

# Semantyka i weryfikacja programów

**Bartosz Klin**

(slajdy Andrzeja Tarleckiego)

Instytut Informatyki

Wydział Matematyki, Informatyki i Mechaniki

Uniwersytet Warszawski

<http://www.mimuw.edu.pl/~klin>

pok. 5680

[klin@mimuw.edu.pl](mailto:klin@mimuw.edu.pl)

Strona tego wykładu:

<http://www.mimuw.edu.pl/~klin/sem18-19.html>

# Program Semantics & Verification

**Bartosz Klin**

(slides courtesy of Andrzej Tarlecki)

Institute of Informatics  
Faculty of Mathematics, Informatics and Mechanics  
University of Warsaw

<http://www.mimuw.edu.pl/~klin>

office: 5680

[klin@mimuw.edu.pl](mailto:klin@mimuw.edu.pl)

This course:

<http://www.mimuw.edu.pl/~klin/sem18-19.html>

## Fixpoints treated seriously

Standard semantics for **while**:

$$\mathcal{S}[\mathbf{while} \ b \ \mathbf{do} \ S] = \mathit{fix}(\Phi) = \bigcup_{n \geq 0} \Phi^n(\emptyset_{\mathbf{State} \rightarrow \mathbf{State}})$$

where  $\Phi: \mathbf{STMT} \rightarrow \mathbf{STMT}$  is defined as follows:

$$\Phi(F) = \mathit{cond}(\mathcal{B}[b], \mathcal{S}[S]; F, \mathit{id}_{\mathbf{State}})$$

- Does a fixed point always exist?

*For “continuous”  $\Phi$ , yes*

- If a fixed point exists, is it unique?

*No, but the least one exists*

## More fixpoint problems

Consider a language where (non-recursive) procedures take procedural parameters.

$$S \in \mathbf{Stmt} ::= \dots \mid \mathbf{call} \ p(q)$$
$$D_P \in \mathbf{PDecl} ::= \mathbf{proc} \ p(q) \ \mathbf{is} \ (S); D_P \mid \varepsilon$$

Let's forbid  $\mathbf{proc} \ p(p) \ \mathbf{is} \ (S)$

$$\mathbf{PEnv} = \mathbf{IDE} \rightarrow (\mathbf{PROC} + \{?\})$$
$$\mathbf{PROC} = \mathbf{PROC} \rightarrow \mathbf{Store} \rightarrow (\mathbf{Store} + \{?\})$$
$$\mathcal{S}[\mathbf{call} \ p(q)] \ \rho_V \ \rho_P = (\rho_P \ p)(\rho_P \ q)$$
$$\mathcal{D}_P[\mathbf{proc} \ p(q) \ \mathbf{is} \ (S); D_P] \ \rho_V \ \rho_P =$$
$$\mathcal{D}_P[D_P] \ \rho_V \ \rho_P [p \mapsto P] \ \mathbf{where} \ P \ Q = \mathcal{S}[S] \ \rho_V \ \rho_P [q \mapsto Q]$$

Oops!  $\mathbf{call} \ p(p)$  makes no mathematical sense now!

## No easy way out

We may try:

$$\mathbf{PROC}_0 = \mathbf{Store} \rightarrow (\mathbf{Store} + \{??\})$$

$$\mathbf{PROC}_1 = \mathbf{PROC}_0 \rightarrow \mathbf{Store} \rightarrow (\mathbf{Store} + \{??\})$$

$$\mathbf{PROC}_2 = \mathbf{PROC}_1 \rightarrow \mathbf{Store} \rightarrow (\mathbf{Store} + \{??\})$$

...

$$\mathbf{PROC} = \bigcup_{n \in \mathbb{N}} \mathbf{PROC}_n$$

But then

$$\mathcal{S}[\mathbf{call} \ p(p)] \ \rho_V \ \rho_P = (\rho_P \ p)(\rho_P \ p)$$

does not fit in the domain.

We need domains such as  $D \cong D \rightarrow D$ .

## Complete partial orders

An ( $\omega$ -chain-)complete partial order, cpo:

$$\mathbf{D} = \langle D, \sqsubseteq, \perp \rangle$$

- $\sqsubseteq \subseteq D \times D$  is a partial order on  $D$  such that each countable chain  $d_0 \sqsubseteq d_1 \sqsubseteq \dots \sqsubseteq d_i \sqsubseteq \dots$  has the least upper bound  $\bigsqcup_{i>0} d_i$  in  $D$
- $\perp \in D$  is the least element w.r.t.  $\sqsubseteq$

**BTW:** Equivalently: all countable *directed* subsets of  $D$  have lub's in  $D$ .

( $\Delta \subseteq D$  is *directed* if for every  $x, y \in \Delta$ , there is  $d \in \Delta$  with  $x \sqsubseteq d$  and  $y \sqsubseteq d$ .)

**BTW:** It is *not* equivalent to require that *all* chains have lub's in  $D$ .

( $C \subseteq D$  is a *chain* if for every  $x, y \in C$ ,  $x \sqsubseteq y$  or  $y \sqsubseteq x$ .)

But it is equivalent to require that *all* countable chains have lub's in  $D$ .

## Examples

Examples	Non-examples	Comments
$\langle \mathcal{P}(X), \subseteq, \emptyset \rangle$	$\langle \mathcal{P}_{fin}(X), \subseteq, \emptyset \rangle$	$\mathcal{P}(X)$ is the set of all subsets, and $\mathcal{P}_{fin}(X)$ of all finite subsets of $X$
$\langle X \rightharpoonup Y, \subseteq, \emptyset_{X \rightharpoonup Y} \rangle$	$\langle X \rightarrow Y, \subseteq, ??? \rangle$	partial and total function spaces
$\langle \mathbf{Nat}^\infty, \leq, 0 \rangle$	$\langle \mathbf{Nat}, \leq, 0 \rangle$	$\mathbf{Nat}^\infty = \mathbf{Nat} \cup \{\omega\}$ ; $n \leq \omega$ , for all $n \in \mathbf{Nat}$
$\langle (\mathbf{R}^+)^\infty, \leq, 0 \rangle$	$\langle (\mathbf{Q}^+)^\infty, \leq, 0 \rangle$	non-negative reals $\mathbf{R}^+$ and rationals $\mathbf{Q}^+$ with “infinity”
$\langle (\mathbf{R}^+)^{\leq a}, \leq, 0 \rangle$	$\langle (\mathbf{Q}^+)^{\leq a}, \leq, 0 \rangle$	their bounded versions
$\langle A^{\leq \omega}, \sqsubseteq, \varepsilon \rangle$	$\langle A^*, \sqsubseteq, \varepsilon \rangle$	$A^{\leq \omega} = A^* \cup A^\omega$ (finite and infinite strings of elements from $A$ , including the empty string $\varepsilon$ ); $\sqsubseteq$ is the prefix ordering

## Continuous functions

Given cpo's  $\mathbf{D} = \langle D, \sqsubseteq, \perp \rangle$  and  $\mathbf{D}' = \langle D', \sqsubseteq', \perp' \rangle$ , a function  $f: D \rightarrow D'$  is

- *monotone* if it preserves the ordering, i.e., for all  $d_1, d_2 \in D$ ,

$$d_1 \sqsubseteq d_2 \text{ implies } f(d_1) \sqsubseteq' f(d_2)$$

- *continuous* if it preserves lub's of all countable chains, i.e., for each chain  $d_0 \sqsubseteq d_1 \sqsubseteq \dots$  in  $D$ ,

$$f(\bigsqcup_{i \geq 0} d_i) = \bigsqcup_{i \geq 0} f(d_i)$$

- *strict* if it preserves the least element, i.e.,

$$f(\perp) = \perp'$$

**BTW:** Continuous functions are monotone; in general they need not be strict.

**BTW:** Monotone functions in general need not be continuous.



## Some intuition?

### Topology

Given a cpo  $\mathbf{D} = \langle D, \sqsubseteq, \perp \rangle$ , define a set  $X \subseteq D$  to be *open* if

- if  $d_1 \in X$  and  $d_1 \sqsubseteq d_2$  then  $d_2 \in X$
- if  $d_0 \sqsubseteq d_1 \sqsubseteq \dots$  is such that  $\bigsqcup_{i \geq 0} d_i \in X$  then  $d_i \in X$  for some  $i \geq 0$ .

This defines a topology on  $D$ :

- $\emptyset$  and  $D$  are open
- intersection of two open sets is open
- union of any family of open sets is open

Given two cpo's  $\mathbf{D} = \langle D, \sqsubseteq, \perp \rangle$  and  $\mathbf{D}' = \langle D', \sqsubseteq', \perp' \rangle$ , a function  $f: D \rightarrow D'$  is continuous if and only if it is continuous in the topological sense, i.e., for  $X' \subseteq D'$  open in  $\mathbf{D}'$ , its co-image w.r.t.  $f$ ,  $f^{-1}(X') \subseteq D$  is open in  $\mathbf{D}$ .

## More intuition?

### *Information theory*

Think of a cpo  $\mathbf{D} = \langle D, \sqsubseteq, \perp \rangle$  as an “information space”.

- if  $d_1 \sqsubseteq d_2$  then  $d_2$  represents “more information” than  $d_1$ ;  $\perp$  is “no information”
- directed sets represent consistent sets of “information pieces”; their lub’s represent “information” that can be derived from the “informations” in the set
- a function is monotone if it yields more information when given more information
- a function is continuous if it deals with information “bit-by-bit”

For a set of elements  $X$ , consider the cpo  $\langle \mathcal{P}(X), \supseteq, X \rangle$  of “informations” about the elements in  $X$  (a set  $I \subseteq X$  represents the property — information — that holds for all the elements in  $I$ , and only for those elements).

## Best intuition?

### *Partial functions*

$$\langle X \rightharpoonup Y, \subseteq, \emptyset_{X \rightharpoonup Y} \rangle$$

- $\emptyset_{X \rightharpoonup Y}$  is nowhere defined
- given two partial functions  $f, g: X \rightharpoonup Y$ ,  $f \subseteq g$  if  $g$  is more defined than  $f$ , but when  $f$  is defined,  $g$  yields the same result
- given a directed set of partial functions  $\mathcal{F} \subseteq X \rightharpoonup Y$ , no two functions in  $\mathcal{F}$  yield different results for the same argument; then  $\sqcup \mathcal{F} = \bigcup \mathcal{F}$ , which is a partial function in  $X \rightharpoonup Y$
- a function  $F: (X \rightharpoonup Y) \rightarrow (X' \rightharpoonup Y')$  is continuous, if  $F(f)(x')$  (for  $f: X \rightharpoonup Y$  and  $x' \in X'$ ) depends only on a finite number of applications of  $f$  to arguments in  $X$ . Typical non-continuous functions: testing definedness, checking infinitely many values, ...

## Fixed point theorem

**Fact:** Given a cpo  $\mathbf{D} = \langle D, \sqsubseteq, \perp \rangle$  and a continuous function  $f: D \rightarrow D$ , there exists the least fixed point  $\text{fix}(f) \in D$  of  $f$ , i.e.,

- $f(\text{fix}(f)) = \text{fix}(f)$
- if  $f(d) = d$  for some  $d \in D$  then  $\text{fix}(f) \sqsubseteq d$

**Proof:**

Define  $f^0(\perp) = \perp$ , and  $f^{i+1}(\perp) = f(f^i(\perp))$  for  $i \geq 0$ . This yields a chain:

$$f^0(\perp) \sqsubseteq f^1(\perp) \sqsubseteq \dots \sqsubseteq f^i(\perp) \sqsubseteq f^{i+1}(\perp) \sqsubseteq \dots$$

Put:

$$\text{fix}(f) = \bigsqcup_{i \geq 0} f^i(\perp)$$

- $f(\text{fix}(f)) = f(\bigsqcup_{i \geq 0} f^i(\perp)) = \perp \sqcup \bigsqcup_{i \geq 0} f(f^i(\perp)) = \bigsqcup_{i \geq 0} f^i(\perp) = \text{fix}(f)$
- Suppose  $f(d) = d$  for some  $d \in D$ ; then  $f^i(\perp) \sqsubseteq d$  for  $i \geq 0$ . Thus  $\text{fix}(f) = \bigsqcup_{i \geq 0} f^i(\perp) \sqsubseteq d$ .

## Proof techniques

Given a cpo  $\mathbf{D} = \langle D, \sqsubseteq, \perp \rangle$  and a continuous function  $f: D \rightarrow D$ .

**Fact:** For any  $d \in D$ , if  $f(d) \sqsubseteq d$  then  $\text{fix}(f) \sqsubseteq d$ .

### Fixed point induction

A property  $P \subseteq D$  is *admissible* if it is preserved by lub's of all countable chains: for any chain  $d_0 \sqsubseteq d_1 \sqsubseteq \dots$ , if  $d_i \in P$  for all  $i \geq 0$  then also  $\bigsqcup_{i \geq 0} d_i \in P$ , and  $\perp \in P$ .

**Fact:** For any admissible  $P \subseteq D$  that is closed under  $f$  (i.e., if  $d \in P$  then  $f(d) \in P$ )

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

## Semantics of while

Recall the (original direct) semantic clause for **while**:

$$\mathcal{S}[\mathbf{while} \ b \ \mathbf{do} \ S] = \mathit{fix}(\Phi)$$

where  $\Phi: \mathbf{STMT} \rightarrow \mathbf{STMT}$  is given by  $\Phi(F) = \mathit{cond}(\mathcal{B}[b], \mathcal{S}[S]; F, \mathit{id}_{\mathbf{State}})$ .

Is  $\mathbf{STMT}$  a cpo?

Is  $\Phi$  continuous?

In this case we can easily check that indeed  $\langle \mathbf{STMT}, \subseteq, \emptyset_{\mathbf{State} \rightarrow \mathbf{State}} \rangle$  is a cpo and  $\Phi: \mathbf{STMT} \rightarrow \mathbf{STMT}$  is continuous.

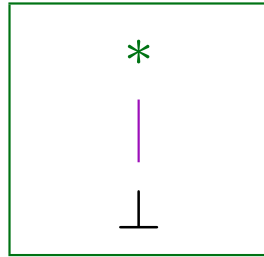
**BUT:** we do not want to have to check this  
each time we use a fixed point definition!

# Domain constructors

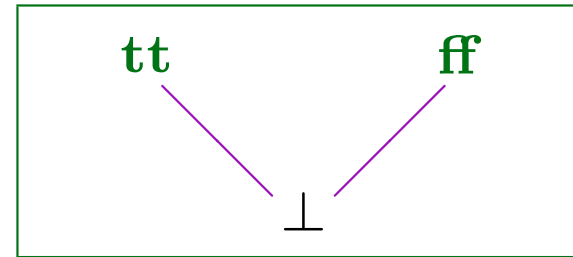
## Basic domains

For any set  $X$ ,  $\mathbf{X}_\perp = \langle X_\perp, \sqsubseteq, \perp \rangle$  is a *flat cpo*, where  $X_\perp = X \cup \{\perp\}$ ,  $\perp$  is a new element,  $\perp \sqsubseteq a$  for all  $x \in X$  and otherwise  $\sqsubseteq$  is trivial.

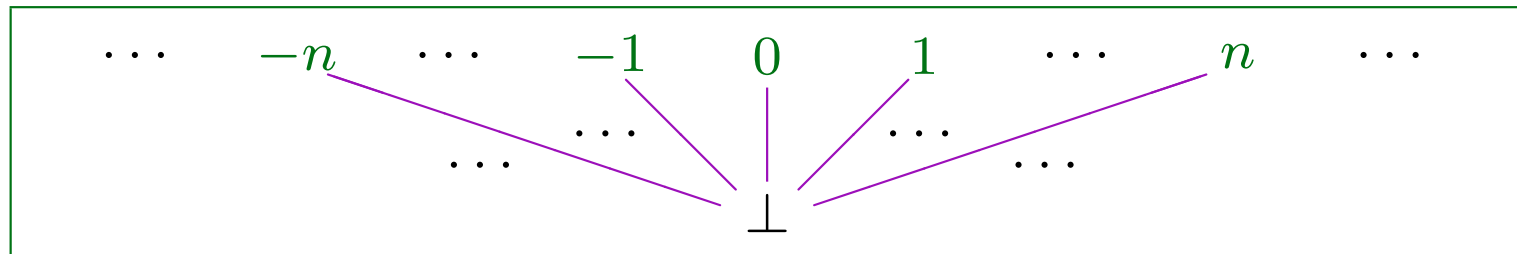
$\{*\}_\perp$ :



$\mathbf{Bool}_\perp$ :



$\mathbf{Int}_\perp$ :



**Fact:** Every monotone function defined on a flat cpo is continuous.

For any cpo's  $\mathbf{D}_1 = \langle D_1, \sqsubseteq_1, \perp_1 \rangle$  and  $\mathbf{D}_2 = \langle D_2, \sqsubseteq_2, \perp_2 \rangle$ :

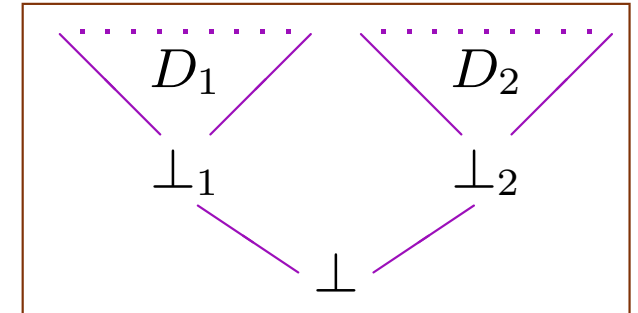
Product

Product of  $\mathbf{D}_1$  and  $\mathbf{D}_2$  is the following cpo:

$$\mathbf{D}_1 \times \mathbf{D}_2 = \langle D_1 \times D_2, \sqsubseteq, \langle \perp_1, \perp_2 \rangle \rangle$$

where for all  $d_1, d'_1 \in D_1$  and  $d_2, d'_2 \in D_2$ ,  $\langle d_1, d_2 \rangle \sqsubseteq \langle d'_1, d'_2 \rangle$  if  $d_1 \sqsubseteq_1 d'_1$  and  $d_2 \sqsubseteq_2 d'_2$ .

Sum



Disjoint sum of  $\mathbf{D}_1$  and  $\mathbf{D}_2$  is the following cpo:

$$\mathbf{D}_1 + \mathbf{D}_2 = \langle (D_1 \times \{1\}) \cup (D_2 \times \{2\}) \cup \{\perp\}, \sqsubseteq, \perp \rangle$$

where for  $d_1, d'_1 \in D_1$ ,  $\langle d_1, 1 \rangle \sqsubseteq \langle d'_1, 1 \rangle$  if  $d_1 \sqsubseteq_1 d'_1$ , for  $d_2, d'_2 \in D_2$ ,  $\langle d_2, 2 \rangle \sqsubseteq \langle d'_2, 2 \rangle$  if  $d_2 \sqsubseteq_2 d'_2$ , and for  $d_1 \in D_1$ ,  $d_2 \in D_2$ ,  $\perp \sqsubseteq \langle d_1, 1 \rangle$  and  $\perp \sqsubseteq \langle d_2, 2 \rangle$ .



To avoid proliferation of bottoms:

### Smashed product

*Smashed product* of  $\mathbf{D}_1$  and  $\mathbf{D}_2$  is the following cpo:

$$\mathbf{D}_1 \otimes \mathbf{D}_2 = \langle (D_1 \setminus \{\perp_1\}) \times (D_2 \setminus \{\perp_2\}) \cup \{\perp\}, \sqsubseteq, \perp \rangle$$

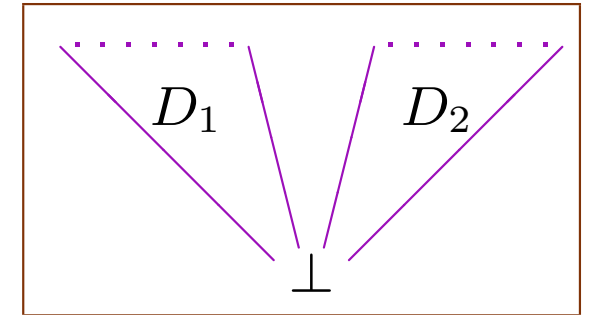
where for all non-bottom  $d_1, d'_1 \in D_1$  and  $d_2, d'_2 \in D_2$ ,  $\langle d_1, d_2 \rangle \sqsubseteq \langle d'_1, d'_2 \rangle$  if  $d_1 \sqsubseteq_1 d'_1$  and  $d_2 \sqsubseteq_2 d'_2$ , and  $\perp \sqsubseteq \langle d_1, d_2 \rangle$ .

### Smashed sum

*Smashed sum* of  $\mathbf{D}_1$  and  $\mathbf{D}_2$  is the following cpo:

$$\mathbf{D}_1 \oplus \mathbf{D}_2 = \langle ((D_1 \setminus \{\perp_1\}) \times \{1\}) \cup ((D_2 \setminus \{\perp_2\}) \times \{2\}) \cup \{\perp\}, \sqsubseteq, \perp \rangle$$

where for all non-bottom  $d_1, d'_1 \in D_1$ ,  $\langle d_1, 1 \rangle \sqsubseteq \langle d'_1, 1 \rangle$  if  $d_1 \sqsubseteq_1 d'_1$ , for  $d_2, d'_2 \in D_2$ ,  $\langle d_2, 2 \rangle \sqsubseteq \langle d'_2, 2 \rangle$  if  $d_2 \sqsubseteq_2 d'_2$ , and  $\perp \sqsubseteq \langle d_1, 1 \rangle$  and  $\perp \sqsubseteq \langle d_2, 2 \rangle$ .



## Function spaces

*Continuous-function space* from  $\mathbf{D}_1$  to  $\mathbf{D}_2$  is the following cpo:

$$[\mathbf{D}_1 \rightarrow \mathbf{D}_2] = \langle [D_1 \rightarrow D_2], \sqsubseteq, \perp \rangle$$

where

- $[D_1 \rightarrow D_2]$  is the set of all *continuous* functions from  $D_1$  to  $D_2$
- for functions  $f, g: D_1 \rightarrow D_2$ ,  $f \sqsubseteq g$  if for each  $d_1 \in D_1$ ,  $f(d_1) \sqsubseteq_2 g(d_1)$
- $\perp(d_1) = \perp_2$  for each  $d_1 \in D_1$ .

$\sqsubseteq$  does not depend on the ordering on  $D_1$

For any set  $X$ , *function space* from  $X$  to  $\mathbf{D}_2$  is the following cpo:

$$(X \rightarrow \mathbf{D}_2) = \langle X \rightarrow D_2, \sqsubseteq, \perp \rangle$$

where  $X \rightarrow D_2$  is the set of total functions from  $X$  to  $D_2$  ordered by  $\sqsubseteq$  as above.

## Domain isomorphism

Cpo's  $\mathbf{D}_1$  and  $\mathbf{D}_2$  are *isomorphic*

$$\mathbf{D}_1 \cong \mathbf{D}_2$$

if there is a bijection between  $D_1$  and  $D_2$  which preserves and reflects the ordering.

**Examples:**

$$\begin{aligned} \mathbf{Bool}_\perp &\cong \{*\}_\perp \oplus \{*\}_\perp \\ \langle X \multimap Y, \subseteq, \emptyset_{X \multimap Y} \rangle &\cong \langle X \multimap Y_\perp, \sqsubseteq, \perp \rangle \end{aligned}$$

*Consider semantic domains up to isomorphism only*

So, we can forget  
(boolean values and)  
partial functions !

It is more difficult to forget natural numbers

**BTW:**

**Informally:**

- $\mathbf{D} \otimes \mathbf{D}'$  admits only “strict” (defined) elements in the pairs
- $\mathbf{D} \times \mathbf{D}'$  admits both “strict” and “undefined” (“unknown”) elements in pairs
- $\mathbf{D}_\perp$  makes all elements in  $\mathbf{D}$  “strict”

**Hence:**

$$\begin{aligned}(\mathbf{D} \times \mathbf{D}')_\perp &\cong \mathbf{D}_\perp \otimes \mathbf{D}'_\perp \\ \mathbf{D} + \mathbf{D}' &\cong \mathbf{D}_\perp \oplus \mathbf{D}'_\perp\end{aligned}$$

Define also:

$$\mathbf{D} \otimes_L \mathbf{D}' \cong \mathbf{D} \otimes \mathbf{D}'_\perp$$

Define:  $\mathbf{D} \oplus_L \mathbf{D}'$ ,  $\mathbf{D} \otimes_R \mathbf{D}'$ ,  $\mathbf{D} \oplus_R \mathbf{D}'$

## Building continuous functions

- Every constant function is continuous
- Partial functions on sets, as used so far, can be replaced by (strict) continuous functions between flat domains; for instance, with a bit of abuse of notation:
  - $\text{ifte}_{\mathbf{D}} \in [\mathbf{Bool}_{\perp} \times \mathbf{D} \times \mathbf{D} \rightarrow \mathbf{D}]$  is given by:

$$\text{ifte}_{\mathbf{D}}(c, d, d') = \begin{cases} \text{ifte}_{\mathbf{D}}(c, d, d') & \text{if } c \neq \perp \\ \perp_{\mathbf{D}} & \text{if } c = \perp \end{cases}$$

- $\_ + \_ \in [\mathbf{Int}_{\perp} \times \mathbf{Int}_{\perp} \rightarrow \mathbf{Int}_{\perp}]$  is given by:

$$n + n' = \begin{cases} n + n' & \text{if } n \neq \perp \text{ and } n' \neq \perp \\ \perp & \text{if } n = \perp \text{ or } n' = \perp \end{cases}$$

## More constructs

- function composition:  $;- \in [[\mathbf{D}_1 \rightarrow \mathbf{D}_2] \times [\mathbf{D}_2 \rightarrow \mathbf{D}_3] \rightarrow [\mathbf{D}_1 \rightarrow \mathbf{D}_3]]$ , i.e.:
  - composition of continuous functions is continuous
  - the composition function is continuous
- indexing:  
 $lift^{\mathbf{I}} \in [[\mathbf{D}_1 \times \dots \times \mathbf{D}_n \rightarrow \mathbf{D}] \rightarrow [[\mathbf{I} \rightarrow \mathbf{D}_1] \times \dots \times [\mathbf{I} \rightarrow \mathbf{D}_n] \rightarrow [\mathbf{I} \rightarrow \mathbf{D}]]]$ , i.e.:
  - indexing a continuous function yields a continuous function
  - the indexing function is continuous
- Given a function  $f: D_1 \times \dots \times D_n \rightarrow D$ ,  $f$  is a continuous function from the product domain  $\mathbf{D}_1 \times \dots \times \mathbf{D}_n$  to  $\mathbf{D}$  if and only if it is continuous w.r.t. each argument separately
  - this justifies the use of lambda-notation to build continuous functions:  
 $\Lambda \in [[\mathbf{D}_0 \times \mathbf{D}_1 \times \dots \times \mathbf{D}_n \rightarrow \mathbf{D}] \rightarrow [\mathbf{D}_1 \times \dots \times \mathbf{D}_n \rightarrow [\mathbf{D}_0 \rightarrow \mathbf{D}]]]$

## ... and more

- continuous-function application is continuous:  $\_(-) \in [[\mathbf{D}_1 \rightarrow \mathbf{D}_2] \times \mathbf{D}_1 \rightarrow \mathbf{D}_2]$
- projections:  $\pi_1 \in [\mathbf{D}_1 \times \mathbf{D}_2 \rightarrow \mathbf{D}_1]$  and  $\pi_2 \in [\mathbf{D}_1 \times \mathbf{D}_2 \rightarrow \mathbf{D}_2]$
- (two-argument pairing, but how to write this sensibly?)
- injections:  $\iota_1 \in [\mathbf{D}_1 \rightarrow \mathbf{D}_1 + \mathbf{D}_2]$  and  $\iota_2 \in [\mathbf{D}_2 \rightarrow \mathbf{D}_1 + \mathbf{D}_2]$ ,
- domain checks:  $is\_in_1 \in [\mathbf{D}_1 + \mathbf{D}_2 \rightarrow \mathbf{Bool}_\perp]$  and  $is\_in_2 \in [\mathbf{D}_1 + \mathbf{D}_2 \rightarrow \mathbf{Bool}_\perp]$
- function pairing:  $\langle \_, \_ \rangle: [[\mathbf{D} \rightarrow \mathbf{D}_1] \times [\mathbf{D} \rightarrow \mathbf{D}_2]] \rightarrow [\mathbf{D} \rightarrow \mathbf{D}_1 \times \mathbf{D}_2]$ , where for  $f \in [\mathbf{D} \rightarrow \mathbf{D}_1]$  and  $g \in [\mathbf{D} \rightarrow \mathbf{D}_2]$ ,  $\langle f, g \rangle = \lambda d:D. \langle f(d), g(d) \rangle$ .
- function sum:  $[-, -]: [[\mathbf{D}_1 \rightarrow \mathbf{D}] \times [\mathbf{D}_2 \rightarrow \mathbf{D}]] \rightarrow [\mathbf{D}_1 + \mathbf{D}_2 \rightarrow \mathbf{D}]$ , where for  $f \in [\mathbf{D}_1 \rightarrow \mathbf{D}]$  and  $g \in [\mathbf{D}_2 \rightarrow \mathbf{D}]$ ,  $[f, g](d) = ifte_{\mathbf{D}}(is\_in_1(d), f(d), g(d))$

also  
their  
smashed  
versions

- the least fixed point operation  $fix(-) \in [[\mathbf{D} \rightarrow \mathbf{D}] \rightarrow \mathbf{D}]$ 
  - for  $\mathbf{D} = [\mathbf{D}_1 \rightarrow \mathbf{D}_2]$ , it follows that the least fixed point of a continuous function on continuous functions is a continuous function...

Enough is enough...

*Not all functions are continuous...*

*Enough functions are continuous...*



## Fixed point equations

Elements of cpo's  $d_1 \in D_1, \dots, d_n \in D_n$  can be defined by writing (sets of) *fixed point equations*

$$d_1 = \Phi_1(d_1, \dots, d_n)$$

...

$$d_n = \Phi_n(d_1, \dots, d_n)$$

where  $\Phi_1 \in [\mathbf{D}_1 \times \dots \times \mathbf{D}_n \rightarrow \mathbf{D}_1], \dots, \Phi_n \in [\mathbf{D}_1 \times \dots \times \mathbf{D}_n \rightarrow \mathbf{D}_n]$ .

This defines  $\langle d_1, \dots, d_n \rangle$  as the least fixed point of

$$\langle \Phi_1, \dots, \Phi_n \rangle \in [\mathbf{D}_1 \times \dots \times \mathbf{D}_n \rightarrow \mathbf{D}_1 \times \dots \times \mathbf{D}_n]$$

The continuous functions used in such definitions may be build using the basic functions and the ways of their composition as discussed so far.

## Domain equations

$$\mathbf{Int} = \{0, 1, -1, 2, -2, \dots\}_\perp$$

$$\mathbf{Bool} = \{\mathbf{tt}, \mathbf{ff}\}_\perp$$

$$\mathbf{State} = \mathbf{Var} \rightarrow \mathbf{Int}$$

$$\mathbf{EXP} = [\mathbf{State} \rightarrow \mathbf{Int}]$$

$$\mathbf{BEXP} = [\mathbf{State} \rightarrow \mathbf{Bool}]$$

$$\mathbf{STMT} = [\mathbf{State} \rightarrow \mathbf{State}]$$

No problem!

Just use the operators to build cpo's  
as discussed above

If definitions of domains turn out to be recursive,  
use the successive approximation technique,  
as above for domain elements

## Recursive domain equations

$$\mathbf{Stream} = A_{\perp} \times \mathbf{Stream}$$

$$\mathbf{Stream}^0 = \{\perp\}$$

$$\mathbf{Stream}^1 = \{\perp \sqsubseteq \langle a_1, \perp \rangle\}$$

$$\mathbf{Stream}^2 = \{\perp \sqsubseteq \langle a_1, \perp \rangle \sqsubseteq \langle a_1, \langle a_2, \perp \rangle \rangle\}$$

...

$$\mathbf{Stream}^n = \{\perp \sqsubseteq \langle a_1, \perp \rangle \sqsubseteq \langle a_1, \langle a_2, \perp \rangle \rangle \sqsubseteq \dots \sqsubseteq \langle a_1, \langle a_2, \langle \dots, \langle a_n, \perp \rangle \dots \rangle \rangle \rangle\}$$

...

$$\begin{aligned} \mathbf{Stream} &= \bigsqcup_{n \geq 0} \mathbf{Stream}^n \\ &= \{\perp \sqsubseteq \langle a_1, \perp \rangle \sqsubseteq \langle a_1, \langle a_2, \perp \rangle \rangle \sqsubseteq \dots \sqsubseteq \langle a_1, \langle a_2, \langle \dots, \langle a_n, \perp \rangle \dots \rangle \rangle \rangle \} \\ &\quad \sqsubseteq \dots \langle a_1, \langle a_2, \langle \dots, \langle a_n, \langle \dots \rangle \rangle \dots \rangle \rangle \rangle \} \end{aligned}$$

where all  $a_1, a_2, \dots, a_n, \dots \in A$ .

$\mathbf{Stream} = A_{\perp} \otimes_L \mathbf{Stream}$   
would yield exactly this

## Problems?

If definitions of domains turn out to be recursive,  
use the successive approximation technique,  
as above for domain elements

*Really?*

*No problem?*

Our problematic domain equation is still unclear:

$$\mathbf{PROC} = [\mathbf{PROC} \rightarrow [\mathbf{Store} \rightarrow \mathbf{Store}]]$$

## Reflexive domains

There is no (non-trivial) *set* that satisfies

$$\mathbf{D} \cong [\mathbf{D} \rightarrow \mathbf{D}]$$

Yet, any form of *self-application* (untyped procedure parameters, dynamic binding, etc) requires a semantic domain of this or similar form.

*Models for  $\lambda$ -calculus*

In particular, this is necessary to model  $\lambda$ -calculus, a formal untyped calculus where every term may be applied to an argument.

**History:** the semantics for ALGOL 60

Christopher Strachey, Dana Scott & many others

## Good naive solution

### *Naive denotational semantics*

- Use standard set-theoretic domain constructors
- Never use “heavy” recursion, as involved in the reflexive domain definition.
- Use naive set-theoretic approximations and set-theoretic unions to solve domain equations.
- This works for well-typed languages with a hierarchy of concepts and domains.

## Solution

*Scott-ery*

- Limit the size of domains: require countable basis plus some technical conditions
- Use continuous functions only
- Define “domain of all domains” where all such domains can be interpreted
- Define continuous functions on this domain to interpret each of the domain constructors
- Write and solve domain equations as fixed point equations in this domain

*Models:  $\mathbf{P}\omega$ ,  $\mathbf{T}\omega$ , information systems, ...*