Lecture 25: Models of CBPV

Lecturer: Max S. New
Scribe: Darren Fitzgerald

April 12, 2023

1 Review of Formal Systems So Far

The overarching theme of this course has been to develop categorical semantics of syntactic systems. Over the course of this semester, we have formalized multiple syntactic systems and constructed some natural mathematical models for them. Some nuances about the system, such as the distinction between contexts and types in STT, cannot be internalized so naturally by its typical model, so we introduce some additional structure to which refines the model to some intermediate notion which is almost tautologically sound and complete.

Similar to the case with IPL and STT, today we will define a sound and complete CT-structure for Call-by-push-value whose model is formed by an adjunction between a category of values and a category of effects. A summary of syntactic systems and their associated semantics, and models can be found below:

Syntax	(Sound and Complete) Semantics	Typical Models
IPL	Propositional CT-Structure	Heyting Algebra
STT	CT-Structure	biCCC
CBPV	CPBV CT-Structure	Adjunction

2 CBPV CT-Structure

2.1 CT-Structure

Recall the following fragment of the judgmental structure of Call-by-push-value. We can see that it contains nearly the same components that should be familiar to us from STT:

- **Value Contexts**: Γ
- **Value Types**: $A_1, A_2, \ldots$
- **Values**: $\Gamma \vdash V : A$

- Implicitly, we should also have a notion of **substitution** between value contexts such that $\Gamma' \xrightarrow{\gamma} \Gamma$

In a sense, we want CBPV to be an extension of STT. It should therefore be an unsurprising result that a similar presentation of the CT-Structure for STT appears as a component of the for Call-by-push-value. We define CBPV's CT-Structure as follows

Definition 1. *A **CBPV CT-Structure** consists of the following data:*

I. A CT-Structure $\mathscr{V}$ representing the values. Recall that this means we have:

- *A cartesian category $\mathscr{V}_c$ where ...*
  - *objects $\Gamma_1, \Gamma_2 \in (\mathscr{V}_c)_0$ represent contexts*
  - *morphisms $\gamma \in \mathscr{V}_c(\Gamma_1, \Gamma_2)$ represent substitutions*
- *A set $\mathscr{V}_T$ which represent types*
- *$forall A \in \mathscr{V}_T$ a predicator $\mathrm{Tm}_{\mathscr{V}} A : \mathscr{V}_c^{op} \to$ **Set***
- *Such that for each type $A \in \mathscr{V}_T$ we have a context $sole A$ representing $\mathrm{Tm}_{\mathscr{V}} A$, i.e., $i : Y(sole A) \cong \mathrm{Tm}_{\mathscr{V}} A$*

II. A category $\mathscr{E}_c$ representing effectful behavior such that

- *objects $\Xi \in (\mathscr{E}_c)_0$ correspond to combinations (referred to as **term contexts**) of value contexts Γ and computation context $\underline{\Delta}$ found in syntax in the judgments of the form $\Gamma \mid \underline{\Delta} \vdash M : B$*
- *morphisms $\xi \in (\mathscr{E}_c)_1(\Xi, \Xi')$ represent substitutions over both the $\underline{\Delta}$ and Γ component of the contexts. Recall that in CBPV we have two rules of substitution in the term judgment,*

$$\frac{\gamma : \Gamma' \to \Gamma \qquad \Gamma \mid \underline{\Delta} \vdash M : \underline{B}}{\Gamma' \mid \underline{\Delta} : M[\gamma]} \qquad \frac{\Gamma \mid \underline{\Delta} \vdash N : \underline{B}' \qquad \Gamma \mid \bullet : \underline{B}' \vdash M : \underline{B}}{\Gamma \mid \underline{\Delta} \vdash M[N]}$$

We can then informally think of morphisms between contexts as internalizing a (pseudo-syntactical) judgment that combines both notions of substitution into a single rule:

$$\frac{(\gamma|\delta) : \Gamma_1 \mid \underline{\Delta}_1 \to \Gamma_2 \mid \underline{\Delta}_2 \qquad \Gamma_2 \mid \underline{\Delta}_2 \vdash M : \underline{B}}{\Gamma_1 \mid \underline{\Delta}_1 \vdash M[\gamma \mid \delta] : B}$$

where δ is a term given in the following cases

$$\delta = \begin{cases} * \text{ if } \underline{\Delta}_1 = \cdot, \underline{\Delta}_2 = \cdot \\ N \text{ if } \underline{\Delta}_1 = \cdot, \underline{\Delta}_2 = \bullet : \underline{B}, \text{ and } \Gamma_2 \mid \Delta_1 \vdash N : \underline{B} \end{cases}$$

where $*$ represents the trivial computation

III. *An object $I \in (\mathscr{E}_c)_0$ representing the empty computation context $\cdot \mid \cdot$. Note that unlike the empty context in a CT structure, I is not necessarily a terminal object in $\mathscr{E}$.*

IV. *A functor $\oslash : \mathscr{V}_c \times \mathscr{E}_c \to \mathscr{E}_c$ (written using infix notation) that internalizes the structure of value contexts concatenating with term contexts in the syntax:*

$$\Gamma \oslash (\Gamma' \mid \underline{\Delta}) = \Gamma, \Gamma' \mid \underline{\Delta}$$

with the following natural isomorphisms given by the following pointwise definitions

- *an "associator"*[1]

$$\alpha^{\Gamma_1,\Gamma_2,\Xi} : \Gamma_1 \times \Gamma_2 \oslash \Xi \cong \Gamma_1 \oslash \Gamma_2 \oslash \Xi$$

satisfying the "pentagon law":

$$\begin{array}{ccccc}
\Gamma_1 \times (\Gamma_2 \times \Gamma_3) \oslash \Xi & \xrightarrow{((\pi_1,\pi_1\pi_2),\pi_2\pi_2)\oslash\Xi} & (\Gamma_1 \times \Gamma_2) \times \Gamma_3 \oslash \Xi & & \\
 & & & \searrow^{\alpha^{\Gamma_1\times\Gamma_2,\Gamma_3,\Xi}} & \\
\downarrow{\scriptstyle \alpha^{\Gamma_1,\Gamma_2\times\Gamma_3,\Xi}} & & & & \Gamma_1 \times \Gamma_2 \oslash \Gamma_3 \oslash \Xi \\
 & & & \swarrow_{\alpha^{\Gamma_1,\Gamma_2,\Gamma_3\oslash\Xi}} & \\
\Gamma_1 \oslash (\Gamma_2 \times \Gamma_3) \oslash \Xi & \xrightarrow[\Gamma_1\oslash\alpha^{\Gamma_2,\Gamma_3,\Xi}]{} & \Gamma_1 \oslash \Gamma_2 \oslash \Gamma_3 \oslash \Xi & &
\end{array}$$

- *a "unitor"*

$$i^{\Xi} : 1 \oslash \Xi \cong \Xi$$

satisfying the "triangle law":

$$\begin{array}{ccc}
1 \times \Gamma \oslash \Xi & \xrightarrow{\alpha^{1,\Gamma,\Xi}} & 1 \oslash \Gamma \oslash \Xi \\
 & \searrow_{\pi_2^{1,\Gamma}\oslash\Xi} & \downarrow{\scriptstyle i^{\Gamma\oslash\Xi}} \\
 & & \Gamma \oslash \Xi
\end{array}$$

Which gives us a natural interpretation of the functor's action on morphisms as well. Given $\gamma : \Gamma_1 \to \Gamma_2$ and $\gamma' \mid \delta : \Gamma_1' \mid \underline{\Delta}_1 \to \Gamma_2' \mid \underline{\Delta}_2$:

$$\gamma \oslash \gamma' \mid \delta = (\gamma, \gamma' \mid \delta) : \Gamma_1, \Gamma_1' \mid \underline{\Delta} \to \Gamma_2, \Gamma_2' \mid \underline{\Delta}$$

Note that the associator and unitor carry the monoidal structure of value contexts into $\mathscr{E}_c$, but in a way that extends to functors *rather than just functions on monoids. In fact, we can think of $\oslash$ paired with the associator and unitor as a generalization of the action of the monoid of value contexts $((\mathscr{V}_c)_0, 1, \times)$ over the term contexts in $\mathscr{E}_c$ to account for morphisms as well.*

[1] note that we can drop the parentheses unambiguously here

V. a set of computation types $\mathscr{E}_{Ty}$

VI. $\forall \underline{B} \in \mathscr{E}_{Ty}$ *we have*

- *a predicator* $\mathrm{Tm}_{\mathscr{E}}\underline{B}$ *on* $\mathscr{E}_c$. *We think of this as modeling the term judgment* $\Gamma \mid \underline{\Delta} \vdash M : B$ *along with the action of substitution and identity/associativity.*
- *a singleton context* $\mathrm{sole}\underline{B} \in (\mathscr{E}_c)_0$ *that represents the predicator* $\mathrm{Tm}_{\mathscr{E}}\underline{B}$*, e.g.* $Y(\mathrm{sole}\underline{B}) \cong \mathrm{Tm}_{\mathscr{E}}\underline{B}$.

  Syntactically, we think of $\mathrm{sole}\underline{B} \in (\mathscr{E}_c)_0$ *as* $\cdot \mid \bullet : \underline{B}$ *where we have an empty value context and singleton computation context.* [2]

This CT-Structure gives a categorical interpretation of the judgmental structure of Call-by-push-value regardless of some initial signature or choice of connectives made. This enables us to describe each type connective and its associated rules more or less in isolation from the base type theory.

2.2 Aside: Theory Extensions from Semantics

When we define a model for syntactic systems, there may be additional structure on the contexts or types that could be interpreted as new types, contexts, or operations on them that were not identified in the image of our original language. For example, in the syntax of CBPV we defined computational contexts as "stoups":

$$\underline{\Delta} ::= \cdot \mid \bullet : \underline{B}'$$

However, there's (intentionally) nothing in the CT-Structure precluding us from expressing objects in $\mathscr{E}_c$ that are not of this form. This ability to express constructions outside the original system is not only intentional, it is a *desirable* outcome: we can design extensions of our type theory by identifying new constructions in the semantics and internalizing them syntactically.

3 Type Structure

Now that we've defined a CT-Structure for CBPV we will internalize the the canonical connectives $\mathrm{Clo}, \times, \to, \mathrm{Ret}$, and $+$ from the syntax. Fix a CT-Structure given by $(\mathscr{V}, \mathscr{E}, I, \oslash)$, then we can define the connectives as follows:

3.1 Closure Types

Definition 2. *Let* $\underline{B} \in \mathscr{E}_{Ty}$ *then a* ***closure type*** *for* $\underline{B}$ *is*

[2] It may look like we defined a CT-Structure $\mathscr{E}$, but we have not defined a product structure over $\mathscr{E}_c$. Therefore, we do not have the conventional notion of syntactic weakening or context concatenation over term contexts.

- *An object* $\mathrm{Clo}\underline{B} \in \mathscr{V}_T$ *and*
- *Terms of type* $\underline{B}$ *are in natural bijection with values of type* $\mathrm{Clo}\underline{B}$*. In other words for some* $\Gamma \in (\mathscr{V}_c)_0$ *we have the following natural bijection*

$$\mathrm{Tm}_{\mathscr{V}}(\mathrm{Clo}\underline{B})(\Gamma) \cong \mathrm{Tm}_{\mathscr{E}}(\underline{B})(\Gamma \oslash I)$$

Recall from the syntax the introduction rule

$$\frac{\Gamma \mid \cdot \vdash M : \underline{B}}{\Gamma \vdash \mathrm{proc}\{M\} : \mathrm{Clo}\underline{B}}$$

This precisely corresponds to backward (right to left) direction of the bijection. The forward direction is given by the universal element

$$\mathrm{Tm}_{\mathscr{E}}(\underline{B})(\mathrm{sole}(\mathrm{Clo}\underline{B} \oslash I))$$

which corresponds to the elimination form

$$\frac{}{x : \mathrm{Clo}\underline{B} \mid \cdot \vdash x.\mathrm{call}() : \underline{B}}$$

3.2 Product Types

Product types are defined exactly the same as they were for STT

Definition 3. *Let* $A_1, A_2 \in \mathscr{V}_T$*. A* ***product type*** *is an element* $(A_1 \times A_2) \in \mathscr{V}_T$ *such that* $\mathrm{sole}(A_1 \times A_2)$ *is a product in* $\mathscr{V}_c$*.*

3.3 Function Types

Definition 4. *Let* $A \in \mathscr{V}_T$ *and* $\underline{B} \in \mathscr{E}_{Ty}$*. A* ***function type*** *is*

- *An element* $A \to \underline{B} \in \mathscr{E}_{Ty}$ *such that*
- *Terms of type* $A \to \underline{B}$ *are in bijection with terms of type* $\underline{B}$ *that can be constructed with a variable of type A in the context. Let* $\Xi \in (\mathscr{E}_c)_0$*. Then we have the following natural bijection:*

$$\mathrm{Tm}_{\mathscr{E}}(A \to \underline{B})\Xi \cong \mathrm{Tm}_{\mathscr{E}}(\underline{B})(\mathrm{sole}A \oslash \Xi)$$

Recall from the syntax the introduction rule for function types:

$$\frac{\Gamma, x : A \mid \underline{\Delta} \vdash M : \underline{B}}{\Gamma \mid \underline{\Delta} \vdash \lambda x.M : A \to \underline{B}}$$

As is the case with closure types, this rule corresponds with the reverse direction of the bijection and composing with the universal property of $\mathrm{sole}(A \to \underline{B})$ *we get that one direction is*

$$Y(\mathrm{sole}(A \to \underline{B}))\Xi \Rightarrow \mathrm{Tm}_{\mathscr{E}}\underline{B}(\mathrm{sole}A \oslash \Xi)$$

By the Yoneda lemma, this is determined by an element $\mathrm{Tm}_{\mathscr{E}}\underline{B}(\mathrm{sole}A \oslash \mathrm{sole}(A \to \underline{B}))$ *which corresponds to the elimination rule:*

$$\frac{}{x : A \mid \bullet : A \to \underline{B} \vdash \bullet : \underline{B}}$$

3.4 Return Types and Sum Types

Definition 5. *Let $A \in \mathscr{V}_T$. A* ***return type*** *is defined as*

- *An object* $\mathrm{Ret}A \in \mathscr{E}_{Ty}$
- *A natural bijection between the set of terms of type $\underline{B}$ when given some notion of a value type A in either the computation context or the value context. Let $\Gamma \in (\mathscr{V}_c)_0$ then*

$$\mathrm{Tm}_{\mathscr{E}}(\underline{B})(\Gamma \oslash \mathrm{sole}(\mathrm{Ret}A)) \simeq \mathrm{Tm}_{\mathscr{E}}(\underline{B})(\Gamma \oslash \mathrm{sole}A \oslash I)$$

This bijection is represented by the following invertible rule

$$\frac{\Gamma, x : A \mid \cdot \vdash N : \underline{B}}{\Gamma \mid \bullet : \mathrm{Ret}A \vdash \mathrm{var}\, x = \bullet \text{ in } N : \underline{B}}$$

Again applying the Yoneda lemma and using sole, *we can derive the introduction form as the universal element* $\mathrm{Tm}_{\mathscr{E}}(\mathrm{Ret}A)(soleA \oslash I)$*:*

$$\frac{}{x : A \mid \cdot \vdash \mathrm{ret}\, x : \mathrm{Ret}A}$$

Definition 6. *Let $A_1, A_2 \in \mathscr{V}_T$. A* ***binary sum type*** *is defined as*

- *An object $A_1 + A_2 \in \mathscr{V}_T$*
- *A natural bijection*

$$\mathrm{Tm}_{\mathscr{V}}(A')(\Gamma \times (A_1 + A_2)) \cong \mathrm{Tm}_{\mathscr{V}}(A')(\Gamma \times A_1) \times \mathrm{Tm}_{\mathscr{V}}(A')(\Gamma \times A_2)$$

- *As well as a natural bijection*

$$\mathrm{Tm}_{\mathscr{E}}(\underline{B})((A_1 + A_2) \oslash \Xi) \cong \mathrm{Tm}_{\mathscr{E}}(\underline{B})(A_1 \oslash \Xi) \times \mathrm{Tm}_{\mathscr{E}}(\underline{B})(A_2 \oslash \Xi)$$

such that the left direction is given by composition with the universal elements $(i_0, i_1) \in \mathrm{Tm}_{\mathscr{V}}(A_1 + A_2)A_1 \times \mathrm{Tm}_{\mathscr{V}}(A_1 + A_2)A_2$

4 Typical models

$$\mathscr{V} \underset{\mathrm{Clo}}{\overset{\mathrm{Ret}}{\rightleftarrows}} \mathscr{E} \quad (\bot)$$

Sometimes it nicer to work with all of the connectives in a type theory and construct the simplest possible model to work with. A typical model for CBPV consists of a bicartesian closed category $\mathscr{V}$ and a category $\mathscr{E}$ paired an interpretation of Ret and Clo such that Ret $\dashv$ Clo. More concretely we list its components:

I. A biCCC $\mathscr{V}$

II. A (usually bicartesian closed) category $\mathscr{E}$

III. An empty computational context $I \in (\mathscr{E}_c)_0$

IV. A functor $\oslash : \mathscr{V} \times \mathscr{E} \to \mathscr{E}$ which satisfies the same unitor and associator natural isomorphisms from the CT-Structure model. However, unlike the CT-Structure $\oslash$ here is in adjoint triality with the following bifunctors.

- $\to: \mathscr{V}^{op} \times \mathscr{E} \to \mathscr{E}$
- $\multimap: \mathscr{E}^{op} \times \mathscr{E} \to \mathscr{V}$

the adjunction of these functors is illustrated in the isomorphism between the following hom sets. Let $A \in \mathscr{V}$ and $B \in E$, then the following bijections are natural in A and B:

$$\mathscr{E}(A \oslash B, B') \cong \mathscr{E}(B, A \to B') \cong \mathscr{V}(A, B \multimap B')$$

Here, giving axioms for $\oslash$ and characterizing $\to$ and $\multimap$ by adjointness is an arbitrary choice: we could just as easily started with axioms for $\to$ or $\multimap$ and define $\oslash$ by adjointness instead. Using $\multimap$ as the primitive is most enticing because this means we can simply define $\mathscr{E}$ to be a *$\mathscr{V}$-enriched* category, a well-studied notion.

Notice that Ret and Clo have not been defined yet. This is because we can describe them as a universal construction using the above bifunctors

- $\mathrm{Ret}A \simeq A \oslash I$
- $\mathrm{Clo}B \simeq I \multimap B$

5 Weak Initiality

The syntax of CBPV itself as before defines a CBPV c-t structure $\mathcal{L}$, the Lindenbaum algebra, which satisfies an analogous weak initiality theorem to STT: it gives a CBPV CT structure homomorphism to any other CBPV CT structure and this homomorphism is unique up to unique natural isomorphism.