

In an applicative structure (C, \cdot) , every element $c \in C$ gives rise to a function

$$\begin{aligned} \chi_c : C &\rightarrow C \\ d &\mapsto c \cdot d \end{aligned}$$

Definition 9: A map $f: C \rightarrow C$ is **representable** if $f = \chi_c$ for some $c \in C$.

We denote the set of representable functions on C by $\text{Rep}(C, \cdot) \subseteq \text{Fun}(C, C)$.

Moreover, two elements $c, d \in C$ are **extensionally equivalent** ($c \sim d$) if

$$\forall e \in C: c \cdot e = d \cdot e.$$

The applicative structure (C, \cdot) is called **extensional** if the quotient projection $\pi: C \rightarrow C/\sim$ is a bijection.

By construction, we have

$$\begin{array}{ccc} C & \xrightarrow{\chi} & \text{Rep}(C, \cdot) \subseteq \text{Fun}(C, C) \\ \pi \downarrow & \cong \nearrow & \\ C/\sim & & \end{array}$$

Proposition 10: Let $M = (C, \cdot, \mathbb{I}, \mathbb{D})$ be a λ -model. Then

$$M \models \lambda\beta\eta \quad \text{iff} \quad (C, \cdot) \text{ is extensional.}$$

(independent of \mathbb{I}, \mathbb{D} !)

Proof: We already know that $M \models \lambda\beta$ from Corollary 8. We are thus to show that

$$(\forall t \in \mathcal{A}^C, x \in \text{Var} \cup \text{FV}(t): M \models \lambda x. \text{App}(t, x) \equiv t) \quad \text{iff} \quad (C, \cdot) \text{ is extensional.}$$

" \Leftarrow " Let $\rho \in \text{Val}(C)$, $c \in C$, $t \in \mathcal{T}$, $x \in \text{Var} - \text{FV}(t)$. Then

$$\begin{aligned} \llbracket \lambda x. \text{App}(t, x) \rrbracket_{\rho}^c &= \llbracket \text{App}(t, x) \rrbracket_{\rho[C/x]} \quad \text{by Def. 1.4} \\ &= \llbracket t \rrbracket_{\rho[C/x]} \cdot \llbracket x \rrbracket_{\rho[C/x]} \quad \text{by Def. 1.3} \\ &= \llbracket t \rrbracket_{\rho} \cdot c, \quad \text{because } x \notin \text{FV}(t). \end{aligned}$$

Thus, $\llbracket \lambda x. \text{App}(t, x) \rrbracket_{\rho} \sim \llbracket t \rrbracket_{\rho}$ f.o. $\rho \in \text{Val}(C)$.

\Rightarrow $\llbracket \lambda x. \text{App}(t, x) \rrbracket_{\rho} = \llbracket t \rrbracket_{\rho}$ f.o. $\rho \in \text{Val}(C)$ by assumption, which was to show.

" \Rightarrow " Let $c, d \in C$ s.t. $c \neq d$. Let $\rho: \text{Var} \rightarrow C$ with $\rho(x) = c$, $\rho(y) = d$.

Let $z \in \text{Var} - \{x, y\}$. Then f.o. $e \in C$:

$$\begin{aligned} \llbracket \text{App}(x, z) \rrbracket_{\rho[C/e]} &= \llbracket x \rrbracket_{\rho[C/e]} \cdot \llbracket z \rrbracket_{\rho[C/e]} \\ &= c \cdot e \\ &= d \cdot e = \dots = \llbracket \text{App}(y, z) \rrbracket_{\rho[C/e]}. \end{aligned}$$

By weak extensionality of \mathcal{M} (Def. 1.5), we see that

$$\llbracket \lambda z. \text{App}(x, z) \rrbracket_{\rho} = \llbracket \lambda z. \text{App}(y, z) \rrbracket_{\rho} \quad \text{in } C.$$

But, by assumption, LHS = $\llbracket x \rrbracket_{\rho} = c$ & RHS = $\llbracket y \rrbracket_{\rho} = d$. So $c = d$.

□

Thus, the property of $(C, \llbracket \cdot \rrbracket)$ to model $\beta\eta$ is a property of the appl. structure (C, \cdot) !

Question: Can we find a characterization of λ -models which is entirely syntax-free? **Yes!**

1st step: Given an opp. structure (C, \cdot) together with a function

$f: C \rightarrow C$, and a valuation $\rho: \text{Var} \rightarrow C$, to define an interpretation

$\llbracket \cdot \rrbracket_\rho: \lambda^C \rightarrow C$ as in Def. 1 we only need to manage λ -abstractions:

1. $\llbracket x \rrbracket_\rho := \rho(x)$ for $x \in \text{Var}$, $\llbracket c \rrbracket_\rho = f(c)$ for $c \in C$,

2. $\llbracket \text{App}(s, t) \rrbracket_\rho := \llbracket s \rrbracket_\rho \cdot \llbracket t \rrbracket_\rho$

3. $\llbracket \lambda x. t \rrbracket_\rho := ?$



Want $a \in C$ s.t. $\forall c \in C: \llbracket \lambda x. c \rrbracket_\rho = a \cdot c$

where each $\llbracket c \rrbracket_\rho$ has been defined in a prior recursive step.

Basically a continuity condition: To define $\llbracket \lambda x. t \rrbracket_\rho$ for each $\rho \in \text{Val}(C)$ individually is not enough. We need to represent the function

$$\llbracket \lambda x. t \rrbracket_\rho: C \rightarrow C,$$

In fact, more generally, if $\text{FV}(t) = \{x_1, \dots, x_n\}$ and $\llbracket t \rrbracket_\rho \in C$ is defined

f.a. $\rho \in \text{Val}(C)$, we need $a \in C$ s.t.

$$\forall a_1, \dots, a_n \in C: \llbracket t \rrbracket_{\rho[x_1 \mapsto a_1, \dots, x_n \mapsto a_n]} = (a \cdot ((a_1) \cdot \dots \cdot a_n)) \cdot c_n$$

to model n -fold λ -abstraction.

E.g. $\llbracket \lambda x. \lambda y. (x y) \rrbracket = C \times C \rightarrow C$

$\forall c \in C \exists a \in C: \llbracket \lambda x. \lambda y. (x y) \rrbracket = \lambda a c \stackrel{!}{=} \lambda_{x(a)c}$

In fact, it suffices to assume the following.

Definition 11: Given a set $S \subseteq \Lambda^e$ of λ -terms, the **combinatory closure** \bar{S} of S is the smallest set T s.t.

1. $S \subseteq T$
2. $s, t \in T \Rightarrow \text{App}(s, t) \in T$.

Let (C, \cdot) be an applicative structure. For $\rho \in \text{Val}(C)$ and some $f: C \rightarrow C$, let $\llbracket \cdot \rrbracket_\rho: \overline{\text{Var} \cup C} \rightarrow C$ be given by

1. $\llbracket x \rrbracket_\rho = \rho(x)$ f.o. $x \in \text{Var}$, $\llbracket c \rrbracket_\rho = f(c)$ f.o. $c \in C$.

2. $\llbracket \text{App}(s, t) \rrbracket_\rho = \llbracket s \rrbracket_\rho \cdot \llbracket t \rrbracket_\rho$ f.o. $s, t \in \overline{\text{Var} \cup C}$.

Say (C, \cdot) is **combinatorially complete** wrt. $f: C \rightarrow C$ if f.o. finite

sets $X = \{x_1, \dots, x_n\}$ of variables, and all $t \in \overline{X \cup C}$,

$\exists a_t \in C \forall \rho \in \text{Val}(C): \llbracket t \rrbracket_\rho = a_t \cdot \rho(x_1) \cdot \dots \cdot \rho(x_n)$

left succedent notation,
 $(\lambda x_1. (\lambda x_2. (\lambda x_n. \dots))) \cdot \rho(x_n)$

Definition 12: A **syntax-free λ -model** is a tuple (C, \cdot, f) where (C, \cdot) is a combinatorially complete applicative structure wrt. $f: C \rightarrow C$, and $\sigma: C/\lambda \rightarrow C$ is a section of the projection $\pi: C/\lambda \rightarrow C$ such that $\sigma \circ \pi: C \rightarrow C$ is representable.

"2
Rep(C, C)

Theorem 13: I. Every λ^C -model $(C, \cdot, \sigma, \mathbb{D})$ gives rise to a syntax-free λ -model (C, \cdot, σ, f) with $\sigma: C/\lambda \rightarrow C$, $[c] \mapsto \underbrace{[\lambda y. \text{App}(x, y)]}_{\sigma \circ \mathbb{D} \circ c}$ for any $c \in \text{Val}(C)$, and $f = \sigma \circ \mathbb{D} \circ c$.

II. Every syntax-free λ -model (C, \cdot, σ, f) gives rise to a λ^C -model $(C, \cdot, \sigma, \mathbb{D})$ s.t.h. for $c \in \text{Val}(C)$,

$$a) \sigma \circ \mathbb{D} \circ c = \sigma(c) \quad \text{f.u. } x \in \text{Var}, \quad \sigma \circ \mathbb{D} \circ c = f(c) \quad \text{f.u. } c \in C$$

$$b) \sigma \circ \mathbb{D} \circ [\text{App}(s, t)] = \sigma \circ \mathbb{D} \circ s \cdot \sigma \circ \mathbb{D} \circ t$$

$$c) \sigma \circ \mathbb{D} \circ [\lambda x. t] = \sigma \left(\underbrace{\sigma \circ \mathbb{D} \circ [c \cdot x]}_{\in \text{Rep}(C, C)} \right)$$

III. The two assignments are mutually inverse.

Proof:

Part I: If (C, \cdot) supports a λ^C -model $(C, \cdot, \sigma, \mathbb{D})$, then (C, \cdot) is combinatorially complete wrt. $\sigma \circ \mathbb{D} \circ c$, as for $X = \{x_1, \dots, x_n\} \in \text{Var}$, $t \in \overline{X \cup C}$, we can set $a t := \sigma \circ \mathbb{D} \circ [\lambda x_1, \dots, \lambda x_n. t]$ for any $c \in \text{Val}(C)$.

The function $\sigma: C/\sim \rightarrow C$ is well-defined, since $c \sim c'$ implies f.a. $d \in C$,

$$[\lambda y. \text{App}(x_1 y)]_{\rho(c/x)} \circ d = [\text{App}(x_1 y)]_{\rho(c/x)} \circ d$$

$$= c \circ d$$

$$= c' \circ d = [\lambda y. \text{App}(x_1 y)]_{\rho(c'/x)} \circ d,$$

i.e. $\sigma(c) = [\lambda y. \text{App}(x_1 y)]_{\rho(c/x)} \sim [\lambda y. \text{App}(x_1 y)]_{\rho(c'/x)} = \sigma(c')$.

$\Rightarrow \sigma(c) = \sigma(c')$ by Ex. II.3 (Berry's Ext. Prop.).

To show that $\sigma: C/\sim \rightarrow C$ is a section of the projection π , we are to show that

$$\forall c \in C \quad c \sim [\lambda y. \text{App}(x_1 y)]_{\rho(c/x)}, \quad (*)$$

But in the proof of Proposition 10 we computed that

$$[\lambda y. \text{App}(x_1 y)]_{\rho(c/x)} \circ d = c \circ d \quad \text{f.a. } c, d \in C,$$

which proves (*).

Lastly, for $c \in C$, we have

$$\sigma \circ \pi(c) = [\lambda y. \text{App}(x_1 y)]_{\rho(c/x)} = \underbrace{[\lambda x. \lambda y. \text{App}(x, y)]_{\rho}}_{[\lambda _]} \circ c,$$

i.e. $\sigma \circ \pi: C \rightarrow C$ is represented by $[\lambda _]$.

Part II: First, we need to show that $\sigma \circ \mathcal{D}_\rho$ is defined on all λ -terms.

We construct $\sigma \circ \mathcal{D}_\rho$ recursively and show that f.e. $t \in \mathcal{N}^C$,

i) $\llbracket t \rrbracket_{\rho} [c/x] = \llbracket t \rrbracket_{\rho} [c'/x]$ f.a. $\rho \in \text{Val}(C)$, $c, c' \in C$ whenever $x \notin \text{FV}(t)$,

ii) $\forall \langle x_1, \dots, x_n \rangle \ni \text{FV}(t) \exists a_t \in C: \forall a_1, \dots, a_n \in C: a_t \cdot c_1 \dots c_n = \llbracket t \rrbracket_{\rho} [a_1/x_1] \dots [a_n/x_n]$.

Case 0: $c \in C$, then $\llbracket c \rrbracket_{\rho} = f(c)$. i) is clear, ii) holds by comp. compl'ness wrt. f .

Case 1: $t \in \overline{\text{Var}}$, then $\llbracket t \rrbracket_{\rho} \in C$ is defined as in Definition I.9 n.th. i) holds.

Condition ii) holds again by combinatorial completeness. ==

Case 2: Let $s, t \in \mathcal{L}^C$ n.th. $\llbracket s \rrbracket_{\rho}, \llbracket t \rrbracket_{\rho} \in C$ are defined f.a. $\rho \in \text{Val}(C)$. Then

$\llbracket \text{App}(s, t) \rrbracket_{\rho} = \llbracket s \rrbracket_{\rho} \cdot \llbracket t \rrbracket_{\rho} \in C$ again satisfies i). For ii), let $\text{FV}(s) \cup \text{FV}(t) = \{x_i, 1 \leq i \leq n\}$

as $a_i \in C$ n.th. $\chi_{as} = \llbracket s \rrbracket_{\rho} [a_i/x_i]$, $\chi_{at} = \llbracket t \rrbracket_{\rho} [a_i/x_i] \in C$,

which exist by the ind. hyp. Then for $u, v \neq x_i$ for $i \leq n$, let $a_{\text{App}} \in C$ n.th.

f.a. $\rho \in \text{Val}(C)$,

$$a_{\text{App}} \cdot c_1 \dots c_n = \llbracket \text{App}(\underbrace{(_ (\text{App}(u, x_1), x_2), _)}_{\in \overline{\text{Var}}}, x_n), (_ (\text{App}(v, x_1), x_2), _)_{x_n}) \rrbracket_{\rho} .$$

Let $a_{\text{App}(s, t)} = a_{\text{App}} \cdot a_s \cdot a_t$. Then for $c_1, \dots, c_n \in C$, and any $\rho \in \text{Val}(C)$, we get

$$\begin{aligned} a_{\text{App}(s, t)} \cdot c_1 \dots c_n &= a_{\text{App}} \cdot a_s \cdot a_t \cdot c_1 \dots c_n \\ &= \llbracket \text{App}(\text{App}(u, \vec{x}), \text{App}(v, \vec{x})) \rrbracket_{\rho} [a_s/u] [a_t/v] [c_i/x_i] \\ &= \llbracket \text{App}(u, \vec{x}) \rrbracket_{\rho} [c_i/x_i] \cdot \llbracket \text{App}(v, \vec{x}) \rrbracket_{\rho} [c_i/x_i] \\ &= (a_s \cdot c_1 \dots c_n) \cdot (a_t \cdot c_1 \dots c_n) \\ &= \llbracket s \rrbracket_{\rho} [c_i/x_i] \cdot \llbracket t \rrbracket_{\rho} [c_i/x_i] = \llbracket \text{App}(s, t) \rrbracket_{\rho} [c_i/x_i]. \end{aligned}$$

Case 3: Suppose $(\tau \Downarrow \rho$ has been defined f.o. $\rho \in \text{Val}(C)$.

Let $\text{FV}(t) = \{x_1, \dots, x_n\}$. By the ind. hyp. we have $a_i \in C$ s.th.

f.o. $\rho \in \text{Val}(C)$, $a_i \cdot \rho(x_i) \sim \rho(x_i) \cdot \rho(s) = (\tau \Downarrow \rho$.

For $\rho \in \text{Val}(C)$, $x \in \text{Var}$, let $\tau \Downarrow \rho \in C \rightarrow C$

$$(\tau \Downarrow \rho) x = \sigma(\lambda a_i \cdot \rho(x_i) \sim \rho(x_n))$$

To verify i), for any $\rho \in \text{Val}(C)$, $\tau \notin \text{FV}(t)$, we have

$$\begin{aligned} (\tau \Downarrow \rho) [d/z] &= \sigma(\lambda a_i \cdot \rho([d/z](x_i)) \sim \rho([d/z](x_n))) \\ &= \sigma(\tau \Downarrow \rho [d/z] C \rightarrow C) \\ &= \tau \Downarrow \rho \text{ whenever } \tau \notin \text{FV}(t) \text{ by the ind. hyp.} \\ &= \sigma(\tau \Downarrow \rho C \rightarrow C) = (\tau \Downarrow \rho) \end{aligned}$$

For ii), let $\ast \Pi \in C$ be s.th. $\chi_\Pi = \sigma \circ \Pi : C \rightarrow C$

$$\ast a_1 = (\text{App}(\text{App}(\text{App}(\sigma, x_1), x_2), \dots), x_n))$$

$$\ast a_{1x.t} = a_1 \Pi \text{ at } \in \overline{\text{Var}} \text{ via combinatory compl.}$$

Then f.o. $c_1, \dots, c_n \in C$, $\rho \in \text{Val}(C)$,

$$a_{1x.t} \cdot a_1 \sim a_1 = a_1 \Pi \text{ at } a_1 \sim a_1$$

$$= \Pi(a_1 \cdot a_1 \sim a_1)$$

$$= \sigma(\lambda a_i \cdot a_i \sim a_i) = (\tau \Downarrow \rho) [c/z]$$

By recursion, for every $\rho \in \text{Val}(C)$ we obtain a map $\sigma \cdot \mathcal{D}_\rho: \Lambda^C \rightarrow C$, s.t.

1. $\sigma \cdot \mathcal{D}_\rho = \rho(x)$ f.a. $x \in \text{Var}$ by construction.

2. $\sigma \cdot \mathcal{D}_\rho = \sigma \cdot \mathcal{D}_\sigma$ f.a. $\sigma \in \text{Val}(C)$ if $\rho(\text{FV}(t)) = \sigma(\text{FV}(t))$ via i).

3. $\sigma \cdot \mathcal{D}_\rho \cdot c = \sigma \cdot \mathcal{D}_\rho(c(x))$ f.a. $t \in \Lambda^C, c \in C$ by case 3.

(Berg's Ext. Prop.) If $\sigma \cdot \mathcal{D}_\rho \sim \sigma \cdot \mathcal{D}_\sigma$, then

$$\sigma(\text{as}(x_1) - e(x_2)) \sim \sigma(\text{at}(y_1) - \sigma(y_2))$$

$$\Rightarrow \quad \quad \quad = \quad \quad \quad \text{because } \sigma \circ \tau \circ \sigma = \sigma$$

$$\Rightarrow \sigma \cdot \mathcal{D}_\rho = \sigma \cdot \mathcal{D}_\sigma.$$

Thus, by Ex. II.3, we deduce that $(C_1, \sigma \cdot \mathcal{D})$ is a λ -model. $\quad =$

Part III: Given a syntax-free λ -model $(C_1, \sigma, \mathcal{D})$, let $\sigma \cdot \mathcal{D}_\rho$ be as in

Part II, and $\bar{\sigma}: C/\sim \rightarrow C$ as in Part I. Then f.a. $c \in C_1$

$$\begin{aligned} \bar{\sigma}(x_c) &= \sigma(\mathcal{D}_\rho(\text{App}(x_1, y)) \cdot \mathcal{D}_\rho(c(x))) = \sigma(\sigma(\text{App}(x_1, y)) \cdot \mathcal{D}_\rho(c(x)) \cdot \sigma^{-1}(y)) \\ &= \sigma(\sigma \cdot \mathcal{D}_\rho(c(x)) \cdot \sigma^{-1}(y)) = \sigma(x_c). \end{aligned}$$

Vice versa, if $(C_1, \sigma \cdot \mathcal{D})$ is a λ -model, we obtain a λ -model

$(C_1, \bar{\sigma} \cdot \mathcal{D})$ via Parts I & II.

Then $\bar{\sigma} \cdot \mathcal{D} = \sigma \cdot \mathcal{D}$ is shown by another induction along the complexity of λ -terms.

The variable & App-cases are immediate. For $t \in \Lambda^C$ s.t. $\bar{\sigma} \cdot \mathcal{D}_\rho = \sigma \cdot \mathcal{D}_\rho$

f.c. $\rho \in \text{Val}(C)$ and $z \in \text{Var}$, we get

$$\overline{(\lambda z.t)\rho} = \sigma(\overline{(\lambda z.t)\rho_{C-1z}})$$

$$= \sigma(\overline{(\lambda z.t)\rho_{C-1z}})$$

$$= \sigma(\overline{\lambda_{C-1z.t}\rho})$$

$$= \overline{(\lambda y. \text{App}(x, y))\rho_{C-1z/x}}$$

$$= \overline{(\lambda y. \text{App}(1z.t, y))\rho} \quad \text{by Berry Ext.}$$

$$= \overline{(\lambda y. t[C-1z])\rho} \quad \text{by } \beta\text{-congruence (remembers } C-1z)$$

$$= \overline{(\lambda z.t)\rho} \quad \text{by } \alpha\text{-congruence.} \quad \square$$

Corollary 14: If (C, \cdot) is combinatorially complete wrt. some $f: C \rightarrow C$ and extensional, then there is exactly one way to extend (C, \cdot) to a λ^C -model.

Proof: If (C, \cdot) is comb. complete wrt. $f: C \rightarrow C$ and extensional, then (C, \cdot, f) is the unique syntax-free λ^C -model on (C, \cdot, f) .

(Here, note that $\text{Id}: C \rightarrow C$ is representable via $\overline{(\lambda x.\rho_{C-1x})}$ & comb. completeness. \square)

Remark 15: Comb. complete applicative structures wrt. $C = \emptyset$ are exactly combinatory algebras in the sense of Curry. That is, appl. structures (C, \cdot) together with distinguished elements $k, s \in C$ s.t.h.

$$\ast \forall c, d \in C: k \cdot c \cdot d = c$$

$$\ast \forall c, d, e \in C: s \cdot c \cdot d \cdot e = (c \cdot e)(d \cdot e).$$