

Formal Methods in software development



a.a.2017/2018

Prof. Anna Labella



See dens.pdf

Function cpo's and domains

Given cpo's (D, \sqsubseteq_D) and (E, \sqsubseteq_E) , the *function cpo* $(D \rightarrow E, \sqsubseteq)$ has underlying set

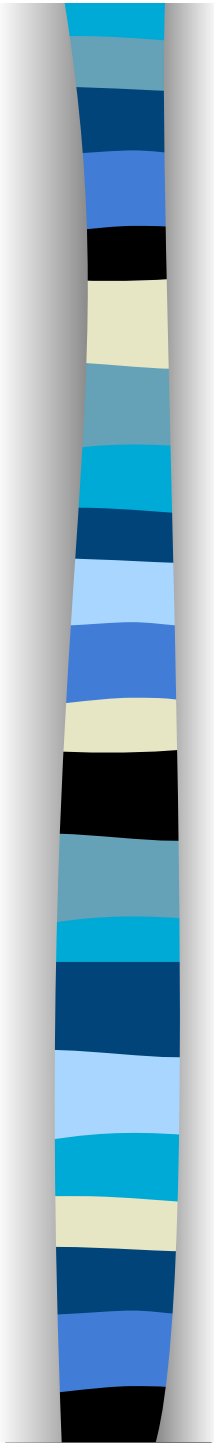
$$D \rightarrow E \stackrel{\text{def}}{=} \{f \mid f : D \rightarrow E \text{ is a } \textit{continuous} \text{ function}\}$$

and partial order: $f \sqsubseteq f' \stackrel{\text{def}}{\iff} \forall d \in D. f(d) \sqsubseteq_E f'(d)$.

Lubs of chains are calculated 'argumentwise' (using lubs in E):

$$\left(\bigsqcup_{n \geq 0} f_n\right)(d) = \bigsqcup_{n \geq 0} f_n(d).$$

If E is a domain, then so is $D \rightarrow E$ and $\perp_{D \rightarrow E}(d) = \perp_E$, all $d \in D$.



Proposition 3.2.1 (Evaluation and ‘Currying’). *Given cpo’s D and E , the function*

$$ev : (D \rightarrow E) \times D \rightarrow E$$

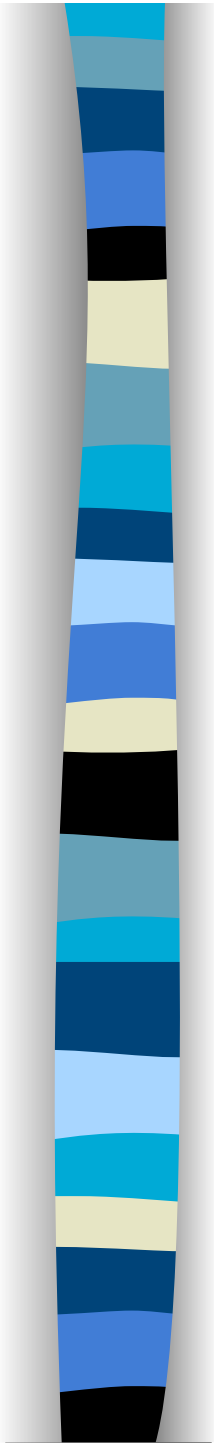
$$ev(f, d) \stackrel{\text{def}}{=} f(d)$$

is continuous. Given any continuous function $f : D' \times D \rightarrow E$ (with D' a cpo), for each $d' \in D'$ the function $d \in D \mapsto f(d', d)$ is continuous and hence determines an element of the function cpo $D \rightarrow E$ that we denote by $cur(f)(d')$. Then

$$cur(f) : D' \rightarrow (D \rightarrow E)$$

$$cur(f)(d') \stackrel{\text{def}}{=} \lambda d \in D . f(d', d)$$

is a continuous function.¹



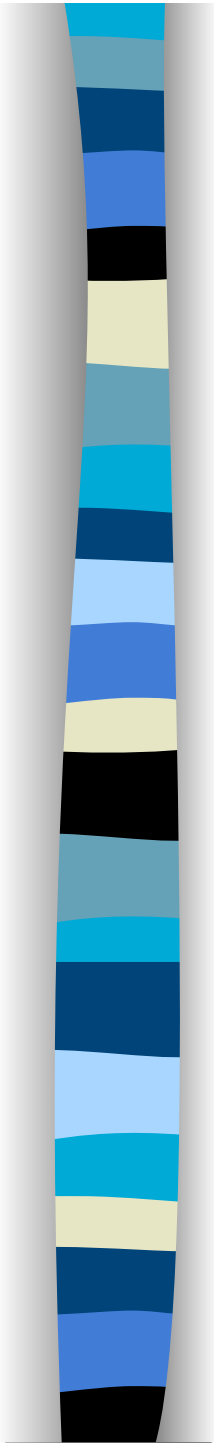
Continuity of the fixpoint operator

Proposition. *Let D be a domain. By Tarski's Fixed Point Theorem (Slide 13) we know that each continuous function $f \in (D \rightarrow D)$ possesses a least fixed point, $\text{fix}(f) \in D$.*

Then the function

$$\text{fix} : (D \rightarrow D) \rightarrow D$$

is continuous.



Discrete cpo's and flat domains

For any set X , the relation of equality

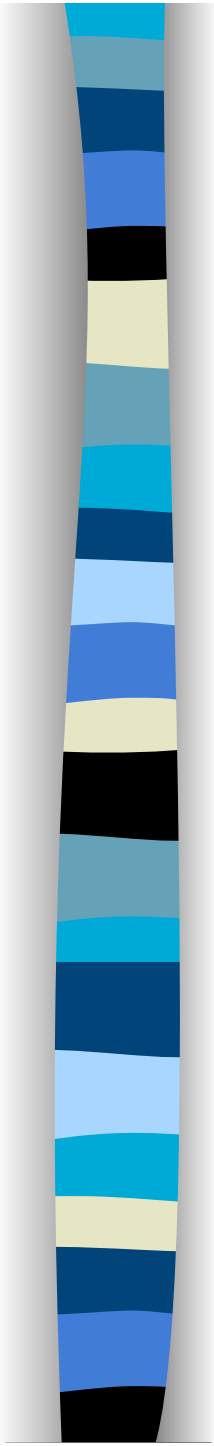
$$x \sqsubseteq x' \stackrel{\text{def}}{\Leftrightarrow} x = x' \quad (x, x' \in X)$$

makes (X, \sqsubseteq) into a cpo, called the *discrete* cpo with underlying set X .

Let $X_{\perp} \stackrel{\text{def}}{=} X \cup \{\perp\}$, where \perp is some element not in X . Then

$$d \sqsubseteq d' \stackrel{\text{def}}{\Leftrightarrow} (d = d') \vee (d = \perp) \quad (d, d' \in X_{\perp})$$

makes (X_{\perp}, \sqsubseteq) into a domain (with least element \perp), called the *flat* domain determined by X .



Proposition 3.3.1. *Let $f : X \rightarrow Y$ be a partial function between two sets. Then*

$$f_{\perp} : X_{\perp} \rightarrow Y_{\perp}$$

$$f_{\perp}(d) \stackrel{\text{def}}{=} \begin{cases} f(d) & \text{if } d \in X \text{ and } f \text{ is defined at } d \\ \perp & \text{if } d \in X \text{ and } f \text{ is not defined at } d \\ \perp & \text{if } d = \perp \end{cases}$$

defines a continuous function between the corresponding flat domains.

Proposition 3.3.2. *For each domain D the function*

$$\text{if} : \mathbb{B}_{\perp} \times (D \times D) \rightarrow D$$

$$\text{if}(x, (d, d')) \stackrel{\text{def}}{=} \begin{cases} d & \text{if } x = \text{true} \\ d' & \text{if } x = \text{false} \\ \perp_D & \text{if } x = \perp \end{cases}$$

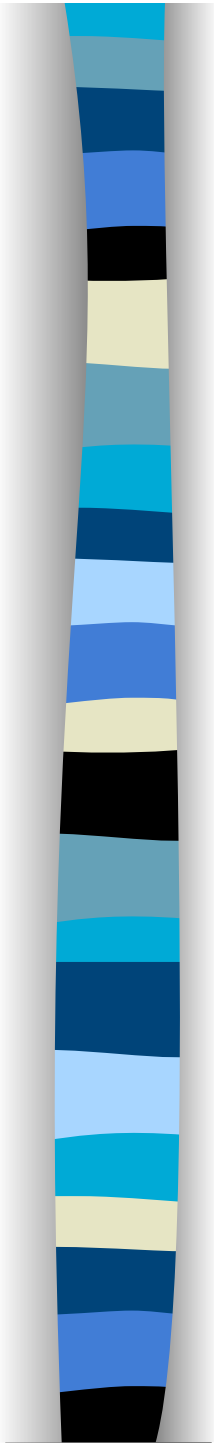
is continuous.



Exercises

Let X be a set and D a domain. Show that every monotone function $f : X_{\perp} \rightarrow D$ is continuous.

Let $f : X \rightarrow Y$ be a partial function between two sets X, Y . Show that $f_{\perp} : X_{\perp} \rightarrow Y_{\perp}$ is continuous and strict.

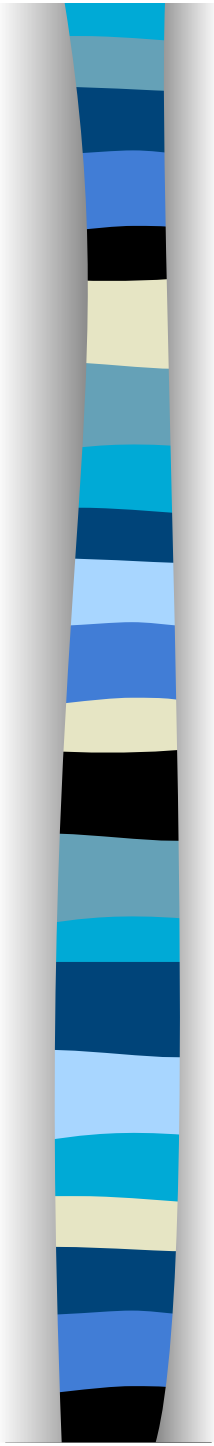


Styles of semantics

Operational. Meanings for program phrases defined in terms of the steps of computation they can take during program execution.

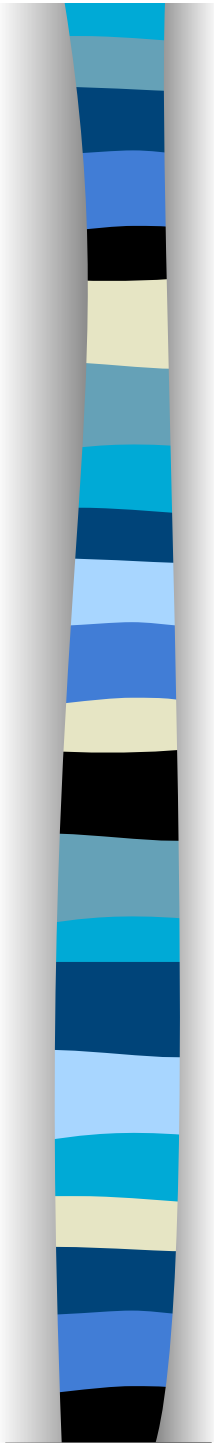
Axiomatic. Meanings for program phrases defined indirectly via the axioms and rules of some logic of program properties.

Denotational. Concerned with giving mathematical *models* of programming languages. Meanings for program phrases defined abstractly as elements of some suitable mathematical structure.



Characteristic features of a denotational semantics

- Each phrase (= part of a program), P , is given a *denotation*, $\llbracket P \rrbracket$ – a mathematical object representing the contribution of P to the meaning of *any* complete program in which it occurs.
- The denotation of a phrase is determined just by the denotations of its subphrases (one says that the semantics is *compositional*).



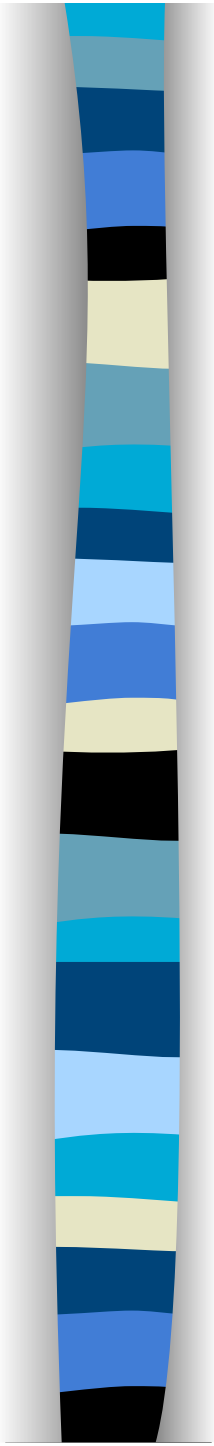
A simple example of compositionality

Given partial functions $\llbracket C \rrbracket, \llbracket C' \rrbracket : State \rightarrow State$ and a function $\llbracket B \rrbracket : State \rightarrow \{true, false\}$, we can define

$$\llbracket \text{if } B \text{ then } C \text{ else } C' \rrbracket = \\ \lambda s \in State. \text{if}(\llbracket B \rrbracket(s), \llbracket C \rrbracket(s), \llbracket C' \rrbracket(s))$$

where

$$\text{if}(b, x, x') = \begin{cases} x & \text{if } b = true \\ x' & \text{if } b = false \end{cases}$$



Denotational semantics of sequential composition

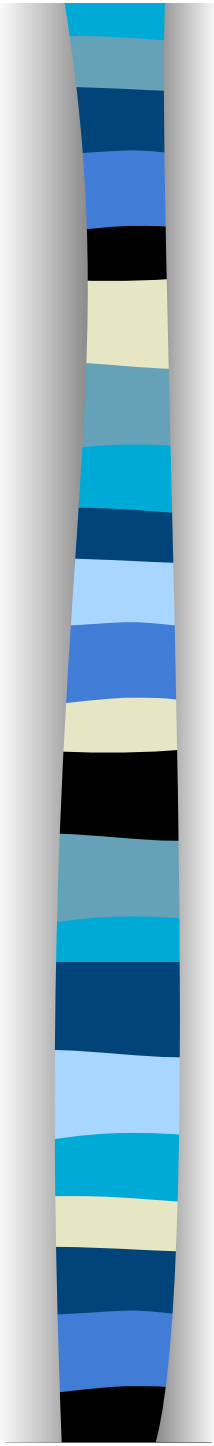
Denotation of sequential composition $C; C'$ of two commands

$$\llbracket C; C' \rrbracket = \llbracket C' \rrbracket \circ \llbracket C \rrbracket = \lambda s \in State. \llbracket C' \rrbracket(\llbracket C \rrbracket(s))$$

given by composition of the partial functions from states to states
 $\llbracket C \rrbracket, \llbracket C' \rrbracket : State \rightarrow State$ which are the denotations of the
commands.

Cf. operational semantics of sequential composition:

$$\frac{C, s \Downarrow s' \quad C', s' \Downarrow s''}{C; C', s \Downarrow s''} .$$



Fixed point property of $\llbracket \text{while } B \text{ do } C \rrbracket$

$$\llbracket \text{while } B \text{ do } C \rrbracket = f_{\llbracket B \rrbracket, \llbracket C \rrbracket}(\llbracket \text{while } B \text{ do } C \rrbracket)$$

where, for each $b : \text{State} \rightarrow \{\text{true}, \text{false}\}$ and $c, w : \text{State} \rightarrow \text{State}$, we define

$$f_{b,c}(w) = \lambda s \in \text{State}. \text{if}(b(s), w(c(s)), s).$$

- Why does $w = f_{\llbracket B \rrbracket, \llbracket C \rrbracket}(w)$ have a solution?
- What if it has several solutions—which one do we take to be $\llbracket \text{while } B \text{ do } C \rrbracket$?



An example

[[while $X > 0$ do ($Y := X * Y ; X := X - 1$)]]

Let

$State \stackrel{\text{def}}{=} \mathbb{Z} \times \mathbb{Z}$ pairs of integers

$D \stackrel{\text{def}}{=} State \rightarrow State$ partial functions.

For $[[\text{while } X > 0 \text{ do } Y := X * Y ; X := X - 1]] \in D$ we seek a minimal solution to $w = f(w)$, where $f : D \rightarrow D$ is defined by:

$$f(w)(x, y) = \begin{cases} (x, y) & \text{if } x \leq 0 \\ w(x - 1, x * y) & \text{if } x > 0. \end{cases}$$



Remember that

$$State \stackrel{\text{def}}{=} \mathbb{Z} \times \mathbb{Z} \quad D \stackrel{\text{def}}{=} State \rightarrow State$$

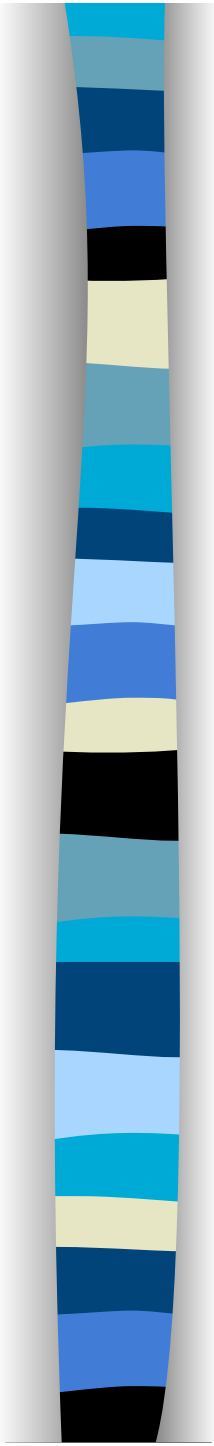
Partial order \sqsubseteq on D :

$w \sqsubseteq w'$ if and only if for all $(x, y) \in State$, if w is defined at (x, y) then so is w' and moreover $w(x, y) = w'(x, y)$.

Least element $\perp \in D$ w.r.t. \sqsubseteq :

$\perp \stackrel{\text{def}}{=} \text{totally undefined partial function}$

(satisfies $\perp \sqsubseteq w$, all $w \in D$).



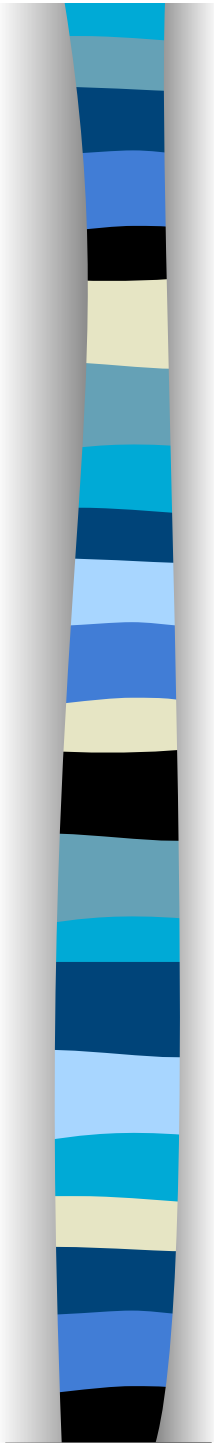
Starting with \perp , we apply the function f over and over again to build up a sequence of partial functions w_0, w_1, w_2, \dots :

$$\begin{cases} w_0 & \stackrel{\text{def}}{=} \perp \\ w_{n+1} & \stackrel{\text{def}}{=} f(w_n). \end{cases}$$

Using the definition of f on Slide 6, one finds that

$$w_1(x, y) = f(\perp)(x, y) = \begin{cases} (x, y) & \text{if } x \leq 0 \\ \text{undefined} & \text{if } x \geq 1 \end{cases}$$

$$w_2(x, y) = f(w_1)(x, y) = \begin{cases} (x, y) & \text{if } x \leq 0 \\ (0, y) & \text{if } x = 1 \\ \text{undefined} & \text{if } x \geq 2 \end{cases}$$



$$w_3(x, y) = f(w_2)(x, y) = \begin{cases} (x, y) & \text{if } x \leq 0 \\ (0, y) & \text{if } x = 1 \\ (0, 2 * y) & \text{if } x = 2 \\ \text{undefined} & \text{if } x \geq 3 \end{cases}$$

$$w_4(x, y) = f(w_3)(x, y) = \begin{cases} (x, y) & \text{if } x \leq 0 \\ (0, y) & \text{if } x = 1 \\ (0, 2 * y) & \text{if } x = 2 \\ (0, 6 * y) & \text{if } x = 3 \\ \text{undefined} & \text{if } x \geq 4 \end{cases}$$

and in general

$$w_n(x, y) = \begin{cases} (x, y) & \text{if } x \leq 0 \\ (0, (!x) * y) & \text{if } 0 < x < n \\ \text{undefined} & \text{if } x \geq n \end{cases}$$

where as usual, $!x$ is the factorial of x . Thus we get an increasing sequence of partial functions

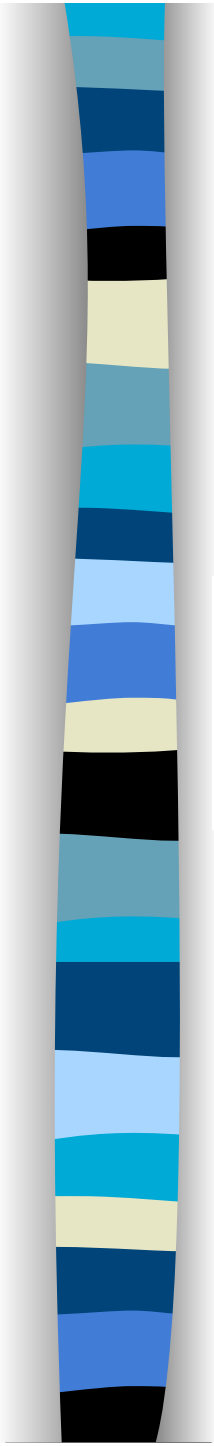
$$w_0 \sqsubseteq w_1 \sqsubseteq w_2 \sqsubseteq \dots \sqsubseteq w_n \sqsubseteq \dots$$

defined on larger and larger sets of states (x, y) and agreeing where they are defined. The union of all these partial functions is the element $w_\infty \in D$ given by

$$w_\infty(x, y) = \begin{cases} (x, y) & \text{if } x \leq 0 \\ (0, (!x) * y) & \text{if } x > 0. \end{cases}$$

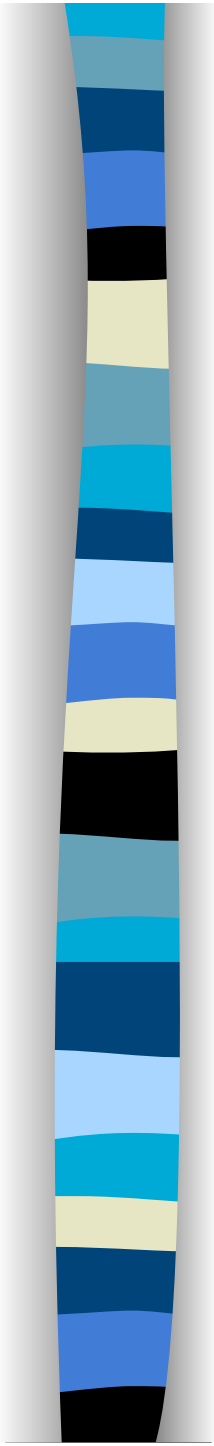
Note that w_∞ is a fixed point of the function f , since for all (x, y) we have

$$\begin{aligned} f(w_\infty)(x, y) &= \begin{cases} (x, y) & \text{if } x \leq 0 \\ w_\infty(x - 1, x * y) & \text{if } x > 0 \end{cases} && \text{(by definition of } f) \\ &= \begin{cases} (x, y) & \text{if } x \leq 0 \\ (0, 1 * y) & \text{if } x = 1 \\ (0, !(x - 1) * x * y) & \text{if } x > 1 \end{cases} && \text{(by definition of } w_\infty) \\ &= w_\infty(x, y). \end{aligned}$$



In fact one can show that w_∞ is the *least* fixed point of f , in the sense that for all $w \in D$

$$(3) \quad w = f(w) \quad \Rightarrow \quad w_\infty \sqsubseteq w.$$



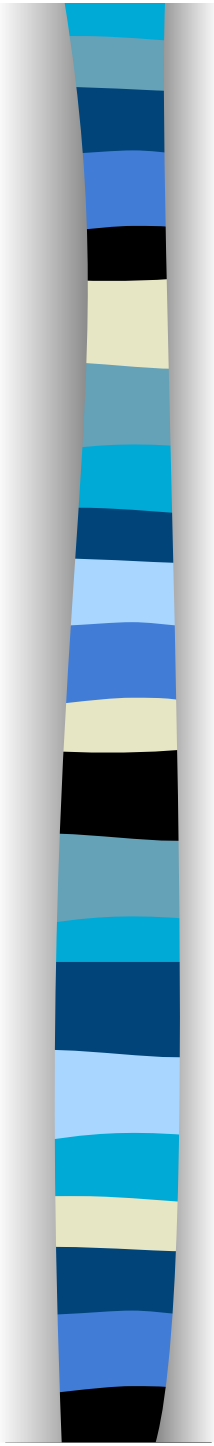
Chain-closed and admissible subsets

Let D be a cpo. A subset $S \subseteq D$ is called *chain-closed* iff for all chains $d_0 \sqsubseteq d_1 \sqsubseteq d_2 \sqsubseteq \dots$ in D

$$(\forall n \geq 0. d_n \in S) \Rightarrow \left(\bigsqcup_{n \geq 0} d_n \right) \in S$$

If D is a domain, $S \subseteq D$ is called *admissible* iff it is a chain-closed subset of D and $\perp \in S$.

A property $\Phi(d)$ of elements $d \in D$ is called *chain-closed/admissible* iff $\{d \in D \mid \Phi(d)\}$ is a chain-closed/admissible subset of D .



Scott's Fixed Point Induction Principle

Let $f : D \rightarrow D$ be a continuous function on a domain D .

For any admissible subset $S \subseteq D$, to prove that the least fixed point of f is in S , i.e. that

$$\text{fix}(f) \in S$$

it suffices to prove

$$\forall d \in D (d \in S \Rightarrow f(d) \in S).$$

Using Scott's induction

Example 4.2.1. Suppose that D is a domain and that $f : (D \times (D \times D)) \rightarrow D$ is a continuous function. Let $g : (D \times D) \rightarrow (D \times D)$ be the continuous function defined by

$$g(d_1, d_2) \stackrel{\text{def}}{=} (f(d_1, (d_1, d_2)), f(d_1, (d_2, d_2))) \quad (d_1, d_2 \in D).$$

Then $u_1 = u_2$, where $(u_1, u_2) \stackrel{\text{def}}{=} \text{fix}(g)$. (Note that g is continuous because we can express it in terms of composition, projections and pairing and hence apply Proposition 3.1.1 and Slide 37: $g = \langle f \circ \langle \pi_1, \langle \pi_1, \pi_2 \rangle \rangle, f \circ \langle \pi_1, \langle \pi_2, \pi_2 \rangle \rangle \rangle$.)

Proof. We have to show that $\text{fix}(g) \in \Delta$ where

$$\Delta \stackrel{\text{def}}{=} \{(d_1, d_2) \in D \times D \mid d_1 = d_2\}.$$

It is not hard to see that Δ is an admissible subset of the product domain $D \times D$. So by Scott's Fixed Point Induction Principle, we just have to check that

$$\forall (d_1, d_2) \in D \times D ((d_1, d_2) \in \Delta \Rightarrow g(d_1, d_2) \in \Delta)$$

or equivalently, that $\forall (d_1, d_2) \in D \times D (d_1 = d_2 \Rightarrow f(d_1, d_1, d_2) = f(d_1, d_2, d_2))$, which is clearly true. \square



Reverting to the example

Given the command `while $X > 0$ do ($Y := X * Y ; X := X - 1$)`

We prove the partial correctness of (7):

$$\forall x, y \geq 0. \text{fix}(f)(x, y) \neq \perp \Rightarrow \text{fix}(f)(x, y) = (0, (!x) * y)$$

$$S \stackrel{\text{def}}{=} \{w \in D \mid \forall x, y \geq 0. w(x, y) \neq \perp \Rightarrow w(x, y) = (0, (!x) * y)\}.$$

It is not hard to see that S is admissible. Therefore, to prove (7), by Scott Induction it suffices to check that $w \in S$ implies $f(w) \in S$, for all $w \in D$. So suppose $w \in S$, that $x, y \geq 0$, and that $f(w)(x, y) \neq \perp$. We have to show that $f(w)(x, y) = (0, (!x) * y)$. We consider the two cases $x = 0$ and $x > 0$ separately.

If $x = 0$, then by definition of f (See Slide 6)

$$f(w)(x, y) = (x, y) = (0, y) = (0, 1 * y) = (0, (!0) * y) = (0, (!x) * y).$$

On the other hand, if $x > 0$, then by definition of f

$$w(x - 1, x * y) = f(w)(x, y) \neq \perp \quad (\text{by assumption})$$

and then since $w \in S$ and $x - 1, x * y \geq 0$, we must have $w(x - 1, x * y) = (0, !(x - 1) * (x * y))$ and hence once again

$$f(w)(x, y) = w(x - 1, x * y) = (0, !(x - 1) * (x * y)) = (0, (!x) * y).$$



Example (cf. CST Pt II, 1988, p4, q3)

Let D be a domain and $p : D \rightarrow \mathbb{B}_\perp$, $h, k : D \rightarrow D$ be continuous functions, with h strict (i.e. $h(\perp) = \perp$).

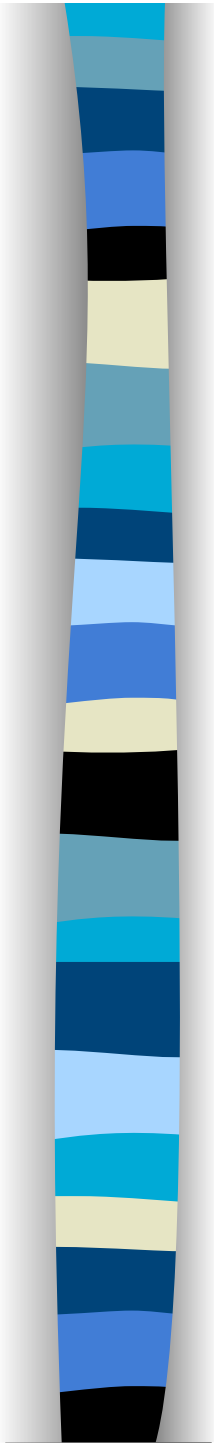
Let $f_1, f_2 : (D \times D) \rightarrow D$ be the least continuous functions such that for all $d_1, d_2 \in D$

$$f_1(d_1, d_2) = \text{if}(p(d_1), d_2, h(f_1(k(d_1), d_2)))$$

$$f_2(d_1, d_2) = \text{if}(p(d_1), d_2, f_2(k(d_1), h(d_2)))$$

$$\text{where } \text{if}(b, d_1, d_2) = \begin{cases} d_1 & \text{if } b = \text{true} \\ d_2 & \text{if } b = \text{false} . \\ \perp & \text{if } b = \perp \end{cases}$$

Then $f_1 = f_2$.



Let D , p , h , and k be as on Slide 22. Defining E to be the function domain $(D \times D) \rightarrow D$, let

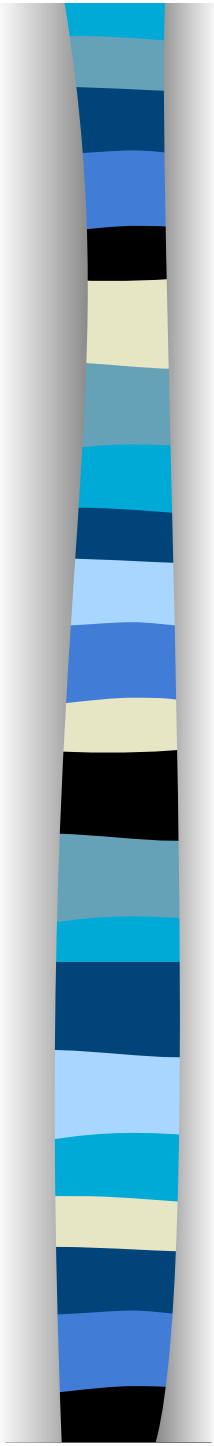
$$g \stackrel{\text{def}}{=} \langle g_1, g_2 \rangle : (E \times E) \rightarrow (E \times E)$$

where $g_1, g_2 : (E \times E) \rightarrow E$ are the continuous functions defined by

$$g_1(u_1, u_2)(d_1, d_2) \stackrel{\text{def}}{=} \begin{cases} d_2 & \text{if } p(d_1) = \text{true} \\ h(u_1(k(d_1), d_2)) & \text{if } p(d_1) = \text{false} \\ \perp & \text{if } p(d_1) = \perp \end{cases}$$

$$g_2(u_1, u_2)(d_1, d_2) \stackrel{\text{def}}{=} \begin{cases} d_2 & \text{if } p(d_1) = \text{true} \\ u_2(k(d_1), h(d_2)) & \text{if } p(d_1) = \text{false} \\ \perp & \text{if } p(d_1) = \perp \end{cases}$$

(all $u_1, u_2 \in E$ and $d_1, d_2 \in D$).



We have to prove that $fix(g)$ in the admissible set Δ

$$\forall (u_1, u_2) \in E \times E ((u_1, u_2) \in \Delta \Rightarrow g(u_1, u_2) \in \Delta)$$

$g_1(u, u)(d_1, d_2) = g_2(u, u)(d_1, d_2)$ holds provided

$$h(u(k(d_1), d_2)) = u(k(d_1), h(d_2))$$

This is not true in general,
hence we restrict ourselves to the set:

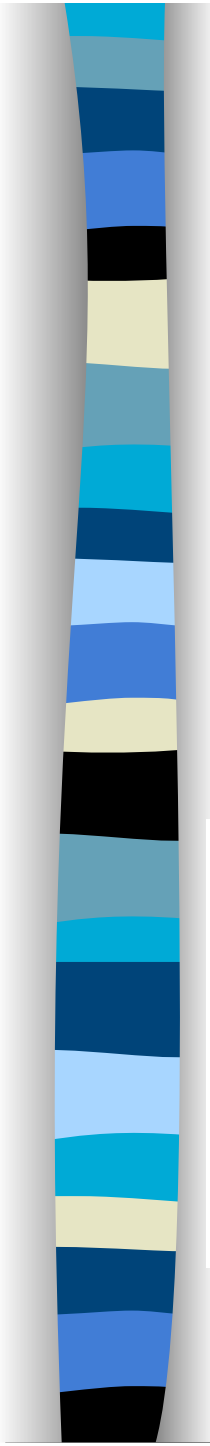
$$S \stackrel{\text{def}}{=} \{(u_1, u_2) \in E \times E \mid u_1 = u_2 \ \& \ \forall (d_1, d_2) \in D \times D \\ h(u_1(d_1, d_2)) = u_1(d_1, h(d_2))\}$$

We first have to check that S is admissible. It is chain-closed because if $(u_{1,0}, u_{2,0}) \sqsubseteq (u_{1,1}, u_{2,1}) \sqsubseteq (u_{1,2}, u_{2,2}) \sqsubseteq \dots$ is a chain in $E \times E$ each of whose elements is in S , then $\bigsqcup_{n \geq 0} (u_{1,n}, u_{2,n}) = (\bigsqcup_{i \geq 0} u_{1,i}, \bigsqcup_{j \geq 0} u_{2,j})$ is also in S since

$$\bigsqcup_{n \geq 0} u_{1,n} = \bigsqcup_{n \geq 0} u_{2,n} \quad (\text{because } u_{1,n} = u_{2,n}, \text{ each } n)$$

and

$$\begin{aligned} h\left(\bigsqcup_{n \geq 0} u_{1,n}\right)(d_1, d_2) &= h\left(\bigsqcup_{n \geq 0} u_{1,n}(d_1, d_2)\right) && \text{function lubs are argumentwise} \\ &= \bigsqcup_{n \geq 0} h(u_{1,n}(d_1, d_2)) && h \text{ is continuous} \\ &= \bigsqcup_{n \geq 0} u_{1,n}(d_1, h(d_2)) && \text{each } (u_{1,n}, u_{2,n}) \text{ is in } S \\ &= \left(\bigsqcup_{n \geq 0} u_{1,n}\right)(d_1, h(d_2)) && \text{function lubs are argumentwise.} \end{aligned}$$



Also, S contains the least element (\perp, \perp) of $E \times E$, because when $(u_1, u_2) = (\perp, \perp)$ clearly $u_1 = u_2$ and furthermore for all $(d_1, d_2) \in D \times D$

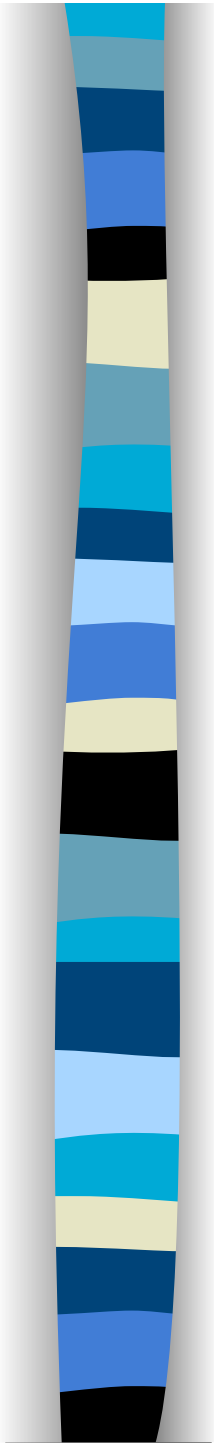
$$\begin{aligned} h(u_1(d_1, d_2)) &= h(\perp(d_1, d_2)) \\ &= h(\perp) && \text{by definition of } \perp \in (D \times D) \rightarrow D \\ &= \perp && h \text{ is strict, by assumption} \\ &= \perp(d_1, h(d_2)) && \text{by definition of } \perp \in (D \times D) \rightarrow D \\ &= u_1(d_1, h(d_2)). \end{aligned}$$

To prove $f_1 = f_2$ it is enough to show that $(f_1, f_2) = \text{fix}(g) \in S$; and since S is admissible, by Scott Induction it suffices to prove for all $(u_1, u_2) \in E \times E$ that

$$(u_1, u_2) \in S \Rightarrow (g_1(u_1, u_2), g_2(u_1, u_2)) \in S.$$

So suppose $(u_1, u_2) \in S$, i.e. that $u_1 = u_2$ and

$$(8) \quad \forall (d_1, d_2) \in D \times D. h(u_1(d_1, d_2)) = u_1(d_1, h(d_2)).$$



It is clear from the definition of g_1 and g_2 on Slide 23 that $u_1 = u_2$ and (8) imply $g_1(u_1, u_2) = g_2(u_1, u_2)$. So to prove $(g_1(u_1, u_2), g_2(u_1, u_2)) \in S$, we just have to check that $h(g_1(u_1, u_2)(d_1, d_2)) = g_1(u_1, u_2)(d_1, h(d_2))$ holds for all $(d_1, d_2) \in D \times D$. But

$$h(g_1(u_1, u_2)(d_1, d_2)) = \begin{cases} h(d_2) & \text{if } p(d_1) = \textit{true} \\ h(h(u_1(k(d_1), d_2))) & \text{if } p(d_1) = \textit{false} \\ h(\perp) & \text{if } p(d_1) = \perp \end{cases}$$

$$g_1(u_1, u_2)(d_1, h(d_2)) = \begin{cases} h(d_2) & \text{if } p(d_1) = \textit{true} \\ h(u_1(k(d_1), h(d_2))) & \text{if } p(d_1) = \textit{false} \\ \perp & \text{if } p(d_1) = \perp. \end{cases}$$

So since $h(h(u_1(k(d_1), d_2))) = h(u_1(k(d_1), h(d_2)))$ by (8), and since $h(\perp) = \perp$, we get the desired result. \square



Exercises

Exercise 4.4.2. Give an example of a subset $S \subseteq D \times D'$ of a product cpo that is not chain-closed, but which satisfies:

- (a) for all $d \in D$, $\{d' \mid (d, d') \in S\}$ is a chain-closed subset of D' ; and
 - (b) for all $d' \in D'$, $\{d \mid (d, d') \in S\}$ is a chain-closed subset of D .
5. Let D, D' be domains. We say that a function $f : D \rightarrow D'$ is a *continuous isomorphism* if it is continuous, bijective, and its inverse $f^{-1} : D' \rightarrow D$ is also continuous.
- (a) Show that if f is continuous and bijective, and f^{-1} is monotone, then f is a continuous isomorphism.
 - (b) Find an example for a continuous and bijective f that is not a continuous isomorphism.