Domain Theory

Part 3: Constructions on Cpos

Dr. Liam O'Connor

based on material from

Graham Hutton, Dana Scott, Joseph E. Stoy, Carl Gunter, Glynn Winskel

March 11, 2024

1 Introduction

In the last lecture, we were introduced to the theory of *complete partial orders*, or cpos, which are the mathematical structure that underlies all of our semantic domains.

However, our semantic domains are not just flat domains like $\mathbb{Z}_\bot$, we also make many *constructions* on our domains, such as the cartesian product or disjoint union, or (continuous) *functions* between domains. In previous lectures, we glossed over the assumption that these constructions are themselves cpos. In this lecture, we will formalise these three constructions and give an overview of their properties, sneaking in some basic category theory along the way.

2 Products

Recall we previously showed that if A and B are posets, then

$$A \times B \triangleq \{(a, b) \mid a \in A \land b \in B\}$$

is a poset under the ordering:

$$(a, b) \sqsubseteq_{A \times B} (a', b') \text{ iff } a \sqsubseteq_A a' \land b \sqsubseteq_B b'$$

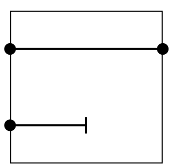
It also has a bottom value if A and B do:

$$\bot_{A \times B} = (\bot_A, \bot_B)$$

In an exercise you may also have shown that $A \times B$ is also a cpo if A and B are cpos, with definition for lubs:

$$\bigsqcup X \;=\; \left(\bigsqcup\{x \mid \exists y.\ (x, y) \in X\}, \bigsqcup\{y \mid \exists x.\ (x, y) \in X\}\right)$$

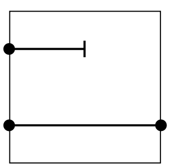
This definition can be made a little more comprehensible by defining the two projection operators for pairs:



π_1

$$\begin{array}{ll} \pi_0 : A \times B \to A & \pi_1 : A \times B \to B \\ \pi_0(x, y) = x & \pi_1(x, y) = y \end{array}$$

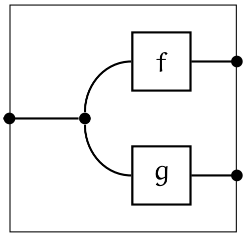
$$\bigsqcup X \;=\; (\bigsqcup\{\pi_0(x) \mid x \in X\}, \bigsqcup\{\pi_1(x) \mid x \in X\})$$



π_2

Destructing pairs is captured by these projection functions, and construction of pairs is captured by the *split* function: If $f : A \to B$ and $B \to C$, then *split*, written $\langle f, g \rangle$, is defined as:

$$\langle f, g \rangle : A \to B \times C$$
$$\langle f, g \rangle \ a = (f(a), g(a))$$



Continuity

Theorem: π_0 and π_1 are continuous. Proof is straightforward and omitted.

Theorem: Assuming f and g are continuous, $\langle f, g \rangle$ is continuous. Proof in two parts:

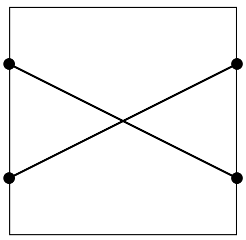
- $\langle f, g \rangle$ *is monotonic*. Let $x \sqsubseteq y \in A$. Then:

$$\begin{array}{rcll} \langle f, g \rangle \ x & = & (f(x), g(x)) & \text{(def)} \\ & \sqsubseteq & (f(y), g(y)) & \text{(monotonicity of f,g)} \\ & = & \langle f, g \rangle \ y & \text{(def)} \end{array}$$

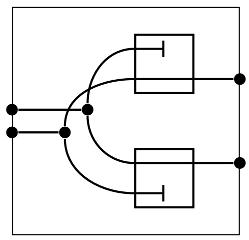
- $\langle f, g \rangle$ *preserves lubs of directed sets*. Let $X \subseteq A$ be directed. Then:

$$\begin{array}{rcll} \langle f, g \rangle \ (\bigsqcup X) & = & (f(\bigsqcup X), g(\bigsqcup X)) & \text{(def)} \\ & = & (\bigsqcup\{f(x) \mid x \in X\}, \bigsqcup\{g(x) \mid x \in X\}) & \text{(continuity of f,g)} \\ & = & \bigsqcup\{\langle f, g \rangle \ x \mid x \in X\} & \text{(def)} \end{array}$$

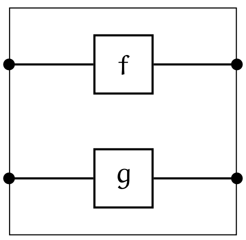
These three "primitive" functions are sufficient to derive other functions on products. It is easy to see the correctness of these functions by thinking in terms of pictures.



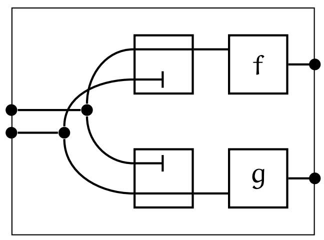
$$swap : A \times B \to B \times A$$
$$swap = \langle \pi_1, \pi_0 \rangle$$



Given $f : A \to C$ and $g : B \to D$, we overload the $\times$ operator to denote the combined function $f \times g : A \times B \to C \times D$, given below:

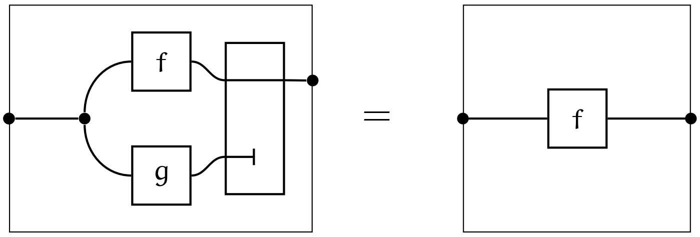


$$f \times g : A \times B \to C \times D$$
$$f \times g = \langle f \circ \pi_0, g \circ \pi_1 \rangle$$

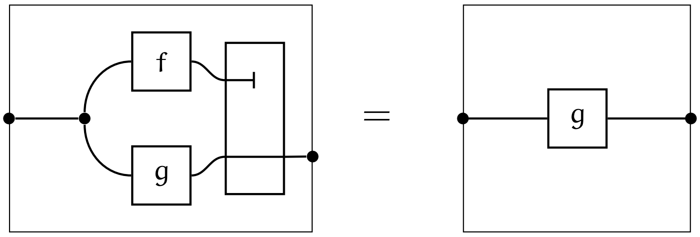


2.1 Universality

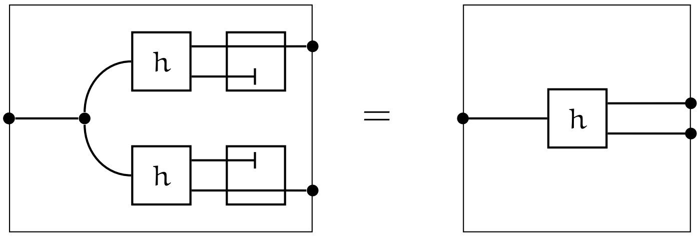
$$\pi_0 \circ \langle f, g \rangle = f$$



$$\pi_1 \circ \langle f, g \rangle = g$$



$$\langle \pi_0 \circ h, \pi_1 \circ h \rangle = h$$



These properties are together equivalent to the universal property for products:

$$(\pi_0 \circ h = f \wedge \pi_1 \circ h = g) \text{ iff } h = \langle f, g \rangle$$

The universal property is easy to remember as a commuting diagram on the category **Cpo**:

$$\begin{tikzcd} & A \arrow[dl, "f"'] \arrow[d, dashed, "\langle f,g \rangle"] \arrow[dr, "g"] & \\ B & B \times C \arrow[l, "\pi_0"] \arrow[r, "\pi_1"'] & C \end{tikzcd}$$

Categories

A category C consists of a collection of things (*objects*) and a collection of arrows between these things (*morphisms*). If there is a morphism $A \xrightarrow{f} B$ and a morphism $B \xrightarrow{g} C$, we also have the composed morphism $A \xrightarrow{g \circ f} C$. This composition operator must be associative. Additionally each object X has an identity morphism $X \xrightarrow{\mathrm{id}_X} X$, such that for an arrow $X \xrightarrow{a} Y$, $a \circ \mathrm{id}_X = \mathrm{id}_Y \circ a = a$. Some examples of categories include:

- **Set**, where objects are sets, morphisms are functions between these sets, composition is function composition ($(g \circ f)(x) \triangleq g(f(x))$) and identity is just $\lambda x.x$.
- **Rel**, where objects are sets, morphisms are relations between these sets, composition is relational composition ($g \circ f \triangleq \{(a, b) \mid \exists i.\ (a, i) \in f \wedge (i, b) \in g\}$) and identity is the diagonal relation (equality).

- **Cpo**, where objects are cpos, morphisms are continuous functions between these cpos. Function compositions preserve continuity (proof below), and identity functions are continuous.

- **Cat**, where objects are *themselves* categories, and the morphisms are functors: structure preserving maps between categories. A functor from a category **C** to a category **D** is a total function F that maps objects of **C** to objects of **D**, and arrows of **C** to arrows of **D** such that:
  - For each $A \xrightarrow{m} B$ in **C**, we have a morphism $F(A) \xrightarrow{F(m)} F(B)$ in **D**.
  - For each object A in **C**, the equation $F(\mathsf{id}_A) = \mathsf{id}_{F(A)}$ holds in **D**.
  - For a pair of morphism $A \xrightarrow{f} B \xrightarrow{g} C$ in **C**, the equation $F(g \circ f) = F(g) \circ F(f)$ holds in **D**.

When working in a category, we can state many theorems compactly using commutative diagrams, as above. In these diagrams, the objects are *vertices* in the diagram, and the morphisms are the *arrows* in the diagram (we typically omit identity morphisms). Looking at the above diagram, we can equate the two paths from A to B and the two paths from A to C, and produce the equations above.

This universal property is called such because *all* the familiar properties of products follow from it (thinking in pictures helps to see how)[1]:

1. $\pi_0 \circ (f \times g) = f \circ \pi_0$
2. $\pi_1 \circ (f \times g) = g \circ \pi_1$
3. $(f \times g) \circ \langle h, i \rangle = \langle f \circ h, g \circ i \rangle$
4. $\langle f, g \rangle \circ h = \langle f \circ h, g \circ h \rangle$
5. $\mathsf{id}_A \times \mathsf{id}_B = \mathsf{id}_{A \times B} \quad (*)$
6. $(f \circ g) \times (h \circ i) = (f \circ h) \times (g \circ i) \quad (**)$

These last two statements $(*)$ and $(**)$ show that $\times$ is a bifunctor on the category **Cpo**, i.e. a functor **Cpo** $\times$ **Cpo** $\to$ **Cpo**.

2.2 Isomorphisms

Definition

Recall an isomorphism between sets consists of a total function $f : X \to Y$ and an inverse $f^{-1} : Y \to X$ such that:

$$f^{-1} \circ f = \mathsf{id}_X \quad \text{and} \quad f \circ f^{-1} = \mathsf{id}_Y$$

Two sets X, Y are isomorphic (written $X \simeq Y$) iff an isomorphism exists between them.

Cpos form a commutative monoid "up to isomorphism" under $\times$ and $\mathbf{1}$ (the cpo consisting of only one element $\perp$). That is, for all cpos A, B and C:

- $A \times \mathbf{1} \simeq A$
- $A \times (B \times C) \simeq (A \times B) \times C$

[1] Indeed, the universal property *characterises* products up to isomorphism.

- $A \times B \simeq B \times A$

Let $A \rightharpoonup B$ denote just the continuous functions from cpo A to cpo B:

$$A \rightharpoonup B \;\triangleq\; \{f : A \to B \mid f \text{ is continuous}\}$$

Another isomorphism we get is between pairs of continuous functions and continuous functions that output pairs:

$$(A \rightharpoonup B) \times (A \rightharpoonup C) \;\simeq\; (A \rightharpoonup (B \times C))$$

Proof follows from the universal property after setting up the isomorphism with functions $f(g_0, g_1) = \langle g_0, g_1 \rangle$ and $f^{-1}(h) = (\pi_0 \circ h, \pi_1 \circ h)$.

Because we have the functor $\times$ on cpos that satisfies the properties of products described above, the category **Cpo** therefore has binary products.

3 Functions

In the previous lecture, specifically in the factorial example, we already implicitly assumed that the space of continuous functions on cpos $A \rightharpoonup B$ is itself a cpo. In this section, we will formalise this construction.

As previously stated, $A \rightharpoonup B$ denotes continuous functions from A to B:

$$A \rightharpoonup B \;\triangleq\; \{f : A \to B \mid f \text{ is continuous}\}$$

If A and B are cpos, the set $A \rightharpoonup B$ is a cpo under the *pointwise* ordering:

$$f \sqsubseteq g \;\text{ iff }\; \forall a \in A.\, f(a) \sqsubseteq g(a)$$

The intuition of this ordering in terms of information is that we increase the information content of a function overall by increasing the information content of any (or many) argument values. To prove this, we must show:

1. $A \rightharpoonup B$ *has a least element.* $\bot_{A \rightharpoonup B}$ is the constant function that returns $\bot_B$, i.e. $\lambda a.\, \bot_B$.
2. $\bigsqcup X$ *exists for all directed* $X \subseteq A \rightharpoonup B$. Our lub operator $\bigsqcup X$ can be just:

$$\Phi(a) = \bigsqcup\{f(a) \mid f \in X\}$$

   a) $\bigsqcup\{f(a) \mid f \in X\}$ *exists in* B. We want to show that the set $\{f(a) \mid f \in X\}$ is directed. Take two values $g(a), h(a) \in \{f(a) \mid f \in X\}$. Since X is directed, there exists a function $k \in X$ such that $g \sqsubseteq k$ and $h \sqsubseteq k$. By applying the definition of $\sqsubseteq$ for functions above, we have $g(a) \sqsubseteq k(a)$ and $h(a) \sqsubseteq k(a)$. Thus $k(a)$ is an upper bound of these two values $g(a), h(a)$, and thus $\{f(a) \mid f \in X\}$ is directed, and thus its lub exists since B is a cpo.

   b) $\Phi(a) = \bigsqcup\{f(a) \mid f \in X\}$ *is continuous.*

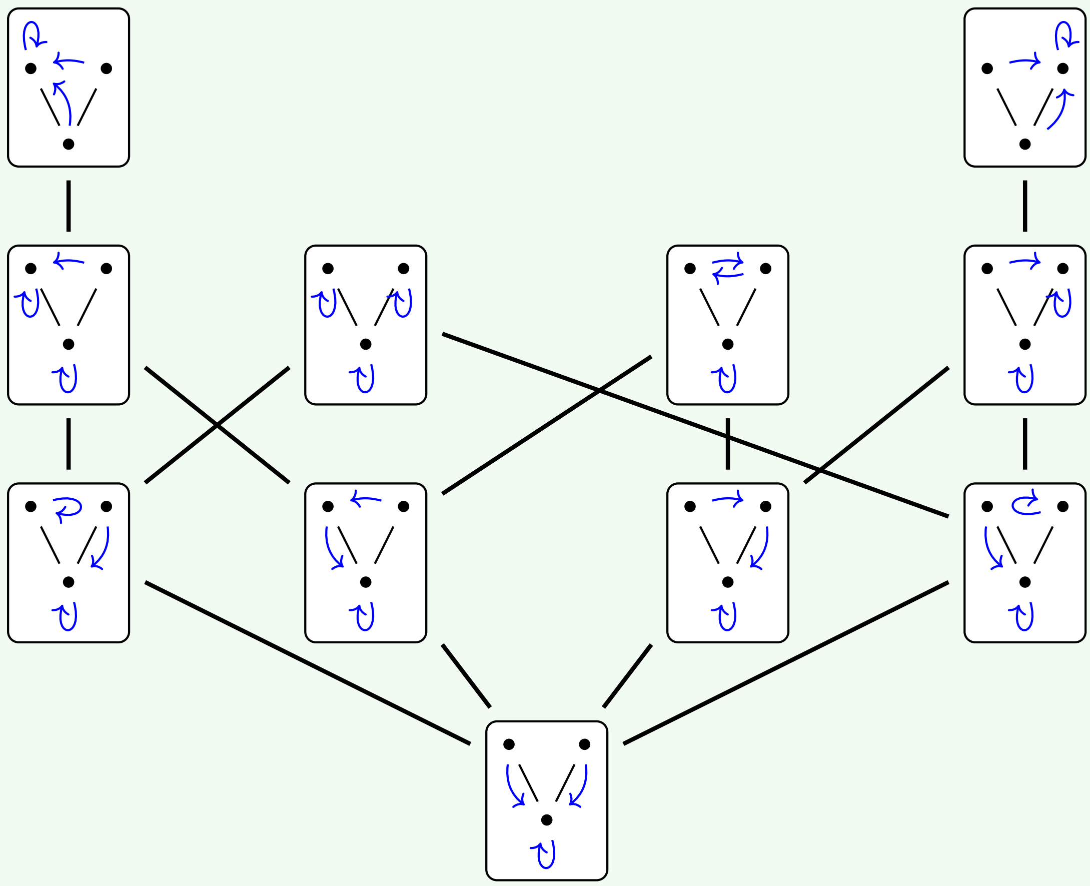
$$\begin{aligned}
\Phi(\bigsqcup Y) &= \bigsqcup\{f(\bigsqcup Y) \mid f \in X\} && \text{(defn. } \Phi\text{)}\\
&= \bigsqcup\{\bigsqcup\{f(y) \mid y \in Y\} \mid f \in X\} && \text{(f is continuous)}\\
&= \bigsqcup\{\bigsqcup\{f(y) \mid f \in X\} \mid y \in Y\} && \text{(swap lubs)}\\
&= \bigsqcup\{\Phi(y) \mid y \in Y\} && \text{(defn. } \Phi\text{)}
\end{aligned}$$

   Monotonicity is similar to prove.

c) Φ *is an upper bound for* $X \subseteq A \rightarrow B$. Let $g \in X$. Then for any $a \in A$ we have $g(a) \in \{f(a) \mid f \in X\}$. Now, by 2a) and lubs, we have $g(a) \sqsubseteq \bigsqcup\{f(a) \mid f \in X\}$, which by the definition of Φ gives $g(a) \sqsubseteq \Phi(a)$. Hence, by the definition of $\sqsubseteq$ on functions, we can conclude $g \sqsubseteq \Phi$.

d) Φ *is the least upper bound for* X. Let $g \in A \rightarrow B$ be an upper bound for $X \subseteq A \rightarrow B$. Then (using $\sqsubseteq$ on functions) $g(a)$ is an upper bound for $\{f(a) \mid f \in X\}$ for all $a \in A$. Now, by 2a) and lubs, we have $\bigsqcup\{f(a) \mid f \in X\} \sqsubseteq g(a)$, i.e. $\Phi(a) \sqsubseteq g(a)$, which, by the definition of $\sqsubseteq$ on functions, allows us to conclude $\Phi \sqsubseteq g$.

Example (on $\mathbb{B}_\perp \to \mathbb{B}_\perp$)



At each step "upwards" on this cpo, one arrow that ends in $\perp$ now points to a non-$\perp$ value (moving up the cpo $\mathbb{B}_\perp$).

3.1 Primitive functions for functions

We will now introduce the primitive functions for functions. We have *apply*, the function that applies a given function to a given argument:

$$\begin{aligned}&\mathit{apply} : ((A \rightarrow B) \times A) \rightarrow B\\ &\mathit{apply}(f, a) = f(a)\end{aligned}$$

And we have the function *curry*, which transforms a function that takes two arguments all at once (as a product) into a function that takes the arguments one at a time. That is, if $f : A \times B \rightarrow C$ then:

$$\begin{aligned}&\mathit{curry}(f) : A \rightarrow (B \rightarrow C)\\ &\mathit{curry}(f)(a)(b) = f(a, b)\end{aligned}$$

Note: As can be seen from the use of the $\rightarrow$ symbol, these functions are both continuous assuming their argument functions are continuous. Proof of this is omitted.

3.2 Universal property for functions

$$\begin{tikzcd}
A \times B \arrow[r, "f"] \arrow[d, dashed, "curry(f)\times id_B"'] & C \\
(B \rightarrow C) \times B \arrow[ur, "apply"'] &
\end{tikzcd}$$

This universal property establishes an important isomorphism:

$$(A \times B) \rightarrow C \quad \simeq \quad A \rightarrow (B \rightarrow C)$$

3.2.1 Functor of functions

The operator $\rightarrow$ we have introduced so far is the object mapping for a binary functor:

$$(\rightarrow) : \mathbf{Cpo}^{op} \times \mathbf{Cpo} \to \mathbf{Cpo}$$

The category $\mathbf{Cpo}^{op}$ is the dual category to $\mathbf{Cpo}$.

Duality

The dual of a category $\mathbf{C}$, written $\mathbf{C}^{op}$, is a category with the same objects as $\mathbf{C}$ but the arrows are reversed, that is, for each morphism $A \xrightarrow{m} B$ in $\mathbf{C}$, there is a morphism $B \xrightarrow{m} A$ in $\mathbf{C}^{op}$.

Thus, a morphism $A \xrightarrow{m} B$ in the category $\mathbf{Cpo}^{op}$ is a continuous function $m : B \rightarrow A$. Thus, the morphism mapping for our functor must take two functions $f : B \rightarrow A$ (the morphism from $\mathbf{Cpo}^{op}$) and $g : C \rightarrow D$ (the morphism from $\mathbf{Cpo}$), and produce $(A \rightarrow C) \rightarrow (B \rightarrow D)$. As with products, we will overload the $\rightarrow$ notation for this mapping as well. Given $f : B \rightarrow A$ and $g : C \rightarrow D$, we have:

$$f \rightarrow g : (A \rightarrow C) \rightarrow (B \rightarrow D)$$
$$f \rightarrow g = curry(g \circ apply \circ (id \times f))$$

(or, operationally:)

$$(f \rightarrow g)(h) = g \circ h \circ f$$

The two functor laws follow as a consequence of the universal property for functions above:

1. $id_A \rightarrow id_B = id_{A \rightarrow B}$
2. $(f \circ g) \rightarrow (h \circ i) = (g \rightarrow h) \circ (f \rightarrow i)$

Thus, our category $\mathbf{Cpo}$ has exponentials.

4 Sums

We shall define a sum construct $A + B$, denoting a disjoint union of two cpos with a new, dedicated $\perp$ value:

$$A + B = \{(0, a) \mid a \in A\} \cup \{(1, b) \mid b \in B\} \cup \{\perp_{A+B}\}$$

This construct is a cpo under the ordering:

$$x \sqsubseteq y \;\text{ iff } \begin{cases} \text{true} & \text{if } x = \perp_{A+B} \\ a \sqsubseteq a' & \text{if } x = (0, a) \text{ and } y = (0, a') \\ b \sqsubseteq b' & \text{if } x = (1, b) \text{ and } y = (1, b') \\ \text{false} & \text{otherwise} \end{cases}$$

Intuitively, this says that the ordering on $A + B$ is the same as A and B separately, except that $\perp_{A+B}$ is less than anything.

4.1 Primitive Functions on Sums

For *constructing* sums, we use two primitive constructor functions, *inl* and *inr*:

$$\begin{aligned} &\mathit{inl} : A \rightarrow (A + B) \\ &\mathit{inl}(x) = (0, x) \end{aligned}$$

$$\begin{aligned} &\mathit{inr} : B \rightarrow (A + B) \\ &\mathit{inr}(y) = (1, y) \end{aligned}$$

For *destructing* sums, we use the *case* function $[f, g] : (A + B) \rightarrow C$, made from functions $f : A \rightarrow C$ and $g : B \rightarrow C$. This function is defined by:

$$\begin{aligned} &[f, g] : (A + B) \rightarrow C \\ &[f, g](x) = \begin{cases} f(a) & \text{if } x = (0, a) \\ g(b) & \text{if } x = (1, b) \\ \perp_C & \text{if } x = \perp_{A+B} \end{cases} \end{aligned}$$

With these, we can define the morphism mapping for the sum bifunctor $(+) : \mathbf{Cpo} \times \mathbf{Cpo} \to \mathbf{Cpo}$. Just as with products and functions, we will overload the $+$ notation. Given functions $f : A \rightarrow C$ and $g : B \rightarrow D$, we have:

$$\begin{aligned} &f + g : (A + B) \rightarrow (C + D) \\ &f + g = [\mathit{inl} \circ f, \mathit{inr} \circ g] \end{aligned}$$

4.2 Weak Universal Properties

We have theorems $[f, g] \circ \mathit{inl} = f$ and $[f, g] \circ \mathit{inr} = g$, i.e.:

$$\begin{tikzcd} & C & \\ A \arrow[ur, "f"] \arrow[r, "\mathit{inl}"'] & A + B \arrow[u, "{[f,g]}"] & B \arrow[ul, "g"'] \arrow[l, "\mathit{inr}"] \end{tikzcd}$$

Note that this diagram is the dual of the product diagram above. That is because sums are dual to products. However, because of our additional $\perp$ value, our sums are only *weakly* universal. For example, consider this scenario with the cpos $\mathbf{2} = \{\top, \perp\}$ and $\mathbf{1} = \{\perp\}$:

$$\begin{tikzcd}
 & \mathbf{2} & \\
\mathbf{1} \arrow[ur, "\lambda x.\ \top"] \arrow[r, "inl"'] & \mathbf{1}+\mathbf{1} \arrow[u, "h"] & \mathbf{1} \arrow[ul, "\lambda x.\ \top"'] \arrow[l, "inr"]
\end{tikzcd}$$

In this case, setting the function $h = \lambda x.\ \top$ makes the diagram commute, but $(\lambda x.\ \top) \neq [\lambda x.\ \top, \lambda x.\ \top]$ due to the different handling of $\perp_{\mathbf{1}+\mathbf{1}}$. Thus we have only a *weak* universal property:

$$(h \circ inl = f \wedge h \circ inr = g) \;\text{ iff }\; [f, g] \sqsubseteq h$$

This also means that we don't have a lot of our expected isomorphisms, e.g:

1. $A + (B + C) \not\simeq (A + B) + C$
2. $(A \rightarrow C) \times (B \rightarrow C) \not\simeq (A + B) \rightarrow C$

So, our category **Cpo** has only *local* sums.

5 Strict Constructions

When giving a semantics to a call-by-name language, the constructions $\rightarrow$, $\times$ and $+$ are exactly what we want. To properly capture call-by-value languages or strict language constructs, however, we also need strict versions of these constructions.

Definitions

Given cpos A and B, then the following are all cpos, with evident orderings:

- $A \circ\!\!\rightarrow B = \{f \in A \rightarrow B \mid f(\perp) = \perp\}$
- $A \otimes B = \{(a, b) \in A \times B \mid a \neq \perp \wedge b \neq \perp\} \cup \{\perp_{A \otimes B}\}$
- $A \oplus B = \{(t, x) \in A + B \mid x \neq \perp\} \cup \{\perp_{A \oplus B}\}$

The operators $\otimes$ and $\oplus$ are often called "smash" product and sum, because the two $\perp$ values are "smashed" together into one $\perp$ value.

The strict cpo constructions satisfy all the usual isomorphisms, including:

- $(A \circ\!\!\rightarrow B) \times (A \circ\!\!\rightarrow C) \;\simeq\; A \circ\!\!\rightarrow (B \otimes C)$
- $(A \otimes B \circ\!\!\rightarrow C) \;\simeq\; A \circ\!\!\rightarrow (B \circ\!\!\rightarrow C)$
- $(A \circ\!\!\rightarrow C) \times (B \circ\!\!\rightarrow C) \;\simeq\; (A \oplus B) \circ\!\!\rightarrow C$

Fact

The category $\mathbf{Cpo}_\perp$ of cpos and *strict* continuous functions between them has products, exponentials *and* sums.

The *lifting* operator $(\cdot)_\perp$, which we have seen before when applied to flat domains, adds a new $\perp$ value to a cpo. With this operator, we can relate the lazy and strict constructions we have introduced today via isomorphism:

- $A \rightarrow B \simeq A_\perp \circ\!\!\rightarrow B$
- $A + B \quad \simeq A_\perp \oplus B_\perp$
- $(A \times B)_\perp \simeq A_\perp \otimes B_\perp$

Exercises

1. Show using the universal property that $\langle f, g\rangle \circ h = \langle f \circ h, g \circ h\rangle$.
2. In the style of my diagram on page 6, draw the cpo $3 \rightarrow 3$, where 3 is the cpo $\{0, 1, 2\}$ ordered by $\leqslant$.
3. Express the function $\mathit{juggle}(a, (b, c)) = ((a, b), c)$ using just our primitive combinators.
4. What is the common (functional programming) name for $\mathit{apply} \circ (f \times \mathsf{id})$?
5. Show that *apply* is continuous.
   Hint: A function $f : A \times B \rightarrow C$ is continuous iff it is continuous in each argument separately, i.e., iff $\forall a \in A.\ f(a, \cdot) : B \rightarrow C$ is continuous, and $\forall b \in B.\ f(\cdot, b) : A \rightarrow C$ is continuous.

Glossary

bifunctor A binary functor, i.e. a functor from two categories to one. Equivalently, a bifunctor is a functor from the product category $\mathbf{C} \times \mathbf{C}$. 4, 8

call-by-name A call-by-name programming language is a language which does not evaluate function arguments before evaluating the function body. Notably, this means that functions may not be strict, i.e. $f(\perp)$ may not be $\perp$. 9

call-by-value A call-by-value programming language is a language which evaluates all function arguments before evaluating a function body. Notably, this means that semantically all functions are strict, i.e. $f(\perp) = \perp$. 9

category A mathematical structure that resembles a (multi-)graph consisting of a class of *objects*, a class of *arrows* or *morphisms* between objects, an associative arrow composition operator and an identity morphism for each object. 1, 3–5, 7, 10, 11

commutative A monoid is commutative if its associative operation also satisfies commutativity, i.e. $a \bullet b = b \bullet a$. 4

composition In a category, the *composition* $f \circ g$ is an arrow $A \xrightarrow{f \circ g} C$ made from the arrows $B \xrightarrow{f} C$ and $A \xrightarrow{g} B$. This operator must be associative. 3, 10

dual The dual of a category $\mathbf{C}$, written $\mathbf{C}^{op}$, is a category with the same objects as $\mathbf{C}$ but the arrows are reversed, that is, for each morphism $A \xrightarrow{m} B$ in $\mathbf{C}$, there is a morphism $B \xrightarrow{m} A$ in $\mathbf{C}^{op}$. 7, 9, 11

exponential Exponentials are the category-theoretic generalisation of *functions*. 7

functor A *functor* is a structure-preserving map between categories. Specifically, a functor from a category $\mathbf{C}$ to a category $\mathbf{D}$ is a total function F that maps objects of $\mathbf{C}$ to objects of $\mathbf{D}$, and arrows of $\mathbf{C}$ to arrows of $\mathbf{D}$ such that:

- For each $A \xrightarrow{m} B$ in $\mathbf{C}$, we have a morphism $F(A) \xrightarrow{F(m)} F(B)$ in $\mathbf{D}$.
- For each object A in $\mathbf{C}$, the equation $F(\mathsf{id}_A) = \mathsf{id}_{F(A)}$ holds in $\mathbf{D}$.
- For a pair of morphism $A \xrightarrow{f} B \xrightarrow{g} C$ in $\mathbf{C}$, the equation $F(g \circ f) = F(g) \circ F(f)$ holds in $\mathbf{D}$.

. 4, 5, 7, 10, 11

identity In a category, an *identity* morphism id_X is associated with each object X such that, for an arrow $X \xrightarrow{f} Y$, $f \circ \mathsf{id}_X = \mathsf{id}_Y \circ f = f$. 3, 4, 10

isomorphism An isomorphism between sets consists of a total function $f : X \to Y$ and an inverse $f^{-1} : Y \to X$ such that:

$$f^{-1} \circ f = \mathsf{id}_X \quad \text{and} \quad f \circ f^{-1} = \mathsf{id}_Y$$

Two sets X, Y are isomorphic (written $X \simeq Y$) iff an isomorphism exists between them . 4, 5, 7, 9–11

monoid A monoid $(S, \bullet, \iota)$ is an algebraic structure consisting of a set S and an associative operation $\bullet$ such that for all $x \in S$, $x \bullet \iota = \iota \bullet x = x$. 4, 10

sum The dual to products, also called coproducts. A *sum* can be thought of as the category-theoretic generalisation of a disjoint union . 8, 9

universal property Also known as the unique extension property, the *universal property* of a particular construction fully characterises the construction up to isomorphism. This means that, once we have the universal property, we can "forget" the exact "implementation" of our construction and reason purely abstractly instead. 3–5, 7, 9, 10