Teoría de Dominios y Modelos Denotacionales

Universidad de Málaga

Curso académico 2018-2019

Prof: Alicia Tocino Sánchez

1. Dcpos y dcpos punteados

Profesora: Alicia Tocino Sánchez. Curso Académico 2018-2019.

Definición 1.1. *Un conjunto P con una relación binaria R es* ***parcialmente ordenado*** *si se cumplen las siguientes condiciones:*

(1) *xRx para cada $x \in P$ (reflexiva)*
(2) *si xRy, yRz entonces xRz para cada $x, y, z \in P$ (transitiva)*
(3) *si xRy, yRx entonces $x = y$ para cada $x, y \in P$ (antisimétrica)*

Un conjunto es ***totalmente ordenado*** *si todos los pares de elementos de P son comparables mediante la relación R.*

Notación 1.2. *La relación R se puede expresar de varias formas: xRy, $x \sqsupseteq y$, $x \leq y$, (x, y). A un conjunto parcialmente ordenado se le llamará* ***poset****.*

Ejemplo 1.3. *Si $P = 1, 2, 3$ es un conjunto entonces $R = \{(1,1), (2,2), (3,3), (1,2), (2,3), (1,3)\}$ es un ejemplo de orden parcial en P.*

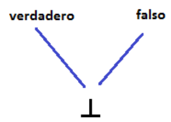
Los conjuntos finitos parcialmente ordenados (poset) se pueden representar mediante **diagramas de Hasse**. Los segmentos unen elementos relacionados del conjunto de modo que es menor el situado en la posición inferior. Se suprimen los segmentos que corresponden a la propiedad transitiva (es decir, si $a \leq b$ y $b \leq c$ entonces no aparecerá el segmento correspondiente a $a \leq c$).

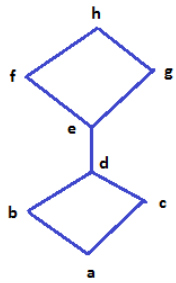
Para evitar preocuparse por resultados no definidos (o indefinidos) se introduce un valor especial, $\perp$, llamado **bottom**, que representa un error, un cálculo que no termina...

Observación 1.4. *Observemos que los conjuntos $S = \{1, 2, 3\}$ y $T = \{1, 3, 2\}$ son iguales. Sin embargo, como listas en un ordenador no serían iguales.*

Ejemplo 1.5.

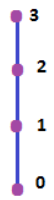
(1) *El siguiente diagrama, verifica que $\perp \leq$ verdadero, $\perp \leq$ falso y verdadero y falso no se pueden comparar.*





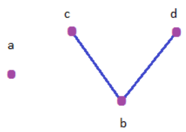
(2) *El siguiente diagrama, verifica que b y c no están relacionados, f y g no están relacionados, $b \leq g$ por la propiedad transitiva.*

(3) *El siguiente diagrama, representa un conjunto totalmente ordenado*



(4) *Con un diagrama similar al anterior se ve que $(\mathbb{N}, \leq)$ es un conjunto totalmente ordenado.*

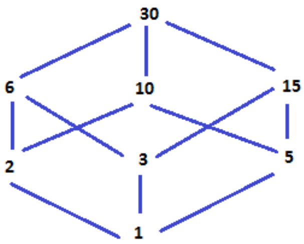
(5) *El siguiente diagrama, representa un conjunto parcialmente ordenado.*

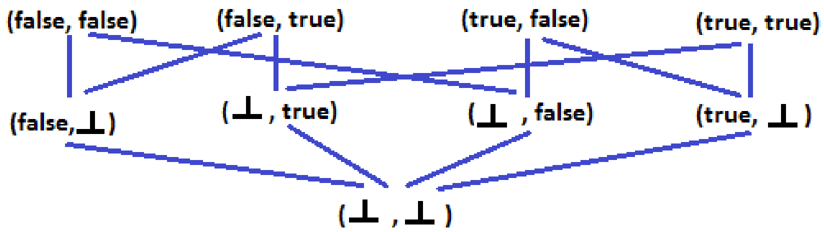


(6) *El conjunto de naturales ordenados por la relación de divisibilidad es parcialmente ordenado. El diagra de Hasse de los divisores de 30, $D(30)$, es el siguiente:*

(7) *Consideramos $\mathbb{B} = \{true, false\}$ y $\mathbb{B}_\perp = \{\perp, true, false\}$. En el producto cartesiano existe el siguiente orden parcial:*

Observación 1.6. *Si A y B son posets, entonces $A \times B = \{(a, b) \,|\, a \in A,\, b \in B\}$ también es un poset con la relación $(a, b) \leq (a', b')$ si y sólo si $a \leq b$ y $a' \leq b'$.*

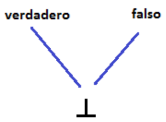




Definición 1.7. *Sea* $(P, \leq)$ *un poset. Un elemento* $x \in P$ *es una* ***cota superior*** *de un subconjunto* $A \subseteq P$ *si* $a \leq x$ *para cada* $a \in A$*. Se dice que* $x \in P$ *es una* ***cota inferior*** *de* $A \subseteq P$ *si* $x \leq a$ *para cada* $a \in A$*.*

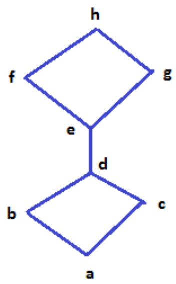
Ejemplo 1.8.

(1) *Del siguiente diagrama, consideramos los siguientes subconjuntos:*



- $A = \{falso\}$*, cota superior* $= \{falso\}$*, cota inferior* $= \{\perp, falso\}$*.*
- $A = \{verdadero, falso\}$*, cota superior* $= \emptyset$*, cota inferior* $= \{\perp\}$*.*

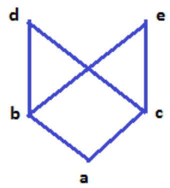
(2) *Del siguiente diagrama, consideramos el subconjunto* $A = \{f, g\}$*, cota superior* $= \{h\}$*, cota*



inferior $= \{\perp\}$.

(3) *Del conjunto de los naturales con el orden usual* $(\mathbb{N}, \leq)$ *consideramos el subconjunto* $A = \{\textit{números pares}\}$, *cota superior* $= \emptyset$, *cota inferior* $= \{0\}$.

(4) *Del siguiente diagrama, consideramos el subconjunto* $A = \{b, c\}$, *cota superior* $= \{d, e\}$,



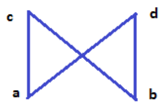
cota inferior $= \{a\}$.

Definición 1.9. *Sea* $A \subseteq P$ *un subconjunto de un poset. Si el conjunto de cotas superiores de* A *tiene mínimo, es decir, si existe* $x \in P$ *cota superior de* A *tal que* $x \leq b$ *para cada* b *en el conjunto de cotas superiores de* A, *se dice que* x *es el* ***supremo*** *de* A.

Notación 1.10. *Denotamos por* $\vee A$ *al supremo de* A. *Si* $A = \{x, y\}$ *entonces* $\vee A = x \vee y$.

Ejemplo 1.11.

(1) *Del siguiente diagrama, consideramos el subconjunto* $A = \{a, b\}$, *cotas superiores* $= \{c, d\}$.

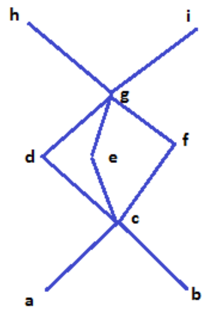


Como $\{c, d\}$ *no tiene mínimo, entonces no existe el supremo.*

(2) *Del siguiente diagrama, consideramos el subconjunto* $A = \{2\}$, *cotas superiores* $= \{2, 3\}$,

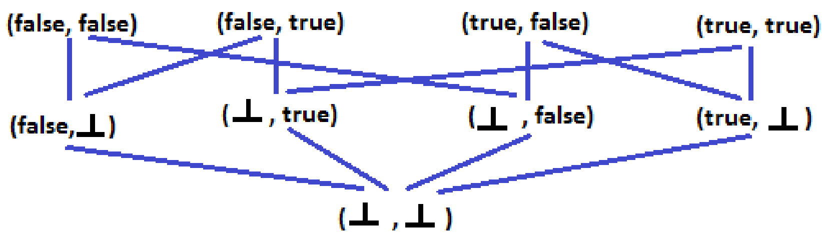
supremo $= \{2\}$.

(3) *Del siguiente diagrama,*



consideramos el subconjunto $A = \{d, e, f\}$*, cotas superiores* $= \{g, h, i\}$*, supremo* $= \{g\}$*. El supremo puede no estar en el subconjunto* A*.*

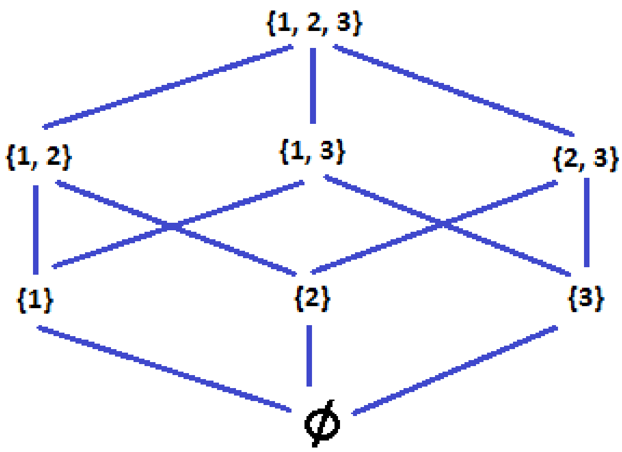
(4) *Sea* $P = \mathbb{B}_\perp \times \mathbb{B}_\perp$*:*



- $(\perp, \mathit{false}) \vee (\mathit{true}, \perp) = (\mathit{true}, \mathit{false})$
- $(\perp, \mathit{false}) \vee (\perp, \perp) = (\perp, \mathit{false})$
- $(\perp, \mathit{false}) \vee (\mathit{true}, \mathit{true})$ *no existe*

(5) *Para cualquier conjunto* X*, se denota por* $\mathcal{P}(X)$ *al conjunto de partes de* X*. Se verifica que* $\mathcal{P}(X)$ *es un poset con la relación de orden parcial dada por la inclusión de conjuntos. El supremos está dado por la unión de conjuntos.*

Si $X = \{1, 2, 3\}$ *entonces* $\mathcal{P}(X) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$.



Si $A = \{\{2\},\{3\}\}$ *entonces cotas superiores* $= \{\{2,3\},\{1,2,3\}\}$ *y supremo* $= \{2\} \vee \{3\} = \{2,3\}$

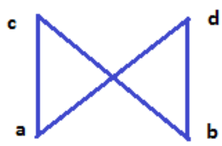
Proposición 1.12. *El supremo de un subconjunto, si existe, es único.*

Proof. Sea $A \subseteq P$ y sean s, s' supremos de A. Denotamos por B al conjunto de cotas superiores de A. Entonces, $s \leq b$ para cada $b \in B$. En particular, $s \leq s'$. De forma análoga, $s' \leq b$ para cada $b \in B$. En particular, $s' \leq s$. De ambas desigualdades tenemos que $s = s'$. □

Definición 1.13. *Sea* P *un poset y* $A \subseteq P$ *un subconjunto. Se dice que* $x \in P$ *es* ***ínfimo*** *de* A *si es el máximo de las cotas inferiores.*

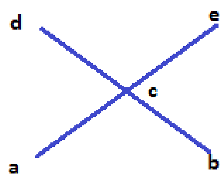
Ejemplo 1.14.

(1) *Del siguiente diagrama,*



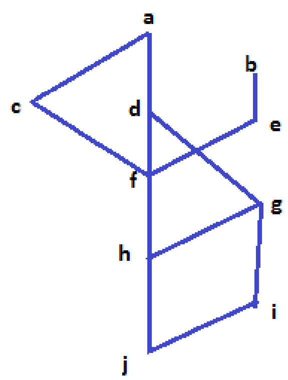
consideramos el subconjunto $A = \{a,b\}$ *no tiene ínfimo porque no tiene cotas inferiores.*

(2) *Del siguiente diagrama,*



consideramos el subconjunto $A = \{d, e\}$, *cotas inferiores* $= \{a, b, c\}$, *ínfimo* $= \{c\}$.

(3) *Del siguiente diagrama,*



consideramos el subconjunto $A = \{c, d, f, g, h\}$

cotas superiores $= \{a\}$

supremo $= \{a\}$

cotas inferiores $= \{h, j\}$

ínfimo $= h$

Proposición 1.15. *El ínfimo de un subconjunto, si existe, es único.*

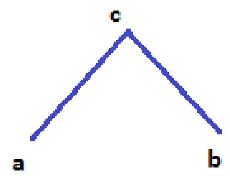
La demostración se propondrá como ejercicio.

Definición 1.16. *Un poset es un* ***retículo*** *si para cada par de elementos existe supremo e ínfimo. Un retículo se dice* ***completo*** *si existe el supremo y el ínfimo para cada subconjunto.*

Ejemplo 1.17.

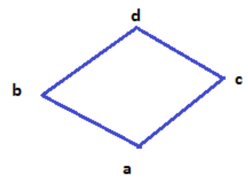
(1) $(\mathbb{N}, \leq)$ *es un retículo, pero no es completo. Para cada* $x, y \in \mathbb{N}$, $x \leq y$, $x =$ *ínfimo,* $y =$ *supremo. En cambio* $A = \{$*números pares*$\}$ *no tiene supremo.*

(2) *El siguiente diagrama,*



no es un retículo porque $\{a, b\}$ *no tiene ínfimo.*

(3) *El siguiente diagrama,*

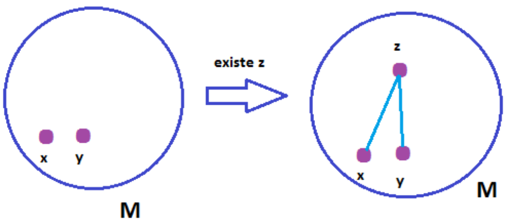


es un retículo completo.

(4) *El siguiente diagrama,*

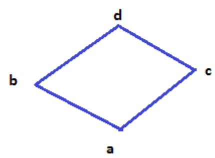
no es un retículo porque $\{c, d\}$ *y* $\{a, b\}$ *no tiene ni ínfimo ni supremos (no son comparables).*

Definición 1.18. *Unb subconjunto no vacío* M *de un poset* P *se dice que es* ***dirigido*** *cuando para cada par de elementos de* M *existe una cota superior en* M*. Es decir, dados dos elementos en* M *podemos encontrar siempre uno mayor (también en* M*).*



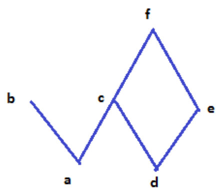
Ejemplo 1.19.

(1) *Del siguiente diagrama,*



$M = \{b, c, d\}$ *es dirigido pero* $M = \{a, b, c\}$ *no es dirigido.*

(2) *El siguiente diagrama,*



$M = \{a, b, c\}$ *no es dirigido pero* $M = \{c, d, e, f\}$ *sí es dirigido y* $\vee M = f$*.*

(3) *El conjunto de partes de* X*,* $\mathcal{P}(X)$*, es dirigido bajo la relación de inclusión.*

(4) *El subconjunto* $A = \{\textit{números pares}\}$ *de* $(\mathbb{N}, \leq)$ *es un conjunto dirigido sin supremo.*

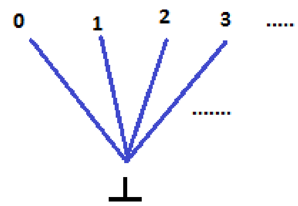
Definición 1.20. *Un poset* P *se dice* ***completo*** *si cada subconjunto dirigido de* P *tiene un supremo en* P*. Es decir,* $\forall M \subseteq P$ *dirigido,* $\exists \vee M$*. Decimos que un poset completo es un* ***dcpo*** *(directed complete partial order). Cuando existe un elemento mínimo,* $\perp$*, en un dcpo se dice que es* ***punteado****.*

Ejemplo 1.21.

(1) *El conjunto* $\{\perp, \top\}$ *con* $\perp \leq \top$ *es un dcpo punteado.*

(2) *El conjunto* $\mathbb{B}_\perp = \{\perp, true, false\}$ *es un dcpo punteado. Se le llama dcpo de los valores de la veracidad.*

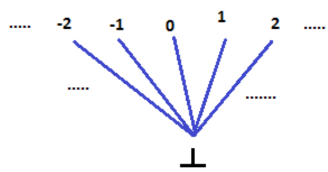
(3) *Consideremos la unión disjunta* $\mathbb{N}_\perp = \mathbb{N} \cup \{\perp\}$ *donde* $\perp \notin \mathbb{N}$ *con el orden parcial mostrado en el diagrama de Hasse siguiente:*



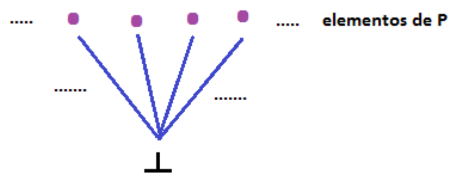
$\mathbb{N}_\perp$ *es un dcpo punteado. Se tiene que* $\perp \in \mathbb{N}_\perp$. *Veamos que todos los conjuntos dirigidos tienen supremo. Los subconjuntos dirigidos son* $\{\perp, n\}$ *donde* $n \in \mathbb{N}$ *y tienen como cota superior* n *cuyo supremo es el propio* n. *Los subconjuntos dirigidos no pueden contener dos naturales porque no tendrían cota superior.*

No hay que confundir el orden de $(\mathbb{N}, \leq)$ *(*$0 \leq 1$*) con el de* $\mathbb{N}_\perp$ *(*$0 \nleq 1$, $1 \nleq 0$, 0 *y* 1 *no son comparables).*

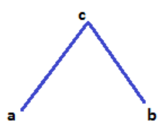
(4) $\mathbb{Z}_\perp = \mathbb{Z} \cup \{\perp\}$ *con el orden parcial del diagrama es un dcpo punteado.*



(5) *Dado un un conjunto cualquiera* P, *podemos considerar* $P_\perp = P \cup \{\perp\}$ *donde* $\perp \notin P$ *es un dcpo punteado con la relación* $x \leq y$ *si y sólo si* $x = y$ *o* $x = \perp$.



(6) *El siguiente diagrama representa un dcpo no punteado.*



(7) *Si* A *y* B *son posets punteados, entonces* $A \times B$ *también es un dcpo punteado donde* $\perp_{A\times B} = (\perp_A, \perp_B)$.

(8) *Sea* $\mathbb{Q}$ *el conjunto de los números racionales, el conjunto* $A = \{1, 1'1, 1'141, 1'414, \ldots\}$ *es una sucesión que converge a* $\sqrt{2} \notin \mathbb{Q}$. *Entonces,* A *no es un dcpo.*

Ejercicios para nota

(1) Sea $T = a, b, c, d, e, f, g$ una lista de tareas para realizar un trabajo, de las que se sabe que unas preceden inmediatamente a otras de la siguiente forma:

$$f \leq a,\ f \leq d,\ e \leq b,\ c \leq f,\ e \leq c,\ b \leq f,\ e \leq g,\ g \leq f.$$

Hallar el orden parcial. ¿Qué tareas pueden realizarse independientemente? Construir un orden si el trabajo lo realiza una sola persona.

(2) Probar que el ínfimo de un subconjunto, si existe, es único.

(3) Probar que si A y B son posets y $X \subseteq A \times B$ es dirigido entonces los conjuntos

$$\pi_0(X) = \{a \in A \,|\, \exists\, b \in B\,,\ (a, b) \in X\} \subseteq A$$

$$\pi_1(X) = \{b \in B \,|\, \exists\, a \in A\,,\ (a, b) \in X\} \subseteq B$$

también son dirigidos.

(4) Mostrar que si A y B son dcpos y $X \subseteq A \times B$ es dirigido entonces $\vee X = (\vee\pi_0(X), \vee\pi_1(X))$. Junto con $\perp_{A\times B} = (\perp_A, \perp_B)$ quedaría probado que el producto cartesiano de dcpos es un dcpo.

(5) Sea R el conjunto formado por dos elementos. Dar un ejemplo de un conjunto $X \subseteq R \times R$ tal que $\pi_0(X)$ y $\pi_1(X)$ son dirigidos pero X no lo es.

(6) Demostrar que $\mathbb{N}$ con su orden usual no es un dcpo, pero sí lo es el conjunto parcialmente ordenado obtenido a partir de $\mathbb{N}$ adjuntándole un elemento mayor que cualquier otro elemento de $\mathbb{N}$.

(7) Demostrar que un retículo completo es un dcpo punteado.

(8) ¿Cuáles de los siguientes son dcpos?
   - $\mathcal{P}(X)$ partes de $\mathbb{N}$ respecto a la inclusión.
   - la familia de subconjuntos finitos de $\mathbb{N}$ respecto a la inclusión.
   - el conjunto $\{\frac{1}{n} : n = 1, 2, \ldots\}$ respecto al orden $\leq$.
   - el conjunto $\{\frac{1}{n} : n = 1, 2, \ldots\}$ respecto al orden $\geq$.

   ¿Cuáles son punteados?

Ejercicios para programar

(1) Desarrollar un programa para introducir un conjunto de números o carácteres P y una relación de orden R entre ellos (dada mediante parejas de elementos) para saber si la relación introducida es un orden parcial.
(2) Obtener todos los órdenes parciales del conjunto de números $P = \{a, b, c, d\}$.
(3) Dado un conjunto de números P, una relación de orden R y un subconjunto $A \subseteq P$, comprobar si el subconjunto A es dirigido o no.
(4) Dado un conjunto de números P, una relación de orden R, comprobar si es un dcpo (es decir, tiene bottom y todo conjunto dirigido tiene supremo).

Podemos hacer que toda función esté bien definida. Supongamos $f : \mathbb{N} \to \mathbb{N}$ una función parcialmente definida. Es decir, no existe la imagen de todos los elementos de $\mathbb{N}$ ($\nexists f(n) \forall n \in \mathbb{N}$). Para hacer que f esté bien definida extendemos f a la función g definida como $f(n)$ cuando $f(n)$ existe y $\perp$ cuando $f(n)$ no exista:

$$\mathbb{N}_\perp \longrightarrow \mathbb{N}_\perp$$

$$n \mapsto f(n)$$

$$n \mapsto \perp$$

Definición 2.1. *Sean D y E posets. Una función $f : D \to E$ se dice* ***monótona creciente*** *si para cada $x, y \in D$ tales que $x \leq y$, entonces $f(x) \leq f(y)$.*

Las funciones monótonas crecientes preservan la estructura de poset (aunque no tienen porque preservar el elemento mínimo $\perp$). Además, la cantidad de información existente en el output debe crecer si la cantidad de información en el input crece.

Proposición 2.2. *Sean D y E posets y $M \subseteq D$ un subconjunto dirigido. Si $f : D \to E$ es una función monótona creciente, entonces $f(M)$ es un subconjunto dirigido de E.*

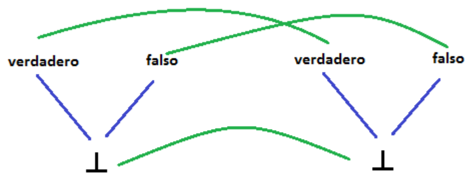
Proof. $f(M) = \{f(x) : x \in M\} \neq \emptyset$ porque $M \neq \emptyset$. Sean a, b dos elementos de $f(M)$ tales que $a = f(x)$, $b = f(y)$ para $x, y \in M$. Como M es dirigido, existe $b \in M$ cota superior de $\{x, y\}$. Como f es monótona creciente,

$$x \leq b, y \leq b \Rightarrow f(x) \leq f(b), f(y) \leq f(b)$$

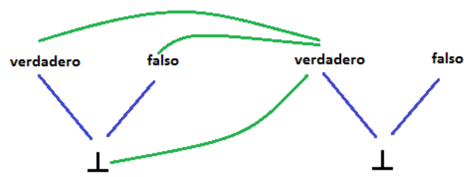
Luego $f(b) \in f(M)$ es cota superior de $\{f(x), f(y)\}$. Por lo tanto, $f(M)$ es un subconjunbto dirigido de E. $\square$

Ejemplo 2.3. *Veamos algunos ejemplos de funciones de $\mathbb{B}_\perp$ en sí mismo.*

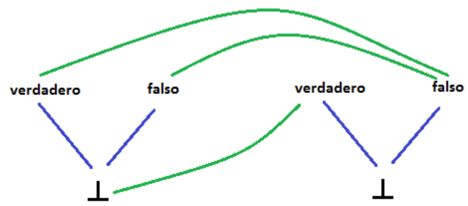
(1) *La función identidad es monótona creciente.*



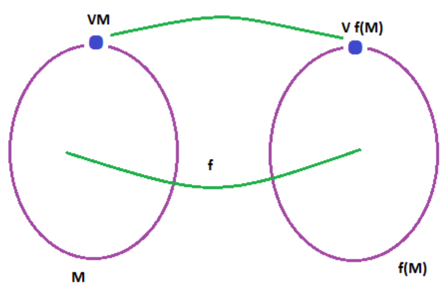
(2) *La función constante es monótona creciente.*



(3) *La función de la imagen no es monótona creciente porque $\perp \leq$ false pero $f(\perp) \nleq f($false$)$.*



Definición 2.4. *Sean D y E dos dcpos. Una función $f : D \to E$ se dice que es* ***continua*** *si es monótona creciente y para todo subconjunto direigido $M \subseteq D$ se tiene que $f(\vee M) = \vee f(M)$.*



Si D y E son dcpos punteados y $f : D \to E$ es una función se dice que f es ***estricta*** *cuando $f(\perp_D) = \perp_E$.*

Notación 2.5. *Denotamos por:*

- $D \to E$ *al conjunto de todas las funciones de* D *en* E. *Si* $f \in (D \to E)$ *entonces* $f : D \to E$.
- $D \circ \to E$ *al conjunto de todas las funciones estrictas de* D *en* E. *Si* $f \in (D \circ \to E)$ *entonces* $f : D \to E$ *y* $f(\perp_D) = \perp_E$.
- $D \xrightarrow{cont} E$ *al conjunto de todas las funciones continuas de* D *en* E.
- $D \circ \xrightarrow{cont} E$ *al conjunto de todas las funciones continuas y estrictas de* D *en* E.

Ejemplo 2.6.

(1) *Las funciones constantes dcpos son continuas. Sean* D *y* E *dcpos y* $f : D \to E$ *tal que* $f(x) = c$ *donde* $c \in E$ *es un elemento fijo.*
   - f *es monótona creciente: para cada* $x, y \in D$, $x \leq y$ *entonces* $f(x) = c \leq c = f(y)$
   - *si* $M \subseteq D$ *es dirigido entonces* $f(M) = \{c\}$ *y por lo tanto* $f(\vee M) = c = \vee f(M)$

(2) *La función identidad de un dcpo* D *(respectivamente dcpo punteado) en sí mismo es continua (respectivamente continua estricta).*
   - f *es monótona creciente: para cada* $x, y \in D$, $x \leq y$ *entonces* $f(x) = x \leq y = f(y)$
   - *si* $M \subseteq D$ *es dirigido entonces* $f(M) = M$ *y por lo tanto* $f(\vee M) = \vee M = \vee f(M)$
   - *si* D *punteado, entonces* $f(\perp_D) = \perp_D$

(3) *La composición de funciones continuas (respectivamente continuas estrictas) entre dcpos (respectivamente dcpos punteados) son continuas (respectivamente continuas estrictas).*

$$f : D \to E \qquad g : E \to F \qquad D, E, F \text{ dcpos}$$

$$g \circ f : D \to F \qquad g \circ f(x) = g(f(x))$$

   - $g \circ f$ *monótona creciente: para cada* $x, y \in D$, $x \leq y$ *entonces* $f(x) \leq f(y)$ *al ser* f *monótona creciente. Como* $f(x), f(y) \in E$ *y* g *es monótona creciente entonces* $g \circ f(x) = g(f(x)) \leq g(f(y)) = g \circ f(y)$
   - *si* $M \subseteq D$ *es dirigido entonces* $g \circ f(\vee M) = g(f(\vee M)) = g(\vee f(M)) = \vee g(f(M)) = \vee g \circ f(M)$
   - $g \circ f(\perp_D) = g(\perp_E) = \perp_F$

(4) *Sea X un conjunto y $S = \mathcal{P}(X)$ el conjunto de partes de X junto son la relación de inclusión donde $\emptyset = \perp$. En ejemplos anteriores vimos que si $M \subseteq S$ dirigido entonces $\vee M = \cup T$ donde $T \in M$.*

Dada una función $f : X \to X'$ podemos incluir una nueva función $f^* : \mathcal{P}(X) \to \mathcal{P}(X')$ tal que $f^*(Y) := \{f(y) \,|\, y \in Y\}$. Se puede probar que f^* es continua entre los dcpos $\mathcal{P}(X)$ y $\mathcal{P}(X')$. Habría que comprobar que:

- $f^*(\cup Y_\alpha) = \cup f^*(Y_\alpha)$ *para ver que* $f^*(\vee M) = \vee f^*(M)$
- $f^*(\emptyset) = \emptyset$

Lema 2.7. *Sean D y E dcpos punteados en los que las restricciones de los órdenes a $D - \{\perp_D\}$ y a $E - \{\perp_E\}$ son triviales (es decir, son cadenas). Sea $f : D \to E$ una función arbitraria. Se tiene entonces que:*

(1) *Si f es monótona creciente entonces f es continua.*

(2) *Si f es estricta entonces f es monótona creciente.*

Proof.

(1) Para ver que f es continua hay que ver que dado $M \subseteq D$ dirigido se tiene que $\vee f(M) = f(\vee M)$ puesto que ya tenemos que es monótona creciente. Sea $M \subseteq D$ dirigido. Como el orden en $D - \{\perp_D\}$ es el trivial entonces M puede constar de un único punto o bien es de la forma $\{\perp_D, d\}$, $d \in D$.

- Si $M = \{d\}$ entonces $f(\vee M) = f(d) = \vee f(d) = \vee f(M)$
- Si $M = \{\perp_D, d\}$ entonces $f(\vee M) = f(d)$ y como f es monótona creciente $\vee f(M) = \vee\{f(\perp_D), f(d)\} = f(d)$

(2) Si f es estricta entonces para cada $x, y \in D$, $x \leq y$ entonces $x \leq y$ o $x = \perp_D$. Por lo tanto, $f(x) = f(y)$ o $f(x) = f(\perp_D) = \perp_E \leq f(y)$.

□

Ejemplo 2.8.

(1) *Cualquier función f de $\mathbb{N}_\perp$ en sí misma tal que $f(\perp) = \perp$ es una función continua. Por el lema anterior, f es estricta, por lo tanto es monótona creciente. Luego es continua.*

(2) *Ejemplo de una función monótona creciente que no es continua. Sea*

$$f : \mathcal{P}(\mathbb{N}) \to \{\perp, \top\} \ \perp \leq \top$$

donde $f(A) = \perp$ *si* A *es finito y* $f(A) = \top$ *si* A *es infinito.*

- *Consideramos un conjunto dirigido de* $\mathcal{P}(X)$, $M = \{\{0\}, \{0, 1\}, \{0, 1, 2\}, \dots, \}$. *Veamos que* $\vee f(M) \neq f(\vee M)$.
  - $\vee M = \mathbb{N}$ *entonces* $f(\vee M) = f(\mathbb{N}) = \top$
  - $f(M) = \{\perp\}$ *entonces* $\vee f(M) = \perp$
- *Veamos que sí es monótona creciente. Sea* $A \subseteq B$. *Distinguimos varios casos:*
  - A *y* B *finitos, entonces* $f(A) = \perp = f(B)$
  - A *finito y* B *infinito, entonces* $f(A) = \perp \leq \top = f(B)$
  - A *y* B *infinitos, entonces* $f(A = \top = f(B))$

Lema 2.9. *Sea* L *un poset,* $\{A_i\}_{i \in I}$ *una familia de subconjuntos de* L *y* $A = \cup_{i \in I} A_i$. *Suponemos que para cada* $i \in I$ *existe el supremo de* A_i, $\vee A_i$ *y suponemos que existe el supremo del conjuntos* $\{\vee A_i \, : \, i \in I\}$. *Entonces existe* $\vee A$ *y además* $\vee A = \vee\{\vee a_i \, : \, i \in I\}$.

Proof. Llamamos $b = \vee\{\vee A_i \, : \, i \in I\}$. Si $a \in A = \cup_{i \in I} A_i$, en particular $a \in A_j$ para algún $j \in I$. Entonces $a \leq \vee A_j \leq \vee\{\vee A_i \, : \, i \in I\} = b$. Luego $a \leq b$ para cada $a \in A$. Por lo tanto b es una cota superior de A.

Sea c otra cota superior de A. Entonces $\vee A_i \leq c$ para cada $i \in I$. De aquí que, $\vee\{\vee A_i \, : \, i \in I\} \leq c$ y por lo tanto $b \leq c$.

Finalmente tenemos que $b = \vee\{\vee A_i \, : \, i \in I\}$ es el supremo de A, $b = \vee A$. □

Ejemplo 2.10. *Puede existir el supremo de cada conjunto pero no el supremo de los supremos. Sea* $\mathbb{N} = \cup_{n \in \mathbb{N}} \{n\}$. *Cada* $\{n\}$ *tiene supremo, pero no existe el supremo de los supremos.*

$$\vee\{\vee\{n\} \, : \, n \in \mathbb{N}\} = \vee\{n \, : \, n \in \mathbb{N}\} = \vee\mathbb{N}$$

Corolario 2.11. *Sea M poset y dirigido, D dcpo y $f : M \times M \to D$ monótona creciente respecto al orden*

$$(a_1, a_2) \leq (b_1, b_2) \text{ si y sólo si } a_1 \leq b_1 \text{ y } a_2 \leq b_2$$

Definimos el conjunto M_x para cada $x \in M$, $M_x = \{f(x, y) : y \in M\} \subseteq D$. Entonces M_x, $\{\vee M_x\}$ y $\{f(x, x) : x \in M\}$ son dirigidos y además

$$\vee f(M \times M) = \vee\{\vee M_x : x \in M\} = \vee\{f(x, x) : x \in M\}.$$

Proof.

- Veamos que M_x es dirigido. Sean $a_1, a_2 \in M_x$ tales que $a_1 = f(x, y_1), a_2 = f(x, y_2)$ para $y_1, y_2 \in M$. Como M es dirigido, existe $b \in M$ cota superior de $\{y_1, y_2\}$. Entonces $(x, y_1) \leq (x, b)$ y $(x, y_2) \leq (x, b)$. Como f es monótona creciente, $f(x, y_1) \leq f(x, b)$ y $f(x, y_2) \leq f(x, b)$. Por lo tanto, $f(x, b) \in M_x$ es una cota superior de $\{a_1, a_2\} = \{f(x, y_1), f(x, y_2)\}$.
- Veamos que $\{\vee M_x : x \in M\}$ es dirigido. Sean $a_1, a_2 \in \{\vee M_x : x \in M\}$. Entonces, $a_1 = \vee M_{x_1}$ y $a_2 = \vee M_{x_2}$. Como M es dirigido, existe $b \in M$ tal que $x_1, x_2 \leq b$.

  $$M_{x_1} = \{f(x_1, y) : y \in M\} \qquad\qquad M_{x_2} = \{f(x_2, y) : y \in M\}$$

  Para cada $y \in M$ cada elemento $f(x_1, y)$ de M_{x_1} es menor o igual que $f(b, y) \in M_b$. De forma análoga, para cada $y \in M$, cada elemento $f(x_2, y)$ de M_{x_2} es menor o igual que $f(b, y)$. De ambas desigualdades tenemos que M_b es una cota superior de M_x. Entonces $\vee M_{x_1}, \vee M_{x_2} \leq \vee M_b$. Luego $\vee M_b \in \{\vee M_x : x \in M\}$ es una cota superior de $\{a, b\}$.
- Veamos que $\{f(x, x) : x \in M\}$ es dirigido. Sean $a, b \in \{f(x, x) : x \in M\}$ tales que $a = f(x_1, x_1)$, $b = f(x_2, x_2)$ para $x_1, x_2 \in M$. Como M dirigido, existe $z \in M$ cota superior de $\{x_1, x_2\}$. Entonces $x_1 \leq x$, $x_2 \leq z$ implica que $(x_1, x_1) \leq (z, z)$, $(x_2, x_2) \leq (z, z)$. Como f es monótona creciente, $f(x_1, x_1) \leq f(z, z)$, $f(x_2, x_2) \leq f(z, z)$. De aquí que, $f(z, z) \in \{f(x, x) : x \in M\}$ es cota superior de $\{a, b\}$.
- Veamos que $\vee f(M \times M) = \vee\{\vee M_x : x \in M\}$. Sabemos que $f(M \times M) = \cup_{x \in M} M_x$. Como M_x dirigido y $M_x \subseteq D$, D dcpo, entonces existe $\vee M_x$. Igualmente, como $\{\vee M_x :$

$x \in M\} \subseteq D$ existe $\vee\{\vee M_x : x \in M\}$. Podemos aplicar el lema anterior para afirmar que existe $\vee f(M \times M)$ y $\vee f(M \times M) = \vee\{\vee M_x : x \in M\}$.

- Veamos que $\vee f(M \times M) = \vee\{f(x,x) : x \in M\}$.
  - Es claro que $f(x,x) \leq \vee f(M \times M)$ por ser $f(x,x)$ un elemento del conjunto para cada $x \in M$. Entonces $\vee\{f(x,x) : x \in M\} \leq \vee f(M \times M)$.
  - Sea $f(x,y)$ un elementos dcualquiera de $f(M \times M)$. Para cada $x, y \in M$ existe $z \in M$ cota superior de $\{x,y\}$. Entonces $(x,y) \leq (z,z)$. Como f es monótona creciente $f(x,y) \leq f(z,z) \leq \vee\{f(x,x) : x \in M\}$. De aquí que $\vee f(M \times M) \leq \vee\{f(x,x) : x \in M\}$.

$\square$

Ejemplo 2.12. *Consideramos $M = \{0,1,\ldots,11\}$, $D = \{0,1,\ldots,121\}$ y la función*

$$f : \{0,1,\ldots,11\} \times \{0,1,\ldots,11\} \to \{0,1,\ldots,121\}$$

definida por $f(n,m) = n \cdot m$. Entonces,

- $\vee f(M \times M) = \vee\{0,1,\ldots,121\} = 121$
- $\vee\{f(x,x) : x \in M\} = \vee\{0,1,4,9,\ldots,121\} = 121$

Definición 2.13. *Sean $f, g \in (D \overset{cont}{\longrightarrow} E)$, D y E posets. Decimos que $f \leq g$ si y sólo si $f(x) \leq g(x)$ para cada $x \in D$. Esto quiere decir que f y g tiene un **orden puntual**.*

Proposición 2.14. *Sean D y E dcpos y $D \overset{cont}{\longrightarrow} E$ el conjunto de todas las funciones continuas de D a E ordenadas puntualmente. Entonces $D \overset{cont}{\longrightarrow} E$ es un dcpo. Además, si E es un dcpo punteado, $D \circ \overset{cont}{\longrightarrow} E$ es un dcpo punteado.*

Proof.

- Veamos que $D \overset{\text{cont}}{\longrightarrow} E$ es un dcpo. Hay que probar quye cada subconjunto dirigido $F \subseteq (D \overset{\text{cont}}{\longrightarrow} E)$ tiene un supremo en $(D \overset{\text{cont}}{\longrightarrow} E)$.

Sea F un subconjunto dirigido de $(D \stackrel{\text{cont}}{\longrightarrow} E)$. Buscamos su supremo en $D \stackrel{\text{cont}}{\longrightarrow} E$. Consideramos el subconjunto M_x para cada $x \in D$,

$$M_x = \{f(x) \,:\, f \in F\} \subseteq E$$

– Veamos que M_x es dirigido. Sean $a_1, a_2 \in M_x$, $a_1 = f_1(x)$, $a_2 = f_2(x)$ para $f_1, f_2 \in F$. Como F es dirigido existe $h \in F$ tal que $f_1 \leq h$, $f_2 \leq h$. Entonces

$$a_1 = f_1(x) \leq h(x) \qquad\qquad a_2 = f_2(x) \leq h(x)$$

Por lo tanto $h(x) \in M_x$ es cota superior de $\{a_1, a_2\}$. Luego M_x es dirigido.

– Como $M_x \subseteq E$ es dirigido y E es un dcpo, entonces M_x tiene supremo para cada $x \in D$.

– Consideremos la aplicación $g \in (D \longrightarrow E)$ definida por $g(x) = \vee M_x = \vee\{f(x) \,:\, f \in F\}$. Veamos que g es continua y $g = \vee F$ y habremos terminado.

  * Veamos que g es monótona creciente. Sean $x, y \in D$, $x \leq y$. Como $f \in F$ es continua, $f(x) \leq f(y)$ para cada $f \in F$. Luego $g(x) = \vee\{f(x) \,:\, f \in F\} \leq \vee\{f(y) \,:\, f \in F\} = g(y)$. Por lo tanto, g es monótna creciente.
  * Dado $M \subseteq D$ dirigido., veamos que $\vee g(M) = g(\vee M)$.

$$g(\vee M) = \vee\{f(\vee M) \,:\, f \in F\} = \vee\{\vee f(M) \,:\, f \in F\}$$

$$\vee g(M) = \vee\{g(x) \,:\, x \in M\} = \vee\{\vee\{f(x) \,:\, f \in F\} \,:\, x \in M\} =$$
$$= \vee\{f(x) \,:\, f \in F, x \in M\}$$

Por el lema anterior sabemos que la igualdad de los conjuntos anteriores se da.

  * Veamos que g es el supremo de F. Por definición g es cota superior de F. Si h es otra cota superior, entonces $f(x) \leq h(x)$ para cada $f \in F$ y para cada $x \in M$. Entonces $\vee\{f(x) \,:\, f \in F\} \leq h(x)$ para cada $x \in M$. Por lo tarto $g(x) \leq h(x)$ para cada $x \in M$.

- Si definimos la función $\perp : D \to E$ como la función constante $\perp(x) = \perp$, entonces $\perp \leq f$ para cada $f \in (D \circ \stackrel{\text{cont}}{\longrightarrow} E)$. Por tanto, $D \circ \stackrel{\text{cont}}{\longrightarrow} E$ es punteado.

$\square$

Lema 2.15. *Sean D un dcpo, E un dcpo punteado y $f, g : D \to E$ funciones continuas. Sea $b : D \to \mathbb{B}_\perp$ una función que toma valores en el dcpo punteado de los valores de veracidad. Sea $h : D \to E$ la función definida por,*

- $h(x) = \perp$ *si* $b(x) = \perp$
- $h(x) = f(x)$ *si* $b(x) = verdadero$
- $h(x) = g(x)$ *si* $b(x) = falso$

Entonces h es continua.

Proof. Hay que ver que h es monótona creciente y que $h(\vee M) = \vee h(M)$ para M dirigido. $\square$

Proposición 2.16. *Sea D un dcpo punteado, b una función continua de D hasta $\mathbb{B}_\perp$ y las funciones siguientes,*

$$\phi, c \in (D \circ \xrightarrow{cont})D \qquad\qquad F(\phi) : D \to D$$

donde $F(\phi)$ es la función que actúa del siguiente modo para cada $x \in D$:

- $F(\phi)(x) = \perp$ *si* $b(x) = \perp$
- $F(\phi)(x) = x$ *si* $b(x) = falso$
- $F(\phi)(x) = (\phi \circ c)(x)$ *si* $b(x) = verdadero$

Se tiene entonces:

(1) *La función $F(\phi)$ es continua estricta.*
(2) *La aplicación $F : (D \circ \xrightarrow{cont} D) \to (D \circ \xrightarrow{cont} D)$ tal que a cada $\phi \in (D \circ \xrightarrow{cont} D)$ le asocio $F(\phi) \in (D \circ \xrightarrow{cont} D)$, es continua.*

Proof.

(1) Para ver que $F(\phi)$ es continua estricta usamos el lema anterior.
(2) Para ver que F es continua probamos que es monótona creciente y que $F(\vee M) = \vee F(M)$ para $M \subseteq (D \circ \xrightarrow{cont} D)$ dirigido.

$\square$

Lema 2.17. *Sean E, E, F dcpos y $f : D \times E \to F$ una función . Entonces, f es continua si y sólo si f es continua en cada variable.*

Proof. Para la demostración hay que considerar las funciones $f_d : E \to F$ definida por $f_d(y) = f(d, y)$ para $d \in D$ fijo y $f_e : D \to F$ definida por $f_e(x) = f(x, e)$ para $e \in E$ fijo. $\square$

Corolario 2.18. *Sean D, E, F dcpos. La función*

$$\circ : (E \overset{cont}{\longrightarrow} F) \times (D \overset{cont}{\longrightarrow} E) \to (D \overset{cont}{\longrightarrow} F)$$

que a cada par de funciones le asocia su composición, es una función continua.

Proof. Usamos el lema anterior y consideramos las aplicaciones,

$$\circ_f : (D \overset{\text{cont}}{\longrightarrow} E) \to (D \overset{\text{cont}}{\longrightarrow} F) \qquad \circ_f\,(g) = f \circ g$$

$$\circ_g : (E \overset{\text{cont}}{\longrightarrow} F) \to (D \overset{\text{cont}}{\longrightarrow} F) \qquad \circ_g\,(f) = f \circ g$$

$\square$

Ejercicios para nota

(1) Sean D y E dcpos y $f : D \to E$ una función sobreyectiva tal que para cualesquiera $x, x' \in D$ las desigualdades $f(x) \leq f(x')$ y $x \leq x'$ son equivalentes. Demostrar que f es continua.

(2) Determinar las aplicaciones continuas que tienen dominio y codominio $\mathbb{N}_\perp$, $f : \mathbb{N}_\perp \to \mathbb{N}_\perp$.

(3) Sea el dcpo punteado $D = [0, 500]$. Para todo $x \in D$ denotamos por $\lfloor x \rfloor$ qal mayor entero del conjunto $\{n \in \mathbb{N} : n \leq x\}$ (es decir, la función *suelo*). La función $f : D \to D$ definida por $f(x) = \lfloor x \rfloor$, ¿es continua?

(4) Sean D y E dcpos, $f : D \to E$ una función continua y biyectiva. Demostrar que si f^{-1} es monótona creciente entonces f^{-1} es también continua.

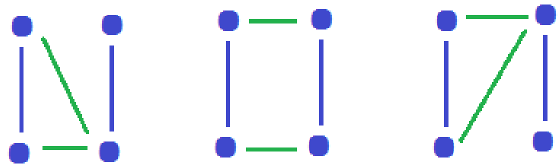
(5) Una función $f : D \to E$ entre dcpos punteados se dice que es un isomorfismo cuando es biyectiva, continua, estricta y su inversa f^{-1} es también continua. Los dcpos punteados se dicen isomorfos cuando existe algún isomorfismo $f : D \to E$.

Dar ejemplos de funciones biyectivas, continuas y estrictas entre dcpos punteados que no son isomorfos.

(6) Sean $f, g, h : \mathcal{P}(\mathbb{N}) \times \mathcal{P}(\mathbb{N}) \to \mathcal{P}(X)$ las aplicaciones dadas por $f(S,T) = S \cup T$, $g(S,T) = S \cap T$ y $h(S,T) = S - T$. ¿Cuáles son funciones continuas?

Ejercicios para programar

(1) Sea $\mathcal{N}$ la cadena de N puntos. Hay una función monótona creciente de $1 \to 1$. Hay tres funciones monótonas crecientes de $2 \to 2$.



Escribir las funciones monótonas crecientes de $3 \to 3$. Escribir un programa (recursivo o no) para calcular el número de funciones crecientes de $\mathcal{N} \to \mathcal{M}$.

(2) Calcular las funciones monótonas creientes entre los siguientes diagramas de Hasse.

(a):

(b):

(c):

(3) Usar el programa realizado en el punto **4** del tema anterior para calcular todas las funciones continuas de un poset P en sí mismo. Comprobar que las funciones del ejercicio anterior son continuas.

3. Algunas construcciones de dcpos

Profesora: Alicia Tocino Sánchez. Curso Académico 2018-2019.

3.1. **Productos directos.** Dados los posets D y E definimos el **producto directo** de D y E como sigue:

$$D \times E = \{(x, y) \,:\, x \in D, y \in E\}$$

El orden parcial usado en $D \times E$ es:

$$(x, y) \leq (x', y') \Longleftrightarrow x \leq x' \,, y \leq y'.$$

Función *fst* y *snd*. Definimos las funciones,

$$\begin{array}{rcl} D \times E & \xrightarrow{\textbf{fst}} & D \\ (x, y) & \longmapsto & \textbf{fst}(x, y) = x \end{array} \qquad \begin{array}{rcl} D \times E & \xrightarrow{\textbf{snd}} & E \\ (x, y) & \longmapsto & \textbf{snd}(x, y) = y \end{array}$$

Recordamos que si $f : S \to T$ es una función entre dos conjuntos, denotamos por $f^* : \mathcal{P}(S) \to \mathcal{P}(T)$ a la función entre el conjunto de partes de S y T.

Sea $L \subseteq D \times E$ un subconjunto dirigido, entonces los siguientes conjuntos son dirigidos:

$$M = \textbf{fst}^*(L) = \{x \in D \,:\, \exists y \in E \,:\, (x, y) \in L\}$$

$$N = \textbf{snd}^*(L) = \{y \in E \,:\, \exists x \in D \,:\, (x, y) \in L\}$$

En particular, si D y E son dcpos punteados, entonces $\vee L = (\vee M, \vee N)$ y $\perp_{D \times E} = (\perp_D, \perp_E)$. Por lo tanto, $D \times E$ es un dcpo punteado.

Dados D, E, F dcpos sabemos que una función $f : D \times E \to F$ es continua si y sólo si es continua en cada componente. Es decir, f es continua si y sólo si:

- la función $\begin{array}{rccl} f_e : & D & \longrightarrow & F \\ & x & \longmapsto & f(x, e) \end{array}$ es continua y
- la función $\begin{array}{rccl} f_d : & E & \longrightarrow & F \\ & y & \longmapsto & f(d, y) \end{array}$ es continua

De esta forma sabemos que las funciones **fst** y **snd** son continuas. Además, dado F dcp y las funciones continuas $f : F \to D$ y $g : F \to E$ existe una funcón continua $< f, g >$ tal que

$\mathbf{fst} \circ < f, g >= f$ y $\mathbf{snd} \circ < f, g >= g$

$$\begin{array}{rccc} < f, g >: & F & \longrightarrow & D \times E \\ & x & \longmapsto & (f(x), g(x)) \end{array}$$

y para cada función continua $h : f \longrightarrow D \times E$ se tiene que,

$$< \mathbf{fst} \circ h, \mathbf{snd} \circ h >= h.$$

Proposición 3.1. *Dados D, E, F dcpos y $f : F \to D$, $g : F \to E$ funciones continuas, existe una única función continua $< f, g >$ que completa el diagrama anterior.*

$$\begin{tikzcd}
D & \\
D \times E \arrow[u, "\boldsymbol{fst}"] \arrow[d, "\boldsymbol{snd}"'] & F \arrow[ul, "f"'] \arrow[l, "<f,g>"'] \arrow[dl, "g"] \\
E &
\end{tikzcd}$$

Esta propiedad se llama propiedad universal del producto.

Entonces $D \times E$ es un producto directo si y sólo si cumple la propiedad universal. Sean $f : D \to D'$ y $g : E \to E'$ funciones continuas. Podemos definir la función,

$$\begin{array}{rccc} f \times g : & D \times E & \longrightarrow & D' \times E' \\ & (x, y) & \longmapsto & (f(x), g(y)) \end{array}$$

como $f \times g =< f \circ \mathbf{fst}, g \circ \mathbf{snd} >$. Podemos resumir la definición de $f \times g$ en el siguiente diagrama.

$$\begin{tikzcd}
D \arrow[r, "f"] & D' \\
D \times E \arrow[u, "\mathbf{fst}"] \arrow[d, "\mathbf{snd}"'] & D' \times E' \arrow[u, "\mathbf{fst}"'] \arrow[d, "\mathbf{snd}"] \arrow[l, "f \times g"'] \\
E \arrow[r, "g"] & E'
\end{tikzcd}$$

Se verifican las siguientes propiedades:

- $id_D \times id_E = id_{D\times E}$
- $(f' \times g') \circ (f \times g) = (f' \circ f) \times (g' \circ g)$

Función apply y curry. Dará una relación entre $\rightarrow$ y $\times$. Sean D, E, F dcpos. Existe una función,

$$\begin{array}{rrcl} \textbf{apply}: & ((E \stackrel{\text{cont}}{\longrightarrow} F) \times E) & \longrightarrow & F \\ & (f, x) & \longmapsto & f(x) \end{array}$$

donde $f : E \rightarrow F$ es una función continua y $x \in E$.

Proposición 3.2. *La función **apply** es continua.*

Dado $f : D \times E \rightarrow F$ existe una función,

$$\begin{array}{rrcl} \textbf{curry}(f): & D & \longrightarrow & (E \stackrel{\text{cont}}{\longrightarrow} F) \\ & x & \longmapsto & \textbf{curry}(f)(x) \end{array}$$

$$\begin{array}{rrcl} \textbf{curry}(f)(x): & E & \longrightarrow & F \\ & y & \longmapsto & \textbf{curry}(f)(x)(y) = f(x, y) \end{array}$$

Observamos que $\textbf{curry}(f)(x) = f_x$.

Proposición 3.3. *La función **curry** es continua. Además, es la única función continua que hace el siguiente diagrama conmutativo.*

$$\begin{tikzcd} D \times E \arrow[r, "f"] \arrow[d, "\boldsymbol{curry(f)}\times id"'] & F \\ ((E \stackrel{cont}{\longrightarrow} F) \times E) \arrow[ur, "\boldsymbol{apply}"'] & \end{tikzcd}$$

Obervemos que la condición de unicidad equivale a la siguiente igualdad entre funciones:

$$\textbf{curry}(\textbf{apply} \circ (h \times id_E)) = h$$

Esta igualdad se sigue del diagrama,

$$\begin{tikzcd} D \times E \arrow[r, "f"] \arrow[d, "h \times id"'] & F \\ ((E \overset{\text{cont}}{\longrightarrow} F) \times E) \arrow[ur, "\mathbf{apply}"'] & \end{tikzcd}$$

donde $f = \mathbf{apply} \circ (h \times id_E)$.

Resumimos todo lo anterior en la siguiente proposición.

Proposición 3.4. *Sean D, E, F dcpos.*

(1) *La función **apply** $: ((E \overset{cont}{\longrightarrow} F) \times E) \to F$ definido por **apply**$(f, x) = f(x)$ es una función continua.*

(2) *Sean $f : D \times E \to F$ continua, $x \in D$ y $f_x(y) = f(x, y)$ para cada $y \in E$. Entonces f_x es una función continua y la función **curry**$(f) : D \to (E \overset{cont}{\longrightarrow})$ dada por **curry**$(f)(x) = f_x$ es continua.*

*Además, **curry**(f) está caracterizado por ser la única función continua $f' : D \to (E \overset{cont}{\longrightarrow})$ que hace conmutativo el diagrama siguiente.*

$$\begin{tikzcd} D \times E \arrow[r, "f"] \arrow[d, "f' \times id"'] & F \\ ((E \overset{cont}{\longrightarrow} F) \times E) \arrow[ur, "\boldsymbol{apply}"'] & \end{tikzcd}$$

Ocurre una situación similar que veremos en las próximas secciones para dcpos punteados y funciones continuas estrictas.

3.2. **Productos smash (o quebrado).** En el producto directo de dcpos punteados $D \times E$ hay elementos de la forma $(x, \perp)$ y $(\perp, y)$. Si $x \neq \perp$ o $y \neq \perp$ entonces $(x, \perp)$ y $(\perp, y)$ son elementos distintos de $D \times E$. En la programaci´ón semántica hay ocasiones en las que queremos identificar los pares $(x, \perp)$ y $(\perp, y)$. Por este motivo existe otra versión de producto llamado **producto smash**.

Definición 3.5. *Para D, E dcpos punteados se define el producto* ***smash*** *como el conjunto,*

$$D \otimes E = \{(x, y) \in D \times E \,:\, x \neq \perp,\, y \neq \perp\} \cup \{\perp_{D\otimes E}\}$$

donde $\perp_{D\otimes E}$ es un nuevo elemento (no es un par).

El orden que se sigue es el mismo que en el producto directo añadiendo que $\perp_{D\otimes E} \leq z$ para cada $z \in D \otimes E$.

Función *smash* y *unsmash*. Existe una función continua y sobreyectiva definida por **smash** : $D \times E \to D \otimes E$ donde:

$$\mathbf{smash}(x, y) = \begin{cases} (x, y) & \text{si } x \neq \perp \text{ y } y \neq \perp \\ \perp_{D\otimes E} & \text{en otro caso} \end{cases}$$

La función **smash** establece una relación entre el producto directo, $D \times E$, y el producto smash, $D \otimes E$.

También existe una función continua e inyectiva definida por **unsmash** : $D \otimes E \to D \times E$ donde:

$$\mathbf{unsmash}(z) = \begin{cases} z & \text{si } z = (x, y) \text{ es un par} \\ (\perp, \perp) & \text{si } z = \perp_{D\otimes E} \end{cases}$$

Para relacionar *smash* y *unsmash* en un mismo diagrama necesitaremos saber qué es una función biestricta.

Definición 3.6. *Decimos que la función $f : D \times E \to F$ donde D, E, F son dcpos punteados, es* ***biestricta*** *si $f(x, y) = \perp$ siempre que $x = \perp$ o $y = \perp$.*

Proposición 3.7. *Si $f : D \times E \to F$ es biestricta y continua entonces $g = f \circ$* ***unsmash*** *es la única función estricta y continua que completa el diagrama.*

$$\begin{tikzcd} D \times E \arrow[d, "\boldsymbol{smash}"'] \arrow[dr, "f"] & \\ D \otimes E \arrow[r, "f \circ \boldsymbol{unsmash}"'] & F \end{tikzcd}$$

Si $f : D \to D'$ y $g : E \to E'$ son estrictas y continuas entonces $f \otimes g = \mathbf{smash} \circ (f \times g) \circ \mathbf{unsmash}$ es la única función estricta y continua que completa el siguiente diagrama:

$$\begin{tikzcd} D \times E \arrow[r, "f\times g"] \arrow[d, "\mathbf{smash}"'] & D' \times E' \arrow[d, "\mathbf{smash}"] \\ D \otimes E \arrow[r, "f\otimes g"'] & D' \otimes E' \end{tikzcd}$$

Observamos que existe una relación entre $\otimes$ y las funciones estrictas continuas.

Al igual que tenemos la relación entre funciones continuas, producto directo, *apply* y *curry* ahora podemos dar una relación entre funciones continuas estrictas, producto smash, *strict_ apply* y *strict_ curry*:

$$\begin{tikzcd} D \otimes E \arrow[r, "f"] \arrow[d, "\mathbf{strict\_curry}(f)\otimes id"'] & F \\ ((E \circ \xrightarrow{\text{cont}} F) \otimes E) \arrow[ur, "\mathbf{strict\_apply}"'] & \end{tikzcd}$$

3.3. Suma y suma fundida.

Definición 3.8. *Sean D, E dcpos. Definimos la* ***suma*** *como el conjunto,*

$$D + E = ((D \times \{0\}) \cup (E \times \{1\})$$

El conjunto $D + E$ es un dcpo respecto al orden parcial dado por,

$$(x, i) \leq (y, j) \iff i = j,\; x \leq y.$$

Es decir, sólo se pueden comparar elementos pertenecientes a un mismo conjunto. No podemos comparar un elemento de D, de la forma $(x, 0)$, con otro de E, de la forma $(y, 1)$, ya que $0 \neq 1$.

Además, las funciones siguientes son continuas:

$$\begin{array}{ccc} D & \xrightarrow{\mathbf{d}} & D + E \\ x & \longmapsto & (x, 0) \end{array} \qquad\qquad \begin{array}{ccc} E & \xrightarrow{\mathbf{e}} & D + E \\ y & \longmapsto & (y, 1) \end{array}$$

Si F es un dcpo y $f_d : D \to F$, $f_e : E \to F$ son continuas, existe una única función $f : D{+}E \to F$ continua que hace conmutativo el diagrama:

$$\begin{tikzcd}
D \arrow[d, "d"'] \arrow[dr, "f_d"] & \\
D+E \arrow[r, "f"'] & F \\
E \arrow[u, "e"] \arrow[ur, "f_e"'] &
\end{tikzcd}$$

Si $f : D \to E'$ y $g : E \to E'$ son funciones continuas, podemos definir la función $f{+}g : D{+}E \to D' + E'$ que hace conmutativo el siguiente diagrama,

$$\begin{tikzcd}
D \arrow[r, "f"] \arrow[d, "d"'] & D' \arrow[d, "d'"] \\
D+E \arrow[r, "f+g"'] & D'+E' \\
E \arrow[u, "e"] \arrow[r, "g"'] & E' \arrow[u, "e'"']
\end{tikzcd}$$

Definición 3.9. *Sean D, E dcpos punteados. Definimos la* ***suma fundida*** *$D \oplus E$ como el conjunto,*

$$D \oplus E = ((D - \{\perp_D\}) \times \{0\}) \cup ((E - \{\perp_E\}) \times \{1\}) \cup \{\perp_{D\oplus E}\}$$

donde $D - \{\perp_D\}$ y $E - \{\perp_E\}$ son los conjuntos D y E sin su elementos mínimo y $\perp_{D\oplus E}$ es un nuevo elemento que no es un par.

La suma fundida, $D \oplus E$, es un dcpo punteado con la relación de orden parcial heredada de $D + E$,

$$(x, i) \leq (y, j) \iff i = j,\ x \leq y$$

$$\perp_{D\oplus E} \leq z \,\forall z \in D \oplus E$$

Funciones *inl* y *inr*. Existen funciones estrictas continuas **inl** $: D\circ \longrightarrow D \oplus E$ y **inr** $: E\circ \longrightarrow D \oplus E$ definidas por:

$$\mathbf{inl}(x) = \begin{cases} (x,0) & \text{si } x \neq \perp \\ \perp_{D\oplus E} & \text{si } x = \perp \end{cases} \qquad \mathbf{inr}(y) = \begin{cases} (y,1) & \text{si } y \neq \perp \\ \perp_{D\oplus E} & \text{si } y = \perp \end{cases}$$

Además, si F es un dcpo punteado y $f : D\circ \to F$, $g : E\circ \to F$ son funciones continuas estrictas, entonces existe una única función continua estricta $[f,g]$ que completa el diagrama:

$$\begin{tikzcd}
D \arrow[d, "\mathbf{inl}"'] \arrow[dr, "f"] & \\
D \oplus E \arrow[r, "{[f,g]}"'] & F \\
E \arrow[u, "\mathbf{inr}"] \arrow[ur, "g"'] &
\end{tikzcd}$$

La función $[f,g]$ está dada por:

$$\mathbf{[f,g]}(z) = \begin{cases} f(x) & \text{si } z = (x,0) \\ g(y) & \text{si } z = (y,1) \\ \perp & \text{si } z = \perp_{D\oplus E} \end{cases}$$

Dadas $f : D\circ \to D'$ y $g : E\circ \to E'$ definimos

$$f \oplus g = [\mathbf{inl} \circ f, \mathbf{inr} \circ g] : D \oplus \circ \to D' \oplus E'$$

haciendo conmutativo el diagrama siguiente.

$$\begin{tikzcd}
D \arrow[r, "f"] \arrow[d, "\mathbf{inl}"'] & D' \arrow[d, "\mathbf{inl}"] \\
D \oplus E \arrow[r, "f\oplus g"'] & D' + E' \\
E \arrow[r, "g"'] \arrow[u, "\mathbf{inr}"] & E' \arrow[u, "\mathbf{inr}"']
\end{tikzcd}$$

3.4. **Elevaciones.**

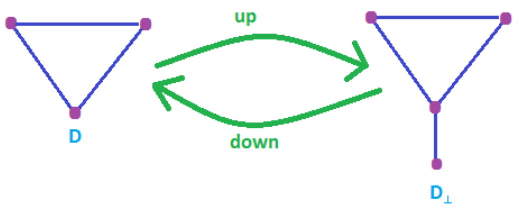
Definición 3.10. *Dado D un dcpo punteado, definimos el* ***elevado*** *de D como el conjunto $D_{bot} = (D \times \{0\}) \cup \{\perp\}$ donde $\perp$ es un nuevo elemento que no es un par.*

El conjunto $D_\perp$ junto con el orden parcial dado por $(x, 0) \leq (y, 0)$ si y sólo si $x \leq y$ y $\perp \leq z$ para cada $x \in D_\perp$ es un dcpo punteado.

Funciones *down* y *up*. Definimos la función continua estricta **down** : $D_\perp \circ \to D$ dada por,

$$\textbf{down}(z) = \begin{cases} x & \text{si } z = (x, 0) \\ \perp_D & \text{en caso contrario} \end{cases}$$

Definimos la función continua (no estricta) **up** : $D \to D_\perp$ dada por $\textbf{up}(x) = (x, 0)$.



Estas funciones cumplen las relaciones:

- $\textbf{down} \circ \textbf{up} = id_D$

$$D \overset{\textbf{up}}{\to} D_\perp \overset{\textbf{down}}{\to} D$$

$$x \mapsto (x, 0) \mapsto x$$

- $\textbf{up} \circ \textbf{down} \neq id_{D_\perp}$

$$D_\perp \overset{\textbf{down}}{\to} D \overset{\textbf{up}}{\to} D_\perp$$

$$z \mapsto x \mapsto (x, 0)$$

$$z \mapsto \perp \mapsto (\perp, 0)$$

Sea $f : D \to E$ función continua entre dcpos punteados, existe una única función continua estricta f^+ que completa el diagrama:

$$\begin{tikzcd}
D \arrow[d, "\mathbf{up}"'] \arrow[dr, "f"] & \\
D_\perp \arrow[r, "f^+"'] & F
\end{tikzcd}$$

Además, dada la función continua $f : D \to E$ definimos la función continua estricta $f_\perp = (\mathbf{up} \circ f)^+ : D_\perp \circ \to E_\perp$.

$$\begin{tikzcd}
D \arrow[r, "f"] \arrow[d, "\mathbf{up}"'] & E \arrow[d, "\mathbf{up}"] \\
D_\perp \arrow[r, "f_\perp"'] & E_\perp
\end{tikzcd}$$

3.5. **Suma separada.**

Definición 3.11. *Sean D, E dcpos punteados. Definimos la **suma separada** como el dcpo punteado $D_\perp \oplus E_\perp$.*

Por las propiedades universales sabemos que $h = [f^+, g^+]$ es la única función continua estricta que completa el diagrama:

$$\begin{tikzcd}
D \arrow[d, "\mathbf{inl} \circ \mathbf{up}"'] \arrow[dr, "f"] & \\
D_\perp \oplus E_\perp \arrow[r, "h"] & F \\
E \arrow[u, "\mathbf{inr} \circ \mathbf{up}"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Puede que h no sea la única función continua que complete el diagrama. Además, dadas $f : D \to D'$ y $g : E \to E'$ continuas, podemos definir $f_\perp \oplus g_\perp : D_\perp \oplus E_\perp \to D'_\perp \oplus E'_\perp$ como la función que hace conmutativo el siguiente diagrama:

$$\begin{tikzcd}
D_\perp \arrow[r, "f_\perp"] \arrow[d, "\mathbf{inl}"'] & D'_\perp \arrow[d, "\mathbf{inl}"] \\
D_\perp \oplus E_\perp \arrow[r, "f_\perp \oplus g_\perp"'] & D'_\perp \oplus E'_\perp \\
E_\perp \arrow[r, "g_\perp"'] \arrow[u, "\mathbf{inr}"] & E'_\perp \arrow[u, "\mathbf{inr}"']
\end{tikzcd}$$

Observación 3.12. *Podemos generalizar todas las construcciones anteriores a mas de dos dcpos:*

$$D_1 \times D_2 \times D_3 \times \dots \qquad D_1 \otimes D_2 \otimes D_3 \otimes \dots$$

$$D_1 + D_2 + D_3 + \dots \qquad D_1 \oplus D_2 \oplus D_3 \oplus \dots \qquad D_{1_\perp} \oplus D_{2_\perp} \oplus D_{3_\perp} \oplus \dots$$

Ejercicios para nota

(1) Sea $D = \{(S,T) \in \mathcal{P}(\mathbb{N}) \times \mathcal{P}(\mathbb{N}) \,:\, S \cap T = \emptyset\}$. Consideramos sobre D el orden parcial inducido por el orden del dcpo producto $\mathcal{P}(\mathbb{N}) \times \mathcal{P}(\mathbb{N})$. Comprobar que D es un dcpo punteado isomorfo ("*similar*") a $(\mathbb{N}_\perp \circ \overset{\text{cont}}{\to} \mathbb{B}_\perp)$ donde $\mathbb{B}_\perp$ es el dcpo de los valores de la verdad $\mathbb{B}_\perp = \{\text{true}, \text{ false}, \perp\}$.

(2) Dar el diagrama de Hasse del orden parcial de los dcpos $II \times II$ y $\mathbb{B}_\perp \otimes \mathbb{B}_\perp$ donde $II = \{\perp, \top\}$ con $\perp \leq \top$ y $\mathbb{B}_\perp$ es el dcpo de los valores de la verdad.

(3) Sea $II = \{\perp, \top\}$ y $D = II \times II \times II \times \ldots$ el correspondiente dcpo producto.

- Sea M el subconjunto de D cuyos elementos son:

  $(\perp, \perp, \perp, \perp, \ldots)$
  $(\top, \perp, \perp, \perp, \ldots)$
  $(\top, \top, \perp, \perp, \ldots)$
  $(\top, \top, \top, \perp, \ldots)$
  $(\top, \top, \top, \top, \ldots)$
  $\vdots$

  Calcular $\vee M$.

- Definimos la función $f : D \to II$ como sigue:

$$f(z) = \begin{cases} \top & \text{si } z \text{ tiene infinitas componentes iguales a } \top \\ \perp & \text{en caso contrario} \end{cases}$$

  ¿Es f monótona? ¿Es f continua? ¿Es f continua en cada componente?

(4) Definir los conceptos de producto directo, producto smash, suma y suma fundida para un número finito de dcpos.

4. Teorema del Punto Fijo y Aplicaciones

Profesora: Alicia Tocino Sánchez. Curso Académico 2018-2019.

Definición 4.1. *Sea D un dcpo punteado y $f : D \to D$ una función. Se dice que $p \in D$ es un* ***punto fijo*** *de f si $f(p) = p$.*

Teorema 4.2. *Sea D un dcpoo punteado.*

(1) *(Teorema de Kleene-Scott) Sea $f : D \to D$ una función continua. Entonces el conjunto $M = \{f^n(\perp) : n = 0, 1, \ldots\}$ es dirigido y $\vee M$ es el menor punto fijo de f. Es decir, $f(\vee M) = \vee M$ y si existe $c' \in D$ tal que $f(c') = c'$, entonces $\vee M \leq c'$.*

(2) *La aplicación fix : $(D \xrightarrow{cont} D) \longrightarrow D$, que a toda función continua de D en D le asocia su menor punto fijo, es continua.*

Proof. Dado el conjunto $M = \{\perp, f(\perp), f^2(\perp), \ldots\}$ se verifica trivialmente que:

- $\perp \leq f(\perp) \leq f^2(\perp) \leq \ldots \leq f^n(\perp) \leq f^{n+1}(\perp) \leq \ldots$
- M es un conjunto dirigido (totalmente ordenado) y por estar dentro de un dcpo punteado existe el supremo de M.

Llamamos c al supremo de M ($c = \vee M$). Veamos que c es un punto fijo de f.

- $f(c) = f(\vee M) = \vee f(M)$ (la última igualdad se da porque f es continua)
- Como $f(M) = \{f(\perp), f^2(\perp), f^3(\perp), \ldots\}$ (sólo falta el elemento mínimo, $\perp$, para ser M) se tiene que $\vee M = \vee f(M)$.

Por lo tanto $\vee M$ es punto fijo de f, $f(\vee M) = \vee M$.

Falta ver que es el menor punto fijo. Supongamos que existe $c' \in D$ tal que $f(c') = c'$. Sabemos que $\perp \leq c'$. Aplicamos f a la desigualdad anterior repetidas veces.

$$\perp \leq c',\ f(\perp) \leq f(c') = c',\ f^2(\perp) \leq f^2(c') = c', \ldots, f^n(\perp) \leq c' \text{ para cada } n \in \mathbb{N}$$

De aquí que c' sea una cota superior de M. Por definición el supremo de M es el mínimo de las cotas superiores, entonces $c = \vee M \leq c'$.

El segundo apartado se propone como ejercicio. □

Aplicación del Teorema del Punto Fijo a funciones recursivas.

Como ejemplo ilustrativo usaremos la función factorial, $\mathrm{fact} : \mathbb{N} \to \mathbb{N}$. La recursividad es una forma más elegante de presentar un programa que usando funciónes iterativas, como el bucle *for*.

$$\mathrm{fact}(n) = \begin{cases} 1 & \text{si } n = 0 \\ n \cdot \mathrm{fact}(n-1) & \text{si } n > 0 \end{cases}$$

¿Cómo sabemos que existe una función *fact* que satisface la ecuación anterior? Aplicando el Teorema del Punto Fijo sabremos que existe y que es única. Para poder aplicar el teorema a una función recursiva debemos realizar los siguientes pasos.

Paso 1. Dada una función $f : \mathbb{N} \to \mathbb{N}$ podemos extenderla a una nueva función estricta.

$$\begin{aligned} f^* : \mathbb{N}_\perp &\longrightarrow \mathbb{N}_\perp \\ \perp &\longmapsto \perp \\ n &\longmapsto f(n) \end{aligned}$$

Paso 2. Sabemos que el conjunto de funciones continuas y estrictas entre dcpos punteados es un dcpo punteado por una Proposición del Tema 2.

Extendemos la función factorial a una función estricta,

$$\begin{aligned} \mathrm{fact} : \mathbb{N}_\perp &\longrightarrow \mathbb{N}_\perp \\ \perp &\longmapsto \perp \\ 0 &\longmapsto 1 \\ n &\longmapsto n \cdot \mathrm{fact}(n-1),\ n > 0 \end{aligned}$$

Sabemos que $\mathbb{N}_\perp$ es un dcpo punteado y que el conjunto de funciones continuas $\mathbb{N}_\perp \circ \stackrel{\text{cont}}{\to} \mathbb{N}_\perp$ es también un dcpo punteado. Para aplicar el Teorema del Punto Fijo construimos la función no recursiva F entre dcpos punteados como sigue.

$$
F : (\mathbb{N}_\perp \circ \stackrel{\text{cont}}{\to} \mathbb{N}_\perp) \longrightarrow (\mathbb{N}_\perp \circ \stackrel{\text{cont}}{\to} \mathbb{N}_\perp) \qquad f \longmapsto F(f)
$$

$$
\begin{aligned}
F(f) : \mathbb{N}_\perp &\longrightarrow \mathbb{N}_\perp \\
\perp &\longmapsto \perp \\
0 &\longmapsto 1 \\
n &\longmapsto n \cdot f(n-1),\ n > 0
\end{aligned}
$$

Debemos probar que F es continua para poder aplicar el Teorema del Punto Fijo (se propone como ejercicio). Aplicando el Teorema del Punto Fijo sabemos que existe un único punto fijo mínimo, f, tal que $F(f) = f$. Entonces existe f tal que $f(0) = 1$ y $f(n) = n \cdot f(n-1)$. Así hemos comprobado que la función factorial existe y es única.

En concreto, para F, definimos su **punto fijo** como $\text{Fix}(F)$ donde:

$$
\begin{aligned}
F : ((\mathbb{N}_\perp \circ \stackrel{\text{cont}}{\to} \mathbb{N}_\perp) \longrightarrow (\mathbb{N}_\perp \circ \stackrel{\text{cont}}{\to} \mathbb{N}_\perp)) &\longrightarrow (\mathbb{N}_\perp \circ \stackrel{\text{cont}}{\to} \mathbb{N}_\perp) \\
F &\longmapsto \text{Fix}(F) = \vee\{\perp, F(\perp), F^2(\perp), \ldots\}
\end{aligned}
$$

Aplicación del Teorema del Punto Fijo para demostrar el Teorema de Scroder-Bernstein. Mostramos otra de las aplicaciones del Teorema del Punto Fijo en la demostración del siguiente resultado.

Teorema 4.3 (de Scroder-Bernstein)**.** *Sean S y T conjuntos. Si $f : S \to T$ y $g : T \to S$ son inyectivas, entonces existe una biyección $h : S \to T$.*

Proof. La siguiente función es continua respecto al orden de inclusión dentro del conjunto de partes.

$$
\begin{aligned}
\mathcal{P}(T) &\longrightarrow \mathcal{P}(T) \\
Y &\longmapsto (T - f(S)) \cup f(g(Y))
\end{aligned}
$$

Como el conjunto de partes $\mathcal{P}(T)$ es un dcpo punteado y la función anterior es continua, podemos aplicar el Teorema del Punto Fijo para afirmar que existe un subconjunto Y tal que $Y = (T - f(S)) \cup f(g(Y))$. En concreto, $T - Y = f(S - g(Y))$ ya que:

$$
T - Y = T - ((T - f(S)) \cup f(g(Y))) = (T - (T - f(S))) \cap (T - (f(g(Y)))) =
$$

$$
= f(S) \cap (T - (f(g(Y)))) = f(S - g(Y))
$$

Definimos $h : S \to T$ por,

$$\mathrm{h}(x) = \begin{cases} y & \text{si } x = g(y) \text{ para algún } y \in Y \\ f(x) & \text{en otro caso} \end{cases}$$

Esto tiene sentido porque g es inyectiva. Además, h es inyectiva ya que está definida mediante f y g (ambas inyectivas). Para ver que es sobreyectiva supongamos $y \in T$. Si $y \in Y$, entonces $h(g(y)) = y$. Si $y \notin Y$, entonces $y \in f(S - g(Y))$ y así $y = f(x) = h(x)$ para algún x. Por lo tanto h es una biyección. $\square$

Ejemplo.

Sea D un dcpo punteado, $\mathbb{B}_\perp$ el dcpo punteado de los valores de verdad, $b : D \to \mathbb{B}_\perp$ una función continua y $c : D \to D$ una función continua estricta. Por una proposición del Tema 2 sabemos que la aplicación $F(\phi) : D \to D$ dada por,

$$F(\phi)(x) = \begin{cases} \perp & \text{si } b(x) = \perp \\ x & \text{si } b(x) = \text{false} \\ (\phi \circ c)(x) & \text{si } b(x) = \text{true} \end{cases}$$

es continua y estricta. Además, la aplicación $F : (D \circ \xrightarrow{\text{cont}} D) \longrightarrow (D \circ \xrightarrow{\text{cont}} D)$ que transforma cada $\phi \in (D \circ \xrightarrow{\text{cont}} D)$ en $F(\phi) \in (D \circ \xrightarrow{\text{cont}} D)$ es continua.

El Teorema del Punto Fijo permite afirmar la existencia de un menor punto fijo de F que es el supremo del conjunto dirigido

$$M = \{\perp, F(\perp), F^2(\perp), \ldots\}$$

Calculamos explícitamente el menor punto fijo de F. Mostramos, para empezar, que para cada $n \in \mathbb{N}$, $\phi \in (D \circ \xrightarrow{\text{cont}} D)$ y $x \in D$, el valor de $F^n(\phi)x$ está dado de la siguiente forma:

- **(i):** $F^n(\phi)(x) = \perp$ si existe un entero i_0 con $0 \leq i_0 < n$, $b \circ c^{i_0}(x) = \perp$ y tal que $b \circ c^i(x) = \text{true}$ para cada i tal que $0 \leq i < i_0$.
- **(ii):** $F^n(\phi)(x) = c^{i_0}(x)$ si existe un entero i_0 con $0 \leq i_0 < n$, $b \circ c^{i_0}(x) = \text{false}$ y tal que $b \circ c^i(x) = \text{true}$ para cada i tal que $0 \leq i < i_0$.
- **(iii):** $F^n(\phi)(x) = \phi \circ c^n(x)$ si se tiene que $b \circ c^i(x) = \text{true}$ para cada i tal que $0 \leq i < n$.

Lo demostramos por inducción sobre n.

- $F^0(\phi)(x) = \phi(x)$ para cada $x \in D$ y para cada $\phi \in (D \circ \xrightarrow{\text{cont}} D)$.
- $F^1(\phi)(x) = F(\phi)(x) = \begin{cases} \perp & \text{si } b(x) = \perp \\ x & \text{si } b(x) = \text{false} \\ (\phi \circ c)(x) & \text{si } b(x) = \text{true} \end{cases}$
- Hipótesis de inducción: suponemos el resultado cierto para $n \in \mathbb{N}$.
- Veamos que el resultado se verifica para $n+1$. Tenemos que,

  $$F^{n+1}(\phi)(x) = F(F^n(\phi))(x) = \begin{cases} \perp & \text{si } b(x) = \perp \\ x & \text{si } b(x) = \text{false} \\ (F^n(\phi) \circ c)(x) & \text{si } b(x) = \text{true} \end{cases}$$

  Si $b(x) = \text{true}$ se tiene que $F^{n+1}(\phi)(x) = F^n(\phi) \circ c(x)$ por la hipótesis de inducción obtenemos:

  - $F^{n+1}(\phi)(x) = \perp$, si existe un entero i_0 con $0 \leq i_0 < n+1$, $b \circ c^{i_0}(x) = \perp$ y para el que se tiene $b \circ c^{i+1}(x) = \text{true}$ para todo entero i tal que $0 \leq i < i_0$.
  - $F^{n+1}(\phi)(x) = c^{i_0}(x)$, si existe un entero i_0 con $0 \leq i_0 < n+1$, $b \circ c^{i_0}(x) = \text{false}$ y para el que se tiene $b \circ c^{i+1}(x) = \text{true}$ para todo entero i tal que $0 \leq i < i_0$.
  - $F^{n+1}(\phi)(x) = \phi \circ c^{n+1}(x)$, si se tiene que $b \circ c^i(x) = \text{true}$ para todo entero i tal que $0 \leq i < n+1$.

  Esto demuestra las igualdades dadas en $(i), (ii), (iii)$.

En el caso particular de $\phi = \perp_{(D \circ \xrightarrow{\text{cont}} D)}$ tenemos que,

$$F^n(\perp_{(D \circ \xrightarrow{\text{cont}} D)})(x) = \begin{cases} c^{i_0}(x) & \text{si existe } i_0 \text{ con } 0 \leq i_0, b \circ c^{i_0}(x) = \text{false} \\ & \text{y } b \circ c^{i_0}(x) = \text{true para } 0 \leq i < i_0 \\ \perp & \text{en otro caso} \end{cases}$$

ya que $\perp \circ c^n(x) = \perp$.

Finalmente,

$$\text{Fix}(F(x)) = \vee\{F^n(\perp)(x) \,:\, n = 0, 1, 2, \ldots\}$$

Y por tanto,

$$\mathrm{Fix}(F(x)) = \begin{cases} c^{i_0}(x) & \text{si existe } i_0 \text{ con } 0 \leq i_0, b \circ c^{i_0}(x) = \text{false} \\ & \text{y } b \circ c^{i_0}(x) = \text{true para } 0 \leq i < i_0 \\ \perp & \text{en otro caso} \end{cases}$$

Ejercicios para nota

(1) Dado D un dcpo punteado, demostrar que la aplicación $Fix : (D \to D) \longrightarrow D$ que asocia a cada función de D en D su menor punto fijo, es continua.

(2) Siguiendo el argumento de aplicar el Teorema del Punto Fijo para aplicarlo a la función factorial recursiva, demostrar que la función F es continua.

$$\begin{array}{rcl} F : (\mathbb{N}_\perp \circ \overset{\text{cont}}{\to} \mathbb{N}_\perp) & \longrightarrow & (\mathbb{N}_\perp \circ \overset{\text{cont}}{\to} \mathbb{N}_\perp) \\ f & \longmapsto & F(f) \end{array} \qquad \begin{array}{rcl} F(f) : \mathbb{N}_\perp & \longrightarrow & \mathbb{N}_\perp \\ \perp & \longmapsto & \perp \\ 0 & \longmapsto & 1 \\ n & \longmapsto & n \cdot f(n-1),\ n > 0 \end{array}$$

(3) Sea D un dcpo y $f : D \to D$ una aplicación continua. Comprobar que los puntos fijos de f constituyen un dcpo respecto del orden parcial inducido por el del dcpo. Mostrar que dicho dcpo es punteado en el caso en que D lo sea.

(4) Sea $G : (\mathbb{N}_\perp \circ \xrightarrow{\text{cont}} \mathbb{N}_\perp) \longrightarrow (\mathbb{N}_\perp \circ \xrightarrow{\text{cont}} \mathbb{N}_\perp)$ la función definida como sigue:

$$G(\phi)(x) = \begin{cases} 0 & \text{si } x = 0 \\ \phi(n-2) + 1 & \text{si } n \neq \perp, 0, 1 \text{ y } \phi(n-2) \neq \perp \\ \perp & \text{en otro caso} \end{cases}$$

Comprobar que G es continua y demostrar que tiene exactamente dos puntos fijos.

Dar la semántica de un lenguaje de programación es explicar la forma en la que puede usarse. Para ello, es preciso describir lo que hacen los programas escritos en dicho lenguaje cuando son ejecutados. Las definiciones precisas existentes sobre la semántica de los lenguajes se pueden dividir en tres grupos.

- La semántica de un lenguaje de programación viene descrita por el funcionamiento de una máquina virtual, más o menos abstracta. Es lo que se conoce como *semántica operacional.*
- Se define un espacio abstracto de significados primitivos (espacio de denotaciones), y se hace corresponder a cada objeto sintáctico primitivo algún objeto de dicho espacio de significados. Además, con el fin de interpretar las expresiones, a cada operador se le debe asociar una función sobre el espacio de denotaciones. A esto se le conoce como *semántica denotacional.*
- Se definen propiedades generales que deben cumplir las estructuras sintácticas y, por lo tanto, el programa. Estas propiedades se suelen describir mediante fórmulas lógicas que deben cumplirse antes (precondiciones) y después (postcondiciones) de ejecutarse la estructura. A esto se le conoce como *semántica axiomática.*

Se detallará la semántica denotacional para un Lenguaje de Programación Simple (LPS) como parte de los lenguajes imperativos en los que existe un único camino posible al ser ejecutados (no contiene errores de escritura). Una de las ventajas de dicha semántica es que se puede predecir el comportamiento de un programa sin necesidad de ejecutarlo. Se dirá que un lenguaje de programación está bien definido cuando la semántica denotacional y la semántica operacional coincidan.

Sintaxis de un LPS.

Primero describiremos la sintaxis de dicho lenguaje dando las *expresiones aritméticas*, las *expresiones lógicas o booleanas* y los *comandos* de los que consta.

(1) **Expresiones Aritméticas**

- $\mathbb{Z}_{\perp}$

- Var = {conjunto infinito de letras}
- $a_1 + a_2$ y $a_1 - a_2$ donde a_1 y a_2 son expresiones aritméticas
- $a_1 \cdot a_2$ donde a_1 y a_2 son exprsiones aritméticas

Si permitiéramos la división entre expresiones aritméticas deberíamos de considerar el conjunto $\mathbb{Q}_\perp$ en vez del conjunto de los números enteros.

(2) **Expresiones Booleanas**

- Valores de verdad: true, false
- Negación: $\neg b$ donde b es una expresión booleana
- Disyunción: $b_1 \vee b_2$ donde b_1 y b_2 son expresiones booleanas
- Menor que: $a_1 < a_2$ son expresiones aritméticas
- Igual que: $a_1 == a_2$ donde a_1 y a_2 son expresiones aritméticas

No incluimos la conjunción $\wedge$ ni la implicación porque se pueden obtener de las anteriores ($b_1 \wedge b_2 \equiv \neg(\neg b_1 \vee \neg b_2)$ y $b_1 \Rightarrow b_2 \equiv \neg(b_1 \vee b_2)$).

(3) **Comandos**

- *Skip* (no hace nada, salta)
- Asignación de expresión aritmética a una variable: $x := a$ donde $x \in$ Var y a es una expresión aritmética
- Condicional: $if\ b\ then\ c_1\ else\ c_2$ donde b es una expresión booleana y c_1, c_2 son comandos
- Composición: $c_1; c_2$ donde c_1 y c_2 son comandos (se ejecuta primero c_1 y después c_2)
- Iteración definida: $for\ a\ do\ c$ donde a es una expresión aritmética y c es un comando
- Iteración indefinida: $while\ b\ do\ c$ donde b es una expresión booleana y c es un comando

Semántica Denotacional de un LPS.

Para dar la semántica denotacional necesitamos conocer el concepto de *estado*. Sabiendo que el computador permite almacenar datos en sus posiciones de memoria, un **estado** caracteriza el contenido de todas las posiciones de memoria de un computador en un momento dado. La *semántica denotacional* asocia a todo comando una aplicación del conjunto de los estados en sí mismo. Dicha aplicación pretende capturar los cambios en los contenidos de memoria que produce el comando al ejecutarse.

Llamamos $\Sigma = (\mathrm{Var} \to \mathbb{Z}_\perp)$ al conjunto de estados del sistema donde $\sigma \in \Sigma$ es una función parcialmente definida que asocia valores de $\mathbb{Z}_\perp$ a las variables de nuestro LPS (σx es el valor de la variable x en el estado σ). A las variables no definidas les asociamos el valor $\perp$.

$$\begin{aligned} \sigma : \mathrm{Var} &\longrightarrow \mathbb{Z}_\perp \\ x &\longmapsto \sigma x \end{aligned}$$

(1) **Semántica denotacional de Expresiones Aritméticas**. Para cada expresión aritmética a definimos la semántica denotacional de a en el estado σ como $[[a]]^A\sigma$ donde:

$$\begin{aligned} [[\,]]^A : \text{Expr. Aritm.} &\longrightarrow (\Sigma \to \mathbb{Z}_\perp) \\ a &\longmapsto [[a]]^A \end{aligned} \qquad\qquad \begin{aligned} [[a]]^A : \Sigma &\longrightarrow \mathbb{Z}_\perp \\ \sigma &\longmapsto [[a]]^A\sigma \end{aligned}$$

Alternativamente, podemos definir $[[a]]^A\sigma$ como:

$$\begin{aligned} [[\,]]^A : (\text{Expr. Aritm.} \times \Sigma) &\longrightarrow \mathbb{Z}_\perp \\ (a,\sigma) &\longmapsto [[a]]^A\sigma \end{aligned}$$

Para simplificar la notación escribiremos $[[a]]\sigma$ para denotar la semántica denotacional de a en el estado σ.

- $n \in \mathbb{Z}_\perp \;\Rightarrow\; [[n]]\sigma = n$
- $x \in \mathrm{Var} \;\Rightarrow\; [[x]]\sigma = \sigma x$ (representa el valor de x en el estado σ)
- $a_1 + a_2 \;\Rightarrow [[a_1 + a_2]]\sigma = [[a_1]]\sigma + [[a_2]]\sigma$ (análogo para la resta)
- $a_1 \cdot a_2 \;\Rightarrow\; [[a_1 \cdot a_2]]\sigma = [[a_1]]\sigma \cdot [[a_2]]\sigma$

Ejemplo 5.1. *Sea la expresión aritmética $a = 3 - (xy+4)^2$ y sea $\sigma \in \Sigma$ tal que $\sigma x = -1$ y $\sigma y = 3$. Se verifica que,*

$$[[a]]\sigma = [[3 - (xy+4)^2]]\sigma = [[3]]\sigma - [[(xy+4)^2]]\sigma = 3 - ([[xy+4]]\sigma)^2 =$$

$$= 3 - ([[xy]]\sigma + [[4]]\sigma)^2 = 3 - ([[x]]\sigma[[y]]\sigma + 4)^2 = 3 - (\sigma x \cdot \sigma y + 4)^2 = 3 - ((-1)3 + 4)^2$$

(2) **Semántica denotacional de Expresiones Booleanas**. Para cada expresión booleana b definimos la semántica denotacional de b en el estado σ como $[[b]]^B\sigma$ donde:

$$\begin{array}{rcl} [[\,]]^B : \text{Expr. Bool.} & \longrightarrow & (\Sigma \to \{\text{true}, \text{false}\}) \\ b & \longmapsto & [[b]]^B \end{array} \qquad \begin{array}{rcl} [[b]]^B : \Sigma & \longrightarrow & \{\text{true}, \text{false}\} \\ \sigma & \longmapsto & [[b]]^B\sigma \end{array}$$

Alternativamente, podemos definir $[[b]]\sigma$ como:

$$\begin{array}{rcl} [[\,]]^B : (\text{Expr. Bool.} \times \Sigma) & \longrightarrow & \{\text{true}, \text{false}\} \\ (b, \sigma) & \longmapsto & [[b]]^B\sigma \end{array}$$

Para simplificar la notación escribiremos $[[b]]\sigma$ para denotar la semántica denotacional de b en el estado σ.

- $[[\text{true}]]\sigma = \text{true}$ y $[[\text{false}]]\sigma = \text{false}$
- $[[\neg b]]\sigma = \begin{cases} \text{true} & \text{si } [[b]]\sigma = \text{false} \\ \text{false} & \text{si } [[b]]\sigma = \text{true} \end{cases}$
- $[[b_1 \vee b_2]]\sigma = \begin{cases} \text{true} & \text{si } [[b_1]]\sigma = \text{true o } [[b_2]]\sigma = \text{true} \\ \text{false} & \text{si } [[b_1]]\sigma = \text{false y } [[b_2]]\sigma = \text{false} \end{cases}$
- $[[a_1 < a_2]]\sigma = \begin{cases} \text{true} & \text{si } [[a_1]]\sigma < [[a_2]]\sigma \\ \text{false} & \text{en caso contrario} \end{cases}$
- $[[a_1 == a_2]]\sigma = \begin{cases} \text{true} & \text{si } [[a_1]]\sigma = [[a_2]]\sigma \\ \text{false} & \text{en caso contrario} \end{cases}$

(3) **Semántica denotacional de Comandos**. Para cada comando c definimos la semántica denotacional de c en el estado σ como $[[c]]^C\sigma$ donde:

$$\begin{array}{rcl} [[\,]]^C : \text{Comandos} & \longrightarrow & (\Sigma \to \Sigma) \\ c & \longmapsto & [[c]]^C \end{array} \qquad \begin{array}{rcl} [[c]]^C : \Sigma & \longrightarrow & \Sigma \\ \sigma & \longmapsto & [[c]]^C\sigma = \sigma' \end{array}$$

Alternativamente, podemos definir $[[c]]^C\sigma$ como:

$$\begin{array}{rcl} [[\,]]^C : (\text{Comandos} \times \Sigma) & \longrightarrow & \Sigma \\ (c, \sigma) & \longmapsto & [[c]]^C\sigma \end{array}$$

Para simplificar la notación escribiremos $[[c]]\sigma$ para denotar la semántica denotacional de c en el estado σ.

- Función semántica de Skip:
$$\begin{aligned}[[Skip]] : \Sigma &\longrightarrow \Sigma \\ \sigma &\longmapsto \sigma\end{aligned} \qquad [[Skip]] = Id_\Sigma \Rightarrow [[Skip]]\sigma = \sigma$$
- Función semántica de asignación de valores:
$$\begin{aligned}[[x := n]] : \Sigma &\longrightarrow \Sigma \\ \sigma &\longmapsto [\sigma \,|\, x : n]\end{aligned} \qquad [[x := n]]\sigma = [\sigma \,|\, x : n]$$
donde $[\sigma \,|\, x : n]$ es un nuevo estado σ' que coincide con σ salvo en $x = n$.
- Función semántica del condicional:
$$[[\mathrm{if}\, b\, \mathrm{then}\, c_1\, \mathrm{else}\, c_2]]\sigma = \begin{cases} [[c_1]]\sigma & \text{si } [[b]]\sigma = \text{true} \\ [[c_2]]\sigma & \text{si } [[b]]\sigma = \text{false} \end{cases}$$
- Función semántica de la composición de comandos:
$$[[c_1; c_2]]\sigma = [[c_2]]([[c_1]]\sigma)$$
- Función semántica de la iteración definida:
$$[[\mathrm{for}\, a\, \mathrm{do}\, c]]\sigma = \begin{cases} ([[c]]^{[[a]]\sigma})\sigma & \text{si } [[a]]\sigma \geq 0 \\ \sigma & \text{si } [[a]]\sigma < 0 \end{cases}$$
- Detallaremos la función semántica de la iteración indefinida $[[\mathrm{while}\, b\, \mathrm{do}\, c]]\sigma =$ en el siguiente aparcado.

Función semántica de la iteración indefinida.

Como *primer intento* para definir la semántica de la iteración indefinida podemos pensar que,

$$\text{while } b \text{ do } c \equiv \text{ if } b \text{ then } (c; (\text{ while } b \text{ do } c))$$

De esta forma su semántica quedaría determinada por la semántica del condicional:

$$[[\text{while } b \text{ do } c]]\sigma = [[\mathrm{if}\, b\, \mathrm{then} (c;\ \text{while } b \text{ do } c)]]\sigma = \begin{cases} [[c;\ \text{while } b \text{ do } c]]\sigma & \text{si } [[b]]\sigma = \text{true} \\ \sigma & \text{si } [[b]]\sigma = \text{false} \end{cases}$$

Si llamamos $c_1 = \text{while } b \text{ do } c$ tendríamos que,

$$[[c_1]]\sigma = \begin{cases} [[c; c_1]]\sigma & \text{si } [[b]]\sigma = \text{true} \\ \sigma & \text{si } [[b]]\sigma = \text{false} \end{cases}$$

Veamos con un ejemplo concreto que la definición dada no nos sirve ya que no tiene una única solución.

Ejemplo 5.2. *Sea* $c_1 = (while \neg(x == 0)\ do\ x := x - 2)$*. Según lo mencionado anteriormente, su función semántica sería:*

$$[[c_1]]\sigma = \begin{cases} [[x := x-2; c_1]]\sigma & si\ [[\neg(x == 0)]]\sigma = true \\ \sigma & en\ caso\ contrario \end{cases} =$$

$$= \begin{cases} [[c_1]][\sigma \,|\, x\ :\ \sigma x - 2]\sigma & si\ [[\neg(x == 0)]]\sigma = true \\ \sigma & en\ caso\ contrario \end{cases}$$

Veamos que hay un conjunto infinito de funciones que verifican la igualdad anterior.

Sea $f : \Sigma \to \Sigma$ *definida como sigue,*

$$f\sigma \overset{(*)}{=} \begin{cases} f[\sigma \,|\, x\ :\ \sigma x - 2]\sigma & si\ [[\neg(x == 0)]]\sigma = true \\ \sigma & en\ caso\ contrario \end{cases}$$

Consideremos ahora otra función $f : \Sigma \to \Sigma$ *definida por,*

$$f\sigma \overset{(\circ)}{=} \begin{cases} [\sigma \,|\, x\ :\ 0] & si\ \sigma x \geq 0\ par \\ \sigma' & si\ \sigma x < 0\ par \\ \sigma'' & si\ \sigma x\ impar \end{cases}$$

Se comprueba que f *satisface la igualdad* $(*)$ *por casos:* $\sigma x > 0$ *par,* $\sigma x > 0$ *impar,* $\sigma x < 0$ *par,* $\sigma x < 0$ *impar y* $\sigma x = 0$ *obteniendo la siguiente conclusión.*

Conclusión*: La ecuación de recurrencia* $(*)$ *no define una única función de* Σ *en* Σ*. Existen infinitas soluciones.*

¿Cómo se arregla la definición de la función semántica de la iteración indefinida, $[[\text{while } b \text{ do } c]]\sigma$? Construyendo una función F adecuada para poder aplicar el **Teorema del Punto Fijo**. Sea F

la siguiente función,

$$\begin{array}{rcl} F : (\Sigma \to \Sigma) & \longrightarrow & (\Sigma \to \Sigma) \\ f & \longmapsto & F(f) : \Sigma \longrightarrow \Sigma \\ & & \quad \sigma \qquad \longmapsto F(f)\sigma \end{array}$$

definida por,

$$F(f)\sigma = \begin{cases} \sigma & \text{si } [[b]]\sigma = \text{false} \\ f([[c]]\sigma) & \text{si } [[b]]\sigma = \text{true} \end{cases}$$

para cada $f \in (\Sigma \to \Sigma)$ y para cada $\sigma \in \Sigma$.

Para poder afirmar que existe $\text{Fix}(F)$ y está bien definido hay que ver:

- $(\Sigma, \leq)$ es un dcpo punteado con el orden puntual propio de las funciones,

  $$\sigma, \tau \in \Sigma, \ \sigma \leq \tau \Leftrightarrow \sigma(x) \leq \tau(x)$$

  donde la función $\perp_\Sigma : \text{Var} \to \mathbb{Z}_\perp$ asigna a cada variable el valor $\perp \in \mathbb{Z}_\perp$, $\perp_\Sigma(x) = \perp_{\mathbb{Z}_\perp}$.
- $(\Sigma \to \Sigma, \leq)$ es un dcpo punteado con el orden,

  $$f_1, f_2 \in (\Sigma \to \Sigma), \ f_1 \leq f_2 \Leftrightarrow f_1(\sigma) \leq f_2(\sigma)$$

- $F : (\Sigma \longrightarrow \Sigma) \longrightarrow (\Sigma \longrightarrow \Sigma)$ descrita anteriormente es continua. (Se propone como ejercicio). Se puede probar utilizando una Proposición del Tema 2 (añadiendo que $\perp$ va a $\perp$).

De esta forma podemos aplicar el Teorema del Punto Fijo de Kleene-Scott para asegurar la existencia de un punto fijo mínimo de F y así concluir que $\mathbf{Fix}(F) = [[\mathbf{while}\ b\ \mathbf{do}\ c]]$ donde,

$$\begin{array}{rcl} \text{Fix} : ((\Sigma \to \Sigma) \longrightarrow (\Sigma \to \Sigma)) & \longrightarrow & (\Sigma \to \Sigma) \\ F & \longmapsto & \text{Fix}(F) = \vee\{\perp, F(\perp), F^2(\perp), \ldots\} \end{array}$$

Si llamamos $f = \text{Fix}(F)$ se cumple que:

$$f\sigma = \begin{cases} \sigma & \text{si } [[b]]\sigma = \text{false} \\ f([[c]]\sigma) & \text{si } [[b]]\sigma = \text{true} \end{cases}$$

En la siguiente proposición se describe una forma más manejable de definir la función semántica de la iteración indefinida.

Proposición 5.3. *Sea Σ el dcpo punteado de los estados, b una expresión booleana y c un comando. Para todo estado $\sigma \in \Sigma$ se tiene:*

(a): $[[\textit{while } b \textit{ do } c]]\sigma = \perp$ *si se satisface alguna de las dos alternativas siguientes:*

- $[[b]][[c]]^i\sigma = \textit{true}$ *para todo* $i \geq 0$ *o*
- $[[b]][[c]]^{i_0}\sigma = \perp$ *para algún* $i_0 \geq 0$ *verificándose además que* $[[b]][[c]]^i\sigma = \textit{true}$ *para cada* $i < i_0$.

(b): $[[\textit{while } b \textit{ do } c]]\sigma = [[c]]^n\sigma$ *si* n *es el menor entero para el que se tiene* $[[b]] \circ [[c]]^n\sigma = \textit{false}$ *y* $[[b]] \circ [[c]]^i\sigma = \textit{true}$ *para todo* $i < n$.

Ejemplos.

Iteración definida (for).

Sean n, m, z expresiones aritméticas. Consideramos el comando:

$$C = (z = 1;\ \text{if } (\neg(m == 0)) \text{ then } (\text{ for } m \text{ do } (z = z \cdot n)))$$

Supongamos que $a, b \in \mathbb{Z}$, $b \geq 0$ y $\sigma \in \Sigma$ tal que $\sigma n = a$ y $\sigma m = b$. Veamos que $[[C]]\sigma$ consiste en hacer a^b.

Supongamos $b = 0$, entonces $\sigma m = 0$ y por lo tanto, no se cumple la condición del condicional.

$$[[C]]\sigma = [[z = 1]]\sigma = [\sigma \,|\, z \,:\, 1] \Longleftrightarrow [[C]]\sigma(z) = 1 = a^0$$

Supongamos $b > 0$, entonces sí se cumple la condición del condicional.

$$[[C]]\sigma = [[\text{if} \neg(m == 0) \text{ then } (\text{ for } m \text{ do } (z = z \cdot n))]]([[z = 1]]\sigma) =$$

$$= [[\text{ for } m \text{ do } z = z \cdot n]][\sigma \,|\, z \,:\, 1] = ([[z = z \cdot n]]^{[[m]][\sigma \,|\, z \,:\, 1]})[\sigma \,|\, z \,:\, 1] =$$

$$= [[z = z \cdot n]]^{\sigma m}[\sigma \,|\, z \,:\, 1] = [[z = z \cdot n]]^b[\sigma \,|\, z \,:\, 1]$$

Debemos probar que $[[z = z \cdot n]]^b[\sigma \,|\, z \,:\, 1] = [\sigma \,|\, z \,:\, a^b]$. Usamos inducción sobre b:

- $b = 1 \Rightarrow [[z = z \cdot n]]^b[\sigma \,|\, z \,:\, 1] = [\sigma \,|\, z \,:\, \sigma z \cdot \sigma n \,,\, z \,:\, 1] = [\sigma \,|\, z \,:\, a]$

- Suponemos que la igualdad se verifica para $b-1$:
  $[[z = z\cdot n]]^{b-1}[\sigma\,|\,z\,:\,1] = [\sigma\,|\,z\,:\,a^{b-1}]$
- Veamos que la igualdad se verifica para b:
  $[[z = z\cdot n]]^{b}[\sigma\,|\,z\,:\,1] = [[z = z\cdot n]]([[z = z\cdot n]]^{b-1}[\sigma\,|\,z\,:\,1]) =$
  $= [[z = z\cdot n]][\sigma\,|\,z\,:\,a^{b-1}] = [\sigma\,|\,z\,:\,\sigma z\cdot\sigma n\,,\,\sigma z\,:\,a^{b-1}] =$
  $= [\sigma\,|\,z\,:\,a^{b-1}\cdot a] = [\sigma\,|\,z\,:\,a^{b}]$

Iteración indefinida (while).
Sean x, y, q, r expresiones aritméticas y $\sigma \in \Sigma$ tal que σx, $\sigma y \in \mathbb{Z}$, $\sigma x \geq 0$ y $\sigma y > 0$. Consideramos el comando:

$$C = (r = x; q = 0;\ \text{while}\ (y \leq r)\ \text{do}\ (r = r - y; q = q + 1))$$

Veamos que $[[C]]\sigma$ transforma q en el cociente de la división de σx por σy donde σr es el resto de la misma.
Sea $C_1 =$ while $(y \leq r)$ do $(r = r - y; q = q + 1)$. Se tiene que:

$$[[C]]\sigma = [[C_1]]([[q = 0]]([[r = x]]\sigma)) = [[C_1]][[q = 0]][\sigma\,|\,r\,:\,\sigma x] = [[C_1]][\sigma\,|\,r\,:\,\sigma x\,,\,q\,:\,0]$$

Sea C_2 el comando $(r = r - y; q = q + 1)$. Llamamos $\sigma' = [\sigma\,|\,r\,:\,\sigma x\,.\,q\,:\,0]$.

$$[[C_2]]\sigma' = [[q = q + 1]]([[r = r - y]]\sigma') = [[q = q + 1]][\sigma'\,|\,r\,:\,\sigma' r - \sigma' y] =$$
$$= [\sigma'\,|\,r\,:\,\sigma' r - \sigma' y\,,\,q\,:\,\sigma' q + 1] = [\sigma\,|\,r\,:\,\sigma x - \sigma y\,,\,q\,:\,1]$$

Se demuestra por inducción sobre $i \geq 0$ que,

$$[[C_2]]^i\sigma' = [[C_2]][[C_2]]^{i-1}\sigma' = [[C_2]][\sigma\,|\,r\,:\,\sigma x - (i-1)\sigma y, q\,:\,i-1] = [\sigma\,|\,r\,:\,\sigma x - i\sigma y,\,q\,:\,i]$$

Por lo tanto:

$$[[y \leq r]] \circ [[C_2]]^i\sigma' = \begin{cases} \text{ture} & \text{si } \sigma y \leq \sigma x - i\sigma y = \sigma r \\ \text{false} & \text{si } \sigma y > \sigma x - i\sigma y = \sigma r \end{cases}$$

Tomando k como el menor entero del conjunto $\{i \in \mathbb{N}\,:\,0 > \sigma x - (i+1)\sigma y\}$ estamos en la hipótesis de la Proposición anterior para calcular la semántica denotacional del *while.*

Las igualdades anteriores y la proposición anterio tenemos que:

$$[[C]]\sigma = [[C_1]]\sigma' = [[C_2]]^k\sigma' = [\sigma \,|\, r \,:\, \sigma x - k\cdot\sigma y,\, q \,:\, k]$$

Por lo tanto:

$$\sigma x - ([[C]]\sigma)(q)\sigma y = \sigma x - k\cdot\sigma y = [[C]]\sigma(r)$$

Por como se ha elegido k se satisfacen las desigualdades:

$$0 > \sigma x - (k+1)\sigma y \qquad\qquad 0 \leq \sigma x - k\sigma y$$

El algoritmo de la división en $\mathbb{Z}$ da que $[[C]]\sigma(q) = k$ coincide con el cociente de la división de σx por σy donde $[[C]]\sigma(r)$ es el resto.

Ejercicios para nota

(1) Probar que el conjunto de estados $\Sigma = (\mathrm{Var} \to \mathbb{Z}_\perp)$ con el orden puntual propio de las funciones es un dcpo punteado.

(2) Probar que la función F,

$$\begin{array}{rcl} F : (\Sigma \to \Sigma) & \longrightarrow & (\Sigma \to \Sigma) \\ f & \longmapsto & F(f) : \Sigma \longrightarrow \Sigma \\ & & \quad\;\; \sigma \;\;\longmapsto F(f)\sigma \end{array}$$

definida por,

$$F(f)\sigma = \begin{cases} \sigma & \text{si } [[b]]\sigma = \text{false} \\ f([[c]]\sigma) & \text{si } [[b]]\sigma = \text{true} \end{cases}$$

para cada $f \in (\Sigma \to \Sigma)$ y para cada $\sigma \in \Sigma$ es una función continua.

(3) Dados $a = (x+y)^2 - 3(x^3 + y^3)$, $\sigma \in \Sigma$ tal que $\sigma x = 1$, $\sigma y = 2$, calcular $[[a]]\sigma$ y $[[a]][\sigma \,|\, x : \sigma x + 3]$.

(4) Usar la semántica denotacional para probar que el comando:

$i = 0$; $b = \text{false}$;
    while $i \leq n$ do
        if $i \cdot i == n$ then $b = \text{true}$;
        $i = i + 1$

sirve para determinar cuándo un entero positivo dado (n) es un cuadrado perfecto.

References

[1] José Antonio Cuenca Mira, *Teoría de dominios y semántica denotacional*, Apuntes del Profesor, 2002.

[2] María de los Ángeles Gómez Molleda, *Notas manuscritas de clase.*

[3] C. A. Gunter y D. S. Scott, *Semantic Domains*, Handbook of Theoretical Computer Science, Volume B, 1990.

[4] José Antonio Jiménez Millán, *Compiladores y procesadores de lenguajes*, Servicio de Publicaciones de la Universidad de Cádiz, 2009.