Towards a Universe of Linear Syntaxes with Binding

James Wood\textsuperscript{1} Bob Atkey\textsuperscript{1}

\textsuperscript{1}University of Strathclyde

VEST, June 2020
Context

- Linearity independent of binding
- Only one traversal over the syntax

---

Idea — stability under weakening$^4$

Two parts:

1. Consolidate all traversals over syntax (e.g., simultaneous renaming, simultaneous substitution, NbE, printing) into a single generic traversal (*kits/semantics*).

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”. Accepted for JFP. 2020.
Generic notion of semantics

- In the intuitionistic case, we have the following fundamental lemma of semantics:

\[
\text{Semantics } \forall C \quad \begin{array}{c}
\Gamma \ni A \rightarrow \forall A \Delta \\
\Gamma \vdash A \rightarrow C \ A \Delta \\
\end{array}
\]

- Semantics \( \forall C \) contains:
  - Proof that \( \forall \) is stable under weakening
  - Ways to interpret term constructors semantically, generic in context:

\[
\begin{align*}
[\text{var}] : \forall[ \forall A \rightarrow C \ A ] \\
[\text{app}] : \forall[ C (A \rightarrow B) \times C A \vdash C B ] \\
[\text{lam}] : \forall[ \Box(\forall A \rightarrow C \ B) \rightarrow C (A \rightarrow B) ]
\end{align*}
\]

\[
\ldots
\]
Generic notion of syntax

- A type system can:
  1. Offer a multitude of term formers. e.g, \texttt{App}, \texttt{Lam}, \ldots
  2. For each term former, require 0 or more premises. \times, \hat{1}
  3. For each premise, maybe bind variables. \Box, \forall

- Variables are a special case.
- Example descriptions:
  - \texttt{App}: \((A \rightarrow B) \times A \Rightarrow B\)
  - \texttt{Lam}: \((A \vdash B) \Rightarrow (A \rightarrow B)\)
Example derivation

\[ P = (A \rightarrow B) \otimes A \]

\[ p : P \vdash p : P \]

\[ p : P, \ f : A \rightarrow B, \ x : A \vdash f : A \rightarrow B \]

\[ p : P, \ f : A \rightarrow B, \ x : A \vdash f \ x : B \]

\[ p : P \vdash \text{let } (f \otimes x) = p \text{ in } f \ x : B \]

\[ \vdash \lambda p. \text{ let } (f \otimes x) = p \text{ in } f \ x : (A \rightarrow B) \otimes A \rightarrow B \]
Example derivation

\[ P = (A \to B) \otimes A \]

\[
\begin{align*}
1p : P & \vdash p : P \\
0p : P, 1f : A \to B, 0x : A \vdash f : A \to B, 0p : P, 0f : A \to B, 1x : A \vdash x : A, 0p : P, 1f : A \to B, 1x : A \vdash f \, x : B, 1p : P & \vdash \text{let } (f \otimes x) = p \text{ in } f \, x : B \\
\vdash \lambda p. \text{let } (f \otimes x) = p \text{ in } f \, x : (A \to B) \otimes A \to B
\end{align*}
\]
Vectors over semirings — addition

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

\[
\begin{align*}
\mathcal{P} \Gamma \vdash M : A \quad & \quad \mathcal{Q} \Gamma \vdash N : B \\
\mathcal{R} \unlhd \mathcal{P} + \mathcal{Q} \quad & \quad \mathcal{R} \Gamma \vdash (M \otimes N) : A \otimes B \\
\mathcal{R} \unlhd 0 \quad & \quad \mathcal{R} \Gamma \vdash (\otimes) : 1
\end{align*}
\]

identity, associativity, commutativity \sim contexts are essentially multisets
Vectors over semirings — multiplication

\[ \Gamma \ni (x : A) \quad R \trianglelefteq \langle x \rangle \]
\[ R\Gamma \vdash x : A \]

\[ P\Gamma \vdash M : A \quad R \trianglelefteq rP \]
\[ R\Gamma \vdash [M] : !rA \]

- \( \langle x \rangle \) — 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

\[
\frac{R \leq 0}{R \Gamma \vdash (\otimes) : 1}
\]

\[
\frac{P \Gamma \vdash M : A \quad Q \Gamma \vdash N : B}{R \leq P + Q}
\]

\[
\frac{\Gamma \ni (x : A) \quad R \leq \langle x \rangle}{R \Gamma \vdash x : A}
\]

\[
\frac{P \Gamma \vdash M : A \quad R \leq rP}{R \Gamma \vdash [M] : !_r A}
\]

- These four are the basic operations of linear algebra, three of them preserved by linear transformations.
- Notice: a variable which is syntactically absent may be given annotation 0.
Generic notion of linear\(^5\) semantics

- **Fundamental lemma of semantics:**

\[
\begin{align*}
\text{Semantics } \mathcal{V} C & \quad \left( i : \Gamma \ni A \right) \rightarrow \mathcal{V} A (\psi_{\Delta}) \quad Q \leq \mathcal{P} \psi \\
\mathcal{P} \Gamma \vdash A \rightarrow C A Q \Delta
\end{align*}
\]

- **Semantics } \mathcal{V} C \text{ contains:}**
  - Proof that } \mathcal{V} \text{ is stable under weakening by 0-use variables}
  - Ways to interpret term constructors semantically, generic in context:

\[
\begin{align*}
[\text{var}] : \forall [ \mathcal{V} A \rightarrow C A ] \\
[\text{app}] : \forall [ C (A \rightarrow B) * C A \rightarrow C B ] \\
[\text{lam}] : \forall [ \Box (\mathcal{V} A \rightarrow C B) \rightarrow C (A \rightarrow B) ]
\end{align*}
\]

\[\ldots\]

\(^5\)for a generic notion of linearity
Generic notion of linear syntax

Multiple premises are handled by bunched implications.\(^6\)

- \( I \, \mathcal{R} \Gamma := \mathcal{R} \sqsubseteq 0 \)
- \( (A \ast B) \, \mathcal{R} \Gamma := \Sigma P, Q. \ (\mathcal{R} \sqsubseteq P + Q) \times A \, P \Gamma \times B \, Q \Gamma \)
- \( (A \multimap B) \, \mathcal{P} \Gamma := \Pi Q, R. \ (\mathcal{R} \sqsubseteq P + Q) \rightarrow A \, Q \Gamma \rightarrow B \, R \Gamma \)
- \( (\!_r A) \, \mathcal{R} \Gamma := \Sigma P. \ (\mathcal{R} \sqsubseteq rP) \times A \, P \Gamma \)

Example description: \((\!_r A \ast (rA \vdash B)) \Rightarrow B\)

- \( \mathcal{P} \Gamma \vdash \!_r A \)
- \( Q \Gamma, \, rx : A \vdash B \)
- \( \mathcal{R} \sqsubseteq P + Q \)

- \( \mathcal{R} \Gamma \vdash B \)

- \[ \text{[bam]} : \forall [ \, C \ (\!_r A) \ast \Box (\!_r (\forall A) \multimap C \ B) \rightarrow C \ B \, ] \]

---

Conclusion

- **Past:**
  - Intuitionistic syntax-to-syntax traversals using kits.
  - Intuitionistic syntax-to-semantics traversals.
  - A generic notion of intuitionistic syntax with binding.

- **Present:**
  - Linear syntax-to-syntax traversals.
  - Using matrices/linear maps to describe compatibility of environments and traversals.

- **Future:**
  - Linear syntax-to-semantics traversals using matrices.
  - Semantics is stable under 0-use weakening.
  - A generic notion of linear syntax written in bunched implications style.
  - [https://github.com/laMudri/generic-lr](https://github.com/laMudri/generic-lr)