Coeffects: Programming languages for rich environments

Tomas Petricek **Dominic Orchard** supervisor: **Alan Mycroft**

University of Cambridge, {name.surname}@cl.cam.ac.uk

Motivation: Why context-tracking matters

- Applications today run in diverse environments, such as mobile phones or the cloud. Different environments provide different capabilities, data with meta-data and other resources.
- Applications access information and resources of the environment. Such context-dependent interactions are often more important than how the application affects or changes the environment.
- Tracking and verifying how computations affect the environment can be done in a unified way using monadic effect systems, but no such mechanism exists for tracking and verifying how computations access and rely on the context.

Example 1: Liveness analysis & optimization

Annotate variable context with *false* (0) if it is definitely not live; *true* (1) if it may be accessed. Unused context can be optimized away.

Context is modelled as dependent **Maybe** type: $C_1 A = A$ and $C_0 A = 1$.

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

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

Example 2: Distributed language with resources

Context carries additional *rebindable resources* that may be accessed. Annotation specifies a set of resources that are available.

Context is represented using a **product** type: $C_r A = A \times (r \to \mathrm{Res})$.

```
fun () →
  let evts = access EventsDatabase
  let date = access CurrentDate
  query evts "SELECT Count(*) WHERE Date > %1" date
```

Resource requirements of a function are split between the call site and the declaration site. Multiple typings are possible, depending on how the function is used.

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

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

Example 3: Efficient data-flow language

Context provides access to previous values of variables. The annotation specifies how many past values may be needed.

Context is represented as a **non-empty list**; the annotation specifies the length of the list: $C_n A = A \times (A_1 \times \ldots \times A_n)$

$$\frac{C^r\Gamma \vdash e_1 : C^t\tau_1 \to \tau_2 \qquad C^s\Gamma \vdash e_2 : \tau_1}{C^{\max(r, s+t)}\Gamma \vdash e_1\ e_2 : \tau_2}$$

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

Effect systems

$$\Gamma \vdash e : \tau \,\&\, \sigma$$

- Track or infer information about what the computation *does* to the environment
- Information σ, such as set of performed memory operations, attached to the result
- Propagate information forward to the overall result
- Modeled as morphisms $\alpha \to \mathcal{C}\beta$ where $\mathcal{C}$ is a monad

Coeffect systems

$$\Gamma @ \sigma \vdash e : \tau$$

- Track or infer information about what the computation *requires* from the environment
- Information σ, such as set of accessed resources, attached to the variable context
- Propagate information backward to the initial input
- Modeled as morphisms $\mathcal{D}\alpha \to \beta$ where $\mathcal{D}$ is a comonad

Unified system: Flat coeffect calculus

Captures the essence of context-dependence tracking. Our unified model identifies common properties of the three examples and has desirable theoretical properties (subject reduction and categorical model)

- Sequential composition given by a monoid ($\oplus, \bot$) or ($\oplus, \top$)
- Context is propagated ($\vee$) and split ($\wedge$) using two additional operators

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

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

$$\frac{x : \tau \in \Gamma}{C^\bot\Gamma \vdash x : \tau} \quad \text{or} \quad \frac{x : \tau \in \Gamma}{C^\top\Gamma \vdash x : \tau}$$

Generalized system: Structural coeffect calculus

We often need to capture fine-grained structure with context requirements corresponding to individual variables (liveness, data-flow, provenance).

- Compose annotations using a product ($\times$) that reflect variable structure
- Write system using structural rules that change annotation accordingly

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

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

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

UNIVERSITY OF CAMBRIDGE