Information Awareness is Super Effective!

Philippa Cowderoy

SPLV 2019

email: *flippa@flippac.org*
twitter: *@flippacpub*

Symbol Games and Hidden Information

Our standard notation hides things from us.

$$\frac{\Gamma \vdash Tp : \tau p \qquad \Gamma \vdash Tf : \tau p \to \tau r}{\Gamma \vdash Tf\ Tp : \tau r}\ App1 \qquad \frac{\Gamma \vdash Tp : \tau p \qquad \Gamma \vdash Tf : \tau f \qquad \tau p \to \tau r = \tau f}{\Gamma \vdash Tf\ Tp : \tau r}\ App2$$

- We are used to $App1$
  - But $App2$ is easier for beginners to understand
- An implicit constraint ($=$) is made explicit
- When checking, τr visibly 'comes from' the constraint

What are Information Effects – What are Effects?

Effects can be seen in relation to the models of computing they break:

General computing	Hello World Anything IO or 'real'
Functional programming	Mutation Control effects
Total programming	Non-termination
Logic programming	Exposing the solver (eg cut)

What are Information Effects – Breaking Isomorphism

- Arose from work on reversible programming
- Introduced by James and Sabry
- Isomorphic programming is a viable model
  - Conservation of information holds
- Conservation of information can be broken by:
  - Creating information
  - Destroying information

What Can Information Effects Tell Us?

Suppose we're writing an interpreter:

- Typical 'fold' pattern
  - Traverse AST, use things at nodes
  - Maybe target an abstract machine
- Syntactic sugar will be destroyed
  - Whether or not explicitly desugared
- Created information is semantics
  - The *meaning* we give the program!

Constraints for the Simply Typed Lambda Calculus

$$\tau = \tau \quad \text{Type equality}$$

$$x : \tau \in \Gamma \quad \text{Binding in context}$$
$$\Gamma' := \Gamma \; ; \; x : \tau \quad \text{Context extension}$$
$$\Gamma \multimap \langle {}^{\Gamma L}_{\Gamma R} \quad \text{Context duplication}$$

- Convention: write $=$ as if 'assigning' to LHS
- Context constraints encode the structural rules
- An alternative interpretation could give us a minimal linear calculus

Information-Aware Simply-Typed λ-Calculus

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau \;\; Var} \qquad \frac{\begin{array}{c}\Gamma f := \Gamma \; ; \; x : \tau p \\ \Gamma f \vdash T : \tau r \\ \tau f = \tau p \to \tau r\end{array}}{\Gamma \vdash \lambda x.T \;\; : \;\; \tau f \;\; Lam}$$

$$\frac{\begin{array}{c}\Gamma \multimap \langle {}^{\Gamma f}_{\Gamma p} \\ \Gamma f \vdash Tf : \tau f \qquad \Gamma p \vdash Tp : \tau p \\ \tau p \to \tau r = \tau f\end{array}}{\Gamma \vdash Tf \;\; Tp : \tau r \qquad App}$$

Information-Aware Simply-Typed λ-Calculus (moded)

Mode: $\Gamma^{+} \vdash T^{+} : \tau^{-}$ (Synthesis or 'typechecking')

$$\frac{x^{-} : \tau^{+} \in \Gamma^{-}}{\Gamma^{+} \vdash x^{+} : \tau^{-}}\ \mathit{Var} \qquad \frac{\begin{array}{c}\Gamma f^{+} := \Gamma^{-}\ ;\ x^{-} : \tau p^{+} \\ \Gamma f^{-} \vdash T^{-} : \tau r^{+} \\ \tau f^{+} = \tau p^{-} \rightarrow^{+} \tau r^{-}\end{array}}{\Gamma^{+} \vdash \lambda x^{+}.T^{+} : \tau f^{-}}\ \mathit{Lam}$$

$$\frac{\begin{array}{c}\Gamma^{-} \dashv\!\left\langle {}^{\Gamma f^{+}}_{\Gamma p^{+}} \right. \\ \Gamma f^{-} \vdash Tf^{-} : \tau f^{+} \qquad \Gamma p^{-} \vdash Tp^{-} : \tau p^{+} \\ \tau p^{-} \rightarrow^{-} \tau r^{+} = \tau f^{-}\end{array}}{\Gamma^{+} \vdash Tf^{+}\ Tp^{+} : \tau r^{-}}\ \mathit{App}$$

New Tricks

Done:

- Information Aware type system presentations
- Telescopic Constraint Trees
  - Derivable from Information Aware rules
  - Represent [ongoing] checking problems

Working on:

- Information Aware elaboration
  - Type-aware translation with sugar
  - One copy to ‘translate’, one to think about
  - Composition – entailment of constraints
- Error reporting

Generalising the Ultimate Answer

A telescopic constraint tree for
$\text{let } id = \lambda x.x \text{ in } id\ 42$:

$$
\begin{array}{l|r}
\{42 : \forall.\mathbb{N}\}, \{\exists \tau a\}\{\exists \sigma\} & \text{Setup\&let} \\
\quad |\{\S,\ \exists \tau b,\ \sigma = \mathit{Gen}(\tau b)\} \ldots & \text{let (LHS)} \\
\quad \ldots \quad \{\exists \tau p,\ \exists \tau r,\ !x : \forall.\tau p,\ \tau b = \tau p \to \tau r\} \ldots & \text{lambda} \\
\qquad\quad \ldots \quad \{\exists \sigma r,\ ?x : \sigma r,\ \sigma r \geq \tau r\} & x \\
\quad |\{!id : \sigma\} \ldots & \text{let (RHS)} \\
\quad \ldots \quad \{\exists \tau f,\ \exists \tau p,\ \tau p \to \tau a = \tau f\} & \text{app} \\
\qquad\qquad |\{\exists \sigma f,\ ?id : \sigma f,\ \sigma f \geq \tau f\} & id \\
\qquad\qquad |\{\exists \sigma p,\ ?42 : \sigma p,\ \sigma p \geq \tau p\} & 42
\end{array}
$$

As in the original talk, the inference rules are left as an exercise for the reader...

Summary

- Information effects help make sense of type systems a lot
- Informal uses too: illumination and inspiration
- Structural proof theory for hackers?
- Notation and ideas ready for serious challenges

Other talks:

- SPLS March 2019:
  Information Aware Type Systems
- Strathclyde MSP101, 2018:
  Telescopic [Constraint] Trees, or:
  Information-Aware Type Systems in Context

Extra: Free Join-the-Dots slide! *Connect +ve to −ve*

$$\frac{x^- : \tau^+ \in \Gamma^-}{\Gamma^+ \vdash x^+ : \tau^-}\ \mathit{Var} \qquad \frac{\begin{array}{c}\Gamma f^+ := \Gamma^- \ ; x^- : \tau p^+ \\ \Gamma f^- \vdash T^- : \tau r^+ \\ \tau f^+ = \tau p^- \to^+ \tau r^-\end{array}}{\Gamma^+ \vdash \lambda x^+ . T^+ \ : \ \tau f^-}\ \mathit{Lam}$$

$$\frac{\begin{array}{c}\Gamma^- \dashv\!\langle {}^{\Gamma f^+}_{\Gamma p^+} \\ \Gamma f^- \vdash Tf^- : \tau f^+ \qquad \Gamma p^- \vdash Tp^- : \tau p^+ \\ \tau p^- \to^- \tau r^+ = \tau f^-\end{array}}{\Gamma^+ \vdash Tf^+ \ Tp^+ : \tau r^-}\ \mathit{App}$$