Circular and non-wellfounded proofs: expressiveness and semantics I

Anupam Das

University of Birmingham

École de Printemps d'Informatique Théorique 2025

Centre Paul-Langevin,
Aussois, France,
22nd May 2025.

Outline

1. Types with fixed points
2. Circular proofs
3. Some results on expressivity
4. Conclusions

Outline

1. Types with fixed points
2. Circular proofs
3. Some results on expressivity
4. Conclusions

Simple types

Types:

$$\sigma, \tau, \ldots \quad ::= \quad \bot \mid 1 \mid X \in \mathsf{Var} \mid \sigma + \tau \mid \sigma \times \tau \mid \sigma \to \tau$$

Types:

$$\sigma, \tau, \ldots \quad ::= \quad \bot \mid 1 \mid X \in \mathsf{Var} \mid \sigma + \tau \mid \sigma \times \tau \mid \sigma \to \tau$$

Curry-Howard viewpoint: *formulas as types*.

Types:

$$\sigma, \tau, \ldots \ ::= \ \bot \mid 1 \mid X \in \mathsf{Var} \mid \sigma + \tau \mid \sigma \times \tau \mid \sigma \to \tau \mid \mu X\, \sigma \mid \nu X\, \sigma$$

In $\mu X\, \sigma$ and $\nu X\, \sigma$, the variable X must occur positively in σ.

Curry-Howard viewpoint: *formulas as types*.

μ and ν : inductive and coinductive data types.

Types:

$$\sigma, \tau, \ldots \ ::= \ \bot \mid 1 \mid X \in \mathsf{Var} \mid \sigma + \tau \mid \sigma \times \tau \mid \sigma \to \tau \mid \mu X\, \sigma \mid \nu X\, \sigma$$

In $\mu X\, \sigma$ and $\nu X\, \sigma$, the variable X must occur positively in σ.

Curry-Howard viewpoint: *formulas as types*.

μ and ν : inductive and coinductive data types.

Example

- $B := 1 + 1$ represents the *Booleans*.
- $N := \mu X(1 + X)$ represents the *natural numbers*.
- $S_\tau := \nu X(\tau \times X)$ represents *infinite streams* over τ.
- $W := \mu X(1 + \nu Y(X \times Y))$ represents the (ω-branching) *well-founded trees*.

Sequents: $\sigma_1, \ldots, \sigma_n \Rightarrow \tau$ (interpret as $\sigma_1 \times \cdots \times \sigma_n \to \tau$)

Each type can be constructed and destructed. E.g.

$$\to_r \frac{\sigma \Rightarrow \tau}{\Rightarrow \sigma \to \tau} \qquad \to_l \frac{\Rightarrow \rho \quad \sigma \Rightarrow \tau}{\rho \to \sigma \Rightarrow \tau}$$

Sequents: $\sigma_1, \ldots, \sigma_n \Rightarrow \tau$ (interpret as $\sigma_1 \times \cdots \times \sigma_n \to \tau$)

Each type can be constructed and destructed. E.g.

$$\to_r \frac{\sigma \Rightarrow \tau}{\Rightarrow \sigma \to \tau} \qquad \to_l \frac{\Rightarrow \rho \quad \sigma \Rightarrow \tau}{\rho \to \sigma \Rightarrow \tau}$$

Curry-Howard viewpoint: *proofs as programs*.

Sequents: $\sigma_1, \ldots, \sigma_n \Rightarrow \tau$ (interpret as $\sigma_1 \times \cdots \times \sigma_n \to \tau$)

Each type can be constructed and destructed. E.g.

$$\to_r \frac{\sigma \Rightarrow \tau}{\Rightarrow \sigma \to \tau} \qquad \to_l \frac{\Rightarrow \rho \quad \sigma \Rightarrow \tau}{\rho \to \sigma \Rightarrow \tau}$$

Curry-Howard viewpoint: *proofs as programs*.

Fixed point rules:

$$\mu_r \frac{\Rightarrow \sigma(\mu X\, \sigma(X))}{\Rightarrow \mu X\, \sigma(X)} \quad \mu_l \frac{\sigma(\tau) \Rightarrow \tau}{\mu X\, \sigma(X) \Rightarrow \tau} \quad \nu_r \frac{\tau \Rightarrow \sigma(\tau)}{\tau \Rightarrow \nu X\, \sigma(X)} \quad \nu_l \frac{\sigma(\nu X\, \sigma(X)) \Rightarrow \tau}{\nu X\, \sigma(X) \Rightarrow \tau}$$

Sequents: $\sigma_1, \ldots, \sigma_n \Rightarrow \tau$ (interpret as $\sigma_1 \times \cdots \times \sigma_n \to \tau$)

Each type can be constructed and destructed. E.g.

$$\to_r \frac{\sigma \Rightarrow \tau}{\Rightarrow \sigma \to \tau} \qquad \to_l \frac{\Rightarrow \rho \quad \sigma \Rightarrow \tau}{\rho \to \sigma \Rightarrow \tau}$$

Curry-Howard viewpoint: *proofs as programs*.

Fixed point rules:

$$\mu_r \frac{\Rightarrow \sigma(\mu X\, \sigma(X))}{\Rightarrow \mu X\, \sigma(X)} \quad \mu_l \frac{\sigma(\tau) \Rightarrow \tau}{\mu X\, \sigma(X) \Rightarrow \tau} \quad \nu_r \frac{\tau \Rightarrow \sigma(\tau)}{\tau \Rightarrow \nu X\, \sigma(X)} \quad \nu_l \frac{\sigma(\nu X\, \sigma(X)) \Rightarrow \tau}{\nu X\, \sigma(X) \Rightarrow \tau}$$

Definition ([Cla09])

μLJ is the extension of usual LJ by the fixed point rules above.

Computational theory given by cut-reduction.

$$\underline{N := \mu X(1 + X)}$$

$$\underline{0} := \;\; \mu_r\,\dfrac{\dfrac{\overline{\Rightarrow 1}}{\Rightarrow 1 + N}}{\Rightarrow N} \qquad\qquad \underline{n + 1} := \;\; \mu_r\,\dfrac{\dfrac{\begin{array}{c}\underline{n}\\ \vdots \\ \Rightarrow N\end{array}}{\Rightarrow 1 + N}}{\Rightarrow N}$$

Examples: natural numbers and streams

$$\underline{N := \mu X(1 + X)}$$

$$\underline{0} := \mu_r \dfrac{\dfrac{\overline{\Rightarrow 1}}{\Rightarrow 1 + N}}{\Rightarrow N} \qquad \underline{n+1} := \mu_r \dfrac{\dfrac{\overset{\nabla\,\underline{n}}{\Rightarrow N}}{\Rightarrow 1 + N}}{\Rightarrow N}$$

$$\underline{\mathsf{add} : N \times N \to N}$$

$$\mu_l \dfrac{\dfrac{\dfrac{\mathsf{id}\,\overline{N \Rightarrow N}}{1, N \Rightarrow N} \qquad \mu_r \dfrac{\dfrac{\mathsf{id}\,\overline{N \Rightarrow N}}{N \Rightarrow 1 + N}}{N \Rightarrow N}}{1 + N, N \Rightarrow N}}{N, N \Rightarrow N}$$

$$\left(\begin{array}{c} \mathsf{add}(0, n) = n \\ \mathsf{add}(m + 1, n) = \mathsf{add}(m, n) + 1 \end{array} \right)$$

Examples: natural numbers and streams

$$\underline{N := \mu X(1 + X)}$$

$$\underline{0} := \quad \mu_r \,\cfrac{\cfrac{\overline{\Rightarrow 1}}{\Rightarrow 1 + N}}{\Rightarrow N} \qquad \underline{n+1} := \quad \mu_r \,\cfrac{\cfrac{\begin{array}{c}\underline{n}\\ \vdots \\ \Rightarrow N\end{array}}{\Rightarrow 1 + N}}{\Rightarrow N}$$

$$\underline{S := \nu Y(N \times Y)}$$

$$\mathtt{hd} := \quad \nu_l \,\cfrac{\cfrac{\mathsf{id}\;\overline{N \Rightarrow N}}{N \times S \Rightarrow N}}{S \Rightarrow N} \qquad \mathtt{tl} := \quad \nu_l \,\cfrac{\cfrac{\mathsf{id}\;\overline{S \Rightarrow S}}{N \times S \Rightarrow S}}{S \Rightarrow S}$$

$$\underline{\mathtt{add} : N \times {\color{blue}N} \to N}$$

$$\mu_l \,\cfrac{\cfrac{\cfrac{\mathsf{id}\;\overline{{\color{blue}N} \Rightarrow N}}{1, {\color{blue}N} \Rightarrow N} \qquad \mu_r \,\cfrac{\cfrac{\mathsf{id}\;\overline{{\color{red}N} \Rightarrow N}}{{\color{red}N} \Rightarrow 1 + N}}{{\color{red}N} \Rightarrow N}}{1 + {\color{red}N}, {\color{blue}N} \Rightarrow N}}{N, {\color{blue}N} \Rightarrow N}$$

$$\left(\begin{array}{r}\mathtt{add}(\mathtt{0}, {\color{blue}n}) = {\color{blue}n} \\ \mathtt{add}(m + 1, {\color{blue}n}) = {\color{red}\mathtt{add}(m, n)} + 1\end{array}\right)$$

$$\underline{N := \mu X(1 + X)}$$

$$\underline{0} := \dfrac{\dfrac{\overline{\Rightarrow 1}}{\Rightarrow 1 + N}}{\Rightarrow N}\,\mu_r \qquad \underline{n+1} := \dfrac{\dfrac{\begin{array}{c}\underline{n}\\ \vdots \\ \Rightarrow N\end{array}}{\Rightarrow 1 + N}}{\Rightarrow N}\,\mu_r$$

$$\underline{S := \nu Y(N \times Y)}$$

$$\mathtt{hd} := \dfrac{\dfrac{\overline{N \Rightarrow N}\ \text{id}}{N \times S \Rightarrow N}}{S \Rightarrow N}\,\nu_l \qquad \mathtt{tl} := \dfrac{\dfrac{\overline{S \Rightarrow S}\ \text{id}}{N \times S \Rightarrow S}}{S \Rightarrow S}\,\nu_l$$

$$\underline{\mathtt{add} : N \times N \to N}$$

$$\dfrac{\dfrac{\dfrac{\overline{N \Rightarrow N}\ \text{id}}{1, N \Rightarrow N} \qquad \dfrac{\dfrac{\overline{N \Rightarrow N}\ \text{id}}{N \Rightarrow 1 + N}}{N \Rightarrow N}\,\mu_r}{1 + N, N \Rightarrow N}}{N, N \Rightarrow N}\,\mu_l$$

$$\underline{f : n \mapsto [n, n+1, \ldots]}$$

$$\dfrac{\dfrac{\overline{N \Rightarrow N}\ \text{id} \qquad \dfrac{\dfrac{\overline{N \Rightarrow N}\ \text{id}}{N \Rightarrow 1 + N}}{N \Rightarrow N}\,\mu_r}{N \Rightarrow N \times N}}{N \Rightarrow S}\,\nu_r$$

$$\left(\begin{array}{c}\mathtt{add}(0, n) = n\\ \mathtt{add}(m+1, n) = \mathtt{add}(m, n) + 1\end{array}\right)$$

$$\left(\ f(n) = n :: f(n+1)\ \right)$$

Aside: interpreting types with fixed points

A set theoretic model

Interpret τ as a set $\tau^{\mathfrak{S}}$:

$$
\begin{aligned}
\bot^{\mathfrak{S}} &:= \varnothing \\
1^{\mathfrak{S}} &:= \varnothing \\
(\sigma + \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \uplus \tau^{\mathfrak{S}} \\
(\sigma \times \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \times \tau^{\mathfrak{S}} \\
(\sigma \to \tau)^{\mathfrak{S}} &:= \{f : \sigma^{\mathfrak{S}} \to \tau^{\mathfrak{S}}\}
\end{aligned}
$$

A set theoretic model

Interpret τ as a set $\tau^{\mathfrak{S}}$:

$$
\begin{aligned}
\bot^{\mathfrak{S}} &:= \varnothing \\
1^{\mathfrak{S}} &:= \varnothing \\
(\sigma + \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \uplus \tau^{\mathfrak{S}} \\
(\sigma \times \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \times \tau^{\mathfrak{S}} \\
(\sigma \to \tau)^{\mathfrak{S}} &:= \{f : \sigma^{\mathfrak{S}} \to \tau^{\mathfrak{S}}\} \\
(\mu X \sigma(X))^{\mathfrak{S}} &:= \textbf{?} \\
(\nu X \sigma(X))^{\mathfrak{S}} &:= \textbf{?}
\end{aligned}
$$

A set theoretic model

Interpret τ as a set $\tau^{\mathfrak{S}}$:

$$
\begin{aligned}
\bot^{\mathfrak{S}} &:= \varnothing \\
1^{\mathfrak{S}} &:= \varnothing \\
(\sigma + \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \uplus \tau^{\mathfrak{S}} \\
(\sigma \times \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \times \tau^{\mathfrak{S}} \\
(\sigma \to \tau)^{\mathfrak{S}} &:= \{f : \sigma^{\mathfrak{S}} \to \tau^{\mathfrak{S}}\} \\
(\mu X \sigma(X))^{\mathfrak{S}} &:= \textbf{?} \\
(\nu X \sigma(X))^{\mathfrak{S}} &:= \textbf{?}
\end{aligned}
$$

No interpretation of, e.g.,
$\nu X\, X$ and $\mu X((X \to \sigma) \to \tau)$.

A set theoretic model

Interpret τ as a set $\tau^{\mathfrak{S}}$:

$$\begin{aligned}
\bot^{\mathfrak{S}} &:= \varnothing \\
1^{\mathfrak{S}} &:= \varnothing \\
(\sigma + \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \uplus \tau^{\mathfrak{S}} \\
(\sigma \times \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \times \tau^{\mathfrak{S}} \\
(\sigma \to \tau)^{\mathfrak{S}} &:= \{f : \sigma^{\mathfrak{S}} \to \tau^{\mathfrak{S}}\} \\
(\mu X \sigma(X))^{\mathfrak{S}} &:= ? \\
(\nu X \sigma(X))^{\mathfrak{S}} &:= ?
\end{aligned}$$

No interpretation of, e.g., $\nu X\, X$ and $\mu X((X \to \sigma) \to \tau)$.

A computability theoretic model

Interpret τ as a set $\tau^{\mathfrak{K}} \subseteq \mathbb{N}$:

$$\begin{aligned}
\bot^{\mathfrak{K}} &:= \varnothing \\
1^{\mathfrak{K}} &:= \{0\} \\
(\sigma_0 + \sigma_1)^{\mathfrak{K}} &:= \{n : \&\, M_n 1\!\downarrow \in \sigma_i^{\mathfrak{K}}\ \&\ M_n 0\!\downarrow i\} \\
(\sigma_0 \times \sigma_1)^{\mathfrak{K}} &:= \{n : M_n i\!\downarrow \in \sigma_i^{\mathfrak{K}}, \text{ for } i < 2\} \\
(\sigma \to \tau)^{\mathfrak{K}} &:= \{n : \forall m \in \sigma^{\mathfrak{K}}\ M_n m\!\downarrow \in \tau^{\mathfrak{K}}\}
\end{aligned}$$

A set theoretic model

Interpret τ as a set $\tau^{\mathfrak{S}}$:

$$
\begin{aligned}
\bot^{\mathfrak{S}} &:= \varnothing \\
1^{\mathfrak{S}} &:= \varnothing \\
(\sigma + \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \uplus \tau^{\mathfrak{S}} \\
(\sigma \times \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \times \tau^{\mathfrak{S}} \\
(\sigma \to \tau)^{\mathfrak{S}} &:= \{f : \sigma^{\mathfrak{S}} \to \tau^{\mathfrak{S}}\} \\
(\mu X \sigma(X))^{\mathfrak{S}} &:= ? \\
(\nu X \sigma(X))^{\mathfrak{S}} &:= ?
\end{aligned}
$$

No interpretation of, e.g., $\nu X\, X$ and $\mu X((X \to \sigma) \to \tau)$.

A computability theoretic model

Interpret τ as a set $\tau^{\mathfrak{K}} \subseteq \mathbb{N}$:

$$
\begin{aligned}
\bot^{\mathfrak{K}} &:= \varnothing \\
1^{\mathfrak{K}} &:= \{0\} \\
(\sigma_0 + \sigma_1)^{\mathfrak{K}} &:= \{n : \&\, M_n 1{\downarrow} \in \sigma_i^{\mathfrak{K}} \,\&\, M_n 0{\downarrow}\ i\} \\
(\sigma_0 \times \sigma_1)^{\mathfrak{K}} &:= \{n : M_n i{\downarrow} \in \sigma_i^{\mathfrak{K}}, \text{for } i < 2\} \\
(\sigma \to \tau)^{\mathfrak{K}} &:= \{n : \forall m \in \sigma^{\mathfrak{K}}\, M_n m{\downarrow} \in \tau^{\mathfrak{K}}\} \\
(\mu X \sigma(X))^{\mathfrak{K}} &:= \mathrm{LFP}[A \mapsto \sigma(A)^{\mathfrak{K}}] \\
(\nu X \sigma(X))^{\mathfrak{K}} &:= \mathrm{GFP}[A \mapsto \sigma(A)^{\mathfrak{K}}]
\end{aligned}
$$

A set theoretic model

Interpret τ as a set $\tau^{\mathfrak{S}}$:

$$
\begin{aligned}
\bot^{\mathfrak{S}} &:= \varnothing \\
1^{\mathfrak{S}} &:= \varnothing \\
(\sigma + \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \uplus \tau^{\mathfrak{S}} \\
(\sigma \times \tau)^{\mathfrak{S}} &:= \sigma^{\mathfrak{S}} \times \tau^{\mathfrak{S}} \\
(\sigma \to \tau)^{\mathfrak{S}} &:= \{f : \sigma^{\mathfrak{S}} \to \tau^{\mathfrak{S}}\} \\
(\mu X \sigma(X))^{\mathfrak{S}} &:= ? \\
(\nu X \sigma(X))^{\mathfrak{S}} &:= ?
\end{aligned}
$$

No interpretation of, e.g., $\nu X\, X$ and $\mu X((X \to \sigma) \to \tau)$.

A computability theoretic model

Interpret τ as a set $\tau^{\mathfrak{K}} \subseteq \mathbb{N}$:

$$
\begin{aligned}
\bot^{\mathfrak{K}} &:= \varnothing \\
1^{\mathfrak{K}} &:= \{0\} \\
(\sigma_0 + \sigma_1)^{\mathfrak{K}} &:= \{n : \& M_n 1\!\downarrow \in \sigma_i^{\mathfrak{K}} \;\&\; M_n 0\!\downarrow i\} \\
(\sigma_0 \times \sigma_1)^{\mathfrak{K}} &:= \{n : M_n i\!\downarrow \in \sigma_i^{\mathfrak{K}}, \text{for } i < 2\} \\
(\sigma \to \tau)^{\mathfrak{K}} &:= \{n : \forall m \in \sigma^{\mathfrak{K}}\, M_n m\!\downarrow \in \tau^{\mathfrak{K}}\} \\
(\mu X \sigma(X))^{\mathfrak{K}} &:= \mathrm{LFP}[A \mapsto \sigma(A)^{\mathfrak{K}}] \\
(\nu X \sigma(X))^{\mathfrak{K}} &:= \mathrm{GFP}[A \mapsto \sigma(A)^{\mathfrak{K}}
\end{aligned}
$$

Q: what do B, N, S_τ, W denote in $\mathfrak{K}$?

Curry-Howard viewpoint relates logic and computation

Curry-Howard viewpoint relates logic and computation:

System	Computation	Logic
simple types	Extended Polynomials	Pure FO Logic
+ N	HO Primitive Recursion (T)	FO Arithmetic (PA)
+ $\forall, \exists$	Polymorphic λ-Calculus (F)	SO Arithmetic (PA2)

Curry-Howard viewpoint relates logic and computation:

System	Computation	Logic
simple types	Extended Polynomials	Pure FO Logic
$+ N$	HO Primitive Recursion (T)	FO Arithmetic (PA)
$+ \mu, \nu$	**?**	**?**
$+ \forall, \exists$	Polymorphic λ-Calculus (F)	SO Arithmetic (PA2)

Curry-Howard viewpoint relates logic and computation:

System	Computation	Logic
simple types	Extended Polynomials	Pure FO Logic
$+ N$	HO Primitive Recursion (T)	FO Arithmetic (PA)
$+ \mu, \nu$	**?**	**?**
$+ \forall, \exists$	Polymorphic λ-Calculus (F)	SO Arithmetic (PA2)

What do fixed point type systems compute?

This may be model-sensitive, but is *robust* for type 1 functions.

Outline

1. Types with fixed points
2. Circular proofs
3. Some results on expressivity
4. Conclusions

Replace μ_l and ν_r by unfoldings:

$$\mu_l' \; \frac{\Gamma, \sigma(\mu X\, \sigma(X)) \Rightarrow \tau}{\Gamma, \mu X\, \sigma(X) \Rightarrow \tau} \qquad \nu_r' \; \frac{\Gamma \Rightarrow \sigma(\nu X\, \sigma(X))}{\Gamma \Rightarrow \nu X\, \sigma(X)}$$

Non-wellfounded typing

Replace μ_l and ν_r by unfoldings:

$$\mu_l' \frac{\Gamma, \sigma(\mu X \, \sigma(X)) \Rightarrow \tau}{\Gamma, \mu X \, \sigma(X) \Rightarrow \tau} \qquad \nu_r' \frac{\Gamma \Rightarrow \sigma(\nu X \, \sigma(X))}{\Gamma \Rightarrow \nu X \, \sigma(X)}$$

- A **coderivation** is generated *coinductively* from rules of $\mu'\mathsf{LJ}$.
- It is **progressing** if every infinite branch has an infinite progressing thread.
  (Precise definition is beyond the scope of this talk.)

Replace μ_l and ν_r by unfoldings:

$$\mu_l' \frac{\Gamma, \sigma(\mu X \sigma(X)) \Rightarrow \tau}{\Gamma, \mu X \sigma(X) \Rightarrow \tau} \qquad \nu_r' \frac{\Gamma \Rightarrow \sigma(\nu X \sigma(X))}{\Gamma \Rightarrow \nu X \sigma(X)}$$

- A **coderivation** is generated *coinductively* from rules of μ'LJ.
- It is **progressing** if every infinite branch has an infinite progressing thread.
  (Precise definition is beyond the scope of this talk.)

Definition

CμLJ is the class of regular progressing coderivations.

Computational theory again given by cut-reduction.

Replace μ_l and ν_r by unfoldings:

$$\mu_l' \frac{\Gamma, \sigma(\mu X \sigma(X)) \Rightarrow \tau}{\Gamma, \mu X \sigma(X) \Rightarrow \tau} \qquad \nu_r' \frac{\Gamma \Rightarrow \sigma(\nu X \sigma(X))}{\Gamma \Rightarrow \nu X \sigma(X)}$$

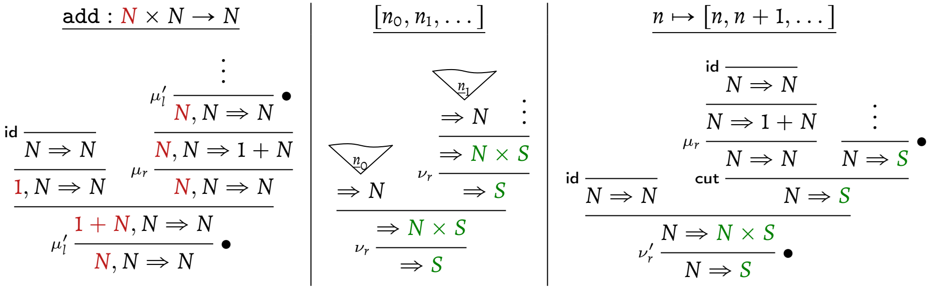
- A **coderivation** is generated *coinductively* from rules of $\mu'\mathsf{LJ}$.
- It is **progressing** if every infinite branch has an infinite progressing thread.
  (Precise definition is beyond the scope of this talk.)

Definition

$\mathsf{C}\mu\mathsf{LJ}$ is the class of regular progressing coderivations.

Computational theory again given by cut-reduction.

NB: cyclic proof checking is decidable, reducing to universality of Büchi automata.


add : N × N → N
id
N ⇒ N
1, N ⇒ N
μ'_l
N, N ⇒ N
μ_r
N, N ⇒ 1 + N
N, N ⇒ N
1 + N, N ⇒ N
μ'_l
N, N ⇒ N
[n_0, n_1, . . .]
n_1
⇒ N
⇒ N × S
ν_r
⇒ S
n_0
⇒ N
⇒ N × S
ν_r
⇒ S
n ↦ [n, n + 1, . . .]
id
N ⇒ N
N ⇒ 1 + N
μ_r
N ⇒ N
N ⇒ S
id
N ⇒ N
cut
N ⇒ S
N ⇒ N × S
ν'_r
N ⇒ S

$$\underline{\mathsf{add} : N \times N \to N}$$

$$\mu_l' \; \dfrac{\dfrac{\dfrac{\mathsf{id}\;\dfrac{}{N \Rightarrow N}}{1, N \Rightarrow N} \qquad \mu_r\;\dfrac{\dfrac{\mu_l'\;\dfrac{\vdots}{N, N \Rightarrow N}\;\bullet}{N, N \Rightarrow 1 + N}}{N, N \Rightarrow N}}{1 + N, N \Rightarrow N}}{N, N \Rightarrow N}\;\bullet$$

$$\underline{[n_0, n_1, \dots]}$$

$$\nu_r \; \dfrac{\dfrac{\dfrac{n_0}{\Rightarrow N} \qquad \nu_r\;\dfrac{\dfrac{\dfrac{n_1}{\Rightarrow N} \quad \vdots}{\Rightarrow N \times S}}{\Rightarrow S}}{\Rightarrow N \times S}}{\Rightarrow S}$$

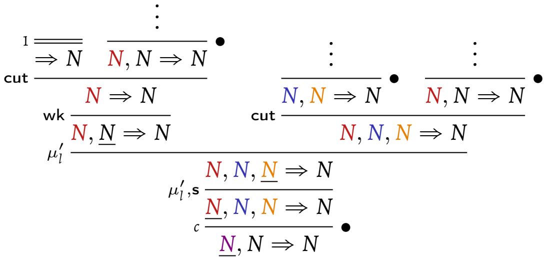
$$\underline{n \mapsto [n, n + 1, \dots]}$$

$$\nu_r' \; \dfrac{\dfrac{\mathsf{id}\;\dfrac{}{N \Rightarrow N} \qquad \mathsf{cut}\;\dfrac{\mu_r\;\dfrac{\dfrac{\mathsf{id}\;\dfrac{}{N \Rightarrow N}}{N \Rightarrow 1 + N}}{N \Rightarrow N} \qquad \dfrac{\vdots}{N \Rightarrow S}\;\bullet}{N \Rightarrow S}}{N \Rightarrow N \times S}}{N \Rightarrow S}\;\bullet$$

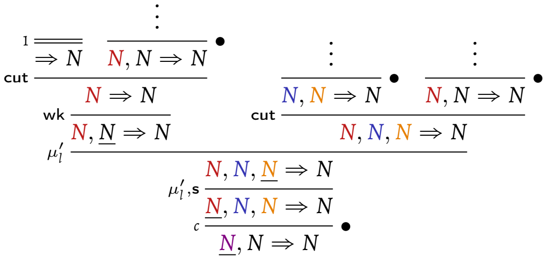
Iteration to cycles:

$$\mu_l \; \dfrac{\sigma(\tau) \Rightarrow \tau}{\mu X \sigma(X) \Rightarrow \tau} \qquad \rightsquigarrow \qquad \mu_l' \; \dfrac{\mathsf{cut}\;\dfrac{\sigma\;\dfrac{\mu_l'\;\dfrac{\vdots}{\mu X \sigma(X) \Rightarrow \tau}\;\bullet}{\sigma(\mu X \sigma(X)) \Rightarrow \sigma(\tau)} \qquad \sigma(\tau) \Rightarrow \tau}{\sigma(\mu X \sigma(X)) \Rightarrow \tau}}{\mu X \sigma(X) \Rightarrow \tau}\;\bullet$$

$$A(0, n) = n + 1$$
$$A(m + 1, 0) = A(m, 1)$$
$$A(m + 1, n + 1) = A(m, A(n + 1, m))$$



$$A(0, n) = n + 1$$
$$A(m + 1, 0) = A(m, 1)$$
$$A(m + 1, n + 1) = A(m, A(n + 1, m))$$



NB: The function A requires iteration at type 1 in finitary derivations.

Some observations

- Circular proofs interpret finite ones.
- Circular proofs can be more succinct than finite ones.

Some observations

- Circular proofs interpret finite ones.
- Circular proofs can be more succinct than finite ones.

Are circular proofs more expressive than finite ones?

…over type 1 functions / …over some/all models.

Some observations

- Circular proofs interpret finite ones.
- Circular proofs can be more succinct than finite ones.

Are circular proofs more expressive than finite ones?

…over type 1 functions / …over some/all models.

A metamathematical approach

- Second-order arithmetic can formalise *metatheory* of infinite proofs.
- Computational interpretations allow us to extract finitary proofs thence.

Outline

1. Types with fixed points
2. Circular proofs
3. Some results on expressivity
4. Conclusions

Definition

(C)T is the restriction of (C)μLJ to just one fixed point N.

Definition

(C)T is the restriction of (C)μLJ to just one fixed point N.

- $N^{\mathfrak{S}} \cong \mathbb{N}$: so $\mathfrak{S}$ reduces to a model of higher-order functionals over $\mathbb{N}$.
- A circular preproof $P : \Gamma \Rightarrow \tau$ represents a partial functional $P^{\mathfrak{S}} : \Gamma^{\mathfrak{S}} \rightarrow \tau^{\mathfrak{S}}$.

Theorem ([KPP21])

(Affine) CT *and (affine)* T *represent the same type 1 functions (in all models).*

Definition

(C)T is the restriction of (C)μLJ to just one fixed point N.

- $N^{\mathfrak{S}} \cong \mathbb{N}$: so $\mathfrak{S}$ reduces to a model of higher-order functionals over $\mathbb{N}$.
- A circular preproof $P : \Gamma \Rightarrow \tau$ represents a partial functional $P^{\mathfrak{S}} : \Gamma^{\mathfrak{S}} \to \tau^{\mathfrak{S}}$.

Theorem ([KPP21])

(Affine) CT *and (affine)* T *represent the same type 1 functions (in all models).*

Write (C)T_n for the restriction of (C)T to just level n types.

Theorem ([Das21])

T_{n+1} *and* CT_n *interpret each other.*
Thus T_{n+1} *and* CT_n *represent the same type 1 functions (in all models).*

Formally, (C)T here is an equational theory with quantifier-free induction.

Definition
(C)T is the restriction of (C)μLJ to just one fixed point N.

- $N^{\mathfrak{S}} \cong \mathbb{N}$: so $\mathfrak{S}$ reduces to a model of higher-order functionals over $\mathbb{N}$.
- A circular preproof $P : \Gamma \Rightarrow \tau$ represents a partial functional $P^{\mathfrak{S}} : \Gamma^{\mathfrak{S}} \to \tau^{\mathfrak{S}}$.

Theorem ([KPP21])
(Affine) CT *and (affine)* T *represent the same type 1 functions (in all models).*

Write (C)T_n for the restriction of (C)T to just level n types.

Theorem ([Das21])
T_{n+1} *and* CT_n *interpret each other.*
Thus T_{n+1} *and* CT_n *represent the same type 1 functions (in all models).*

Formally, (C)T here is an equational theory with quantifier-free induction.

Proposition ([Das21])
T *and* CT *represent the same functionals in* $\mathfrak{S}$.

Proposition (Totality)

If P is progressing, then $P^{\mathfrak{S}}$ is total.

Proposition (Totality)

If P is progressing, then $P^{\mathfrak{S}}$ is total.

Proof sketch.

Let P_0 be a coderivation of $\Gamma_0 \Rightarrow \tau_0$.

- Suppose not P_0 total, and let $\vec{a}_0 \in \Gamma_0^{\mathfrak{S}}$ s.t. $P_0^{\mathfrak{S}} \vec{a}_0 \uparrow$.
- Each rule preserves totality, so we can build an infinite branch $B = (P_i)_{i<\omega}$ and inputs $\vec{a}_i$ such that $P_i \vec{a}_i \uparrow$.

Proposition (Totality)

If P is progressing, then $P^{\mathfrak{S}}$ is total.

Proof sketch.

Let P_0 be a coderivation of $\Gamma_0 \Rightarrow \tau_0$.

- Suppose not P_0 total, and let $\vec{a}_0 \in \Gamma_0^{\mathfrak{S}}$ s.t. $P_0^{\mathfrak{S}} \vec{a}_0 \uparrow$.
- Each rule preserves totality, so we can build an infinite branch $B = (P_i)_{i<\omega}$ and inputs $\vec{a}_i$ such that $P_i \vec{a}_i \uparrow$.
- Any thread $(N^j)_{j<\omega}$ in B induces a non-increasing sequence $\mathbf{a} = (a_{i_j})_{j<\omega} \in \mathbb{N}^\omega$.

Proposition (Totality)

If P is progressing, then $P^{\mathfrak{S}}$ is total.

Proof sketch.

Let P_0 be a coderivation of $\Gamma_0 \Rightarrow \tau_0$.

- Suppose not P_0 total, and let $\vec{a}_0 \in \Gamma_0^{\mathfrak{S}}$ s.t. $P_0^{\mathfrak{S}} \vec{a}_0 \uparrow$.
- Each rule preserves totality, so we can build an infinite branch $B = (P_i)_{i<\omega}$ and inputs $\vec{a}_i$ such that $P_i \vec{a}_i \uparrow$.
- Any thread $(N^j)_{j<\omega}$ in B induces a non-increasing sequence $\mathbf{a} = (a_{i_j})_{j<\omega} \in \mathbb{N}^\omega$.
- $\mathbf{a}$ must converge, by well-foundedness of $\mathbb{N}$, so $(N^j)_{j<\omega}$ is not progressing. □

Proposition (Totality)

If P is progressing, then $P^{\mathfrak{S}}$ is total.

Proof sketch.

Let P_0 be a coderivation of $\Gamma_0 \Rightarrow \tau_0$.

- Suppose not P_0 total, and let $\vec{a}_0 \in \Gamma_0^{\mathfrak{S}}$ s.t. $P_0^{\mathfrak{S}} \vec{a}_0 \uparrow$.
- Each rule preserves totality, so we can build an infinite branch $B = (P_i)_{i<\omega}$ and inputs $\vec{a}_i$ such that $P_i \vec{a}_i \uparrow$.
- Any thread $(N^j)_{j<\omega}$ in B induces a non-increasing sequence $\mathbf{a} = (a_{i_j})_{j<\omega} \in \mathbb{N}^\omega$.
- $\mathbf{a}$ must converge, by well-foundedness of $\mathbb{N}$, so $(N^j)_{j<\omega}$ is not progressing. □

NB: this proof is highly non-constructive! How can we extract induction invariants?

Before dissecting these results, let us set up our toolbox:

Proof theoretic ingredients:

- Totality at type 1 is a Π^0_2 property $(\forall m \exists n\, Pm \downarrow n)$.
  $\rightsquigarrow$ all proofs are constructive!
- Formalising this proof bounds the proof theoretic strength of CT.

Before dissecting these results, let us set up our toolbox:

Proof theoretic ingredients:

- Totality at type 1 is a Π^0_2 property ($\forall m \exists n \, Pm \downarrow n$).
  $\rightsquigarrow$ all proofs are constructive!
- Formalising this proof bounds the proof theoretic strength of CT.

(Higher-order) recursion theoretic ingredients:

- Totality can proved wrt. HO computability models, e.g. $\mathfrak{K}$.
  $\rightsquigarrow$ formalisation in subsystems of SO arithmetic.
- Proof checking is decidable, provably.
  $\rightsquigarrow$ reverse mathematics of *ω-automaton theory* [KMPS19, Das20].

Before dissecting these results, let us set up our toolbox:

Proof theoretic ingredients:

- Totality at type 1 is a Π^0_2 property ($\forall m \exists n\ Pm \downarrow n$).
  $\rightsquigarrow$ all proofs are constructive!
- Formalising this proof bounds the proof theoretic strength of CT.

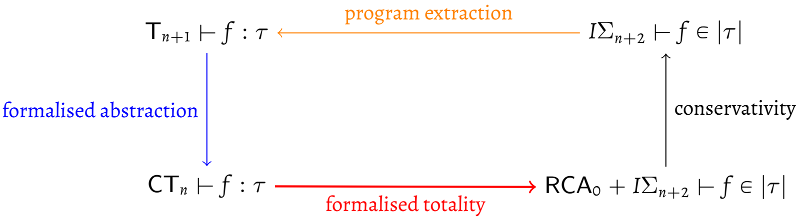
(Higher-order) recursion theoretic ingredients:

- Totality can proved wrt. HO computability models, e.g. $\mathfrak{K}$.
  $\rightsquigarrow$ formalisation in subsystems of SO arithmetic.
- Proof checking is decidable, provably.
  $\rightsquigarrow$ reverse mathematics of *ω-automaton theory* [KMPS19, Das20].

Textbook interpretations:

- SO theories are conservative extensions of appropriate FO ones.
- *Computational interpretations* allow T to realise FO theorems.

For f of type 1:



- formalised abstraction requires a careful *partial evaluation* result.
- Confluence of CT in $\mathsf{RCA}_0 \implies$ determinism.
- formalised totality arithmetises the totality argument for CT.
    - *Reverse mathematics* of ω-automaton theory [KMPS19, Das20].
    - Must formalise totality argument in a HO computability model $|\cdot|$.
- program extraction is textbook.

WHAT ABOUT (C)μLJ?

What about (C)μLJ?

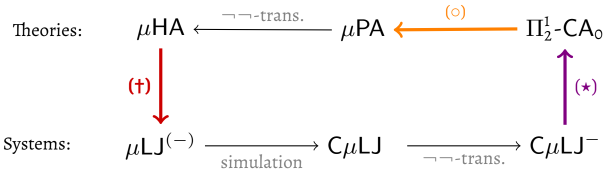
Theorem ([CD23])

μLJ and CμLJ define just the functions provably recursive in Π^1_2-CA_0.

WHAT ABOUT (C)μLJ?

Theorem ([CD23])

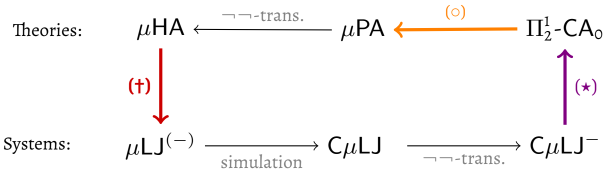
μLJ and CμLJ define just the functions provably recursive in Π^1_2-CA_0.



What about (C)μLJ?

Theorem ([CD23])

μLJ and CμLJ define just the functions provably recursive in Π^1_2-CA$_0$.



(⋆) Formalisation of semantics by *fixed points as fixed points*:

- Novel reverse mathematics of ordinal and fixed point theory, building on [Das21, DM23].

(○) A complex black box result due to [Mö02].

(†) Realisability interpretation by *fixed points as SO types*.

- (Considerable) specialisation of HA2 $\to$ F.

Outline

1. Types with fixed points
2. Circular proofs
3. Some results on expressivity
4. Conclusions

Metamathematics provides a **powerful toolbox** for understanding circular proofs.

Metamathematics provides a **powerful toolbox** for understanding circular proofs.

- Classifying absolute and relative expressivity of circular systems.
- Exposing new connections with classical topics.
- Fuelling new results of independent interest.

Metamathematics provides a **powerful toolbox** for understanding circular proofs.

- Classifying absolute and relative expressivity of circular systems.
- Exposing new connections with classical topics.
- Fuelling new results of independent interest.

NB: considerable precursory work in *cyclic arithmetic* [Sim17, BT17, Das20].

Metamathematics provides a **powerful toolbox** for understanding circular proofs.

- Classifying absolute and relative expressivity of circular systems.
- Exposing new connections with classical topics.
- Fuelling new results of independent interest.

NB: considerable precursory work in *cyclic arithmetic* [Sim17, BT17, Das20].

Motto: *circular proofs are more expressive than finite ones… up to a point!*

Metamathematics provides a **powerful toolbox** for understanding circular proofs.

- Classifying absolute and relative expressivity of circular systems.
- Exposing new connections with classical topics.
- Fuelling new results of independent interest.

NB: considerable precursory work in *cyclic arithmetic* [Sim17, BT17, Das20].

Motto: *circular proofs are more expressive than finite ones… up to a point!*

- What about weak systems? some work in complexity theory e.g. [CD22].
- Further development of the model theory of circular proofs.

Metamathematics provides a **powerful toolbox** for understanding circular proofs.

- Classifying absolute and relative expressivity of circular systems.
- Exposing new connections with classical topics.
- Fuelling new results of independent interest.

NB: considerable precursory work in *cyclic arithmetic* [Sim17, BT17, Das20].

Motto: *circular proofs are more expressive than finite ones… up to a point!*

- What about weak systems? some work in complexity theory e.g. [CD22].
- Further development of the model theory of circular proofs.

Thank you.

Stefano Berardi and Makoto Tatsuta.
Equivalence of inductive definitions and cyclic proofs under arithmetic.
In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017.

Gianluca Curzi and Anupam Das.
Cyclic implicit complexity.
In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 19:1–19:13. ACM, 2022.

Gianluca Curzi and Anupam Das.
Computational expressivity of (circular) proofs with fixed points.
In *LICS*, pages 1–13, 2023.

Pierre Clairambault.
Least and greatest fixpoints in game semantics.
In Ralph Matthes and Tarmo Uustalu, editors, *FICS '09, Coimbra, Portugal, September 12-13, 2009*, pages 39–45. Institute of Cybernetics, 2009.

Anupam Das.
On the logical complexity of cyclic arithmetic.
Logical Methods in Computer Science, Volume 16, Issue 1, January 2020.

Anupam Das.
A circular version of Gödel's T and its abstraction complexity.
CoRR, abs/2012.14421, 2021.

Anupam Das and Lukas Melgaard.
Cyclic proofs for arithmetical inductive definitions.
In Marco Gaboardi and Femke van Raamsdonk, editors, *8th International Conference on Formal Structures for Computation and Deduction, FSCD 2023, July 3-6, 2023, Rome, Italy*, volume 260 of *LIPIcs*, pages 27:1–27:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

Leszek Aleksander Kolodziejczyk, Henryk Michalewski, Pierre Pradic, and Michal Skrzypczak.
The logical strength of Büchi's decidability theorem.
Log. Methods Comput. Sci., 15(2), 2019.

Denis Kuperberg, Laureline Pinault, and Damien Pous.
Cyclic proofs, system T, and the power of contraction.
Proc. ACM Program. Lang., 5(POPL):1–28, 2021.

Michael Möllerfeld.
Generalized inductive definitions. The μ-calculus and Π^1_2-comprehension.
PhD thesis, 2002.
University of Münster,
https://nbn-resolving.de/urn:nbn:de:hbz:6-85659549572.

Alex Simpson.
Cyclic arithmetic is equivalent to peano arithmetic.
In Javier Esparza and Andrzej S. Murawski, editors, *FOSSACS '17, Held as Part of ETAPS '17, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10203 of *Lecture Notes in Computer Science*, pages 283–300, 2017.