_6 & = & {[}\ u_{61} & u_{62} & u_{63} & u_{64}\ {]} & \\
 & \vdots & & & & &
\end{tikzcd}
$$

Proof. Suppose it does not halt...



Proof. Suppose it does not halt...

$$M_1 = [\; u_{11} \quad u_{12} \quad u_{13} \;]$$

Merge $\Downarrow$

$$M_2 = [\; u_{21} \quad u_{22} \quad u_{23} \quad u_{24} \;]$$

Merge $\Downarrow$

$$M_3 = [\; u_{31} \quad u_{32} \quad u_{33} \quad u_{34} \quad u_{35} \;]$$

Reduce $\Downarrow$

$$M_4 = [\; u_{41} \quad u_{42} \quad u_{43} \quad u_{44} \quad u_{45} \;]$$

IdElim $\Downarrow$

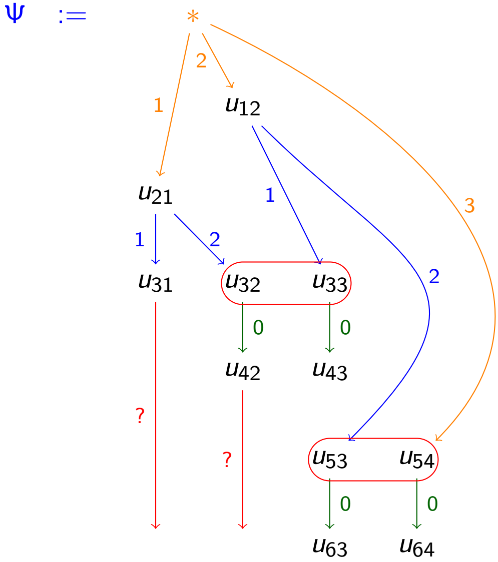
$$M_5 = [\; u_{51} \quad u_{52} \quad u_{53} \quad u_{54} \;]$$

Reduce $\Downarrow$

$$M_6 = [\; u_{61} \quad u_{62} \quad u_{63} \quad u_{64} \;]$$

$\vdots$

Proof. Suppose it does not halt...



- Ψ is an infinite finitely branching tree.

- Ψ is an infinite finitely branching tree.
- The set $\mathcal{B}_\infty(\Psi)$ of its infinite branches is non-empty. (Kőnig)

- Ψ is an infinite finitely branching tree.
- The set $\mathcal{B}_\infty(\Psi)$ of its infinite branches is non-empty. (Kőnig)
- $\mathcal{B}_\infty(\Psi)$ is lexicographically ordered, it is a complete lattice.

- Ψ is an infinite finitely branching tree.
- The set $\mathcal{B}_\infty(\Psi)$ of its infinite branches is non-empty. (Kőnig)
- $\mathcal{B}_\infty(\Psi)$ is lexicographically ordered, it is a complete lattice.
- Infinite branches of Ψ, correspond to infinite paths in Π. Therefore, they satisfiy the guard condition!

$$\mathcal{B}_\infty(\Psi) = \mu\text{-branches} \cup \nu\text{-branches}$$

- Ψ is an infinite finitely branching tree.
- The set $\mathcal{B}_\infty(\Psi)$ of its infinite branches is non-empty. (Kőnig)
- $\mathcal{B}_\infty(\Psi)$ is lexicographically ordered, it is a complete lattice.
- Infinite branches of Ψ, correspond to infinite paths in Π. Therefore, they satisfiy the guard condition!

$$\mathcal{B}_\infty(\Psi) = \mu\text{-branches} \cup \nu\text{-branches}$$

Lemma (F.–S., 2013)

1. *The least infinite branch of Ψ is a ν-branch.*
2. *Let E be a nonempty collection of ν-branches and let $\gamma = \bigvee E$. Then γ is a ν-branch.*
3. *If β is a ν-branch, then there exists another ν-branch $\beta' \succ \beta$.*

Let

$$E = \text{All the } \nu\text{-branches}$$

So what?

Let

$$E = \text{All the } \nu\text{-branches}$$

By 1 $E \neq \varnothing$.

So what?

Let

$$E = \text{All the } \nu\text{-branches}$$

By 1 $E \neq \varnothing$. Let $\gamma = \bigvee E$. By 2, γ is a ν-branch.

So what?

Let

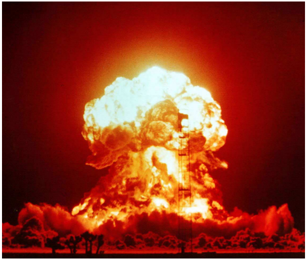
$$E = \text{All the } \nu\text{-branches}$$

By 1 $E \neq \varnothing$. Let $\gamma = \bigvee E$. By 2, γ is a ν-branch. Hence by 3, there is another ν-branch $\gamma' \succ \gamma$.

Let

$$E = \text{All the } \nu\text{-branches}$$

By 1 $E \neq \varnothing$. Let $\gamma = \bigvee E$. By 2, γ is a ν-branch. Hence by 3, there is another ν-branch $\gamma' \succ \gamma$. But then, $\gamma' \in E$ and therefore $\gamma' \preceq \bigvee E = \gamma$.



$\square$

Cut-eliminating infinite proof-trees

- We can cut eliminate a cut-free infinite proof against a fixed circular proof Π.
- We obtain a cut-free infinite proof.

Cut-eliminating infinite proof-trees

- We can cut eliminate a cut-free infinite proof against a fixed circular proof Π.
- We obtain a cut-free infinite proof.

Considering that for any μ-definable set X,

$$X \quad \simeq \quad \text{Winning strategies for } \oplus \text{ in some game}$$

Cut-eliminating infinite proof-trees

- We can cut eliminate a cut-free infinite proof against a fixed circular proof Π.
- We obtain a cut-free infinite proof.

Considering that for any μ-definable set X,

$$\begin{aligned} X &\simeq \text{Winning strategies for } \oplus \text{ in some game} \\ &\simeq \text{Cut-free infinite valid proofs of } 1 \vdash X, \end{aligned}$$

- We can cut eliminate a cut-free infinite proof against a fixed circular proof Π.
- We obtain a cut-free infinite proof.

Considering that for any μ-definable set X,

$$\begin{aligned} X &\simeq \text{Winning strategies for } \oplus \text{ in some game} \\ &\simeq \text{Cut-free infinite valid proofs of } 1 \vdash X, \end{aligned}$$

Cut-elimination is a generic algorithm for computing all the μ-definable functions.

Computability problems

- What are the set-theoretic functions that one can denote with circular proofs and computed with cut-elimination?

Computability problems

- What are the set-theoretic functions that one can denote with circular proofs and computed with cut-elimination?

Primitive recursive $\leq$ μ-definable $\leq$ Recursive

Computability problems

- What are the set-theoretic functions that one can denote with circular proofs and computed with cut-elimination?

  Primitive recursive $\leq$ μ-definable $\leq$ Recursive

- Which of those bounds are strict?

Computability problems

- What are the set-theoretic functions that one can denote with circular proofs and computed with cut-elimination?

Primitive recursive $\leq$ μ-definable $\leq$ Recursive

- Which of those bounds are strict?
- Is the Ackermann function definable?

Computability problems

- What are the set-theoretic functions that one can denote with circular proofs and computed with cut-elimination?

Primitive recursive $\leq$ μ-definable $\leq$ Recursive

- Which of those bounds are strict?
- Is the Ackermann function definable?
- What about streams?

Computability problems

- What are the set-theoretic functions that one can denote with circular proofs and computed with cut-elimination?

Primitive recursive $\leq$ μ-definable $\leq$ Recursive

- Which of those bounds are strict?
- Is the Ackermann function definable?
- What about streams?
- And trees?

Computability problems

- What are the set-theoretic functions that one can denote with circular proofs and computed with cut-elimination?

Primitive recursive $\leq$ μ-definable $\leq$ Recursive

- Which of those bounds are strict?
- Is the Ackermann function definable?
- What about streams?
- And trees?

Regular tree $<$ μ-definable

Computability problems

- What are the set-theoretic functions that one can denote with circular proofs and computed with cut-elimination?

$$\text{Primitive recursive} \leq \mu\text{-definable} \leq \text{Recursive}$$

- Which of those bounds are strict?
- Is the Ackermann function definable?
- What about streams?
- And trees?

$$\text{Regular tree} < \mu\text{-definable}$$

- What about higher order pushdown trees?

Proof-theoretic problems

Proof-theoretic problems

- How can we enrich the proof system (and corresponding model) while keeping a productive cut-elimination procedure?

Proof-theoretic problems

- How can we enrich the proof system (and corresponding model) while keeping a productive cut-elimination procedure?
- Find links with existing circularities (get a categorical perspective)

Proof-theoretic problems

- How can we enrich the proof system (and corresponding model) while keeping a productive cut-elimination procedure?
- Find links with existing circularities (get a categorical perspective)
  - Add contexts, linear logic $s_1 \dots s_m \vdash t_1 \dots t_n$ (Baelde)

Proof-theoretic problems

- How can we enrich the proof system (and corresponding model) while keeping a productive cut-elimination procedure?
- Find links with existing circularities (get a categorical perspective)
  - Add contexts, linear logic $s_1 \ldots s_m \vdash t_1 \ldots t_n$ (Baelde)
  - Add modalties $\Box t, \Diamond t, \ldots$ (Walukiewicz)

Proof-theoretic problems

- How can we enrich the proof system (and corresponding model) while keeping a productive cut-elimination procedure?
- Find links with existing circularities (get a categorical perspective)
  - Add contexts, linear logic $s_1 \ldots s_m \vdash t_1 \ldots t_n$ (Baelde)
  - Add modalties $\Box t, \Diamond t, \ldots$ (Walukiewicz)
  - Add first order. (Brotherston–Simpson, Roșu, Lismont)

Proof-theoretic problems

- How can we enrich the proof system (and corresponding model) while keeping a productive cut-elimination procedure?
- Find links with existing circularities (get a categorical perspective)
  - Add contexts, linear logic $s_1 \ldots s_m \vdash t_1 \ldots t_n$ (Baelde)
  - Add modalties $\Box t, \Diamond t, \ldots$ (Walukiewicz)
  - Add first order. (Brotherston–Simpson, Roșu, Lismont)
- Philosophical question: What is the meaning of circularity in mathematical reasoning?

Merci!

Thank you!

Dank u!