Bauer 1.1

1. algebraic theories (today)
2. Programming language (tomorrow) (w/o equationality)
3. reasoning
4. applications

Algebraic Theories

E a group $(G, u, \cdot, ()^{-1})$

$u \in G$

$\cdot : G \times G \to G$

$()^{-1} : G \to G$

for $x \in G$

$u \cdot x = x = x \cdot u$

$(x_1 \cdot x_2) \cdot x_3 = x_1 \cdot (x_2 \cdot x_3)$

$x \cdot x^{-1} = u = x^{-1} \cdot x$

form := operations and equations

Alternatively $(G, u, \cdot)$ monoid + $\forall x \exists y \; x \cdot y = u = y \cdot x$

eqn — axiom, not eqn

a group is a monoid w an axiom

if this lemma is satisfied — L if $xy = u = yx$ and $xz = u = zx$ then $y = z$

- having just equations, no axioms, is better than having axioms.

D A signature $\Sigma = \{(op_i, n_i)\}_{i \in I}$ where op_i are operation symbols and $n_i \in \mathbb{N}$ is the arity of op_i

D a term in context $x_1, \dots, x_k$ is either
- one of the variables x_i or
- $op_i(t_1, \dots, t_n)$ where $t_1, \dots, t_n$ are terms in context $x_1, \dots, x_k$

read as an inductive defn.

D An algebraic (equational) theory T is $(\Sigma_T, \mathcal{E}_T)$ where Σ_T is a signature and $\mathcal{E}_T$ is a set of equations.

an eqn is $x_1, \dots, x_k \mid l = r$ where l, r are Σ_T terms in $x_1, \dots, x_k$

E group $\Sigma_{group} = \{(u, 0), (m, 2), (i, 1)\}$

$x \mid m(u(), x) = x$

$x \mid m(x, i(x)) = u()$

$\vdots$

Es
- ring

E
- empty theory

E
- theory of pointed set
  - signature $= \Sigma_{ps} = \{(\cdot, 0)\}$
  - no equations

E
- semilattice

$\Sigma_{semilattice} = \{(\bot, 0), (\vee, 2)\}$

$\bot \vee x = x$

$x \vee (y \vee z) = (x \vee y) \vee z$

$x \vee x = x$

Bauer 1.2

E field $+, -, \times, ()^{+}, 0, 1$

problem : 0^{+} is undefined.

② Interpretation and models

An interpretation I of T is given by

- a carrier set $|I|$
- for each $(op_i, n_i) \in \Sigma_T$ a map
$$[\![op_i]\!]_I : \underbrace{|I| \times \cdots \times |I|}_{n_i} \longrightarrow |I|$$
- Each term $x_1, \ldots, x_k \mid t$ is interpreted as a map $[\![x_1, \ldots, x_k \mid t]\!] : |I|^k \longrightarrow |I|$

it follows $[\![x_1, \ldots, x_k \mid x_i]\!] : |I|^k \to |I|^k$ is π_i the i^{th} projection.

- $[\![x_1, \ldots, x_k \mid op_i(t_1, \ldots, t_{n_i})]\!]_I$ is composition
$$|I|^k \xrightarrow{([\![t_1]\!]_I, \ldots, [\![t_{n_i}]\!]_I)} |I|^{n_i} \xrightarrow{[\![op_i]\!]_I} |I|$$

D A T-model is an interpretation M of theory T s.t. for every $x_1, \ldots, x_k \mid l = r$ in $\mathcal{E}_T$

the maps
$$[\![x_1, \ldots, x_k \mid l]\!]_M \quad |M|^k \to |M|$$
$$[\![x_1, \ldots, x_k \mid r]\!]_M \quad |M|^k \to |M|$$
are equal

E A model M of the theory of a pointed set

- a carrier set $|M|$
- a map $[\![\cdot]\!]_M \quad |M|^0 \longrightarrow |M|$
$$1 \longrightarrow |M|$$

Isomorphically : (S, s) where S is a set and $s \in S$

E every theory T has the trivial model M : $|M| = 1$

$[\![op_i]\!]_M : 1^{n_i} \longrightarrow 1$

Bauer 1,3

$\underline{\underline{E}}$ if M and L are T-models

$$|M \times L| = |M| \times |L|$$

where
$$[\![op_i]\!]_{M\times L} : (|M| \times |L|)^{n_i} \longrightarrow |M| \times |L|$$

$$[\![op_i]\!]_{M\times L}(a_1, \dots, a_{n_i}) = \left([\![op_i]\!]_M(\pi_1 a_1, \dots, \pi_n a_{n_i}),\ [\![op_i]\!]_L(\pi_2 a_1, \dots, \pi_n a_{n_i})\right) \in |M \times L|$$

③ Free Models

Given theory T and set X

Say that a T-model M together w a map $\eta : X \to |M|$ is freely generated by X when the following holds.

$$\begin{tikzcd} X \arrow[r, "\eta"] \arrow[dr, "f"'] & {|M|} \arrow[d, "\bar{f}"] \\ & {|L|} \end{tikzcd}$$

$\forall f\ \exists! \bar{f}$ s.t. this diagram,

where $\bar{f}$ is a T-homomorphism and unique.

$\underline{\underline{D}}$ a T-homomorphism $f : M \to L$

for every op_i in T

$$f\left([\![op_i]\!]_M(a_1, \dots, a_{n_i})\right) = [\![op_i]\!]_L(f(a_1), \dots, f(a_{n_i}))$$

$\underline{\underline{V}}$ $(M, \eta : X \to |M|)$ is called $\begin{cases} \text{free model over } X \\ \text{free model generated by } X \end{cases}$

$\underline{\underline{E}}$ Dfn. $P_{<\omega}(X) = \{ S \subseteq X \mid S \text{ finite} \}$

claim: $(P_{<\omega}(X), \emptyset, \cup)$ is the free semilattice generated by X

w $\eta : X \to P_{<\omega}(X)$

$\eta : x \longmapsto \{x\}$

check
$$\begin{tikzcd} X \arrow[r, "\{-\}"] \arrow[dr, bend right] & P_{<\omega}(X) \arrow[d, "\bar{f}"] \\ & L \end{tikzcd}$$

w $\bot, \vee \in L$

$\bar{f}(\emptyset) = \bot$

$\bar{f}(S_1 \cup S_2) = \bar{f}(S_1) \vee \bar{f}(S_2)$

$\bar{f}(\{x\}) = f(x)$

note! $\bar{f}$ satisfies, but how do you know if it's unique

Scanned by CamScanner

Bauer 1.4

Free model $Free_T(X)$

non-infinite

① $Tree_T(X)$ set of well-founded trees

defined inductively

- for every $x \in X$ there is tree return $x \in Tree_T(X)$
- if $(op_i, n_i) \in \Sigma_T$ and $t_{n,1}, \dots, t_{n_i} \in Tree_T(X)$ then

then we construct a tree op_i is root and n_i subtrees.

② Define $\approx_T$ on $Tree_T(X)$ to be the least equivalence relation s.t.

- it's a congruence w respect to forming trees, and
- it validates the equations of the theory.

HW

Bauer 2.1

theory so far

- Signature $\Sigma = \left\{(op_i, n_i)\right\}_{i \in I}$
- term $x_1, \ldots, x_n \mid t$
- equation $x_1, \ldots, x_n \mid l = r \in \mathcal{E}_T$
- theory T $T = (\Sigma_T, \mathcal{E}_T)$
- Model M
- Free Model $\mathrm{Free}_T(X)$

$[n] := \{0, 1, \ldots, n-1\}$

NB $\underbrace{|I| \times \cdots \times |I|}_{n} = |I|^k$

B^A is set of functions $A \to B$

$X \times \cdots \times X \cong X^{[n]}$

④ General arities and parameters

operation symbol op_i n_i

$$[\![op_i]\!]_I : \underbrace{|I| \times \cdots \times |I|}_{n_i} \longrightarrow |I|$$

an arity can be any set whatsoever

Signature

$$\Sigma = \left\{ (op_k, A_k) \right\}_{k \in I}$$

symbols sets

$$[\![op_i]\!] : |I|^{A_i} \longrightarrow |I|$$

E.g. Vector Space V

$v + w$ :: $+ : 1 \times V \times V \to V$ unit parameter, if there are no others

0

$-v$

sv where s is scalar.

or $\cdot : \mathbb{R} \times V \to V$

this is the idea of parameter

we will replace arglists w K which represents functions.

$op_i\cancel{(\ldots t_a \ldots)}_{a \in A_i} \mapsto op_i(K)$

or $op_i(\lambda a. t_a)$

Parameters $[\![op]\!]_I : P \times |I|^{A} \longrightarrow |I|$

parameter

signature $\Sigma := \left\{ op_i : P_i \rightsquigarrow A_i \right\}_{i \in I}$

symbol set parameters set arity

(instead of terms) trees $\mathrm{Tree}_\Sigma(X)$: $(\mathrm{return}\ x) \in \mathrm{Tree}_\Sigma(X)$ for $x \in X$

$op_i(p, K) \in \mathrm{Tree}_\Sigma(X)$ for $p \in P_i$, $K : A_i \to \mathrm{Tree}_\Sigma(X)$

Bauer 2.2

Signature $\Sigma = \left\{ \underset{\underset{\text{symbols}}{\uparrow}}{op_i} : \underset{\underset{\substack{\text{set}\\\text{parameters}}}{\uparrow}}{P} \rightsquigarrow \underset{\underset{\substack{\text{set}\\\text{arities}}}{\uparrow}}{A_i} \right\}_{i \in I}$

(instead of terms) Trees $\quad \mathrm{Tree}_\Sigma(x) = \begin{cases} \text{return } x \in \mathrm{Tree}_\Sigma(x) & \text{for } x \in X \\ op_i(p, K) \in \mathrm{Tree}_\Sigma(x) & \text{for } p \in P_i,\ K : A_i \to \mathrm{Tree}_\Sigma(x) \end{cases}$

Vars $\quad X = x_1, x_2, \ldots, x_n \ (= \vec{x})$

Equations $\mathcal{E}_\Sigma \quad \vec{X} \mid l = r$ where $l, r \in \mathrm{Tree}_\Sigma(x)$

Theory $T \quad T = (\Sigma_T, \mathcal{E}_T)$

Interpretation $I = \begin{cases} \text{carrier set } |I| \\ \text{for each } op_i : P \rightsquigarrow A_i \text{, give } [\![op_i]\!]_I : P_i \times |I|^{A_i} \to |I| \end{cases}$

extend to interpretation of trees/terms

For $t \in \mathrm{Tree}_\Sigma(X)$

$$[\![t]\!]_I : |I|^X \to |I|$$

$$[\![\text{return } x]\!]_I(\gamma) = \gamma(x) \quad \text{for } x \in X$$

$$[\![op_i(p, K)]\!]_I(\gamma) = [\![op_i]\!]_I \left(p, \lambda a \in A_i, [\![K(a)]\!]_I(\gamma)\right)$$

Model as before.

$$\mathrm{Free}_T(X) = \mathrm{Tree}_\Sigma(x) / \approx_T$$

where $\approx_T$ is the least congruence enforcing equations $\mathcal{E}_T$

Bauer 2.3

5 Computational effects as algebraic operations

effects are stuff not in λ-c

print,
read,
choice,
continuations,
etc.

Computations:

pure   return v ← doesn't matter whether you compute it now, later, once, twice

effectful $op(p, K)$

parameter

the rest of computation awaiting the result of op.

R: nontermination is a computational effect.
very interesting

E: $l{+}{+}$ i.e. $l = l+1$

pointer (lookup)

incr, store result.

$$l{+}{+} := \text{lookup}\left(l,\ \lambda x.\ \text{update}\left(\overbrace{(l, x+1)}^{\text{parameter}},\ \lambda\_.\ \text{return } x\right)\right)$$

← there is no runtime env, or free vars

$$\text{print} := \text{print}\left(\text{"Hello World!"},\ \lambda\_.\ \text{return}()\right)$$

$$\Sigma_{PL} = \left\{ \text{lookup} : \text{Locations} \rightsquigarrow \text{States},\ \text{update} : \text{Locations} \times \text{States} \rightsquigarrow 1,\ \text{print} : \text{String} \rightsquigarrow 1 \right\}$$

Bauer 2.4

State holding elements of a set S

R explicit continuation-passing style

T is theory of effects

return values

R $\text{Free}_T(V)$ satisfies these equations and no others

Signature

$put : S \rightsquigarrow 1$

$get : 1 \rightsquigarrow S$

$$get((), \lambda x. get((), \lambda y. K(x,y))) = get((), \lambda z. K(z,z))$$

$$get((), \lambda x. put(x, K(x))) = K()$$

$$put(s, get((), K)) = put(s, \lambda\_. K(s))$$

$$put(s, \lambda\_. put(t, K)) = put(t, K)$$

the equations governing states

R the free model $\text{Free}_T(V)$ is the set of computations w effects described by theory T and returning values from set V

(continuations)

exceptions

$abort : 1 \rightsquigarrow \emptyset$

$[\![abort]\!]_M : 1 \times |M|^{\emptyset} \rightsquigarrow |M|$

this means non-resumable.

R handling is not algebraic!

Java exception handlers aren't here.

but we have a generalized handler

$\text{Free}_{state}(int) \cong State \to State \times Int$

like the state monad (Haskell)

there is a Monad structure on the free Models

$\text{Free}_{state}(V) = \text{Tree}_{state}(V) / \sim_{state}$

Bauer 2.5

6 General operation and sequencing

- generic operation $\overline{op}(p) = op(p, \lambda x.\, \text{return } x)$

- sequencing $\text{do } x \leftarrow t \text{ in } t_2(x)$ (Haskell)

  $\text{let } x = t_1 \text{ in } t_2(x)$ (ML)

  ~~$x = t_1 ; t_2$~~ (C) not really

  ~~put(...~~

D

$\text{do } x \leftarrow \text{return } v \text{ in } t_2(x) = t_2(v)$

$\text{do } x \leftarrow op(p, k) \text{ in } t_2(x) = op(p, \lambda y.\, \text{do } x \leftarrow k(y) \text{ in } t_2(x))$

produce a y and feed it into k

$$op(p, \lambda x.\, k(x)) = \text{do } x \leftarrow \overline{op}(p) \text{ in } k(x).$$

$\overline{op} : P \to A$

$op : P \rightsquigarrow A$

Bauer 3.1

review

Signature $\Sigma = \{op_k : P_k \rightsquigarrow A_k\}_{k \in I}$

Trees/terms $Tree_\Sigma(V) = \begin{cases} return\ v \quad (pure) \quad v \in V \\ op_i(p, K) \quad p \in P_i \quad K : A_i \to Tree_\Sigma(x) \quad (effect) \end{cases}$

Interpretation/model M carrier $|M|$

$[\![op_i]\!]_M : P_i \times |M|^{A_i} \longrightarrow |M|$

$Free_T(V) = Tree_{\Sigma_T}(V) / \approx_T$ (computations)

Transformation of Computation

$|Free_T(V)| \longrightarrow |Free_{T'}(V')|$ is a T-homomorphism if we turn cod into M

where $|M| = |Free_{T'}(V')|$ and for $op_k : P_k \longrightarrow A_k$ we need

$[\![op_k]\!]_M : P_k \times |Free_{T'}(V')|^{A_k} \longrightarrow |Free_{T'}(V')|$ s.t. $\mathcal{E}_T$ are satisfied by M

handler H

So the T-homomorphism we want is $H : |Free_T(V)| \longrightarrow M$

given by the maps $[\![op_i]\!]_M$ as above and a map $\gamma : V \longrightarrow |Free_{T'}(V')|$

H is a handler

e.g. $H([return\ v]) = \gamma(v)$, $H([op_i(p,K)]) = \underset{c_i}{\underset{\shortparallel}{[\![op_i]\!]_M}}(p, H \circ K)$

$\underline{\underline{N}}$ $A \xrightarrow[K]{} |Free_T(V)|$ handler $\{ return\ x \mapsto r(x),\ op_i(x,K) \mapsto c_i(x,K) \}$

$\underline{\underline{N}}$ for $H(c)$ where $c \in |Free_T(V)|$ with H handle c

with H handle return $v = r(v)$

with H handle $op(p,K) = c_i(p_i, \lambda x.$ with H handle $Kx)$

Bauer 3.2

"top-level handler" is not a thing, the word is <u>CoModel</u>

A T-model in a category $\mathbb{C}$ is a T-model in $\mathbb{C}^{op}$

in $\mathbb{C} = Set$ we get

a T-cointerpretation W is:

- a carrier set $|W|$
- for each $op_i : P_i \longrightarrow A_i$ a cooperation
  $[\![op_i]\!]^W : P_i \times |W| \longrightarrow A_i \times |W|$

R exponentials are dual w products

- extend to interpret trees
  T- comodel W as cointerpretation that validates the equations

E

$print : String \rightsquigarrow 1$ $\quad$ $read : 1 \rightsquigarrow String$

$[\![print]\!]^W : String \times |W| \longrightarrow 1 \times |W|$ $\quad$ $[\![read]\!]^W : 1 \times |W| \longrightarrow String \times |W|$

$rnd : 1 \rightsquigarrow bool$

for Model M and comodel W

D Tensor $M \otimes W = M \times W /_{\sim_T}$ where $\left([\![op_i]\!]_M(p, K), w\right) \sim_T \left(K(a), w'\right)$

$[\![op_i]\!]^W(p, w) = (a, w')$

Model is like software
Comodel/World is like hardware

<u>Combining theories</u> T and T'

D ① Coproduct $T \oplus T'$

$$\Sigma_{T \oplus T'} = \Sigma_T + \Sigma_{T'}$$

$$\mathcal{E}_{T \oplus T'} = \mathcal{E}_T + \mathcal{E}_{T'}$$

Distributivity:

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

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

② tensor $T \otimes T'$ — totally not the prior tensor.

$$\Sigma_{T \otimes T'} = \Sigma_T + \Sigma_{T'}$$

$$\mathcal{E}_{T \otimes T'} = \mathcal{E}_T + \mathcal{E}_{T'} \text{ w distributivity}$$