Diagrams in symmetric monoidal categories

are they symmetric monoidal?

Let \( C \) be a category with an internal hom complimented by a left adjoint. That is, for objects \( A, B \) in \( C \), \( \text{Hom}(A, B) \) is again an object of \( C \), and there is a symmetric product \( \otimes \) in \( C \) such that for all objects \( A, B, C \) in \( C \)

\[
\text{Hom}(A \otimes B, C) \cong \text{Hom}(A, \text{Hom}(B, C)).
\]

This is a particularly nice attribute for a category to possess thus earning it the distinguished title symmetric monoidal. The authors were checking that this condition held for various categories to familiarize themselves with the properties of adjoints and quickly got bored by the repetitive arguments. As the categories got more complicated, we found that the arguments boiled down to previous arguments for the categories from which the complicated ones were constructed. That is, the argument for simplicial sets worked because the condition held for sets, and a simplicial set is just a particular diagram of sets. There must be some properties of the category we could check first that would imply the existence of this left adjoint. So we had a question: what is requisite of the category \( C \) in order for diagrams in \( C \) to have an internal-hom and a symmetric product with an adjunction? Can we construct the product as the left adjoint of the internal-hom?

Let \( D \) be a small category, \( C \) a symmetric monoidal category, and \( F : D \to C \) a functor. This can be viewed as filling a particular diagram with objects of \( C \), because a small category looks like a diagram made up of dots and arrows. We want to show that the category of functors \( D \to C \) (that is, the category of all the diagrams in \( C \) of the same shape) has an internal-hom that has a left adjoint (because \( C \) has this structure). We will impose various conditions on the category \( C \) along the way.

We will use the notation \( A \bullet \) for a functor \( A : D \to C \) and use \( A(D) \) to represent an object of \( C \) in the diagram (for \( D \in \text{obj}(D) \)). This notation is similar to that used for simplicial sets and chain complexes as a reminder that the objects in our category are functors. Our constant source of inspiration will be the category of simplicial sets, but our reality check will be the nearly trivial dumbbell diagram

\[
\begin{array}{c}
D \\
\downarrow^d \\
D'.
\end{array}
\]

With all these different categories and homs floating around, some notation is in order. The subscript will always denote the category the objects are in. The set hom, which is a result of \( C \) being a category, will be denoted by \( \text{hom}_C(A, B) \). When a category has an internal-hom, that is, when we can define a hom such that it is an object in the category and not merely a set, we will denote the maps by \( \text{Hom}_C(A, B) \). A category is called enriched when its hom-sets have more structure than a set. For example, when each hom-set of \( C \) forms a group, we say \( C \) is enriched in groups. When a category has an enriched hom-set, we will indicate the enriching category in context and denote the hom-set by \( \text{Hom}_C(A, B) \). Note that the internal-hom is just a category enriched over itself and we have given this distinguished notation. Finally, we will denote the category of functors \( D \to C \) with natural transformations by \( C^D \).

Perhaps we should note that everything we say about simplicial sets is completely standard. Nothing is new. We explore the constructions that others have developed long ago in order to generalize to functor categories.

1 Some essential category theory

Define limits and colimits. Discuss equalizers, coequalizers, products, coproducts. Can move the properties from the left adjoint section here.
2 Simplicial Sets

Definition 2.1 A simplicial set is a contravariant functor from the category $\Delta$ of finite totally ordered sets with order preserving maps to the category of Sets.

We may also characterize a simplicial set as a sequence of sets $X_0, X_1, \ldots$ with face maps $d_i : X_n \to X_{n-1}$ and degeneracy maps $s_i : X_n \to X_{n+1}$, $0 \leq i \leq n$, satisfying compatibilities:

(i) $d_id_j = d_{j-1}d_i$ for $i < j$

(ii) $s_is_j = s_{j+1}s_i$ for $i \leq j$

(iii) $d_is_j = s_{j-1}d_i$ for $i < j$

$d_is_j = 1$ if $i < j$

$d_is_j = d_{j+1}s_j$ for $i > j + 1$

Similarly, $d_i$ has coface maps $d' : n \to n - 1$ and degeneracy maps $s' : n \to n + 1$ for $0 \leq i \leq n$ where $d_i$ omits the element $i$ and $s_i$ shifts all entries greater than $i$ up one spot and puts an $i$ in the empty slot.

A simplicial set $X_*$ can be represented by a diagram (which is the shape of $\Delta$ filled with sets and set maps):

\[
\begin{array}{cccc}
X_0 & \xleftarrow{d_0} & X_1 & \xleftarrow{d_0} X_2 & \ldots \\
\xrightarrow{s_0} & & \xrightarrow{s_0} & & \\
X_0 & \xleftarrow{d_1} & X_1 & \xleftarrow{d_1} X_2 & \ldots \\
\xrightarrow{s_1} & & \xrightarrow{s_1} & & \\
& & & & \\
\end{array}
\]

Note that to see sSet as a functor from a small category $\mathcal{D}$ to a symmetric monoidal category $\mathcal{C}$ (the context of this exposition), we just let $\mathcal{D}$ be $\Delta^{op}$ and $\mathcal{C}$ be Set.

Note that the opposite ordinal category, $\Delta^{op}$, has face maps $d_i : n \to n - 1$ and degeneracy maps $s_i : n \to n + 1$ for $0 \leq i \leq n$ where $d_i$ omits the element $i$ and $s_i$ shifts all entries greater than $i$ up one spot and puts an $i$ in the empty slot.

That is,

\[
d_i(0, 1, \ldots, n) = (0, \ldots, \hat{i}, \ldots, n)
\]

and

\[
s_i(0, 1, \ldots, n) = (0, \ldots, i, \ldots, n).
\]

If you are not convinced that these maps satisfy the identities above, go to a private place and check them. In fact, every morphism in $\Delta^{op}$ can be written as a composition of the maps $d_i$ and $s_j$.

Similarly, $\Delta$ has coface maps $d'_i : n - 1 \to n$ and codegeneracy maps $s'_i : n + 1 \to n$ for $0 \leq i \leq n$ where $d'_i$ omits the element $i$ and $s'_i$ sends both $i$ and $i + 1$ to $i$.

That is,

\[
d'_i(j) = \begin{cases} j & \text{if } j < i \\ j + 1 & \text{if } j \geq i \end{cases}
\]

and

\[
s'_i(j) = \begin{cases} j & \text{if } j \leq i \\ j - 1 & \text{if } j > i \end{cases}
\]

We learn more about objects in a category by considering maps between them that preserve structure, so let’s describe the morphisms of simplicial sets.

Definition 2.2 Let $X_*$ and $Y_*$ be simplicial sets. A simplicial map $f : X_* \to Y_*$ is a sequence of maps $f_n : X_n \to Y_n$ that commute with the face and degeneracy maps. So $f_{n+1}s_i = s_if_n$ and $f_n d_i = d_if_{n+1}$.

These levelwise maps are crucial to defining the internal-hom of sSet. We call them levelwise because we only ask for maps straight across for each $n \geq 0$:
We will look at the internal-hom of \( sSet \) after we have generalized these levelwise maps to diagram categories.

## 3 An Objectwise Hom

These levelwise maps in \( sSet \) are crucial to defining the internal-hom of \( sSet \), so we would like to generalize this idea, before trying to figure out the internal-hom. To build an internal-hom in a general diagram category, we need first to define an objectwise hom and force it to commute with all the "structure" of \( D \). These are like the levelwise simplicial maps for \( sSet \).

For the stupid diagram

\[
\begin{array}{ccc}
D & \rightarrow & D' \\
\downarrow^{d} & & \downarrow^{d'} \\
D' & & \\
\end{array}
\]

with \( A, B : D \rightarrow C \), we want to define the arrow from the \( A \) diagram to the \( B \) diagram

\[
\begin{array}{ccc}
A(D) & \rightarrow & B(D) \\
\downarrow^{A(d)} & & \downarrow^{B(d)} \\
A(D') & \rightarrow & B(D') \\
\end{array}
\]

We clearly want a map for each dot in the diagram and some consistency with the diagram maps. That is, if \( f_D : A(D) \rightarrow B(D) \) and \( f_{D'} : A(D') \rightarrow B(D') \), then we want \( f_{D'} \circ A(d) = B(d) \circ f_D \), that is, the following diagram commutes:

\[
\begin{array}{ccc}
A(D) & \rightarrow & B(D) \\
\downarrow^{A(d)} & & \downarrow^{B(d)} \\
A(D') & \rightarrow & B(D') \\
\end{array}
\]

For every dot \( D \) in \( D \), we want an element of \( \text{Hom}_\mathcal{C}(A(D), B(D)) \) (this hom-set is an object of \( \mathcal{C} \) because \( \mathcal{C} \) has an internal-hom), so an objectwise map between the diagrams \( A \) and \( B \) is an element of the product \( \prod_{D \in D} \text{Hom}_\mathcal{C}(A(D), B(D)) \), because we need a map for every object of \( D \). For this to be an object in \( \mathcal{C} \), we need \( \mathcal{C} \) to have products.

So let \( \mathcal{C} \) have products.

**Definition 3.1** A product is... (this belongs in section 1...)

We want only the collections of maps which commute with the diagram maps. So we will probably need to throw out some maps from this product. We’ll take a subset of the maps, but when we do this, we may throw out too many maps. For example, if \( \mathcal{C} \) is \( \text{Gp} \), in whittling down our hom-set we may throw out an inverse and no longer have a group. We need our hom-set to be an object of \( \mathcal{C} \) still, so we will get another prereq for \( \mathcal{C} \).

We want to isolate the elements of the product which commute with our diagram. Let’s think about the boring case again:

\[
\begin{array}{ccc}
A(D) & \rightarrow & B(D) \\
\downarrow^{A(d)} & & \downarrow^{B(d)} \\
A(D') & \rightarrow & B(D') \\
\end{array}
\]

We want \( (f, f') \) to be in the hom-set of diagrams only if \( f' \circ A(d) = B(d) \circ f \). So there are two maps \( A(D) \rightarrow B(D') \) that we wish would agree, and these two maps are obtained by pre-composition or post-composition with (essentially) the diagram map \( d \). Given a map \( f \) of \( \text{Hom}_\mathcal{C}(A(D), B(D)) \), we may
define \( d^*(f) = B(d) \circ f \) in \( \text{Hom}_C(A(D), B(D')) \), and given a map \( f' \) in \( \text{Hom}_C(A(D'), B(D')) \) we may define \( d_*(f') = f' \circ A(d) \) in \( \text{Hom}_C(A(D), B(D')) \). So we have

\[
\text{Hom}_C(A(D), B(D)) \times \text{Hom}_C(A(D'), B(D')) \xrightarrow{d^*, \ d_*} \text{Hom}_C(A(D), B(D'))
\]

To choose just the maps in the product which agree on both maps \( d_* \) and \( d^* \) is to ask for the limit of the diagram. If you’re not brushed up on your limits, there’s no need to panic because in this situation the limit is just the equalizer. Recall, for sets \( A, B \) with maps \( A \xrightarrow{f} B \), the equalizer is the subset \( C = \{ x \in A : f(x) = g(x) \} \). So the equalizer for homs is a “subset” of the product made up of elements that commute with the diagram map \( d \).

In general for any diagram (not just our dumbbell), our diagram hom-set will be the equalizer of

\[
\prod_{D \in D} \text{Hom}_C(A(D), B(D)) \xrightarrow{d^*, \ d_*} \prod_{D \in D'} \text{Hom}_C(A(D), B(D')).
\]

Let’s do a slightly harder example to check that this makes sense. Say we have the diagram:

\[
\begin{array}{ccc}
D & \xrightarrow{\delta} & F \\
\downarrow{\epsilon} & & \downarrow{B(\epsilon)} \\
E & & B(E)
\end{array}
\]

A map between diagrams of this shape would look like the following

\[
\begin{array}{ccc}
A(D) & \xrightarrow{A(\delta)} & A(F) \\
\downarrow{A(\epsilon)} & & \downarrow{B(\epsilon)} \\
A(E) & & B(E)
\end{array}
\]

\[
\begin{array}{ccc}
\quad & \quad & \quad \\
\quad & \quad \quad (f_D, f_E, f_F) \\
\quad & \quad \quad \quad \downarrow{f_D} \\
\quad & \quad \quad \quad B(D)
\end{array}
\]

So \( (f_D, f_E, f_F) \in \prod_{D \in D} \text{Hom}_C(A(D), B(D)) \times \text{Hom}_C(A(E), B(E)) \times \text{Hom}_C(A(F), B(F)) \) can be mapped to \( \text{Hom}_C(A(D), B(D)) \times \text{Hom}_C(A(D), B(D')) \times \text{Hom}_C(A(D), B(D')) \) in two natural ways. We can precompose with the structure maps: \( (f_D, f_E, f_F) \mapsto (f_E A(\epsilon), f_F A(\delta)) \) or postcompose: \( (f_D, f_E, f_F) \mapsto (B(\epsilon) f_D, B(\delta) f_D) \). Requiring these two maps to agree is asking for \( f_E A(\epsilon) = B(\epsilon) f_D \) and \( f_F A(\delta) = B(\delta) f_D \), that is, the object maps are chosen to consistently commute with the structure maps of the diagram. This is what we wanted.

This is nearly our objectwise hom! We need only that this is an object of \( \mathcal{C} \), so we must require \( \mathcal{C} \) to have equalizers or more generally, limits. So if \( \mathcal{C} \) has products and equalizers, we can define an objectwise hom on the category \( \mathcal{C}^D \) by

\[
\text{hom}_{\mathcal{C}^D}(A, B) = \text{equalizer} \left[ \prod_{D \in D} \text{Hom}_C(A(D), B(D)) \xrightarrow{d^*, \ d_*} \prod_{D \in D'} \text{Hom}_C(A(D), B(D')) \right]
\]

This is like the simplicial maps of \( sSet \) in that we have a map for each dot of \( \Delta \) and the maps commute with the structure maps of \( \Delta \). We have not tried to make this look like a diagram yet. It is just a consistent choice of maps defined on each object of the diagram that commute with all the arrows of the diagram.

We should note that all this work shows the construction of the set of natural transformations between the functors \( A \) and \( B \). That is, we’ve found the morphisms in the category \( \mathcal{C}^D \). Why didn’t we just state that the objectwise-hom was just the natural transformations? By going through the details we’ve gained information on when this set of natural transformations is actually an object of \( \mathcal{C} \): when \( \mathcal{C} \) has products and equalizers.
4 The Internal Hom

Recall that the category of simplicial sets is the functor category $\Delta^{op} \to \text{Set}$ so by the last section we know the levelwise maps between two simplicial sets form a set (an object of $\mathcal{C}$), but not necessarily an entire simplicial set. To get the simplicial structure on hom$_{sSet}(X_\bullet, Y_\bullet)$, more maps are needed. One must define a set for each $n \geq 0$ (for each object of $\Delta^{op}$) and check that the structure maps behave. Tried and true is the definition

$$[\text{Hom}_{sSet}(X_\bullet, Y_\bullet)]_n = \text{hom}_{sSet}(X_\bullet \times \Delta[n], Y_\bullet)$$

To help generalize this, we should note that $\Delta[n] = \text{hom}_{\Delta^{op}}(n, -)$ and recall that the simplicial product is levelwise so $[A_\bullet \times B_\bullet]_n = A_n \times B_n$.

There could be an example here. Really simple. Like explain the 1-level maps.

To generalize to our category $\mathcal{C}^D$, we need to define an object of $\mathcal{C}$ for each object $\ast \in D$. Let’s follow the lead of $sSet$ and define:

$$[\text{Hom}_{\mathcal{C}^D}(X_\bullet, Y_\bullet)](\ast) = \text{hom}_{\mathcal{C}^D}(X_\bullet \times \text{hom}_D(\ast, \ast), Y_\bullet)$$

So we will need to define what is meant by $X_\bullet \times \text{hom}_D(\ast, \ast)$. Since $D$ need not have interesting hom-sets, this is an object of $\mathcal{C}$ times a set for each object $D$ of $\mathcal{D}$. We want the product to be an object of $\mathcal{C}$. Let $K$ be a set. How could we define $\mathcal{C} \times K$? A naive guess may be to take $|K|$ copies of $\mathcal{C}$. We like products to be left adjoints of homs, so let’s try to think of it that way.

We have a functor $\mathcal{C} \to \text{Set} : C \mapsto \text{hom}_\mathcal{C}(C, -)$. If this functor has a left adjoint, it would look like a product (a functor $\mathcal{C} \times \text{Set} \to \mathcal{C}$) as we can see from the adjunction $\text{hom}_\mathcal{C}(C \times K, C') \cong \text{Hom}_{\text{Set}}(K, \text{hom}_\mathcal{C}(C, C'))$ for $C'$ an object of $\mathcal{C}$.

An element of $\text{Hom}_{\text{Set}}(K, \text{hom}_\mathcal{C}(C, C'))$ is a labelling of a bunch of maps $C \to C'$ by elements of $K$. This is like having $|K|$ maps from $C$ to $C'$. These must factor through $\bigsqcup K C \to C'$ by universal property of the coproduct. So if $\mathcal{C}$ has coproducts, we can identify the left adjoint as the coproduct. Now our naive guess is validated. The product is given by $C \times K = \bigsqcup K C$.

Thus we can identify $X_\bullet \times \text{hom}_D(\ast, \ast)$ as a new diagram in $\mathcal{C}$. Every object $D \in D$ is assigned an object $X(D) \times \text{hom}_D(\ast, D)$ of $\mathcal{C}$. We can think of this object as a coproduct of copies of $X(D)$ labelled by the maps in $\text{hom}_D(\ast, D)$. Let $d : D \to D'$ be a map of $\mathcal{D}$, then we have a map $X(d) : X(D) \to X(D')$ on the first factor of the product, and we may postcompose with $d$ on the second factor so that the map $\alpha \in \text{hom}_D(\ast, D)$ is mapped to $d \circ \alpha \in \text{hom}_D(\ast, D')$. Thus $d$ induces a map $X(d) \times d_\ast$ on the product so that $X_\bullet \times \text{hom}_D(\ast, \ast)$ has the form of a diagram in $\mathcal{C}$. By the previous section, it makes sense to talk about the maps between this diagram and $Y_\bullet$.

So the definition above is in fact a diagram in $\mathcal{C}$! We have defined an internal-hom:

$$[\text{Hom}_{\mathcal{C}^D}(X_\bullet, Y_\bullet)](\ast) = \text{hom}_{\mathcal{C}^D}(X_\bullet \times \text{hom}_D(\ast, -), Y_\bullet)$$

The right hand side is the objectwise-hom we defined in the last section. Using the characterization of the product above and the definition of objectwise-hom, we can expand this definition to:

$$\text{Hom}_{\mathcal{C}^D}(X_\bullet, Y_\bullet)(\ast) = \text{equalizer} \left( \bigsqcup_{D \in \mathcal{D}} \text{Hom}_\mathcal{C} \left( \bigsqcup_{\text{hom}_D(\ast, D)} X(D), Y(D) \right) \Rightarrow \bigsqcup_{D \to D'} \text{Hom}_\mathcal{C} \left( \bigsqcup_{\text{hom}_D(\ast, D)} X(D), Y(D') \right) \right)$$

Let’s check the details with our dumbbell diagram.

$$\begin{array}{ccc}
D & \xrightarrow{d} & D' \\
\downarrow & & \downarrow \\
 D' & & D'
\end{array}$$

First note that $(X_\bullet \times \text{hom}_D(D, -))(D) = X(D) \times \{id\} = X(D)_{id}$ and similarly $(X_\bullet \times \text{hom}_D(D, -))(D') = X(D')_{d}$. $(X_\bullet \times \text{hom}_D(D', -))(D) = \emptyset$, and $(X_\bullet \times \text{hom}_D(D', -))(D') = X(D')_{id}$. So for $D$, the set of
diagram maps of the top square is the objectwise hom of $C^D$ (which is an object of $C$), and similarly the set of diagram maps of the bottom square is the objectwise hom corresponding to $D'$. The map $d^*$ is induced by precomposition with $id \times d^*$ on each factor. (It maps the object $X(D)$ of $C$ labelled by $\alpha$ to the same object $X(D)$ labelled by $d^*(\alpha) = \alpha \circ d$.)

\[
\begin{array}{cccc}
X(D)_d & \text{diagram maps} & Y(D) \\
\downarrow \quad & & \downarrow \\
X(D')_d & \quad & Y(D') \\
\end{array}
\]

\[
\begin{array}{cccc}
\emptyset & \text{diagram maps} & Y(D) \\
\downarrow \quad & & \downarrow \\
X(D')_d & \quad & Y(D') \\
\end{array}
\]

Our dumbbell is looking more like a bench press. Let $(f_D, f_{D'})$ be an element of $\hom_{C^D}(X_\bullet \times \hom_{D^D}(D, \bullet), Y_\bullet)$, that is, $f_D : X(D)_d \to Y(D)$ and $f_{D'} : X(D')_d \to Y(D')$ such that $f_{D'} \circ (X(d) \times d_\ast) = Y(d) \circ f_D$. Then

\[
d^*(f_D, f_{D'}) = [\emptyset \to X(D)_d \xrightarrow{f_D} Y(D), X(D')_d \xrightarrow{id \times d^*} X(D')_d \xrightarrow{f_{D'}} Y(D')]
\]

We will not try to spell out the details for $sSet$. The mess would be overwhelmingly discouraging, and that is not our aim.

5 The Left Adjoint

So we have characterized when a diagram category has an internal-hom, and described this diagram in gory detail with respect to the underlying diagram. Now we will (try to) describe the left adjoint (a symmetric product) and what is necessary of $C$ in order to have it. This promises to be a nightmare, because we will consider the diagram maps into the diagram maps we just struggled to wrap our brains around.

Recall, what we wish to explore is the adjoint relationship:

\[
\hom_{C^D}(A_\bullet \otimes B_\bullet, C_\bullet) \cong \hom_{C^D}(A_\bullet, \hom_{C^D}(B_\bullet, C_\bullet))
\]

There will be a lot of symbol pushing here, and it may be nice to have a list of the properties we will use.

1. A map $X \to E$ into the equalizer $E$ of the diagram $A \xrightarrow{\alpha} B$ gives a unique map $X \to A$.
2. Dually, a map $C \to Y$ out of the coequalizer $C$ of the diagram $A \xrightarrow{\alpha} B$ gives a unique map $B \to Y$.
3. $\hom(\prod A_\alpha, B) \cong \prod \hom(A_\alpha, B)$
4. $\hom(A, \prod B_\alpha) \cong \prod \hom(A, B_\alpha)$
5. Left adjoints commute with colimits. Right adjoints commute with limits.
6. Therefore limits commute with limits. Colimits commute with colimits. Therefore coproducts commute and products commute.
Our description of the internal-hom for $C^D$ replaces the functor $B_\bullet : \mathcal{D} \to \mathcal{C}$ with a bifunctor $B(-) \times \text{hom}_D(\bullet , -) : \mathcal{D}^{op} \times \mathcal{D} \to \mathcal{C}$.

Let the string of isomorphisms commence.

We want to specify a map $\text{Hom}_{C^D}(A_\bullet , \text{Hom}_{C^D}(B_\bullet , C_\bullet ))$ and see if we can construct an adjoint map. For every $\triangledown \in \mathcal{D}$ we have

\[
\text{Hom}_{C^D}(A_\bullet , \text{Hom}_{C^D}(B_\bullet , C_\bullet ))(\triangledown) = \text{eq} \left[ \prod_{\triangledown \in \mathcal{D}} \text{Hom} \left( \prod_{\text{hom}_D(\triangledown, \star)} A(\star), \text{Hom}_{C^D}(B_\bullet , C_\bullet )(\star) \right) \right] \equiv \prod_{\star \to 0} \text{Hom} \left( \prod_{\text{hom}_D(\triangledown, \star)} A(\star), \text{Hom}_{C^D}(B_\bullet , C_\bullet )(\star) \right)
\]

\[
\subseteq \prod_{\triangledown \in \mathcal{D}} \text{Hom} \left( \prod_{\text{hom}_D(\triangledown, \star)} A(\star), \text{Hom}_{C^D}(B_\bullet , C_\bullet )(\star) \right)
\]

\[
= \prod_{\triangledown \in \mathcal{D}} \text{Hom} \left( \prod_{\text{hom}_D(\triangledown, \star)} A(\star), \text{eq} \left[ \prod_{D \in \mathcal{D}} \text{Hom} \left( \prod_{\text{hom}_D(\star, D)} B(D), C(D) \right) \right] \prod_{D \to D'} \text{Hom} \left( \prod_{\text{hom}_D(\star, D')} B(D), C(D') \right) \right)
\]

\[
\subseteq \prod_{\triangledown \in \mathcal{D}} \text{Hom} \left( \prod_{\text{hom}_D(\triangledown, \star)} A(\star), \prod_{D \in \mathcal{D}} \text{Hom} \left( \prod_{\text{hom}_D(\star, D)} B(D), C(D) \right) \right)
\]

\[
= \text{Hom} \left( \prod_{\triangledown \in \mathcal{D}} \prod_{\text{hom}_D(\triangledown, \star)} A(\star), \prod_{D \in \mathcal{D}} \text{Hom} \left( \prod_{\text{hom}_D(\star, D)} B(D), C(D) \right) \right)
\]

\[
= \prod_{D \in \mathcal{D}} \text{Hom} \left( \prod_{\triangledown \in \mathcal{D}} \prod_{\text{hom}_D(\triangledown, \star)} A(\star) \prod_{\text{hom}_D(\star, D)} B(D), C(D) \right)
\]

\[=
\]

Let the bifunctor $A(\star) \times \text{hom}_D(D, \star) = \prod_{\text{hom}_D(D, \star)} A(\star)$ be denoted by $\tilde{A}(D, \star) : \mathcal{D}^{op} \times \mathcal{D} \to \mathcal{C}$.

We have $\tilde{A} \otimes \hat{B}(\cdot) = \text{and } A \otimes \tilde{B}(\cdot)$.

Define a map $\phi : A(\star) \times \text{hom}_D(X, \star) \otimes \text{hom}_D(Y, \star) \to A(\star) \otimes \text{hom}_D(X, Y) \times \text{hom}_D(X, Y)$ by $\phi(a \times \alpha \otimes \beta) = (a \times \beta \otimes \beta \circ \alpha)$.