1 Bicartesian Doubly Closed Category

Given a category $\mathcal{C}$, its presheaf category ($\widehat{\mathcal{C}} := [\mathcal{C}^{op}, Set]$) is bicartesian closed. Given a monoidal category $(\mathcal{C}, \otimes_C, I_C)$, its presheaf category is bicartesian closed and monoidal closed via the Day convolution. The monoidal product is given by:

$$(P \otimes^{Day} Q)(x) = \int^{y,z} \mathcal{C}[x, y \otimes_C z] \times P(y) \times Q(z)$$

The Day monoidal product has the universal property that any maps out of it are in bijective correspondence with a family of maps natural in x and y (Agda): [1]

$$\widehat{\mathcal{C}}[P \otimes^{Day} Q, R] \;\cong\; \widehat{\mathcal{C} \times \mathcal{C}}[P \overline{\times} Q, R \circ \otimes_C] \;\cong\; \Pi_{x,y:\, ob\; C}\; Set[P(x) \times Q(y), R(x \otimes_C y)]$$

The monoidal closed structure is given by:

$$(P \mathrel{-\!\!*} Q)(X) = \widehat{\mathcal{C}}[P, Q(X, -)]$$

With the universal property that the closed structure is right adjoint to the tensor (Agda):

$$\widehat{\mathcal{C}}[A \otimes_C B, C] \cong \widehat{\mathcal{C}}[A, B \mathrel{-\!\!*} C] \tag{1}$$

Bicartesian doubly closed categories have been used in the denotational semantics of bunched type theories [5][1][4].

2 Towards Bunched Call By Push Value with Dynamic Store

Categorical models of dynamic store use presheaf categories to model the dependence of the heap structure on a current *world* [3][6][2]. Seemingly none of these existing models attempt to combine a call by push value language with the separating type connectives, $\otimes$ and $\mathrel{-\!\!*}$, used in bunched type theories. Our investigation into possible models of such a language have run into some potential issues. To illustrate this, we will start with the model for a call by push value language with dynamic store presented in chapter 7 of Levy's thesis.

2.1 Definitions

Let $(C, \otimes_C, I_C)$ be a monoidal category, the value category be $\mathcal{V} := [C, Set]$, computation category $\mathcal{C} := [C^{op}, Set]$, and use the *standard* monad for ground dynamic store with $F : \mathcal{V} \to \mathcal{C}$ as:

$$F(A)(x) := \Sigma_{y:ob\; C} \Sigma_{f:C[x,y]} A(y)$$

and $U : \mathcal{C} \to \mathcal{V}$ as :

$$U(\underline{B})(x) := \Pi_{y:ob\; C} \Pi_{f:C[x,y]} \underline{B}(y)$$

The oblique morphisms in this model are given by families of maps:

$$\mathcal{O}[A, \underline{B}] := \Pi_{x:ob\; C} Set[A(x), \underline{B}(x)]$$

we have the following isomorphims:

$$\mathcal{V}[A, U(\underline{B})] \cong \mathcal{O}[A, \underline{B}] \cong \mathcal{C}[F(A), \underline{B}]$$

And we can attempt to define a computation separating function by:

$$(A \mathrel{-\!\!*} \underline{B})(x) := \Pi_{y:ob\; C} Set[A(y), \underline{B}(x \otimes_C y)]$$

2.2 Problems with an Abstract Monoidal Category

Before committing to the category of worlds used in Levy's model, we will work with an arbitrary monoidal category $(C, \otimes_C, I_C)$.

[1] here $\overline{\times}$ is the *external* product

2.2.1 Issue 1: Universal Property of Tensor for Oblique Morphisms

Let's attempt to show the following:

$$\mathcal{O}[P \otimes Q, \underline{R}] \cong \mathcal{O}\times[P \overline{\times} Q, \underline{R} \circ \otimes_C]$$

where

$$\mathcal{O}\times[P \overline{\times} Q, \underline{R} \circ \otimes_C] := \Pi_{x,y:ob\ C} Set[P(x) \times Q(y), \underline{R}(x \otimes_C y)]$$

A problem arises when trying to define the backwards map of this isomorphims. Given $m : \mathcal{O}\times[P \overline{\times} Q, \underline{R} \circ \otimes_C]$ and $x : ob\ C$, we need to define a map $Set[(P \otimes Q)(x), \underline{R}(x)]$. This is a map out of a coequalizer [2] which we can attempt to give as a map induced from:

$$(f : y \otimes_C z \to x, p : P(y), q : Q(z)) \mapsto ? : \underline{R}(x)$$

However, using the data we currently have, we can only construct

$$m(y)(z)(p, q) : \underline{R}(y \otimes_C z)$$

and since $\underline{R}$ is contravariant, we can't use $\underline{R}(f) : \underline{R}(x) \to \underline{R}(y \otimes_C z)$. This is not surprising since the proof of this universal property in the value category $\mathcal{V}[P \otimes Q, R] \cong \mathcal{V}\times[P \overline{\times} Q, R \circ \otimes_C]$ uses the functorial action of R on f (see here) [3] So by swapping the variance of R (now $\underline{R}$ since it is from the computation category) this proof should break. Seemingly, this proof won't go through when we assume a generic monoidal category C. Perhaps we can recover this property if we work with a specific concrete category?

2.2.2 Issue 2: Universal Property of the Separating Function Type

This is just another perspective on the variance issue above. We'd like to show

$$\mathcal{O}[P \otimes Q, \underline{R}] \cong \mathcal{O}[P, Q \mathrel{-\!\!*} \underline{R}]$$

Since we don't have the universal property of tensor for oblique morphisms, we can try to get at this proof via the universal property of tensor in the value category. Note that we have

$$\mathcal{O}[P \otimes Q, \underline{R}] \cong \mathcal{V}[P \otimes Q, U(\underline{R})] \cong \mathcal{V}\times[P \overline{\times} Q, U(\underline{R}) \circ \otimes_C]$$

and

$$\mathcal{O}[P, Q \mathrel{-\!\!*} \underline{R}] \cong \mathcal{V}[P, U(Q \mathrel{-\!\!*} \underline{R})]$$

So we can try to show

$$\mathcal{V}\times[P \overline{\times} Q, U(\underline{R}) \circ \otimes_C] \cong \mathcal{V}[P, U(Q \mathrel{-\!\!*} \underline{R})]$$

Again, we fail to define the backwards direction of this isomorphim due to a variance issue with $\underline{R}$. Given $m : \mathcal{V}[P, U(Q \mathrel{-\!\!*} \underline{R})]$, it suffices to construct a map $eval : \mathcal{V}\times[U(Q \mathrel{-\!\!*} \underline{R}) \overline{\times} Q, U(\underline{R}) \circ \otimes_C]$ with components

$$(x, y)(f : U(Q \mathrel{-\!\!*} \underline{R})(x), q : Q(y)) \mapsto ?\ :\ (U(\underline{R}))(x \otimes_C y)$$

unfolding some of the definitions, we have

$$\begin{aligned} &f : \Pi_{z:ob\ C} \Pi_{g:C[x,z]} (\Pi_{w:ob\ C} Set[Q(w), \underline{R}(z \otimes_c w)]) \\ &? : \Pi_{z:ob\ C} \Pi_{g:C[x \otimes_C y, z]} (R(z)) \end{aligned}$$

Thus we have to define $? : \underline{R}(z)$ from the following data:

$$\begin{aligned} &x, y, z : ob\ C \\ &q : Q(y) \\ &f : \Pi_{z:ob\ C} \Pi_{g:C[x,z]} (\Pi_{w:ob\ C} Set[Q(w), \underline{R}(z \otimes_c w)]) \\ &g : C[x \otimes_C y, z] \end{aligned}$$

The *obvious* thing do to would be to use $f(x)(id_x)(y)(q) : \underline{R}(x \otimes_C y)$ and $\underline{R}(g)$, but the variance of $\underline{R}$ is working against us.

[2] since coends in Set can be encoded as coequalizers

[3] note the difference in variance is due to the fact this proof is for presheaves and not covariant presheaves

3 Issue

There is a variance issue when trying to add a **computational** separating function type to Levy's dynamic store model[3] [4][5]. Take the category of worlds to be $\mathcal{W} := FinSet_{mono}$, the value category to be $\mathcal{V} := [\mathcal{W}, Set]$ and the computation category to be $\mathcal{C} := [\mathcal{W}^{op}, Set]$. Value judgments $\Gamma \vdash_v M : A$ are denoted as morphisms in $\mathcal{V}$. Computation judgments $\Gamma \vdash_c M : B$ are denoted as families of maps $\forall (w : ob\ W) \rightarrow Set[[\![\Gamma]\!](w), [\![B]\!](w)]$. Note that we are dropping the storage part (S) of Levy's monad. The monoidal structure on $\mathcal{W}$ given by disjoint union yields a monoidal structure on $\mathcal{V}$ via the Day convolution[6].

$$(A \otimes_D B)_0(w_1) = \int^{w_2, w_3} \mathcal{W}[w_2 \otimes w_3, w_1] \times A(w_2) \times B(w_3)$$

The separating function in the **value category**$(A, B : ob\ \mathcal{V})$ is given by:

$$(A \mathrel{-\!\!*} B)_0(w) = \mathcal{V}[[\![A]\!], [\![B]\!](w \otimes \_)]$$

And we have that:

$$\mathcal{V}[A \otimes_D B, C] \cong \mathcal{V}[A, B \mathrel{-\!\!*} C] \tag{2}$$

The **computational** function type $(A : ob\ \mathcal{V}, B : ob\ \mathcal{C})$ is given by:

$$(A \rightarrow B)_0(w) = Set[[\![A]\!](w), [\![B]\!](w)]$$

We can try to define the **computational** separating function $(A : ob\ \mathcal{V}, B : ob\ \mathcal{C})$ as :

$$(A \mathrel{-\!\!*} B)_0(w) = \forall (w' : ob\ W) \rightarrow Set[[\![A]\!](w'), [\![B]\!](w \otimes w')]$$

which is a contravariant functor. We should expect the following isomorpism of types(in Set?):

$$(A \otimes_D B) \rightarrow C \cong A \rightarrow B \mathrel{-\!\!*} C$$

given by:

$$\begin{aligned}
&fun : ((A \otimes_D B) \rightarrow C) \rightarrow (A \rightarrow B \mathrel{-\!\!*} C) \\
&fun\ M\ w_1\ (a : [\![A]\!](w_1))\ w_2\ (b : [\![B]\!](w_2)) = M(w_1 \otimes w_2)(id_{w_1 \otimes w_2}, a, b) \\
&\\
&inv : (A \rightarrow B \mathrel{-\!\!*} C) \rightarrow ((A \otimes_D B) \rightarrow C) \\
&inv\ M\ w_1\ (w_2, w_3, f : w_2 \otimes w_3 \rightarrow w_1, a : [\![A]\!](w_2), b : [\![B]\!](w_3)) = [\![B]\!]_1(f)(M\ w_2\ a\ w_3\ b)
\end{aligned}$$

However, the variance of $[\![B]\!]$ gives us $[\![B]\!]_1(f) : [\![B]\!](w_1) \rightarrow [\![B]\!](w_2 \otimes w_3)$ which is the opposite direction that *we want*[7].

3.1 Our Model

I was able to derive an *inverse* (likely not able to show the isomorphism) in our model, but it felt like a hack and involves an arbitrary choice. Without reproducing all the details here, the gist is the following:

$$\begin{aligned}
&s2p : \mathcal{V}[A \otimes_D B, A \times B] \\
&s2p(w_1)(w_2, w_3, f : w_2 \otimes w_3 \hookrightarrow w_1, a, b) = [\![A]\!]_1(inl\ ; f)(a), [\![B]\!]_1(inr\ ; f)(b) \\
&\\
&inv : (A \rightarrow B \mathrel{-\!\!*} C) \rightarrow ((A \otimes_D B) \rightarrow C) \\
&inv\ M\ w\ s = [\![B]\!]_1(inl\ or\ inr)(M\ w\ (\pi_1\ p)\ w\ (\pi_2\ p)) \\
&\quad where \\
&\qquad p : [\![A \times B]\!](w) \\
&\qquad p = s2p\ w\ s
\end{aligned}$$

[4] Chapter 6

[5] The following issue exists in our setup too.

[6] *covariant* Day convolution given by taking the monoidal structure on $\mathcal{W}^{op}$ and then applying the day convolution

[7] Meaning this is how the isomorpism goes in (1)

4 A possible way forward

I'm starting to look at a weaker version of the setup in section 2.4 of [6] which is a model of SystemF$_{\mu}^{ref}$. I think we had already worked out the computational separating function for an algebra model of CBPV.

References

[1] Biering, B. On the Logic of Bunched Implications.

[2] Kammar, O., Levy, P. B., Moss, S. K., and Staton, S. A monad for full ground reference cells. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)* (June 2017), pp. 1–12.

[3] Levy, P. *Call-By-Push-Value: A Functional/Imperative Synthesis.* 01 2004.

[4] O'Hearn, P. On bunched typing. *Journal of Functional Programming 13*, 4 (July 2003), 747–796.

[5] Pym, D. J. *The Semantics and Proof Theory of the Logic of Bunched Implications*, vol. 26 of *Applied Logic Series.* Springer Netherlands, Dordrecht, 2002.

[6] Sterling, J., Gratzer, D., and Birkedal, L. Denotational semantics of general store and polymorphism, Apr. 2023. arXiv:2210.02169 [cs].