From Substitution to Semantics for a Family of Substructural Type Systems

James Wood[1]   Bob Atkey[1]

[1]University of Strathclyde

VEST, June 2020

Context

$$\begin{tikzcd}
& \text{McB05[1]} \arrow[dl, "\text{linearity}"'] \arrow[dr, "\text{semantics}"] & \\
\text{WA20[2]} \arrow[dr] & & \text{AACMM18/20[3]} \arrow[dl] \\
& \text{WIP} &
\end{tikzcd}$$

- Linearity independent of binding (de Bruijn indices)
- Only one traversal over the syntax

[1] Conor McBride. *Type-preserving renaming and substitution*. 2005.

[2] James Wood and Robert Atkey. "A Linear Algebra Approach to Linear Metatheory". Linearity/TLLA. 2020.

[3] Guillaume Allais et al. "A Type and Scope Safe Universe of Syntaxes with Binding: Their Semantics and Proofs". JFP. 2020.

Idea — stability under structurality[4]

Two parts:

1. Consolidate all traversals over syntax (e.g, simultaneous renaming, simultaneous substitution, NbE, printing) into a single generic traversal.
2. Build typing rules from small building blocks, so that they admit a generic semantics by construction.

[4] Guillaume Allais et al. "A Type and Scope Safe Universe of Syntaxes with Binding: Their Semantics and Proofs". JFP. 2020.

Starter — substitution via kits

Renaming

$$\frac{\forall A.\ A \in \Gamma \to \color{red}{A \in \Delta}}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}$$

Substitution

$$\frac{\forall A.\ A \in \Gamma \to \color{red}{A \dashv \Delta}}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}$$

$$\mathrm{subst}\ \sigma\ (\mathrm{lam}\ M) = \mathrm{lam}\ (\mathrm{subst}\ ?\ M)$$

$$\begin{cases} \sigma : \forall A.\ A \in \Gamma \to A \dashv \Delta \\ M : Y \dashv \Gamma, X \\ ? : \forall A.\ A \in X, \Gamma \to A \dashv X, \Delta \end{cases}$$

Starter — substitution via kits

Renaming

$$\frac{\forall A.\ A \in \Gamma \to A \in \Delta}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}$$

Substitution

$$\frac{\forall A.\ A \in \Gamma \to A \dashv \Delta}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}$$

$$\text{subst } \sigma \ (\text{lam } M) = \text{lam } (\text{subst } (\text{bind } \sigma) \ M) \quad \begin{cases} \sigma : \forall A.\ A \in \Gamma \to A \dashv \Delta \\ M : Y \dashv \Gamma, X \\ \text{bind } \sigma : \forall A.\ A \in X, \Gamma \to A \dashv X, \Delta \end{cases}$$

$$\text{bind } \sigma \ \text{new} = ?$$

$$\text{bind } \sigma \ (\text{old } i) = ?$$

Starter — substitution via kits

Renaming

$$\frac{\forall A.\ A \in \Gamma \to {\color{red} A \in \Delta}}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}$$

Substitution

$$\frac{\forall A.\ A \in \Gamma \to {\color{red} A \dashv \Delta}}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}$$

$$\text{subst } \sigma \ (\text{lam } M) = \text{lam } (\text{subst } (\text{bind } \sigma) \ M) \quad \begin{cases} \sigma : \forall A.\ A \in \Gamma \to A \dashv \Delta \\ M : Y \dashv \Gamma, X \\ \text{bind } \sigma : \forall A.\ A \in X, \Gamma \to A \dashv X, \Delta \end{cases}$$

$$\text{bind } \sigma \text{ new} = \text{var new}$$

$$\text{bind } \sigma \ (\text{old } i) = ?$$

Starter — substitution via kits

Renaming

$$\frac{\forall A.\ A \in \Gamma \to \color{red}{A \in \Delta}}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}$$

Substitution

$$\frac{\forall A.\ A \in \Gamma \to \color{red}{A \dashv \Delta}}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}$$

$$\text{subst } \sigma \ (\text{lam } M) = \text{lam } (\text{subst } (\text{bind } \sigma) \ M) \quad \begin{cases} \sigma : \forall A.\ A \in \Gamma \to A \dashv \Delta \\ M : Y \dashv \Gamma, X \\ \text{bind } \sigma : \forall A.\ A \in X, \Gamma \to A \dashv X, \Delta \end{cases}$$

$$\text{bind } \sigma \text{ new} = \text{var new}$$

$$\text{bind } \sigma \ (\text{old } i) = \text{rename } \rho \ (\sigma \ i) \qquad \left\{ \rho : \forall A.\ A \in \Delta \to A \in X, \Delta \right.$$

Generic syntactic traversal

Generalise over renaming and substitution.

$$\frac{\mathrm{Kit}\ \mathcal{V} \qquad \forall A.\ A \in \Gamma \to \mathcal{V}\ A\ \Delta}{\forall A.\ A \dashv \Gamma \to A \dashv \Delta}\ \mathrm{trav}$$

The Kit contains $\mathit{kit.tm}$, $\mathit{kit.vr}$, and $\mathit{kit.str}$.

$$\mathrm{trav}\ \sigma\ (\mathrm{var}\ i) = \mathit{kit.tm}\ (\sigma\ i) \qquad \left\{ \mathit{kit.tm} : \forall A, \Gamma.\ \mathcal{V}\ A\ \Gamma \to A \dashv \Gamma \right.$$

$$\mathrm{trav}\ \sigma\ (\mathrm{lam}\ M) = \mathrm{lam}\ (\mathrm{trav}\ (\mathrm{bind}\ \sigma)\ M) \qquad \left\{ \mathrm{bind}\ \sigma : \forall A.\ A \in X, \Gamma \to \mathcal{V}\ A\ (X, \Delta) \right.$$

$$\mathrm{bind}\ \sigma\ \mathrm{new} = \mathit{kit.vr}\ \mathrm{new} \qquad \left\{ \mathit{kit.vr} : \forall A, \Gamma.\ A \in \Gamma \to \mathcal{V}\ A\ \Gamma \right.$$

$$\mathrm{bind}\ \sigma\ (\mathrm{old}\ i) = \mathit{kit.str}\ \rho\ (\sigma\ i) \qquad \begin{cases} \mathit{kit.str} : \Gamma \subseteq \Delta \to \mathcal{V}\ A\ \Gamma \to \mathcal{V}\ A\ \Delta \\ \rho : \forall A.\ A \in \Delta \to A \in X, \Delta \end{cases}$$

Generic semantic traversal

- We have the following fundamental lemma of semantics:

$$\frac{\text{Semantics } \mathcal{V}\ \mathcal{C} \qquad \overbrace{\forall A.\ A \in \Gamma \to \mathcal{V}\ A\ \Delta}^{\text{environment}}}{\underbrace{\forall A.\ A \dashv \Gamma \to \mathcal{C}\ A\ \Delta}_{\text{traversal}}}\ \text{sem}$$

- Semantics $\mathcal{V}$ $\mathcal{C}$ contains:
  - A proof that $\mathcal{V}$ is stable under structurality
  - Ways to interpret term constructors semantically, generic in context:

$$[\![\text{var}]\!] : \forall[\ \mathcal{V}\ A \dot{\to} \mathcal{C}\ A\ ] \qquad [\![\text{app}]\!] : \forall[\ \Box(\mathcal{C}\ (A \to B)) \dot{\times} \Box(\mathcal{C}\ A) \dot{\to} \mathcal{C}\ B\ ]$$

$$[\![\text{lam}]\!] : \forall[\ \Box(\mathcal{V}\ A \dot{\to} \mathcal{C}\ B) \dot{\to} \mathcal{C}\ (A \to B)\ ] \qquad \ldots$$

Generic semantic traversal

- We have the following fundamental lemma of semantics:

$$\frac{\mathrm{Semantics}\ \mathcal{V}\ \mathcal{C} \qquad \Gamma \overset{\mathcal{V}}{\Rightarrow} \Delta}{\underbrace{\forall A.\ A \dashv \Gamma \to \mathcal{C}\ A\ \Delta}_{\mathrm{traversal}}}\ \mathrm{sem} \qquad\qquad \Gamma \overset{\mathcal{V}}{\Rightarrow} \Delta = \forall A.\ A \in \Gamma \to \mathcal{V}\ A\ \Delta$$

- $\mathrm{Semantics}\ \mathcal{V}\ \mathcal{C}$ contains:
  - A proof that $\mathcal{V}$ is stable under structurality
  - Ways to interpret term constructors semantically, generic in context:

$$[\![\mathrm{var}]\!] : \forall[\ \mathcal{V}\ A \mathrel{\dot{\to}} \mathcal{C}\ A\ ] \qquad [\![\mathrm{app}]\!] : \forall[\ \Box(\mathcal{C}\ (A \to B)) \mathrel{\dot{\times}} \Box(\mathcal{C}\ A) \mathrel{\dot{\to}} \mathcal{C}\ B\ ]$$

$$[\![\mathrm{lam}]\!] : \forall[\ \Box(\mathcal{V}\ A \mathrel{\dot{\to}} \mathcal{C}\ B) \mathrel{\dot{\to}} \mathcal{C}\ (A \to B)\ ] \qquad \ldots$$

Generic semantic traversal

- We have the following fundamental lemma of semantics:

$$\frac{\text{Semantics } \mathcal{V}\ \mathcal{C} \qquad \Gamma \overset{\mathcal{V}}{\Rightarrow} \Delta}{\underbrace{\forall A.\ A \dashv \Gamma \to \mathcal{C}\ A\ \Delta}_{\text{traversal}}}\ \text{sem}$$

$$\Gamma \overset{\mathcal{V}}{\Rightarrow} \Delta = \forall A.\ A \in \Gamma \to \mathcal{V}\ A\ \Delta$$

- Semantics $\mathcal{V}\ \mathcal{C}$ contains: $\quad (\square\mathcal{T})\ \Gamma = \forall \Delta.\ \Gamma \subseteq \Delta \to \mathcal{T}\ \Delta$
  - A proof that $\mathcal{V}$ is stable under structurality
  - Ways to interpret term constructors semantically, generic in context:

$$[\![\text{var}]\!] : \forall[\ \mathcal{V}\ A \mathbin{\dot\to} \mathcal{C}\ A\ ] \qquad [\![\text{app}]\!] : \forall[\ \square(\mathcal{C}\ (A \to B)) \mathbin{\dot\times} \square(\mathcal{C}\ A) \mathbin{\dot\to} \mathcal{C}\ B\ ]$$

$$[\![\text{lam}]\!] : \forall[\ \square(\mathcal{V}\ A \mathbin{\dot\to} \mathcal{C}\ B) \mathbin{\dot\to} \mathcal{C}\ (A \to B)\ ] \qquad \ldots$$

Generic notion of syntax

- A type system can:
  1. Offer a multitude of term formers. e.g, $\textsc{App}$, $\textsc{Lam}$, ...
  2. For each term former, require 0 or more premises. $\dot{\times}$, $\dot{1}$
  3. For each premise, maybe bind variables. $\Box$, $\mathcal{V}$
- Variables are a special case.
- Example descriptions:
  - $\textsc{App}_{A,B}$: $(A \to B) \times A \Longrightarrow B$
  - $\textsc{Lam}_{A,B}$: $(A \vdash B) \Longrightarrow (A \to B)$

Generic generic semantic traversal

If we are given a $\mathcal{V}$-value for each newly bound variable in Γ, we can produce a computation.

$$\mathrm{Kripke}\ \mathcal{V}\ \mathcal{C}\ \Gamma\ A = \Box((\Gamma \overset{\mathcal{V}}{\Rightarrow} -) \mathbin{\dot\to} \mathcal{C}\ A)$$

Let d be the description of a type system. $[\![d]\!]$ is one layer of its syntax.

$$[\![\mathrm{var}]\!] : \forall[\ \mathcal{V}\ A \mathbin{\dot\to} \mathcal{C}\ A\ ]$$

$$[\![\mathrm{con}]\!] : \forall[\ [\![d]\!]\ (\mathrm{Kripke}\ \mathcal{V}\ \mathcal{C})\ A \mathbin{\dot\to} \mathcal{C}\ A\ ]$$

Given $[\![\mathrm{var}]\!]$ and $[\![\mathrm{con}]\!]$, we can produce a similar traversal to before.

$$\mathrm{sem}\ \sigma\ (\mathrm{var}\ i) = [\![\mathrm{var}]\!]\ (\sigma\ i)$$

$$\mathrm{sem}\ \sigma\ (\mathrm{con}\ t) = [\![\mathrm{con}]\!]\ (\mathrm{map}\ (\mathrm{bind}\ \sigma)\ t)$$

Example derivation

$$P = (A \multimap B) \otimes A$$

$$\dfrac{\dfrac{\dfrac{}{p : P \vdash p : P} \qquad \dfrac{\dfrac{}{\begin{array}{c} p : P,\ \ f : A \multimap B, \\ x : A \vdash f : A \multimap B \end{array}} \qquad \dfrac{}{\begin{array}{c} p : P,\ \ f : A \multimap B, \\ x : A \vdash x : A \end{array}}}{p : P,\ \ f : A \multimap B,\ \ x : A \vdash f\ x : B}}{p : P \vdash \mathrm{let}\ (f \,{}_{\otimes}\, x) = p\ \mathrm{in}\ f\ x : B}}{\vdash \lambda p.\ \mathrm{let}\ (f \,{}_{\otimes}\, x) = p\ \mathrm{in}\ f\ x : \underbrace{(A \multimap B) \otimes A}_{P} \multimap B}$$

Example derivation

$$P = (A \multimap B) \otimes A$$

$$\dfrac{\dfrac{}{1p : P \vdash p : P} \quad \dfrac{\dfrac{}{\begin{array}{c}0p : P, 1f : A \multimap B,\\ 0x : A \vdash f : A \multimap B\end{array}} \quad \dfrac{}{\begin{array}{c}0p : P, 0f : A \multimap B,\\ 1x : A \vdash x : A\end{array}}}{0p : P, 1f : A \multimap B, 1x : A \vdash f\ x : B}}{\dfrac{1p : P \vdash \text{let } (f \otimes x) = p \text{ in } f\ x : B}{\vdash \lambda p.\ \text{let } (f \otimes x) = p \text{ in } f\ x : \underbrace{(A \multimap B) \otimes A}_{P} \multimap B}}$$

Vectors over semirings — addition

Semiring operations (operating on annotations on individual variables) are lifted to vector operations (operating on contexts-worth of variables).

$$\frac{\mathcal{P}\gamma \vdash M : A \qquad \mathcal{Q}\gamma \vdash N : B \qquad \mathcal{R} = \mathcal{P} + \mathcal{Q}}{\mathcal{R}\gamma \vdash (M \,{}_{\otimes}\, N) : A \otimes B}$$

$$\frac{\mathcal{R} = 0}{\mathcal{R}\gamma \vdash ({}_{\otimes}) : 1}$$

identity, associativity, commutativity $\sim$ contexts are essentially multisets

Vectors over semirings — multiplication

$$\frac{(x : A) \in \gamma \qquad \mathcal{R} = \langle x|}{\mathcal{R}\gamma \vdash x : A}$$

$$\frac{\mathcal{P}\gamma \vdash M : A \qquad \mathcal{R} = r\mathcal{P}}{\mathcal{R}\gamma \vdash [M] : !_r A}$$

- $\langle x|$ — basis vector. The variable x can be used once, and every other variable can be discarded.
- 'M' for "Multiplication", also for "Modality"

Vectors over semirings

$$\frac{\mathcal{R} = 0}{\mathcal{R}\gamma \vdash ({}_{\otimes}) : 1} \qquad \frac{\mathcal{P}\gamma \vdash M : A \qquad \mathcal{Q}\gamma \vdash N : B \qquad \mathcal{R} = \mathcal{P} + \mathcal{Q}}{\mathcal{R}\gamma \vdash (M \,{}_{\otimes}\, N) : A \otimes B}$$

$$\frac{(x : A) \in \gamma \qquad \mathcal{R} = \langle x|}{\mathcal{R}\gamma \vdash x : A} \qquad \frac{\mathcal{P}\gamma \vdash M : A \qquad \mathcal{R} = r\mathcal{P}}{\mathcal{R}\gamma \vdash [M] : !_r A}$$

- These four are the basic operations of linear algebra.
- 0, $+$, and $r\,\cdot$ are preserved by linear transformations.
- Notice: we can consistently add 0-use variables and maintain typing.

Vectors over semirings

$$\frac{\mathcal{R} = 0}{\mathcal{R}\gamma \vdash (_{\otimes}) : 1}$$

$$\frac{\mathcal{P}\gamma \vdash M : A \qquad \mathcal{Q}\gamma \vdash N : B \qquad \mathcal{R} = \mathcal{P} + \mathcal{Q}}{\mathcal{R}\gamma \vdash (M \,_{\otimes}\, N) : A \otimes B}$$

$$\frac{(x : A) \sqsubseteq \mathcal{R}\gamma}{\mathcal{R}\gamma \vdash x : A}$$

$$\frac{\mathcal{P}\gamma \vdash M : A \qquad \mathcal{R} = r\mathcal{P}}{\mathcal{R}\gamma \vdash [M] : !_r A}$$

- These four are the basic operations of linear algebra.
- 0, $+$, and $r\,\cdot$ are preserved by linear transformations.
- Notice: we can consistently add 0-use variables and maintain typing.

Generic notion of linear syntax

Multiple premises are handled by bunched implications.[5]

- $\mathfrak{I}\ \mathcal{R}\gamma := \mathcal{R} = 0$
- $(\mathcal{T} * \mathcal{U})\ \mathcal{R}\gamma := \Sigma\mathcal{P}, \mathcal{Q}.\ (\mathcal{R} = \mathcal{P} + \mathcal{Q}) \times \mathcal{T}\ \mathcal{P}\gamma \times \mathcal{U}\ \mathcal{Q}\gamma$
- $(r \cdot \mathcal{T})\ \mathcal{R}\gamma := \Sigma\mathcal{P}.\ (\mathcal{R} = r\mathcal{P}) \times \mathcal{T}\ \mathcal{P}\gamma$
- $(\mathcal{T} \mathrel{-\!*} \mathcal{U})\ \mathcal{P}\gamma := \Pi\mathcal{Q}, \mathcal{R}.\ (\mathcal{R} = \mathcal{P} + \mathcal{Q}) \to \mathcal{T}\ \mathcal{Q}\gamma \to \mathcal{U}\ \mathcal{R}\gamma$

Example description: $(!_r A * (rA \vdash B)) \Longrightarrow B$

- $\dfrac{\mathcal{P}\gamma \vdash !_r A \qquad \mathcal{Q}\gamma, rx : A \vdash B \qquad \mathcal{R} = \mathcal{P} + \mathcal{Q}}{\mathcal{R}\gamma \vdash B}$
- $[\![\mathrm{bam}]\!] : \forall[\ \square(\mathcal{C}\ (!_r A)) * \square(r \cdot (\mathcal{V}\ A) \mathrel{-\!*} \mathcal{C}\ B) \mathrel{\dot\to} \mathcal{C}\ B\ ]$

[5] Arjen Rouvoet et al. "Intrinsically-Typed Definitional Interpreters for Linear, Session-Typed Languages". In: *CPP 2020*. New Orleans, LA, USA, 2020, pp. 284–298. ISBN: 9781450370974. DOI: 10.1145/3372885.3373818.

Linear Kripke

We're *adding in* extra $\mathcal{V}$-values, so use $-\!\!*$.

$$\mathrm{Kripke}\ \mathcal{V}\ \mathcal{C}\ \Gamma\ A = \Box((\Gamma \overset{\mathcal{V}}{\Rightarrow} -) \mathrel{-\!\!*} \mathcal{C}\ A)$$

Desiderata for environments:

- $\left(\cdot \overset{\mathcal{V}}{\Rightarrow} -\right) \simeq \mathfrak{I}$
- $\left(\Gamma, \Delta \overset{\mathcal{V}}{\Rightarrow} -\right) \simeq \left(\Gamma \overset{\mathcal{V}}{\Rightarrow} -\right) * \left(\Delta \overset{\mathcal{V}}{\Rightarrow} -\right)$
- $\left(rA \overset{\mathcal{V}}{\Rightarrow} -\right) \simeq r \cdot (\mathcal{V}\ A)$

Linear environments

Renaming

$$\underbrace{1C, 2A, 4A}_{\mathcal{P}\gamma} \overset{\sqsubseteq}{\Rightarrow} \underbrace{6A, 0B, 1C, 0D}_{\mathcal{Q}\delta}$$

$$\overbrace{\begin{pmatrix} 6 & 0 & 1 & 0 \end{pmatrix}}^{\mathcal{Q}} = \overbrace{\begin{pmatrix} 1 & 2 & 4 \end{pmatrix}}^{\mathcal{P}} \begin{array}{cc} \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{l} \delta \ni C \\ \delta \ni A \\ \delta \ni A \end{array} \end{array}$$

Substitution

$$1(A \otimes B) \overset{\dashv}{\Rightarrow} 1A, 1B$$

$$\begin{pmatrix} 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \end{pmatrix} A, B \vdash A \otimes B$$

Generally

- Pick a matrix Ψ such that:
- $\mathcal{Q} = \mathcal{P}\Psi$
- $\forall A, \mathcal{P}'.\ A \sqsubseteq \mathcal{P}'\gamma \rightarrow \mathcal{V}\ A\ (\mathcal{P}'\Psi)\delta$

Linear environments

Renaming

$$\underbrace{1C, 2A, 4A}_{\mathcal{P}\gamma} \sqsubseteq \underbrace{6A, 0B, 1C, 0D}_{\mathcal{Q}\delta}$$

$$\overbrace{\begin{pmatrix}6 & 0 & 1 & 0\end{pmatrix}}^{\mathcal{Q}} = \overbrace{\begin{pmatrix}1 & 2 & 4\end{pmatrix}}^{\mathcal{P}} \begin{array}{cc} \begin{pmatrix}0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\end{pmatrix} & \begin{array}{l}\delta \sqsupseteq C\\ \delta \sqsupseteq A\\ \delta \sqsupseteq A\end{array}\end{array}$$

Substitution

$$1(A \otimes B) \overset{\dashv}{\Rightarrow} 1A, 1B$$

$$\begin{pmatrix}1 & 1\end{pmatrix} = \begin{pmatrix}1\end{pmatrix}\begin{pmatrix}1 & 1\end{pmatrix} A, B \vdash A \otimes B$$

Generally

- Pick a matrix Ψ such that:
- $\mathcal{Q} = \mathcal{P}\Psi$
- $\forall A, \mathcal{P}'.\ A \sqsubseteq \mathcal{P}'\gamma \to \mathcal{V}\ A\ (\mathcal{P}'\Psi)\delta$

Generic notion of linear semantics

- Fundamental lemma of semantics:

$$\frac{\mathrm{Semantics}\ \mathcal{V}\ \mathcal{C} \qquad \Gamma \overset{\mathcal{V}}{\Rightarrow} \Delta}{\underbrace{A \dashv \Gamma \to \mathcal{C}\ A\ \Delta}_{\text{traversal}}}$$

- $\mathrm{Semantics}\ \mathcal{V}\ \mathcal{C}$ contains:
  - A proof that $\mathcal{V}$ is stable under structurality
  - $[\![\mathrm{var}]\!] : \forall[\ \mathcal{V}\ A \mathrel{\dot{\to}} \mathcal{C}\ A\ ]$
  - $[\![\mathrm{con}]\!] : \forall[\ \ [\![d]\!]\ (\mathrm{Kripke}\ \mathcal{V}\ \mathcal{C})\ A \mathrel{\dot{\to}} \mathcal{C}\ A\ ]$
- Traversal: similar to before, but with more algebra

Showing off — generic usage checker

- Let $\mathcal{V} = \sqsubseteq$ and $\mathcal{C}\ A\ \gamma = \forall \mathcal{R}.\ \mathrm{List}\ (\mathrm{Tm}_{d}\ A\ \mathcal{R}\gamma)$.
- We need a way to non-deterministically invert the semiring operations 0, 1, $+$, and $r \cdot -$.
- For example, $3 \rightsquigarrow [0 + 3, 1 + 2, 2 + 1, 3 + 0]$.
- Custom monadic handling of descriptions ($\dot{1}$, $\dot{\times}$, $\mathfrak{I}$, $*$, $r \cdot -$).
- Traverse an unannotated term, with guess annotations for the free variables.

Showing off — classical linear type theory

- Adapt Herbelin's *arborescente* presentation of $\mu\tilde{\mu}$-calculus[6].
- Where we previously had types A, B, C, &c., we have *conclusions* of the form $A\ \mathrm{term}$, $A\ \mathrm{coterm}$, or $\mathrm{command}$.
- (co)Variables are hypothetical (co)terms.
- Example rules:
  - $\langle v \| e \rangle$: $A\ \mathrm{term} * A\ \mathrm{coterm} \Longrightarrow \mathrm{command}$
  - $\mu\alpha.\ c$: $(A\ \mathrm{coterm} \vdash \mathrm{command}) \Longrightarrow A\ \mathrm{term}$

[6] Hugo Herbelin. *C'est maintenant qu'on calcule, au cœur de la dualité*. Habilitation. 2005.

Conclusion

- Adapted an intuitionistic framework to track usage information
- Linear metatheory in a natural deduction style
- First statement of linear simultaneous substitution (to my knowledge)
- Agda framework: `https://github.com/laMudri/generic-lr`
- Future work:
  - Write the paper!
  - Recursion/inductive types (implemented)
  - Testing the limits of expressibility
  - Intuitionistic requires intuitionistic, linear requires bunched — what else?