A canonical bidirectional typing discipline through polarised System L ($\mu\tilde{\mu}$-calculus)

Zanzi Mihejevs, Glaive

Motivation

- I come from Functional Programming
- Algebras, algebraic effects, monads, applicatives
- The dual story is often incomplete, the duals are not as common
- Hot take: We are missing some key primitives of coalgebras
- Programming with dual calculi is hard, so I decided to build a type-checker

What do we mean by canonical?

- Expressivity: linear logic - $\otimes$, $\oplus$, $⅋$, &, $\neg$, ~, $\forall$, $\exists$
  - $A \multimap B := \neg A ⅋ B$
  - $A \leftarrow B = A \otimes {\sim}B$
- No choices - each connective's check/synth discipline is fully determined by its polarity + 'chirality'
- The only annotations are on the shifts between check/synth
- Structurally recursive - no unification, no extraneous syntax

What is Chirality?

- Derived from the greek 'χείρ', meaning 'hand'
- An object or a system is chiral if it is mirror image is not identical to itself
- Refers to a kind of 'assymetric duality'.
- In our case, we will talk about the chirality between producer terms and consumer terms
- Slogan: the ~~principal~~ **Dominant** chirality is **Checkable**, the ~~auxilliary~~ **Sinister** chirality is **Synthesisable**

Chirality in System L / μμ̃

- Producer terms have a distinguished *output*:

$$\mathit{\Gamma} \vdash t : A | \mathit{\Delta}$$

- Consumer 'co-terms' have a distinguished *input*:

$$\mathit{\Gamma} | e : A \vdash \mathit{\Delta}$$

- A *cut* between a producer and a consumer:

$$\frac{\mathit{\Gamma} \vdash t : A, \mathit{\Delta} \quad \mathit{\Gamma}', e : A \vdash \mathit{\Delta}'}{\langle t | e \rangle : (\mathit{\Gamma}, \mathit{\Gamma}' \vdash \Delta, \Delta')}(\mathrm{Cut})$$

Introducing System L

Core rules:

$$\frac{}{x : A \vdash x : A \mid} \; VR \qquad \frac{}{\mid \alpha : A \vdash \alpha : A} \; VL$$

$$\frac{c : (\Gamma \vdash \alpha : A, \Delta)}{\Gamma \vdash \mu\alpha.c : A \mid \Delta} \; AR \qquad \frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma \mid \tilde{\mu}x.c : A \vdash \Delta} \; AL$$

$$\frac{\Gamma \vdash v : A \mid \Delta \quad \Gamma' \mid e : A \vdash \Delta'}{\langle v \| e \rangle : (\Gamma', \Gamma \vdash \Delta', \Delta)} \; Cut$$

- Non confluent - critical pair

CBV fragment - positive types

$$\frac{\Gamma \vdash v : A \mid \Delta}{\Gamma \vdash \iota_2\,(v) : A \oplus B \mid \Delta}\ \oplus R_1 \qquad\qquad \frac{\Gamma \vdash v : B \mid \Delta}{\Gamma \vdash \iota_1\,(v) : A \oplus B \mid \Delta}\ \oplus R_2$$

$$\frac{c : (\Gamma, x : A \vdash \Delta) \quad c' : (\Gamma, y : B \vdash \Delta)}{\Gamma \mid \tilde{\mu}[\iota_1\,(x).c \mid \iota_2\,(y).c'] : A \oplus B \vdash \Delta}\ \oplus L$$

$$\frac{\Gamma \vdash v : A \mid \Delta \quad \Gamma \vdash v' : B \mid \Delta}{\Gamma, \Gamma' \vdash (v, v') : A \otimes B \mid \Delta, \Delta'}\ \otimes R \qquad\qquad \frac{c : (\Gamma, x : A, y : B \vdash \Delta)}{\Gamma \mid \tilde{\mu}[(x, y).c] : A \otimes B \vdash \Delta}\ \otimes L$$

CBN fragment - negative types

$$\frac{c : (\Gamma \vdash \alpha : A, \Delta) \quad c' : (\Gamma \vdash \beta : B, \Delta)}{\Gamma \vdash \mu(\pi_1\,[\alpha].c \mid \pi_2\,[\beta].c') : A \mathbin{\&} B \mid \Delta}\ \&R$$

$$\frac{\Gamma \mid e : A \vdash \Delta}{\Gamma \mid \pi_1\,[e] : A \mathbin{\&} B \vdash \Delta}\ \&L_1 \qquad \frac{\Gamma \mid e : B \vdash \Delta}{\Gamma \mid \pi_2\,[e] : A \mathbin{\&} B \vdash \Delta}\ \&L_2$$

$$\frac{c : (\Gamma \vdash \alpha : A, \beta : B, \Delta)}{\Gamma \vdash \mu([\alpha, \beta].c) : A \parr B \mid \Delta}\ \parr R \qquad \frac{\Gamma \mid e : A \vdash \Delta \quad \Gamma \mid e' : B \vdash \Delta}{\Gamma, \Gamma' \mid [e, e'] : A \parr B \vdash \Delta, \Delta'}\ \parr L$$

Full calculus - negation

$$\frac{\Gamma \mid e : A \vdash \Delta}{\Gamma \vdash {\sim}(e) : {\sim}A \mid \Delta}\ {\sim}R \qquad \frac{c : (\Gamma \vdash \alpha : A, \Delta)}{\Gamma \mid \mu({\sim}(\alpha).c) : {\sim}A \vdash \Delta}\ {\sim}L$$

$$\frac{c : (\Gamma, x : A \vdash \Delta)}{\Gamma \vdash \mu(\neg[x].c) : \neg A \mid \Delta}\ \neg R \qquad \frac{\Gamma \vdash v : A \mid \Delta}{\Gamma \mid \neg[v] : \neg A \vdash \Delta}\ \neg L$$

- Adding polarity means that the identity function becomes not well-kinded

Full-calculus - shifts

$$\frac{\Gamma \vdash v_- : A_- \mid \Delta}{\Gamma \vdash \downarrow(v_-) : \downarrow A_- \mid \Delta} \downarrow R \qquad \frac{c : (\Gamma, x^- : A_- \vdash \Delta)}{\Gamma \mid \tilde{\mu}[\downarrow(x^-).c] : \downarrow A_- \vdash \Delta} \downarrow L$$

$$\frac{c : (\Gamma \vdash \alpha^+ : A_+, \Delta)}{\Gamma \vdash \mu(\uparrow[\alpha^+].c) : \uparrow A_+ \mid \Delta} \uparrow R \qquad \frac{\Gamma \mid e_+ : A_+ \vdash \Delta}{\Gamma \mid \uparrow[e_+] : \uparrow A_+ \vdash \Delta} \uparrow L$$

- Wouldn't it be nice if these shifts coincided with bidi-shifts?
- This is foreshadowing

How to type-check System L?

- The cruicial question is how to go under binders
- Pair introduction is checkable
- We expect Pair elimination to be synthesisable
- But it has binders

$$\frac{c : (\Gamma, x^+ : A_+, y^+ : B_+ \vdash \Delta)}{\Gamma \mid \tilde{\mu}[(x^+, y^+).c] : A_+ \otimes B_+ \vdash \Delta} \otimes L$$

- In lambda calculus, the only binding form is checkable, so going under a binder is easy

Noam's reverse bidirectional typing

- instead of assuming a typing context, we *discover* the types as we go along

$$\frac{\Gamma, x_1 \Leftarrow A_1, x_2 \Leftarrow A_2 \vdash e' \Rightarrow C \qquad \Delta \vdash e \Leftarrow A_1 \otimes A_2}{\Gamma, \Delta \vdash \mathsf{let}\,\langle x_1, x_2 \rangle = e\,\mathsf{in}\,e' \Rightarrow C} \quad .$$

- Cruicially, variables become checkable rather than synthesisable

$$\frac{}{x \Leftarrow A \vdash x \Leftarrow A}$$

Generalise to cuts between positive types

- Checking positive types:

$$\frac{\Gamma \vdash \mathrm{producer} \overset{check}{\Leftarrow} A \mid \Delta \quad \Gamma' \mid A \overset{synth}{\Leftarrow} \mathrm{pattern} \vdash \Delta'}{\langle \mathrm{producer} \mid \mathrm{pattern} \rangle : (\Gamma, \Gamma' \vdash \Delta, \Delta')}\,(\mathrm{Cut})$$

- The causal flow is right-to-left, from the sinister (pattern) chirality to the dominant (producer) chirality

Flip the recipe for negative types

- Checking negative types:

$$\frac{\Gamma \vdash \text{co-pattern} \overset{synth}{\Rightarrow} A \mid \Delta \quad \Gamma' \mid A \overset{check}{\Rightarrow} \text{consumer} \vdash \Delta'}{\langle \text{co-pattern} \mid \text{consumer} \rangle : (\Gamma, \Gamma' \vdash \Delta, \Delta')}(\text{Cut})$$

- The causal flow is left-to-right, from the sinister (co-pattern) chirality to the dominant (consumer) chirality

Negation preserves the dominant chirality

Negation Left ($\neg L$)

$$\frac{\Gamma \vdash V_+ \overset{check}{\Leftarrow} A_+; \Delta}{\Gamma; \neg[V_+] \overset{check}{\Rightarrow} \neg A_+ \vdash \Delta}\ \neg L$$

Negation Right ($\neg R$)

$$\frac{c : (\Gamma, x^+ \overset{synth}{\Leftarrow} A_+ \vdash \Delta)}{\Gamma \vdash \mu(\neg[x^+].\, c) \overset{synth}{\Rightarrow} \neg A_+ \mid \Delta}\ \neg R$$

Tilde Negation Right ($\sim R$)

$$\frac{\Gamma; E_- \overset{check}{\Rightarrow} A_- \vdash \Delta}{\Gamma \vdash \sim (E_-) \overset{check}{\Leftarrow} \sim A_-; \Delta}\ \sim R$$

Tilde Negation Left ($\sim L$)

$$\frac{c : (\Gamma \vdash \alpha^- \overset{synth}{\Rightarrow} A_-, \Delta)}{\Gamma \mid \tilde{\mu}[\sim (\alpha^-).\, c] \overset{synth}{\Leftarrow} \sim A_- \vdash \Delta}\ \sim L$$

Shifts flip the dominant chirality

Down Shift Right ($\downarrow R$)

$$\frac{\Gamma \vdash v_- \overset{check}{\Rightarrow} A_- \mid \Delta}{\Gamma \vdash \downarrow (v_-) \overset{synth}{\Leftarrow} \downarrow A_-; \Delta} \downarrow R$$

Down Shift Left ($\downarrow L$)

$$\frac{c : (\Gamma, x^- \overset{synth}{\Rightarrow} A_- \vdash \Delta)}{\Gamma \mid \tilde{\mu}[\downarrow (x^-). c] \overset{check}{\Leftarrow} \downarrow A_- \vdash \Delta} \downarrow L$$

Up Shift Right ($\uparrow R$)

$$\frac{c : (\Gamma \vdash \alpha^+ \overset{synth}{\Leftarrow} A_+, \Delta)}{\Gamma \vdash \mu(\uparrow [\alpha^+]. c) \overset{check}{\Rightarrow} \uparrow A_+ \mid \Delta} \uparrow R$$

Up Shift Left ($\uparrow L$)

$$\frac{\Gamma \mid e_+ \overset{check}{\Leftarrow} A_+ \vdash \Delta}{\Gamma; \uparrow [e_+] \overset{synth}{\Rightarrow} \uparrow A_+ \vdash \Delta} \uparrow L$$

Conclusion

- Typing algorithm for polymorphic substructural type theory
- The information flow goes from the sinister to the dominant chirality
- The shifts from System L coincides with the shifts from bidirectional typing

Future work

- Type operators - type-level sequent calculus?
- Codebruijn - recovering copying and deleting of variables
- Dependent types
- Subtyping and duotyping
- Categorical semantics is very subtle