Tracking **context-dependent** properties using **coeffects**

Tomas Petricek
University of Cambridge

The Big Picture

- Track properties of computations
  - Language allows everything
  - Extend **existing programs**
  - Annotate **libraries**
  - Track information **using types**
- Different kinds of properties
  - **Effects** - what computations do
  - **Coeffects** - how computations use context

Introducing **effect** and **coeffect** systems

Effect systems

When to use effect systems?

$$\Gamma \vdash e : \tau \;\&\; \boldsymbol{\sigma}$$

Typing judgment

Given variables Γ

… expression e has a type τ

… and performs effects $\boldsymbol{\sigma}$

Tracking **memory operations**

Primitive operations have effects

$$\frac{r\text{: ref}_p \in \Gamma \qquad \Gamma \vdash e: \tau \; \& \; \boldsymbol{\sigma}}{\Gamma \vdash r \leftarrow e: \text{unit} \; \& \; \boldsymbol{\sigma} \cup \{\mathbf{w}(\boldsymbol{p})\}}$$

Composition combines effects

$$\frac{\Gamma \vdash e_1: \tau_1 \; \& \; \boldsymbol{\sigma_1} \qquad \Gamma, x: \tau_1 \vdash e_2: \tau_2 \; \& \; \boldsymbol{\sigma_2}}{\Gamma \vdash \textbf{let}\; x = e_1 \;\textbf{in}\; e_2 : \tau_2 \; \& \; \boldsymbol{\sigma_1} \cup \boldsymbol{\sigma_2}}$$

Coeffect systems

When to use effect systems?

$$\Gamma \,@\, \boldsymbol{\sigma} \vdash e : \tau$$

Typing judgment

Given variables Γ

… with additional context $\boldsymbol{\sigma}$

… expression e has a type τ

Distributed programming

Primitives with limited modalities

$$\Gamma \,@\, \{\text{server}, \text{client}\} \vdash \mathit{writeFile} : \text{string} \rightarrow \text{unit}$$

$$\Gamma \,@\, \{\text{client}, \text{phone}\} \vdash \mathit{readInput} : \text{unit} \rightarrow \text{string}$$

Composition combines coeffects

$$\frac{\Gamma \,@\, \boldsymbol{\sigma_1} \vdash e_1 : \tau_1 \qquad \Gamma, x : \tau_1 \,@\, \boldsymbol{\sigma_2} \vdash e_2 : \tau_2}{\Gamma \,@\, \boldsymbol{\sigma_1} \cap \boldsymbol{\sigma_2} \vdash \mathbf{let}\; x = e_1 \;\mathbf{in}\; e_2 : \tau_2}$$

Effect and **coeffect** systems

Effect systems

- Annotations on the result
- Propagate information forward
- Correspond to **monads**

Coeffect systems

- Annotations on the context
- Propagate information backward
- Correspond to **comonads**

The marriage of
coeffects and **comonads**

Categorical semantics approach

Interpret **expressions** in context

$$x_1 : \tau_1, \ldots, x_n : \tau_n \vdash e : \tau$$

As **functions** of context

$$[\![\tau_1 \times \cdots \times \tau_n]\!] \to [\![\tau]\!]$$

Additional structure over **result**

Additional structure over **domain**

Monadic lambda calculus

Monadic type for effects

$$\frac{r\colon \mathrm{ref}_p \in \Gamma \qquad \Gamma \vdash e\colon \mathbf{IO}\ \boldsymbol{\tau}}{\Gamma \vdash r \leftarrow e\colon \mathbf{IO\ unit}}$$

Composition combines effects

$$\frac{\Gamma \vdash e_1\colon \mathbf{IO}\ \boldsymbol{\tau_1} \qquad \Gamma, x\colon \tau_1 \vdash e_2\colon \mathbf{IO}\ \boldsymbol{\tau_2}}{\Gamma \vdash \mathbf{let}\ x = e_1\ \mathbf{in}\ e_2 : \mathbf{IO}\ \boldsymbol{\tau_2}}$$

Type means there are **some IO effects**

The marriage of **effects** and **monads**

Capture effects using **tagged** monads

$$\frac{r : \text{ref}_p \in \Gamma \qquad \Gamma \vdash e : \boldsymbol{M^{\sigma}}\tau}{\Gamma \vdash r \leftarrow e : \boldsymbol{M^{\sigma \otimes \{\mathrm{w}(p)\}}}\text{unit}}$$

Composition combines effects

$$\frac{\Gamma \vdash e_1 : \boldsymbol{M^{\sigma_1}\tau_1} \qquad \Gamma, x : \tau_1 \vdash e_2 : \boldsymbol{M^{\sigma_2}\tau_2}}{\Gamma \vdash \textbf{let}\, x = e_1 \textbf{ in } e_2 : \boldsymbol{M^{\sigma_1 \otimes \sigma_2}\tau_2}}$$

As **precise type** as in effect systems

The marriage of **effects** and **monads**

Tagged monad structure over the result

Tag $\boldsymbol{r}$ captures the effects

$$\tau_1 \to \boldsymbol{M}^{\boldsymbol{r}}\tau_2$$

Defines **composition**

$$(\tau_1 \to \boldsymbol{M}^{\boldsymbol{r}}\tau_2) \to (\tau_2 \to \boldsymbol{M}^{\boldsymbol{s}}\tau_3) \to \left(\tau_1 \to \boldsymbol{M}^{\boldsymbol{r}\otimes\boldsymbol{s}}\tau_3\right)$$

And **pure** computations

$$\tau \to \boldsymbol{M}^{\boldsymbol{1}}\tau$$

The marriage of coeffects and comonads

Capture context using tagged comonads

$$\mathsf{C}^{\{\text{client},\text{phone}\}}\Gamma \vdash \mathit{readInput} : \text{unit} \to \text{string}$$

Composition combines coeffects

$$\frac{C^{\sigma_1}\Gamma \vdash e_1 : \tau_1 \qquad C^{\sigma_2}(\Gamma, x:\tau_1) \vdash e_2 : \tau_2}{\mathsf{C}^{\sigma_1 \otimes \sigma_2}\Gamma \vdash \textbf{let}\; x = e_1 \;\textbf{in}\; e_2 : \tau_2}$$

Tagged comonadic lambda calculus

The marriage of coeffects and comonads

Tagged comonad structure over the domain

Tag r captures the coeffects

$$C^r \tau_1 \to \tau_2$$

Defines **composition**

$$(C^r \tau_1 \to \tau_2) \to (C^s \tau_2 \to \tau_3) \to \left(C^{r \otimes s} \tau_1 \to \tau_3\right)$$

And **pure** computations

$$C^1 \tau \to \tau$$

More **examples** and the **lambda abstraction**

Distributed programming

Tagged with **sets** of environments

$$\mathsf{C}^{\{\text{client,phone}\}}\Gamma \vdash \mathit{read} : \text{string} \rightarrow \text{string}$$
$$\mathsf{C}^{\{\text{server,client}\}}\Gamma \vdash \mathit{write} : \text{string} \rightarrow \text{unit}$$

Tags combined using **intersection**

Lambda abstraction in a **pure** context

$$\frac{C^{\{\text{client}\}}(\Gamma, x : \text{unit}) \vdash \mathit{write}\ (\mathit{read}\ x) : \text{unit}}{\mathsf{C}^{\{\text{server,client,phone}\}}\Gamma \vdash \lambda x.\, \mathit{write}\ (\mathit{read}\ x) : C^{\{\text{client}\}}\text{unit} \rightarrow \text{unit}}$$

Introducing implicit parameters

Configuration problem

- Parameterize function deep in the call tree
- Without adding parameters to all functions

Implicit parameters *?param*

let print = λ*prefix* →
  if length *prefix* > ***?width*** **then**
    λ*str* → append *prefix str* ***?width ?size***
  else …

Implicit parameters

Tagged with **sets** of parameters

$$\mathsf{C}^{\{?\mathbf{width},?\mathbf{size}\}}\Gamma \vdash \text{append} \ldots\ ?\text{width}\ ?\text{size}\ :\ \text{string}$$

Tags combined using **intersection**

Lambda abstraction **combines** contexts

$$
\begin{array}{l}
\textbf{let}\ \text{print} = \lambda \textit{prefix} \rightarrow \\
\quad \textbf{if}\ \text{length}\ \textit{prefix} > \textit{?width}\ \textbf{then} \\
\quad\quad \lambda \textit{str} \rightarrow \text{append}\ \textit{prefix}\ \textit{str}\ \textit{?width}\ \textit{?size}\ \ \ldots
\end{array}
$$

$$\text{print} :\ \mathsf{C}^{\{?\mathbf{width}\}}\text{string} \rightarrow (\mathsf{C}^{\{?\mathbf{size}\}}\text{string} \rightarrow \text{string})$$

Type system for **coeffects**

Comonadic **coeffect** typing

$$\frac{\mathcal{C}^{r}\Gamma \vdash e_1 : \mathcal{C}^{t}\tau_1 \to \tau_2 \qquad \mathcal{C}^{s}\Gamma \vdash e_2 : \tau_1}{\mathbf{C}^{r \otimes s \otimes t}\Gamma \vdash e_1\ e_2 : \tau_2}$$

$$\frac{x : \tau \in \Gamma}{\mathbf{C}^{1}\Gamma \vdash x : \tau}$$

$$\frac{\mathcal{C}^{r \otimes s}\ (\Gamma, x : \tau_1) \vdash e : \tau_2}{\mathbf{C}^{s}\Gamma \vdash \lambda x. e\ : \mathcal{C}^{r}\tau_1 \to \tau_2}$$

Structure of the tags

Monoid with binary operation $(\boldsymbol{M}, \otimes, \mathbf{1})$

Type preservation requires

- Idempotence $\boldsymbol{r} \otimes \boldsymbol{r} = \boldsymbol{r}$
- Symmetry $\boldsymbol{r} \otimes \boldsymbol{s} = \boldsymbol{s} \otimes \boldsymbol{r}$
- Partial order $\forall r \in M.\ \mathbf{1} \sqsubseteq \boldsymbol{r}$

Unrestricted reduction needs **set-like** tags

- Is there a more **fine-grained structure**?

Comparing **coeffects** and **effects**

Comonadic abstraction **captures context**

$$\frac{\boldsymbol{C}^{r \oplus s}\,(\Gamma, x\!:\tau_1) \vdash e : \tau_2}{\mathbf{C}^{s}\Gamma \vdash \lambda x.\,e \;:\; \boldsymbol{C}^{r}\tau_1 \to \tau_2}$$

Monadic abstraction is **always pure**

$$\frac{\Gamma, x\!:\tau_1 \vdash e : \boldsymbol{M}^{r}\tau_2}{\Gamma \vdash \lambda x.\,e \;:\; \boldsymbol{M}^{1}(\tau_1 \to \boldsymbol{M}^{r}\tau_2)}$$

Compare **implicit parameters** and **reader** monad

More precise **structural coeffects** system

Tracking properties per variable

Example: checked array indexing

$$\mathbf{C}^{3\times 5}(x{:}\,A, y{:}\,A) \vdash \boldsymbol{x}.\mathbf{3} + \boldsymbol{y}.\mathbf{5} + \boldsymbol{y}.\mathbf{3} : \mathrm{int}$$

Structural rules manipulate tags

Tags correspond to variables

$$\frac{\mathbf{C}^{r\times s}\,(\Gamma_1, \Gamma_2) \vdash e{:}\,\tau}{\mathbf{C}^{s\times r}(\Gamma_2, \Gamma_1) \vdash e{:}\tau} \qquad \frac{\mathbf{C}^{r}\,\Gamma \vdash e{:}\,\tau}{\mathbf{C}^{r\times 1}(\Gamma, x{:}\,\tau') \vdash e{:}\tau}$$

Tracking properties per variable

Contraction rule combines coeffects

$$\frac{\mathbf{C}^{\mathbf{3\times 5}}\ (x{:}\,A, y{:}\,A) \vdash \boldsymbol{y}.\mathbf{2} + \boldsymbol{x}.\mathbf{3} + \boldsymbol{y}.\mathbf{5} : \mathrm{int}}{\mathbf{C}^{\mathbf{max(3,5)}}(z{:}\,A) \vdash \boldsymbol{z}.\mathbf{2} + \boldsymbol{z}.\mathbf{3} + \boldsymbol{z}.\mathbf{5} : \mathrm{int}}$$

Use **two different** operations

- $\times$ for product structure
- $\otimes$ for combining coeffects

$$\frac{\mathbf{C}^{\boldsymbol{r\times s}}\ (x{:}\,\tau, y{:}\,\tau) \vdash e{:}\,\tau_1}{\mathbf{C}^{\boldsymbol{r\otimes s}}(z{:}\,\tau) \vdash e[z/x][z/y]{:}\,\tau_1}$$

More precise coeffects

Generalized application rule

$$\frac{C^{r}\Gamma_1 \vdash e_1 : C^{t}\tau_1 \to \tau_2 \qquad C^{s}\Gamma_2 \vdash e_2 : \tau_1}{C^{r\times(t\otimes s)}(\Gamma_1, \Gamma_2) \vdash e_1\ e_2 : \tau_2}$$

Point-wise (or scalar) application of $\otimes$

$$\frac{\begin{array}{c} \mathrm{C} \vdash \lambda y.\,y.\,5 : \mathrm{C}^{5}\mathrm{A} \to \mathrm{int} \\ \mathrm{C}^{0\times 0}\ (x{:}\,A, z{:}\,A) \vdash (\textbf{if}\ \ldots \textbf{then}\ x\ \textbf{else}\ z) : \mathrm{A} \end{array}}{\mathrm{C}^{5\times 5}(x{:}\,A, z{:}\,A) \vdash (\lambda y.\,y.\,5)(\textbf{if}\ \ldots \textbf{then}\ x\ \textbf{else}\ z) : \mathrm{int}}$$

More precise **coeffects**

Tag structure with two operations

- $\times$ for product structure
- $\otimes$ for combining coeffects
- Distributivity law $(\boldsymbol{a} \times \boldsymbol{b}) \otimes \boldsymbol{c} = (\boldsymbol{a} \otimes \boldsymbol{c}) \times (\boldsymbol{b} \otimes \boldsymbol{c})$

Future work

- Does it generalize simple version?
- Refined categorical semantics
- **Application:** Secure information flow
- **Application:** Multi-stage programming

Conclusions

Summary

Introducing **coeffects**

- Context-dependent properties
- Modeled using **comonads**
- Distributed, dynamic scoping, multi-stage, security

Tracking information

- Simple **set-like** structures for **context properties**
- **Precise structure** associates data with **variables**

Backup slides

Categorical semantics for coeffects

Categorical **semantics** for coeffects

Operations of a **tagged comonad**

$$(\mathcal{C}^{r}\tau_1 \to \tau_2) \to (\mathcal{C}^{s}\tau_2 \to \tau_3) \to (\mathcal{C}^{r \otimes s}\tau_1 \to \tau_3)$$

$$\mathcal{C}^{1}\tau \to \tau$$

With additional structure

- **Combine** contexts for abstraction
- **Split** contexts for application

Semantics of **lambda abstraction**

Combine the **inner** and **outer** scope

Use **monoidal tagged comonad**

$$[\![\lambda x. e]\!] = \text{curry}\,([\![e]\!] \circ \text{combine})$$

$$\text{combine} : \ C^r \tau_1 \times C^s \tau_2 \rightarrow C^{r \otimes s}(\tau_1 \times \tau_2)$$

Other **variations of tags** are possible!

Restrict one context or require **equal** tags

Future work

Could be done in the **monadic** setting

Semantics of **application**

Evaluate both expressions and apply

$$[\![e_1\ e_2]\!] = \mathrm{ev} \circ \langle [\![e_1]\!], \mathrm{cobind}\ [\![e_2]\!] \rangle$$

Split context between two functions

$$f : C^r \tau \to \tau_1 \qquad \langle f, g \rangle : C^{r \otimes s} \tau \to (\tau_1, \tau_2)$$
$$g : C^s \tau \to \tau_2$$

Tagged inverse of the combine operation

$$\mathrm{split}_{r,s} : C^{r \otimes s}(\tau_1 \times \tau_2) \to C^r \tau_1 \times C^s \tau_2$$