Algebraic Effects and Handlers

Andrej Bauer
University of Ljubljana

July 20, 2018

1 Review

- Signature $\Sigma = \{\mathtt{op}_i : P_i \Rightarrow A_i\}_{i \in I}$ Note: $\Rightarrow$ does not mean function.
- Trees/Terms $\mathtt{Tree}_\Sigma(V)$:
  - $\mathtt{return}\ v$ with $v \in V$
  - $\mathtt{op}_i(p, \kappa)$ with $p \in P_i, \kappa : A_i \to \mathtt{Tree}_\Sigma(X)$
- Interpretation/Model M:
  - carrier $|M|$
  - $[\![\mathtt{op}_i]\!]_M : P_i \times |M|^{A_i} \to |M|$
- $\mathtt{Free}_T(V) = \mathtt{Tree}_{\Sigma_T}(V)/ \approx_T$ computations

2 Transformation of Computations

What does this mean:

$$|\mathtt{Free}_T(V)| \to |\mathtt{Free}_{T'}(V')|?$$

Homomorphism, but from where? From the domain. T-Homomorphism.

$$\mathtt{Free}_T(V) \to \underset{\texttt{T-Model}}{M} \text{ where } |M| = |\mathtt{Free}_{T'}(V')|$$

and for $\mathtt{op}_i : P_i \to A_i$ we need:

$$[\![\mathtt{op}_i]\!]_M : P_i \times |\mathtt{Free}_{T'}(V')|^{A_i} \to |\mathtt{Free}_{T'}(V')|$$

such that $\mathcal{E}_T$ are satisfied by M.

2.1 Definition: Handler

A handler H given by:

- the maps $[\![\mathtt{op}_i]\!]_M$ as above
- a map $r : V \to |\mathtt{Free}_{T'}(V')|$
  I.e.: $H([\mathtt{return}\ v]) = r(v)$
  and $H([\mathtt{op}_i(p,\kappa)]) = [\![\mathtt{op}_i]\!]_M(p, H \circ \kappa)$

2.1.1 Notation

handler$\{\mathtt{return}\ x \mapsto r(x), \mathtt{op}_i(x,\kappa) \mapsto [\![\mathtt{op}_i]\!]_M(x,\kappa) = C_i(x,\kappa)\}$

2.1.2 Notation

for $H(c)$ where $c \in |\mathtt{Free}_T(V)|$: with H handle c

2.1.3 Rewrite with New Notation (Code)

```
with H handle return v = r(v)
with H handle op(p,κ) = C_i(p,λx.with H handle κx)
```

2.2 Comodels

A T-comodel in a category $\mathbb{C}$ is a T-model in $\mathbb{C}^{\mathtt{op}}$
In $\mathbb{C} = \mathrm{Set}$ we get

- A T-cointerpretation W is
  - carrier set $|W|$
  - for each $\mathtt{op}_i : P_i \to A_i$ a cooperation
    $[\![\mathtt{op}_i]\!]^W : P_i \times |W| \to A_i \times |W|$
    Why?
    $P_i \times |M|^{A_i} \to |M|$ carry $|M|^{A_i} \to |M|^{P_i}$
    dualize $P_i \times |M| \to A_i \times |M|$
    extend to interpret trees $\implies$ T-comodel W
    a cointerpretation that validates the equations

2.2.1 Examples

print: $\mathtt{String} \to 1$
$[\![\mathtt{print}]\!]^W : \mathtt{String} \times |W| \to 1 \times |W|$
read: $1 \to \mathtt{String}$
$[\![\mathtt{read}]\!]^W : 1 \times |W| \to \mathtt{String} \times |W|$
rnd: $1 \to \mathtt{Bool}$

2.3 Model M and Comodel W

Tensor $M \otimes W = M \times W / \sim_T$ where
$([\![\mathtt{op}_i]\!]_M(p,\kappa), w) \sim_T (\kappa(a), w')$
$[\![\mathtt{op}_i]\!]^W(p,w) = (a, w')$

3 Combining Theories

Consider two theories T and T'

1. Coproduct $T \oplus T'$
   $\Sigma_{T \oplus T'} = \Sigma_T + \Sigma_{T'}$
   $\mathcal{E}_{T \oplus T'} = \mathcal{E}_T + \mathcal{E}_{T'}$

2. Tensor $T \otimes T'$
   $\Sigma_{T \otimes T'} = \Sigma_T + \Sigma_{T'}$
   $\mathcal{E}_{T \otimes T'} = \mathcal{E}_T + \mathcal{E}_{T'}+$ distributive laws

Distributive Law:

$$op(p.\lambda x.op'(p', \lambda y.\kappa(x,y))) = op'(p'.\lambda y.op(p, \lambda x.\kappa(x,y)))$$

where $\mathtt{op} \in \Sigma_T$ and $\mathtt{op}' \in \Sigma_{T'}$

4 Designing a Programming Language

1. Change math terminology to program terminology (familiar)

2. Reuse existing concepts

3. Add missing features (recursion, etc) and make pretty (reorganize)

4. Provide operational semantics

5. Provide typing rules

4.1 Terminology

- Free Σ v to computations
- v from generators to values
- sets of generators to value types
- free models to computation types

4.2 Syntax

$$
\begin{array}{lll}
\mathrm{v} & ::= & \mathrm{x} \mid \mathrm{false} \mid \mathrm{true} \mid \mathrm{h}\ \text{(handler)} \mid \lambda \mathrm{x.c}\ \text{(function)} \\
\mathrm{h} & ::= & \mathrm{handler}\ \{\mathbf{return}\ \mathrm{x} \mapsto \mathrm{c\_ret},\ \mathrm{op}_i(\mathrm{x,k})\ \mathrm{to}\ \mathrm{c\_i}\} \\
\mathrm{c} & ::= & \mathbf{return}\ \mathrm{v} \\
 & \mid & \mathbf{if}\ \mathrm{v}\ \mathbf{then}\ \mathrm{c1}\ \mathbf{else}\ \mathrm{c2} \\
 & \mid & \mathrm{v1}\ \mathrm{v2} \\
 & \mid & \mathrm{with}\ \mathrm{v}\ \mathrm{handle}\ \mathrm{c} \\
 & \mid & \mathbf{do}\ \mathrm{x} \leftarrow \mathrm{c1}\ \mathbf{in}\ \mathrm{c2} \\
 & \mid & \mathrm{op}(\mathrm{v},\ \lambda \mathrm{x.c})\ (\mathrm{op}\ \mathrm{v}) \\
 & \mid & \mathrm{fix}\ \mathrm{x.c}
\end{array}
$$

The rest should be online.