Coeffects

Types for tracking context-dependence

Tomas Petricek, University of Cambridge
tomas.petricek@cl.cam.ac.uk | @tomaspetricek

Joint work with **Dominic Orchard** and **Alan Mycroft**

Properties of computations

Effects on the environment

$\Gamma \vdash e : \boldsymbol{M}^{\boldsymbol{\sigma}}\tau$

Memory, network, I/O, …

Effect systems and monads

Requirements from the environment

$\mathbf{C}^{\boldsymbol{\sigma}}\Gamma \vdash e : \tau$

Resources, meta-data, contexts…

Coeffect systems and comonads

Effects $\neq$ Coeffects

Lambda abstraction rule

> *"In the rule for abstraction, the effect is empty because evaluation immediately returns the function, with no side effects. The effect on the function arrow is the same as the effect for the function body, because applying the function will have the same side effects as evaluating the body.* [Wadler and Thiemann, 2003]

Not true for *context-dependent* properties!

Motivating examples

Implicit parameters are coeffects

Dynamically scoped parameters **?param**

```
let ?culture = "en-US"

let print = fun (num) →
   printNumber num ?culture ?format
```

Abstraction splits requirements

$$\frac{C^{r \cup s}(\Gamma, x : \tau_1) \vdash e : \tau_2}{C^{r}\Gamma \vdash \lambda x. e : C^{s}\tau_1 \rightarrow \tau_2}$$

Implicit parameters are coeffects

Dynamically scoped parameters **?param**

```
let ?culture = "en-US"

let print = fun (num) →
   printNumber num ?culture ?format
```

Different typing for different uses

$$C^{\{?\text{culture}\}}\Gamma \vdash \text{print} : C^{\{?\text{format}\}}\text{int} \rightarrow \text{unit}$$

$$C^{\emptyset}\Gamma \vdash \text{print} : C^{\{?\text{format},\ ?\text{culture}\}}\text{int} \rightarrow \text{unit}$$

Liveness is a coeffect

Is the entire variable context live or dead?

$$\frac{x : \tau \in \Gamma}{C^{1}\Gamma \vdash x : \tau} \qquad \frac{n \in \{0,1,2,\ldots\}}{C^{0}\Gamma \vdash n : \iota}$$

Application propagates contexts

foo 42

(**fun** x → x) 42

(**fun** x → 42) y

$$\frac{C^{r}\Gamma \vdash e_1 : C^{t}\tau_1 \rightarrow \tau_2 \qquad C^{s}\Gamma \vdash e_2 : \tau_1}{C^{r \vee (s \wedge t)}\Gamma \vdash e_1\ e_2 : \tau_2}$$

Semantics of liveness

Context with liveness annotation

$$[\![C^r \Gamma \vdash e : \tau]\!] \;=\; C^r(\tau_1 \times \cdots \times \tau_n) \to \tau$$

Modeled using indexed **Maybe** type

Definitely dead: Nothing $\qquad C^0 \tau = 1$

Maybe live: Just of context $\qquad C^1 \tau = \tau$

Impossible using a monad!

$$[\![C^r \Gamma \vdash e : \tau]\!] \;=\; (\tau_1 \times \cdots \times \tau_n) \to M^\sigma \tau$$

Structural coeffects for liveness

Track liveness of individual variables

Use structural type system

Product × that mirrors variable structure

$$\frac{C^{r \times s}(\Gamma, x : \tau_1) \vdash e : \tau_2}{C^{r}\Gamma \vdash \lambda x. e : C^{s}\tau_1 \rightarrow \tau_2}$$

Add contraction, exchange, weakening etc.

$$\frac{C^{r \times s}(x : \tau, y : \tau) \vdash e : \tau'}{C^{r \vee s}(z : \tau) \vdash \{z/x\}\{z/y\}e : \tau'}$$

$$\frac{C^{r \times s}(\Gamma_1, \Gamma_2) \vdash e : \tau}{C^{s \times r}(\Gamma_2, \Gamma_1) \vdash e : \tau}$$

Efficient data-flow language

Discrete-time computations over streams

Access past value using **prev** keyword

How many past values need to cache?

$$\frac{\boldsymbol{C}^{\boldsymbol{r}}\,\Gamma \vdash e : \tau}{\boldsymbol{C}^{\boldsymbol{r}+1}\,\Gamma \vdash \mathbf{prev}\; e : \tau}$$

$$\frac{\boldsymbol{C}^{\min(\boldsymbol{r},\boldsymbol{s})}(\Gamma, \mathrm{x} : \tau_1) \vdash e : \tau_2}{\boldsymbol{C}^{\boldsymbol{r}}\,\Gamma \vdash \lambda x.\, e : \boldsymbol{C}^{\boldsymbol{s}}\tau_1 \rightarrow \tau_2}$$

$$\frac{\boldsymbol{C}^{\boldsymbol{r}}\,\Gamma \vdash e_1 : \boldsymbol{C}^{\boldsymbol{t}}\tau_1 \rightarrow \tau_2 \qquad \boldsymbol{C}^{\boldsymbol{s}}\,\Gamma \vdash \mathrm{e}_2 : \tau_1}{\boldsymbol{C}^{\max(\boldsymbol{r},\boldsymbol{s}+\boldsymbol{t})}\,\Gamma \vdash e_1\; e_2 : \tau_2}$$

Semantics of data-flow language

Comonadic semantics [Uustalu & Vene, 2008]

Non-empty list
Over the domain

$$\mathbf{NEList}\,\tau = \tau \times (1 + \mathbf{NEList}\,\tau)$$

Indexed comonadic semantics

Indexed **List** type
Index specifies length

$$C^0\tau = \tau$$
$$C^1\tau = \tau \times (\tau)$$
$$C^2\tau = \tau \times (\tau \times \tau)$$
$$\ldots$$

Some more theory

Coeffect systems

Flat and structural variants

- Both are useful for some applications
- Flat: Implicits, dataflow, cross-compilation
- Structural: Liveness, security, taintedness

Semantics using indexed comonads

- Generalize ordinary comonads
- With more structure for context passing

Flat coeffect system

Coeffect algebra $(S, \oplus, \sqcap, \sqcup, \varepsilon)$ formed by a monoid $(S, \oplus, \varepsilon)$, semi-lattice $(S, \sqcup)$ and a binary operation $\sqcap$.

$$\frac{x : \tau \in \Gamma}{C^{\varepsilon}\Gamma \vdash x : \tau} \qquad \frac{C^{r}\Gamma \vdash e_1 : C^{t}\tau_1 \to \tau_2 \qquad C^{s}\Gamma \vdash e_2 : \tau_1}{C^{r \sqcup (s \oplus t)}\Gamma \vdash e_1\, e_2 : \tau_2}$$

$$\frac{C^{r \sqcap s}(\Gamma, x : \tau_1) \vdash e : \tau_2}{C^{r}\Gamma \vdash \lambda x. e : C^{s}\tau_1 \to \tau_2} \qquad \frac{C^{r}\Gamma \vdash e : \tau}{C^{s}\Gamma \vdash e : \tau}\,(r \leq s)$$

Semantics using indexed comonads

Family $\boldsymbol{C}^{\boldsymbol{r}} A$ of mappings (data types) with

$$\varepsilon : \boldsymbol{C}^{\boldsymbol{\varepsilon}} A \rightarrow A$$
$$\circ : (\boldsymbol{C}^{\boldsymbol{r}} A \rightarrow B) \rightarrow (\boldsymbol{C}^{\boldsymbol{s}} B \rightarrow C) \rightarrow (\boldsymbol{C}^{\boldsymbol{r} \oplus \boldsymbol{s}} A \rightarrow C)$$

and for context propagation

$$m_{r,s} : \boldsymbol{C}^{\boldsymbol{r}} A \times \boldsymbol{C}^{\boldsymbol{s}} B \rightarrow \boldsymbol{C}^{(\boldsymbol{r} \sqcap \boldsymbol{s})} (A \times B)$$
$$n_{r,s} : \boldsymbol{C}^{(\boldsymbol{r} \sqcup \boldsymbol{s})} (A \times B) \rightarrow \boldsymbol{C}^{\boldsymbol{r}} A \times \boldsymbol{C}^{\boldsymbol{s}} B$$

Structural coeffect system

Different operations for context passing

$$m_{r,s} : \boldsymbol{C}^{\boldsymbol{r}} A \times \boldsymbol{C}^{\boldsymbol{s}} B \to \boldsymbol{C}^{(\boldsymbol{r}\times\boldsymbol{s})}(\text{Node}(A, B))$$

$$n_{r,s} : \boldsymbol{C}^{(\boldsymbol{r}\times\boldsymbol{s})}(\text{Node}(A, B)) \to \boldsymbol{C}^{\boldsymbol{r}} A \times \boldsymbol{C}^{\boldsymbol{s}} B$$

$$\Delta_{r,s} : \boldsymbol{C}^{(\boldsymbol{r}\sqcup\boldsymbol{s})} A \to \boldsymbol{C}^{\boldsymbol{r}} A \times \boldsymbol{C}^{\boldsymbol{s}} A$$

Structural application & abstraction

$$\frac{\boldsymbol{C}^{\boldsymbol{r}}\Gamma_1 \vdash e_1 : \boldsymbol{C}^{\boldsymbol{t}}\tau_1 \to \tau_2 \quad \boldsymbol{C}^{\boldsymbol{s}}\Gamma_2 \vdash \mathrm{e}_2 : \tau_1}{\boldsymbol{C}^{\boldsymbol{r}\times(\boldsymbol{s}\oplus\boldsymbol{t})}(\Gamma_1, \Gamma_2) \vdash e_1\ e_2 : \tau_2}$$

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

Conclusions

Why coeffects matter

Important properties not connected before

- Liveness, security tainting, provenance
- Implicits or resource usage, cross-compilation
- Efficient data-flow

What coeffects are the right way

- Types for standard lambda calculus
- Indexed comonads for wide-spread use?

BACKUP SLIDES

Syntactic equational theory

Using substitution $(\lambda x. e_2)e_1 \to e_2[x \leftarrow e_1]$

Function "values" may still need context

So consider call-by-name evaluation

If $C^r\Gamma \vdash e : \tau$ and $e \to e'$ then $C^r\Gamma \vdash e : \tau$ holds

– If $\varepsilon = \top$ (for example, liveness)

– If $\sqcup = \sqcap = \oplus$ (for example, implicit params)

But dataflow does not use substitution