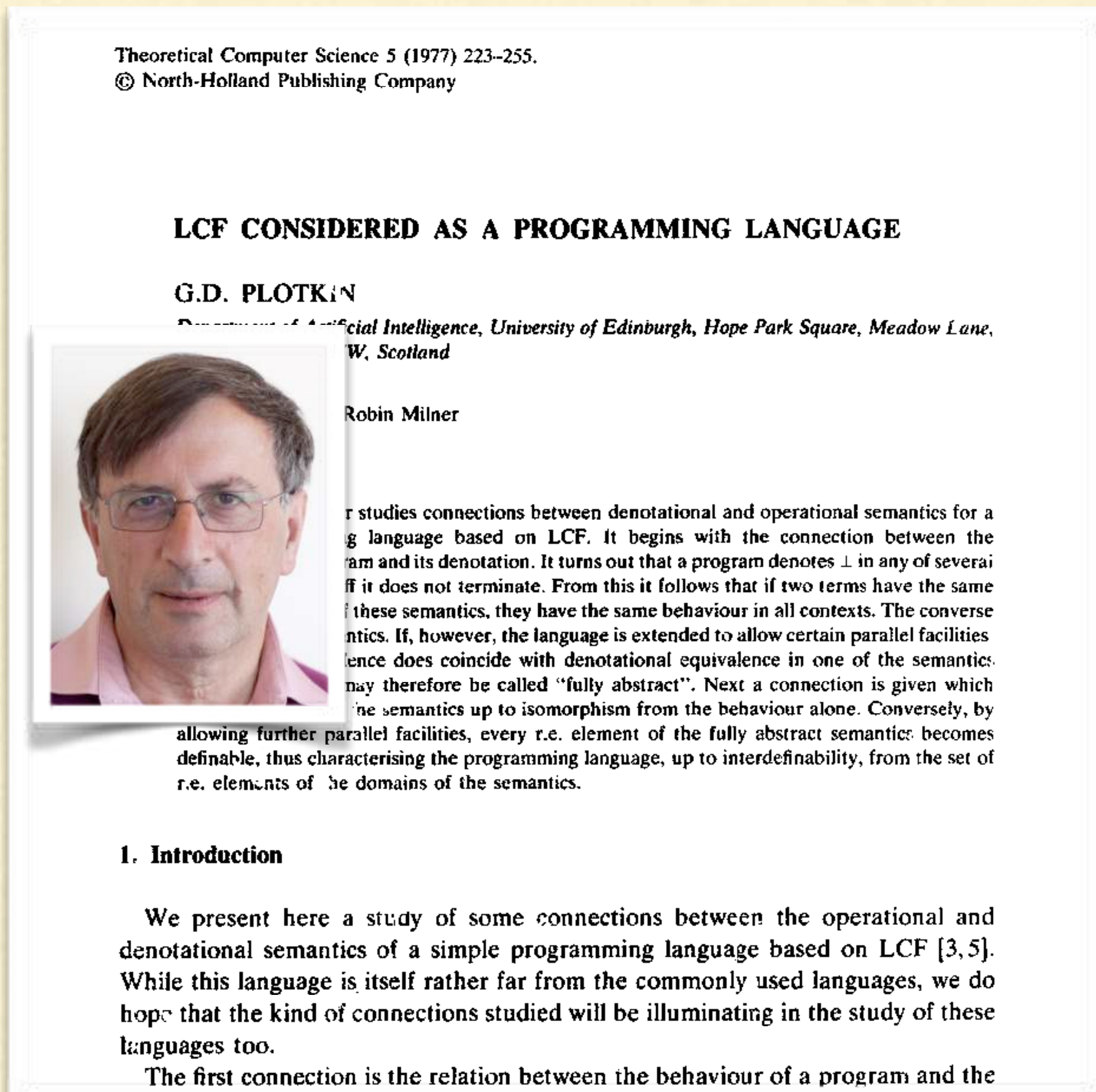
A PROBABILISTIC AND NON-DETERMINISTIC CALL-BY-PUSH-VALUE LANGUAGE

Jean Goubault-Larrecq

- PCF, probabilistic choice, and the trouble with **V**
- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction

PLOTKIN'S PCF (1977)



Theoretical Computer Science 5 (1977) 223–255.

LCF CONSIDERED AS A PROGRAMMING LANGUAGE

G.D. PLOTKIN
...cial Intelligence, University of Edinburgh, Hope Park Square, Meadow Lane, ...W, Scotland

...Robin Milner

...r studies connections between denotational and operational semantics for a ...g language based on LCF. It begins with the connection between the ...am and its denotation. It turns out that a program denotes ⊥ in any of several ...f it does not terminate. From this it follows that if two terms have the same ...these semantics, they have the same behaviour in all contexts. The converse ...ntics. If, however, the language is extended to allow certain parallel facilities ...ence does coincide with denotational equivalence in one of the semantics ...nay therefore be called "fully abstract". Next a connection is given which ...he semantics up to isomorphism from the behaviour alone. Conversely, by allowing further parallel facilities, every r.e. element of the fully abstract semantics becomes definable, thus characterising the programming language, up to interdefinability, from the set of r.e. elements of the domains of the semantics.

1. Introduction

We present here a study of some connections between the operational and denotational semantics of a simple programming language based on LCF [3, 5]. While this language is itself rather far from the commonly used languages, we do hope that the kind of connections studied will be illuminating in the study of these languages too.

The first connection is the relation between the behaviour of a program and the

- Types $\sigma, \tau, \ldots ::= \mathbf{int} \mid \sigma \to \tau$
- Terms $M, N, \ldots ::= x_\tau$
$$\mid MN$$
$$\mid \lambda\, x_\sigma . M$$
$$\mid \mathbf{rec}\ x_\sigma . M$$
$$\mid \underline{n}$$
$$\mid \mathbf{succ}\ M$$
$$\mid \mathbf{pred}\ M$$
$$\mid \mathbf{ifz}\ M\ N\ P$$
- (All terms are typed. Call by name.)

PLOTKIN'S PCF (1977)

- Types $\sigma, \tau, \ldots ::= \mathbf{int} \mid \sigma \to \tau$
- Terms $M, N, \ldots ::= x_\tau$
$$\begin{aligned} &\mid MN \\ &\mid \lambda\, x_\sigma \,.\, M \\ &\mid \mathbf{rec}\ x_\sigma \,.\, M \\ &\mid \underline{n} \\ &\mid \mathbf{succ}\ M \\ &\mid \mathbf{pred}\ M \\ &\mid \mathbf{ifz}\ M\ N\ P \end{aligned}$$
- (All terms are typed. Call by name.)

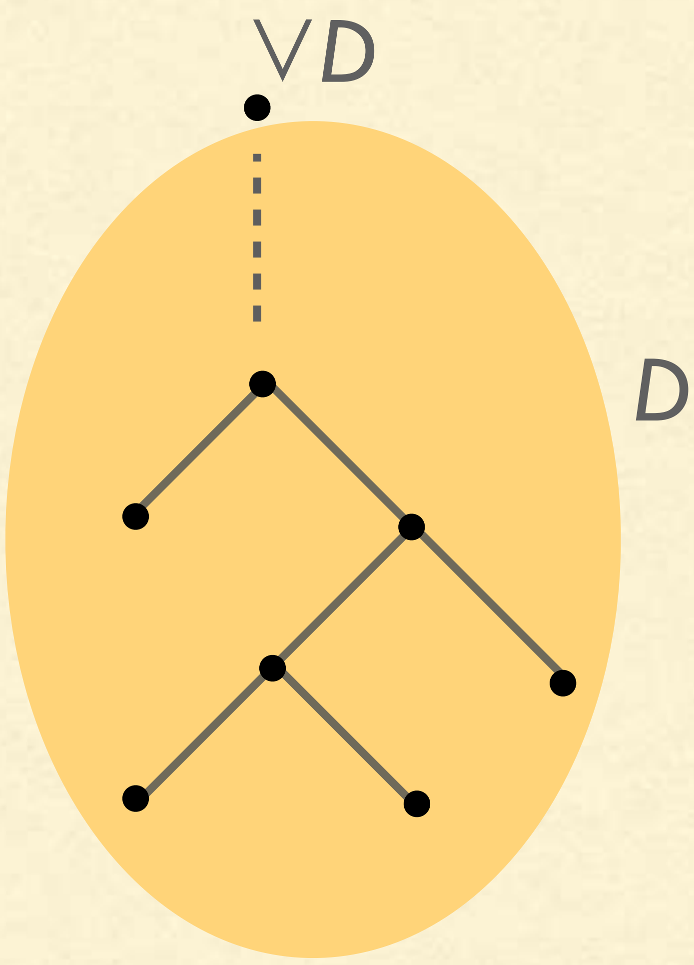
- An **operational** semantics:
$$M \to^* N$$
- A **denotational** semantics:
$$[\![M]\!]$$
- **Adequacy**:
for every ground M : **int**,
$$[\![M]\!] = n \text{ iff } M \to^* \underline{n}$$

PLOTKIN'S PCF (1977)

- An **operational** semantics:
  $$M \to^* N$$
- A **denotational** semantics:
  $$[\![M]\!]$$
- **Adequacy**:
  for every ground M : **int**,
  $$[\![M]\!] = n \text{ iff } M \to^* \underline{n}$$
- **Contextual preordering**:
  $M \lesssim N$ iff
  for every context C : **int**,
  $$C[M] \to^* \underline{n} \;\Rightarrow\; C[N] \to^* \underline{n}$$
- **Fact**: if $[\![M]\!] \leq [\![N]\!]$ then $M \lesssim N$
- Converse is **full abstraction**.
  Fails for PCF, works for PCF+**por**

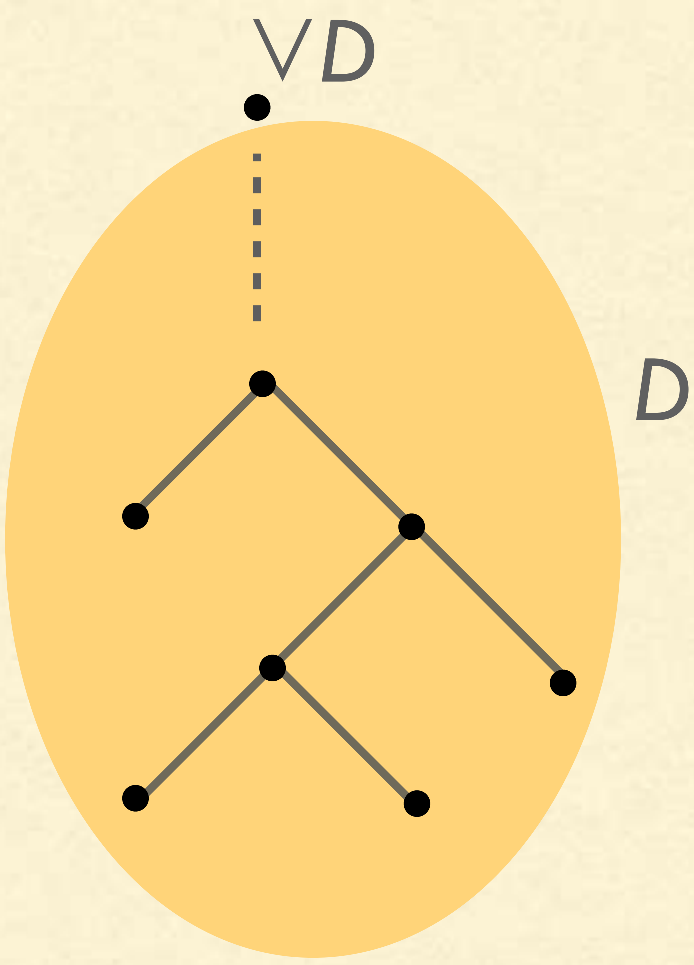
DCPOS

- Every type τ interpreted as a **dcpo** $[\![\tau]\!]$...
  = poset in which every directed family D has a supremum $\bigvee D$



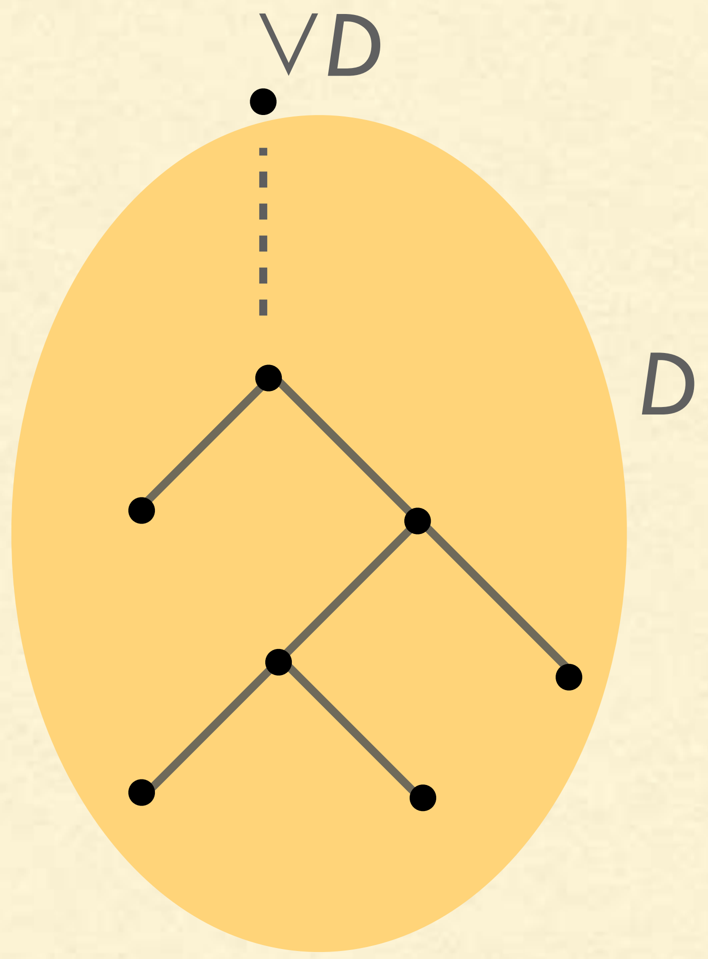
DCPOS

- Every type τ interpreted as a **dcpo** $[\![\tau]\!]$…
  = poset in which every directed family D
  has a supremum $\bigvee D$
- $[\![\textbf{int}]\!] = \mathbb{Z}_\perp$ ($\perp \leq n$, all n incomparable)



DCPOS

- Every type τ interpreted as a **dcpo** $[\![\tau]\!]$…
  = poset in which every directed family D
  has a supremum $\bigvee D$
- $[\![\mathbf{int}]\!] = \mathbb{Z}_\perp$ ($\perp \leq n$, all n incomparable)
- $[\![\sigma \to \tau]\!] = [[\![\sigma]\!] \to [\![\tau]\!]]$,
  dcpo of Scott-continuous maps : $[\![\sigma]\!] \to [\![\tau]\!]$
  (monotonic + preserves directed sups)



THE SEMANTICS OF PCF

- Types $\sigma, \tau, \ldots ::= \mathbf{int} \mid \sigma \to \tau$
- Terms $M, N, \ldots ::= x_\tau$
  $\mid MN$
  $\mid \lambda\, x_\sigma\, .\, M$
  $\mid \mathbf{rec}\ x_\sigma\, .\, M$
  $\mid \underline{n}$
  $\mid \mathbf{succ}\ M$
  $\mid \mathbf{pred}\ M$
  $\mid \mathbf{ifz}\ M\ N\ P$

$\in [\![\sigma \to \tau]\!]$ $\in [\![\sigma]\!]$

- $[\![MN]\!] = [\![M]\!]([\![N]\!])$
  $[\![\lambda\, x_\sigma\, .\, M]\!] = (V \mapsto [\![M]\!][x_\sigma{:=}V])$

$\in [\![\sigma]\!]$ $\in [\![\tau]\!]$

- Meaningful since **Dcpo** is a Cartesian-closed category

CARTESIAN-CLOSEDNESS

- $[\![MN]\!] = [\![M]\!]([\![N]\!])$
  $[\![\lambda\, x_\sigma\, .\, M]\!] = (V \mapsto [\![M]\!][x_\sigma{:=}V])$

  $\in [\![\sigma \to \tau]\!]$ $\in [\![\sigma]\!]$ $\in [\![\sigma]\!]$ $\in [\![\tau]\!]$

- Meaningful since **Dcpo** is a Cartesian-closed category

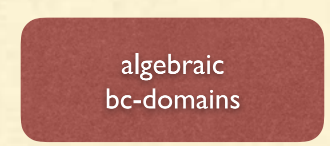
- In order to prove full abstraction (with por), we require to be able to **approximate** elements of $[\![\tau]\!]$ by **definable** elements $[\![M]\!]$.

- In the case of PCF, each $[\![\tau]\!]$ is an **algebraic bc-domain**, making that possible.

- Cartesian-closed… good.

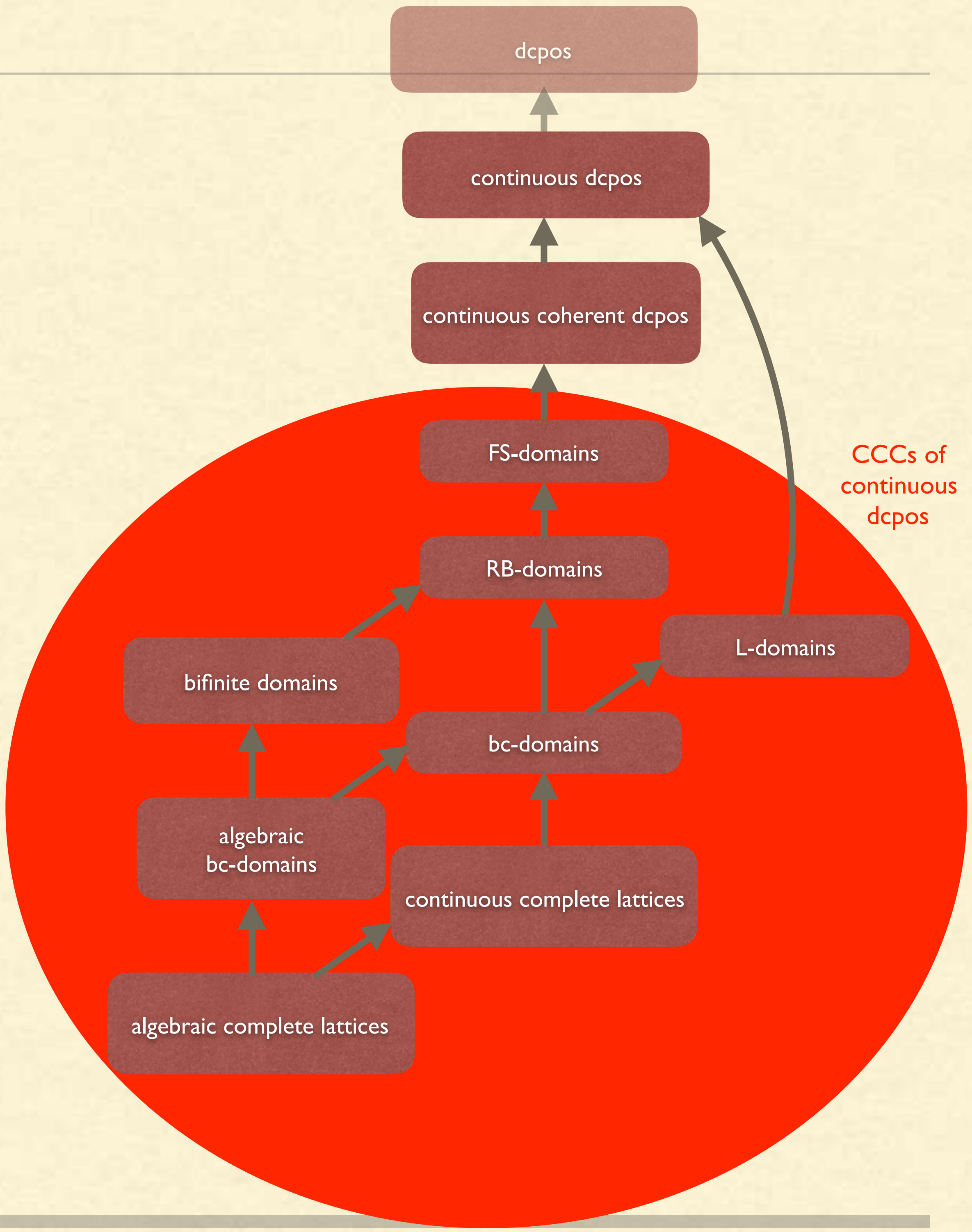
CCCS OF CONTINUOUS DCPOS

- In order to prove full abstraction (with por), we require to be able to **approximate** elements of $[\![\tau]\!]$ by **definable** elements $[\![M]\!]$.
- In the case of PCF, each $[\![\tau]\!]$ is an **algebraic bc-domain**, making that possible.
- Cartesian-closed… good.



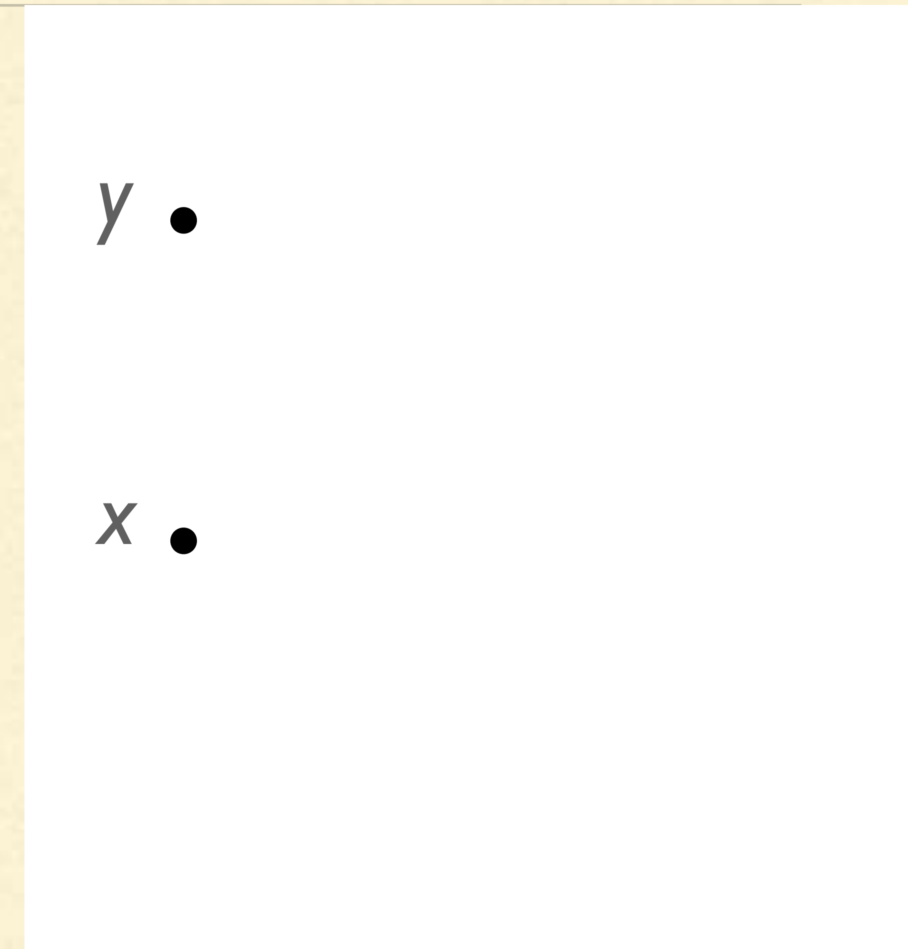
CCCS OF CONTINUOUS DCPOS

- In order to prove full abstraction (with por), we require to be able to **approximate** elements of $[\![\tau]\!]$ by **definable** elements $[\![M]\!]$.
- In the case of PCF, each $[\![\tau]\!]$ is an **algebraic bc-domain**, making that possible.
- Cartesian-closed… good.
- Many other CCCs would fit.



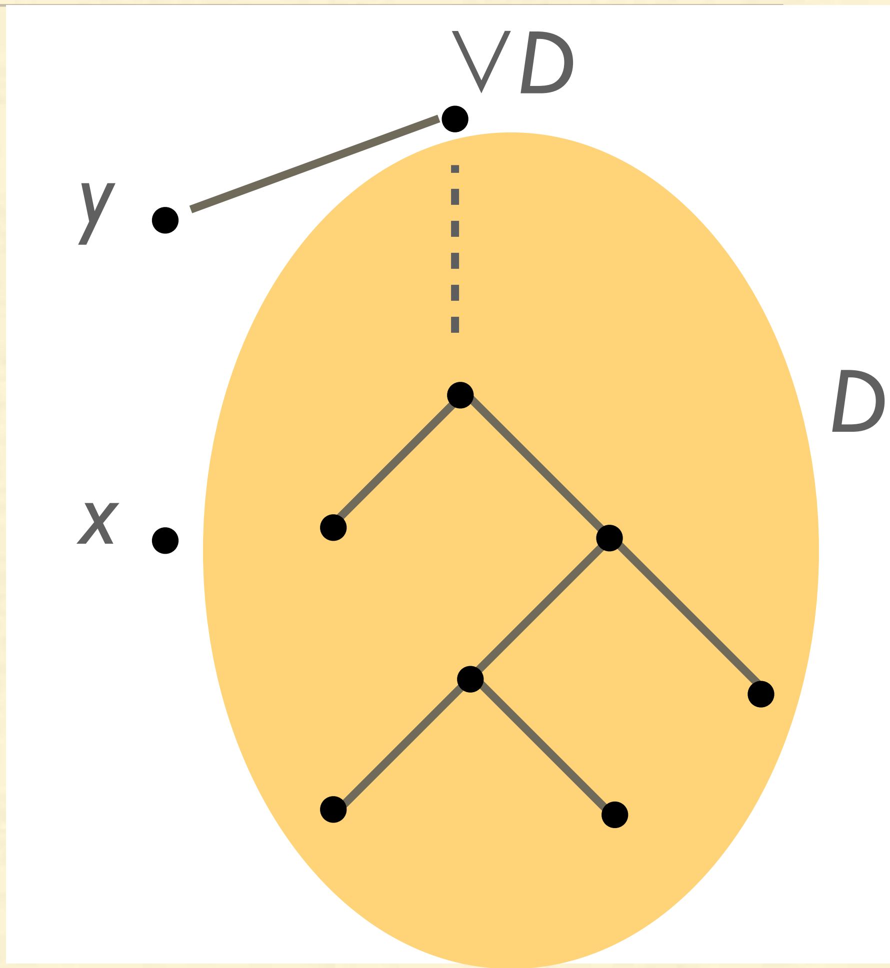
CONTINUOUS DCPOS

- **Approximation** (**way-below**):
  $x \ll y$ iff for every directed D such that $y \leq \bigvee D$,
  x is already below some element of D



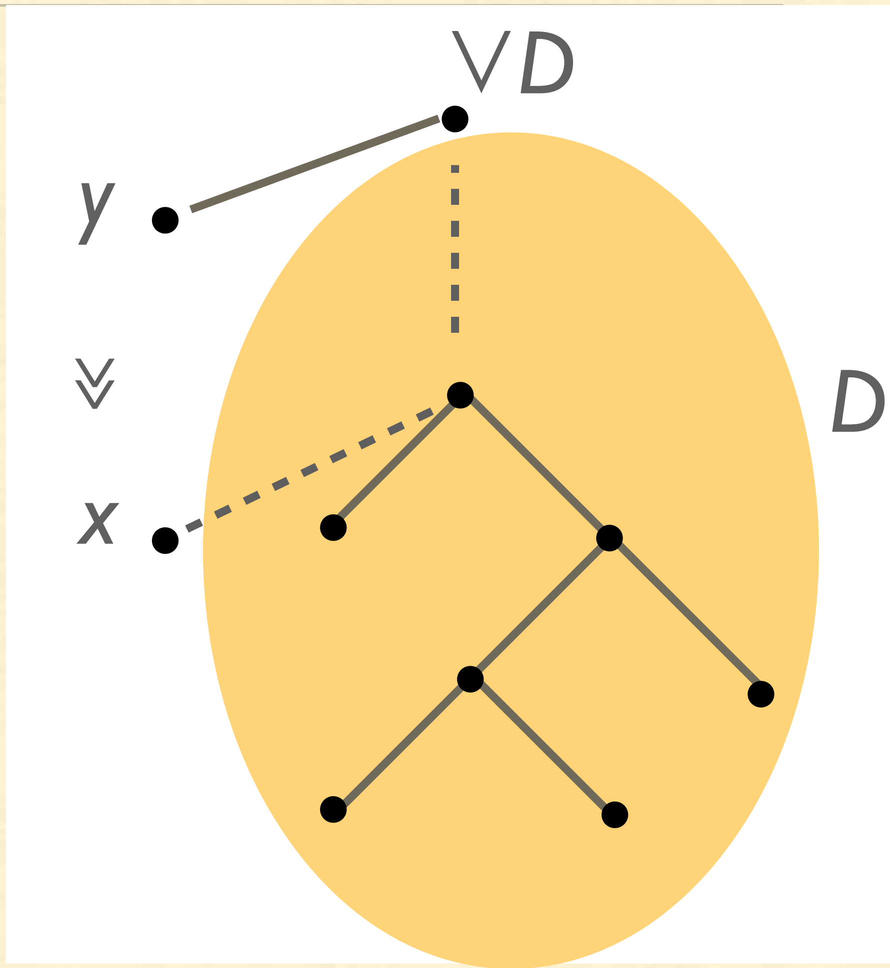
CONTINUOUS DCPOS

- **Approximation** (**way-below**):
  $x \ll y$ iff for every directed D such that $y \leq \bigvee D$,
  x is already below some element of D



CONTINUOUS DCPOS

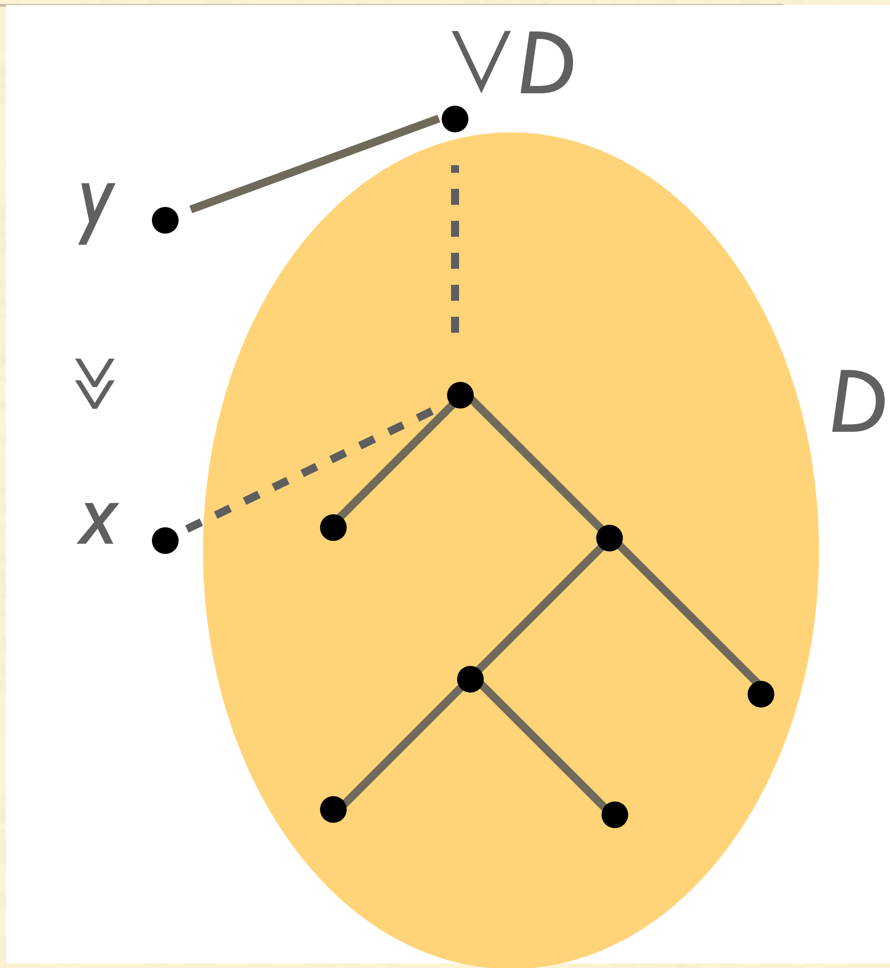
- **Approximation** (**way-below**):
  $x \ll y$ iff for every directed D such that $y \leq \bigvee D$,
  x is already below some element of D



CONTINUOUS DCPOS

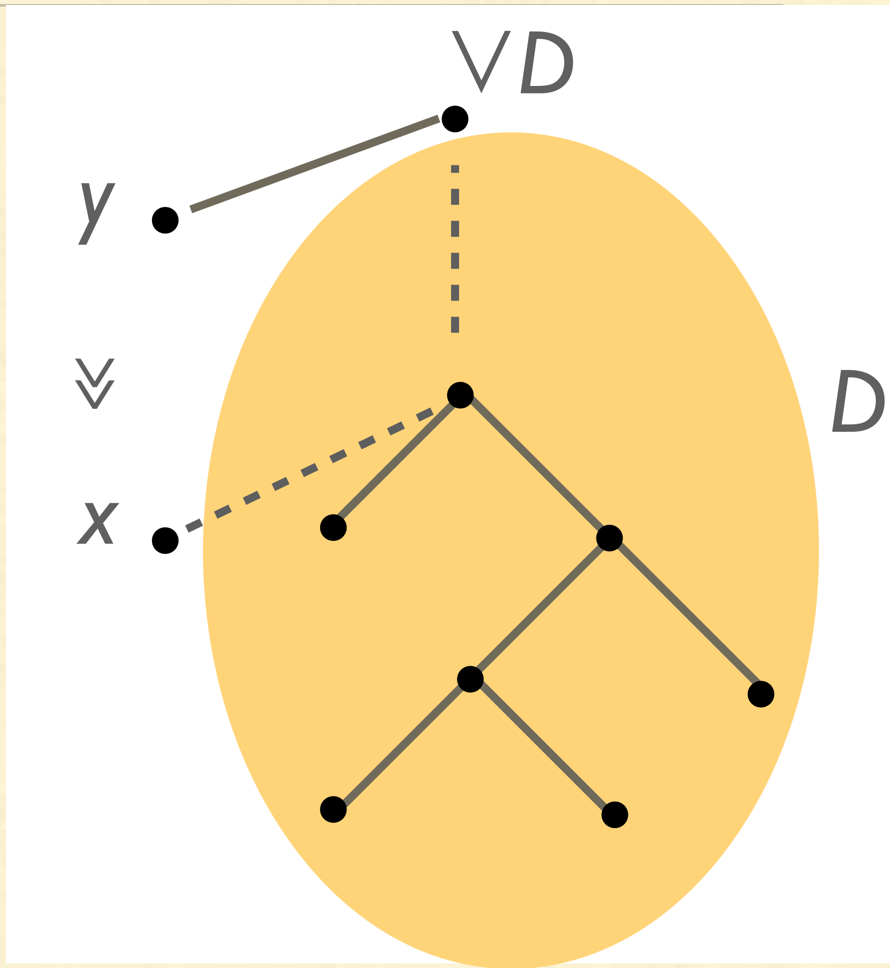
- **Approximation** (**way-below**):
  $x \ll y$ iff for every directed D such that $y \leq \bigvee D$,
  x is already below some element of D

- A **basis** B of a dcpo X iff for every x,
  $\{b \in B \mid b \ll x\}$ directed and has x as sup
  A dcpo X is **continuous** iff has a basis



CONTINUOUS DCPOS

- **Approximation** (**way-below**):
  $x \ll y$ iff for every directed D such that $y \leq \bigvee D$,
  x is already below some element of D



- A **basis** B of a dcpo X iff for every x,
  $\{b \in B \mid b \ll x\}$ directed and has x as sup

  A dcpo X is **continuous** iff has a basis

- Ex: the finite subsets of A form a basis of $\mathbf{P}(A)$ with inclusion
  $\mathbb{N}$ forms a basis of $\mathbb{N} \cup \{\infty\}$
  $\mathbb{Q}_+$ forms a basis of $\mathbb{R}_+ \cup \{\infty\}$ ($x \ll y$ iff x=0 or x<y here)

ADDING PROBABILITIES

- Types
  $\sigma, \tau, \ldots ::= \mathbf{int} \mid \sigma \to \tau \mid \mathbf{V}\tau$

- Terms $M, N, \ldots ::= \ldots$
  $\mid M \oplus N$
  $\mid \mathbf{ret}\ M$
  $\mid \mathbf{do}\ x_\sigma \leftarrow M; N$

ADDING PROBABILITIES

- Types
  $\sigma, \tau, \ldots ::= \mathbf{int} \mid \sigma \to \tau \mid \mathbf{V}\tau$

Monadic type of subprobability valuations over τ

- Terms $M, N, \ldots ::= \ldots$
  $\mid M \oplus N$
  $\mid \mathbf{ret}\ M$
  $\mid \mathbf{do}\ x_\sigma \leftarrow M; N$

ADDING PROBABILITIES

- Types
  $\sigma, \tau, \ldots ::= \mathbf{int} \mid \sigma \to \tau \mid \mathbf{V}\tau$

Monadic type of subprobability valuations over τ

- Terms $M, N, \ldots ::= \ldots$
  $\mid M \oplus N$
  $\mid \mathbf{ret}\ M$
  $\mid \mathbf{do}\ x_\sigma \leftarrow M; N$

with M, N: $\mathbf{V}\tau$,
choose between M and N
with probability 1/2

ADDING PROBABILITIES

- Types
  $\sigma, \tau, \ldots ::= \mathbf{int} \mid \sigma \to \tau \mid \mathbf{V}\tau$

Monadic type of subprobability valuations over τ

- Terms $M, N, \ldots ::= \ldots$
  $\mid M \oplus N$
  $\mid \mathbf{ret}\ M$
  $\mid \mathbf{do}\ x_\sigma \leftarrow M; N$

with M, N: $\mathbf{V}\tau$,
choose between M and N
with probability 1/2

monadic constructions:
$M{:}\tau \Rightarrow \mathbf{ret}\ M{:}\mathbf{V}\tau$

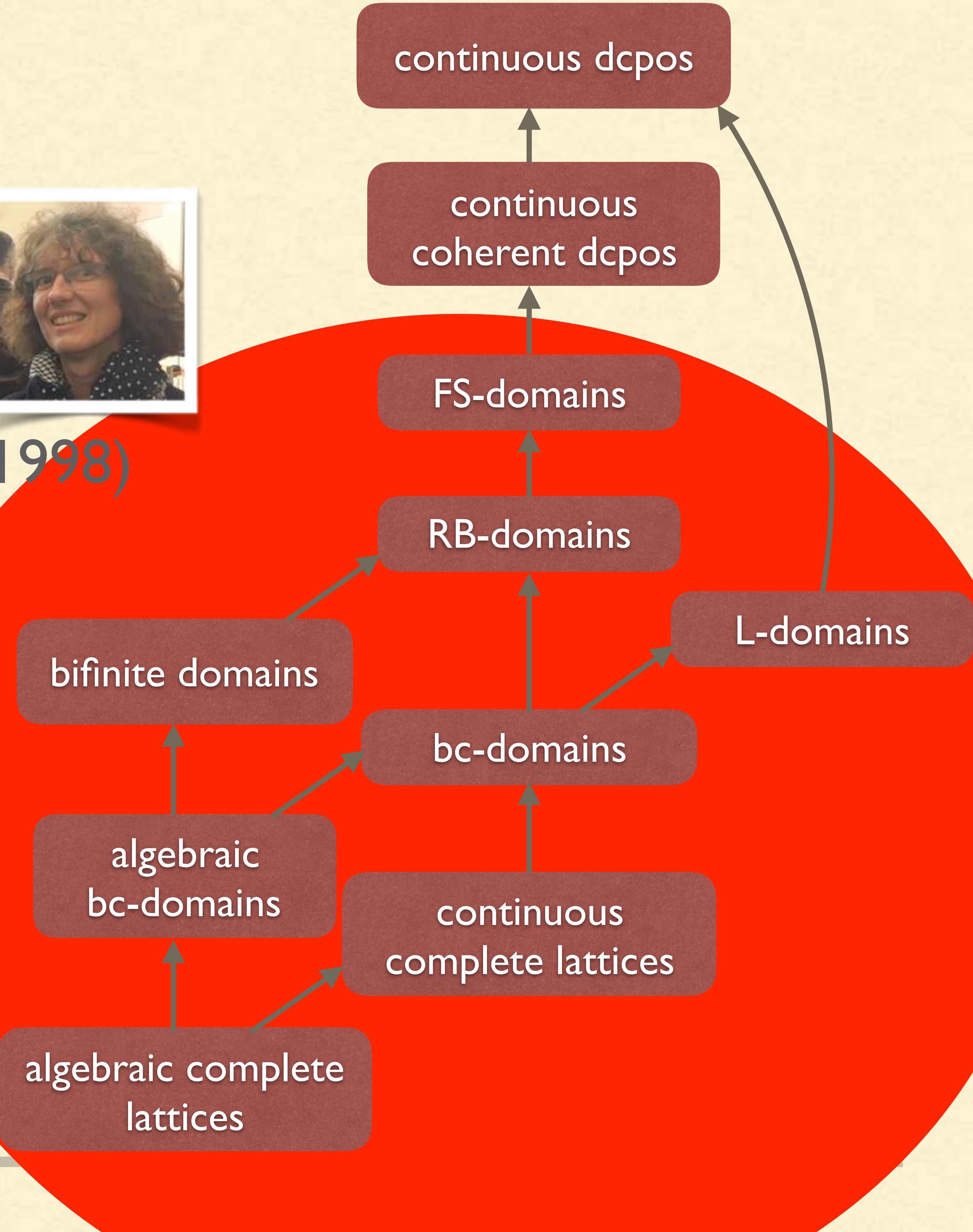
$M{:}\mathbf{V}\sigma \quad N{:}\mathbf{V}\tau \Rightarrow \mathbf{do}\ x_\sigma \leftarrow M; N : \mathbf{V}\tau$

(Moggi 1991)

THE TROUBLE WITH V

(Jung, Tix 1998)

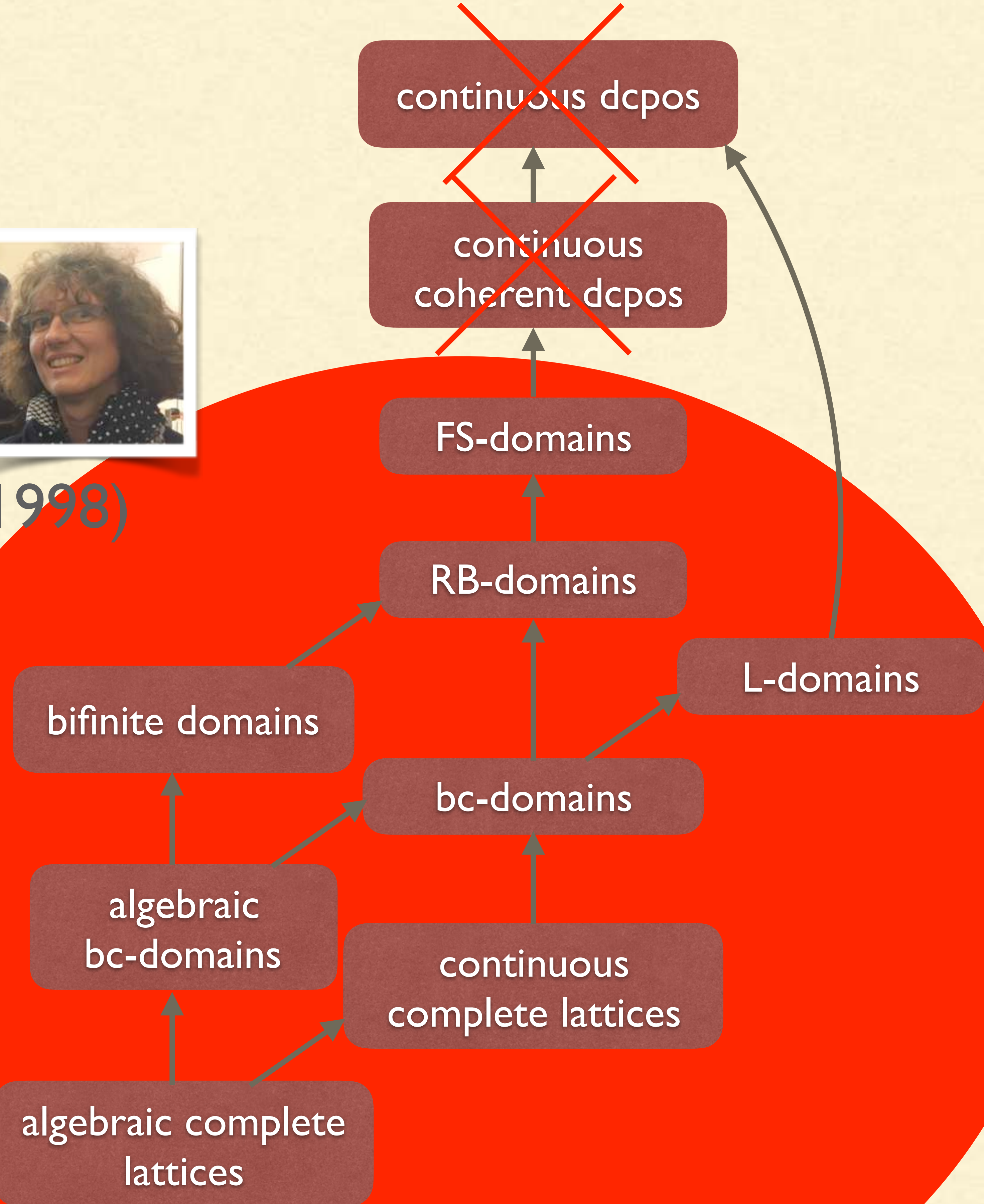
- Look for a category of continuous dcpos that is…



THE TROUBLE WITH V

(Jung, Tix 1998)

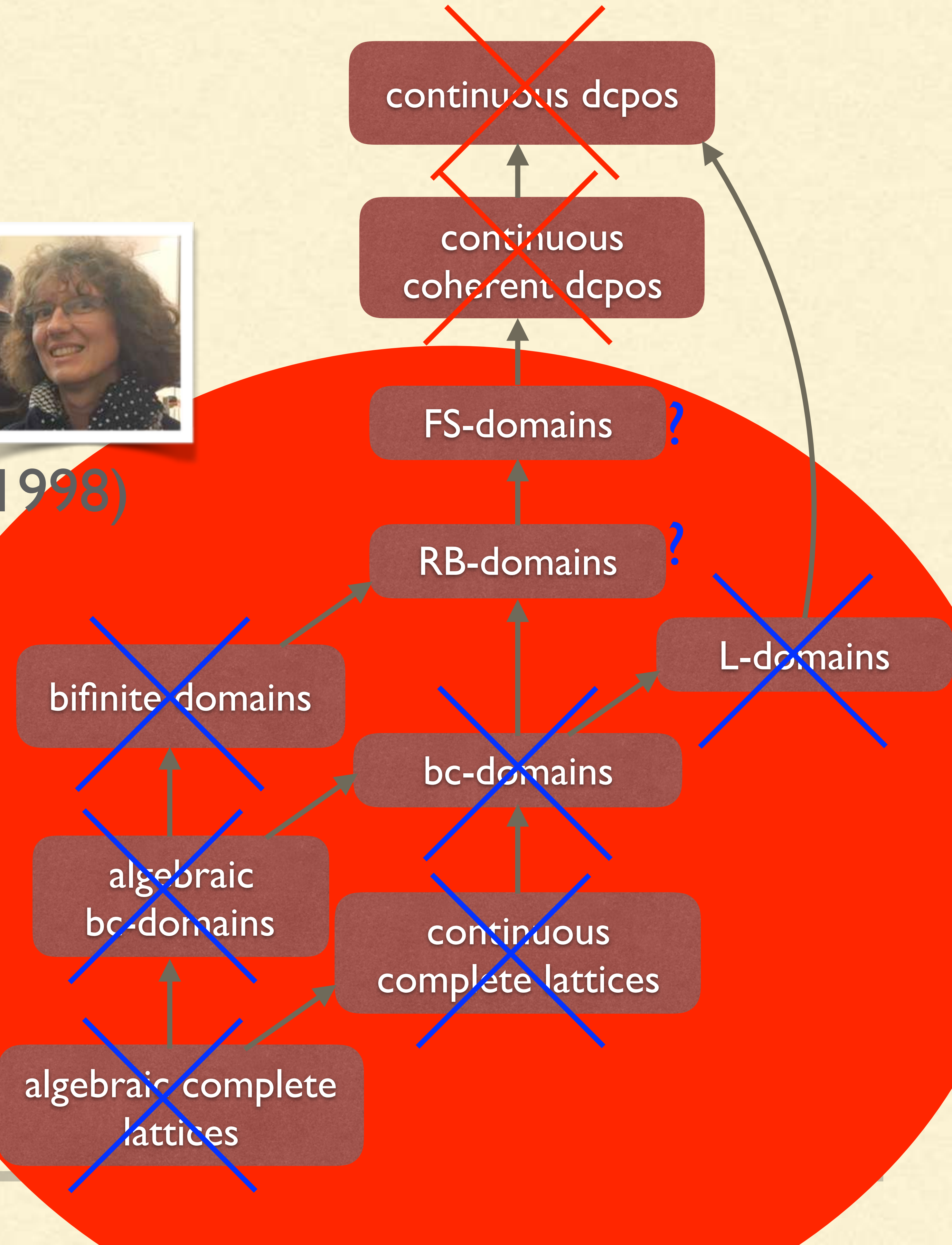
- Look for a category of continuous dcpos that is…
- Cartesian-closed



THE TROUBLE WITH $\mathbf{\vee}$

(Jung, Tix 1998)

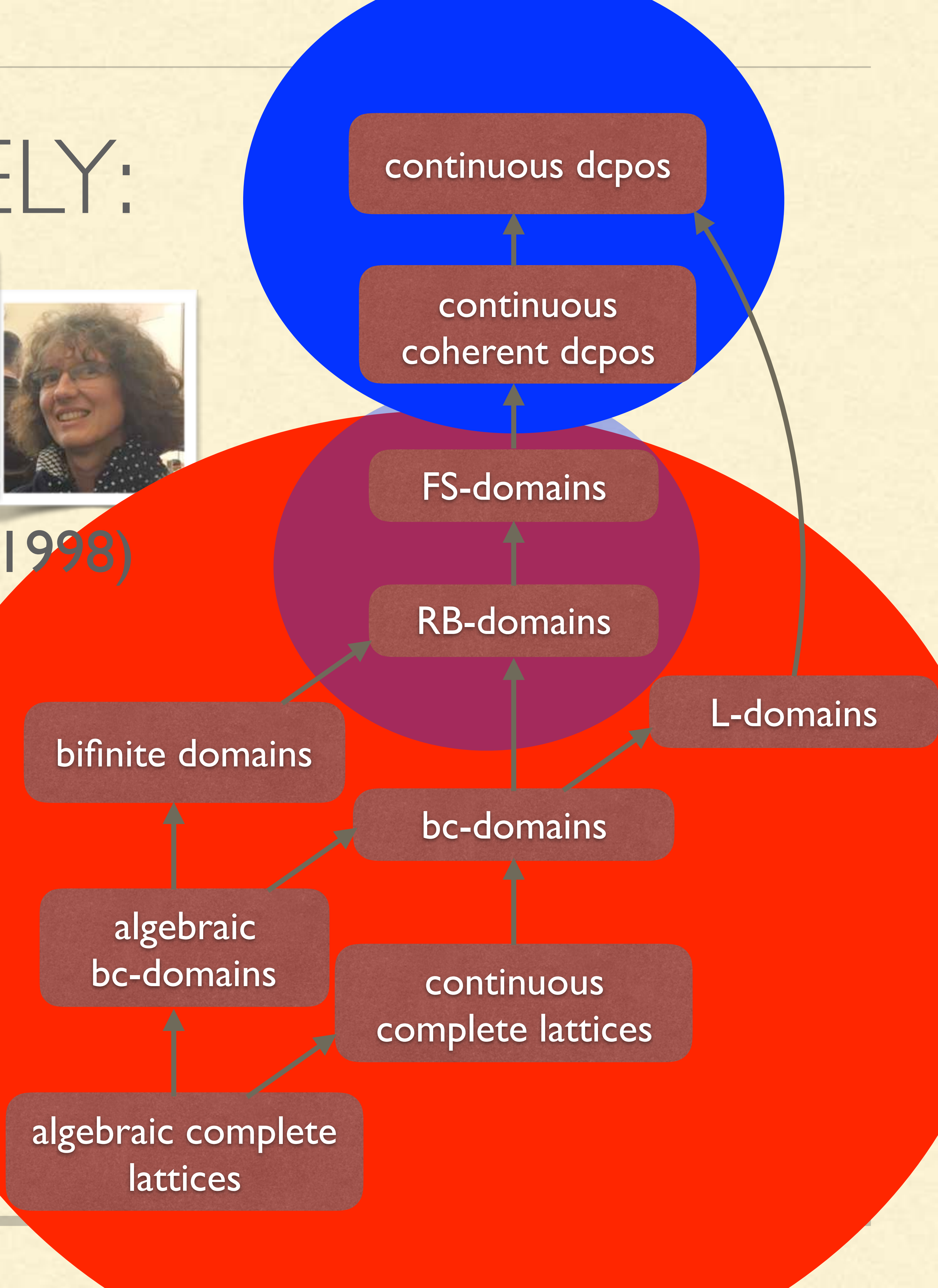
- Look for a category of continuous dcpos that is…
- Cartesian-closed
- closed under $\mathbf{\vee}$



MORE POSITIVELY:

(Jung, Tix 1998)

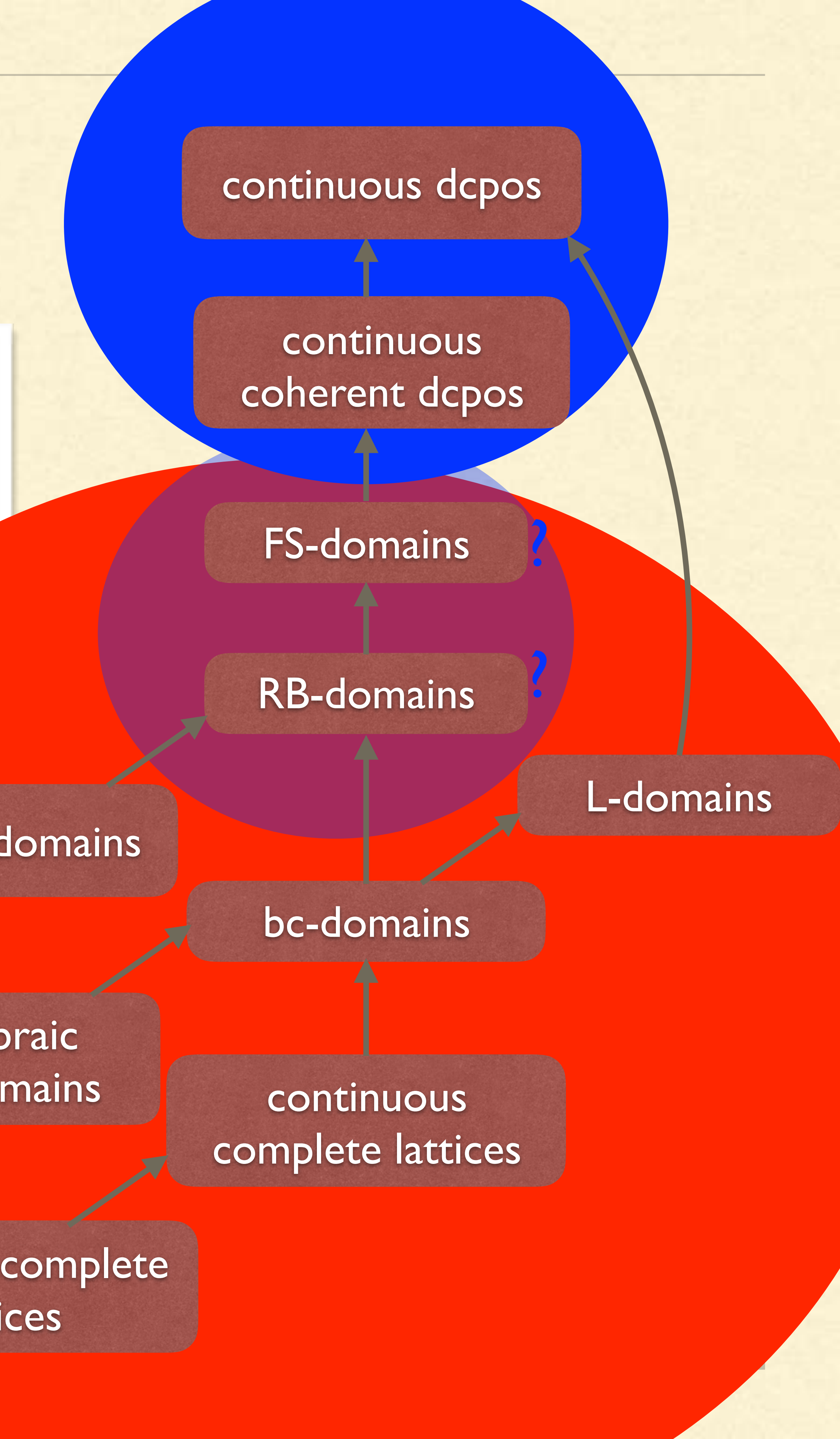
- Look for a category of continuous dcpos that is:
- Cartesian-closed
- closed under **V**



MORE POSITIVELY:

(Jung, Tix 1998)

- Look for a category of continuous dcpos that is:
- Cartesian-closed
- closed under **V**



OTHER SOLUTIONS (1)

- Change categories entirely. E.g., reason in **probabilistic coherence spaces**
- Equationally **fully abstract** semantics (Ehrhard, Pagani, Tasson 14)
- also for call-by-push-value (Ehrhard, Tasson 19)
- probabilistic choice 'built-in'

OTHER SOLUTIONS (2)

- Change categories, and opt for **QCB spaces/predomains** (Battenfeld 06)
  … Cartesian-closed, and has a probabilistic choice monad

OTHER SOLUTIONS (2)

- Change categories, and opt for **QCB spaces/predomains** (Battenfeld 06)
  … Cartesian-closed, and has a probabilistic choice monad

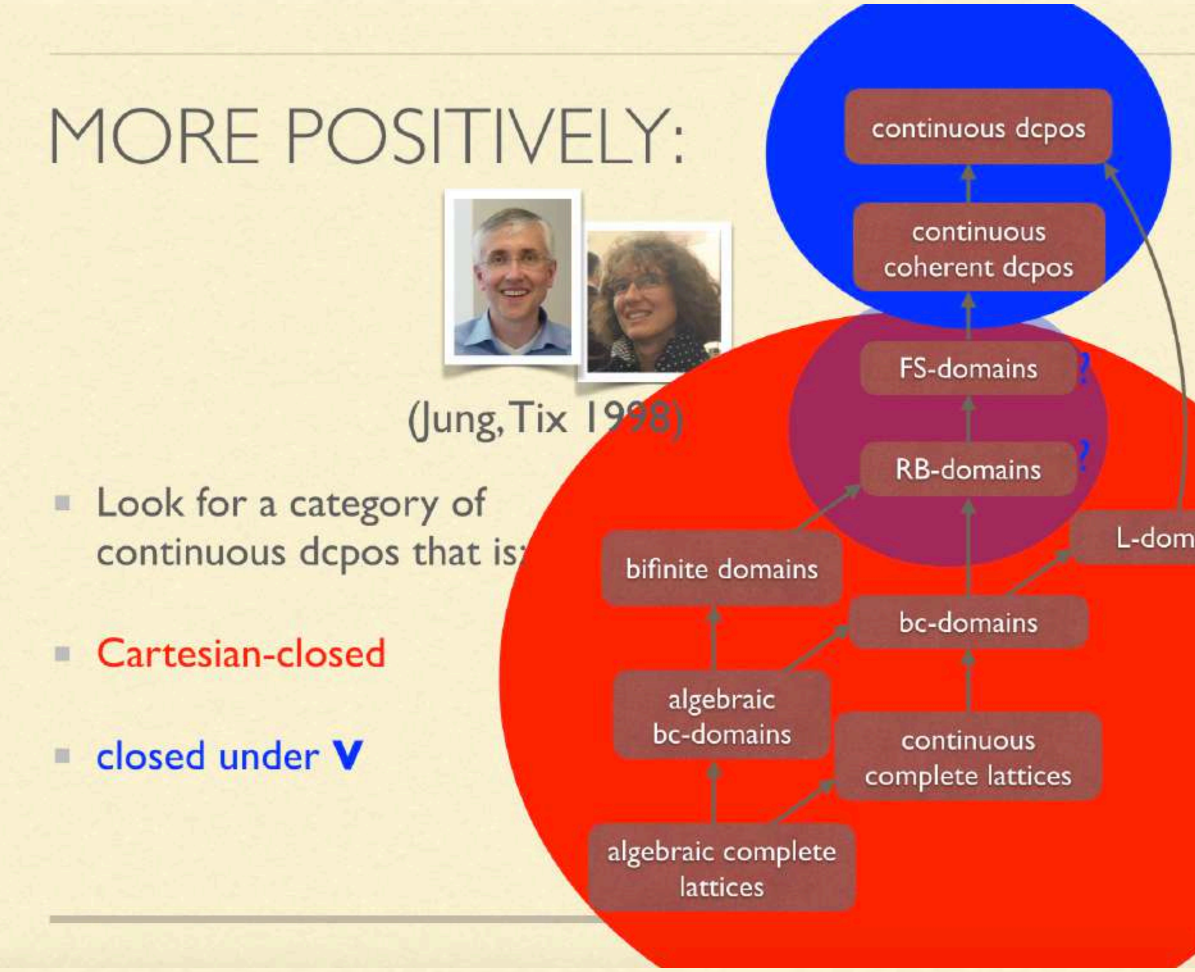
- Changes categories, and opt for **quasi-Borel spaces/domains**
  (Heunen, Kammar, Staton, Yang 17; Vákár, Kammar, Staton 19)
  … Cartesian-closed,
  and closed under a 'laws of random variables' functor





BACK TO DOMAINS

- There is no need to leave domain theory after all
- An easy solution using **call-by-push-value**
- will also handle the mix with **demonic non-determinism**



- PCF, probabilistic choice, and the trouble with $\mathbf{V}$
- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction

- PCF, probabilistic choice, and the trouble with **V**
- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction

TWO KINDS OF TYPES?

- No such problem with two kinds of types:
  $\sigma, \tau, \ldots ::= \mathbf{int} \mid \ldots \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \ldots \mid \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

CALL-BY-PUSH-VALUE

- No such problem with two kinds of types:
  $\sigma, \tau, \ldots ::= \textbf{int} \mid \textbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \mathbf{F}\sigma \mid \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- This is the type structure of Paul B. Levy's **call-by-push-value** (except for the **V** construction)

Call-By-Push-Value: A Subsuming Paradigm
(extended abstract)

Paul Blain Levy*

Department of Computer Science, Queen Mary and Westfield College
LONDON E1 4NS pbl@dcs.qmw.ac.uk

Abstract. Call-by-push-value is a new paradigm that subsumes the call-by-name and call-by-value paradigms, in the following sense: both operational and denotational semantics for those paradigms can be seen as arising, via translations that we will provide, from similar semantics for call-by-push-value.
To explain call-by-push-value, we first discuss general operational ideas, especially the distinction between values and computations, using the principle that "a value is, a computation does". Using an example program, we see that the lambda-calculus primitives can be understood as push/pop commands for an operand-stack.
We provide operational and denotational semantics for a range of computational effects and show their agreement. We hence obtain semantics for call-by-name and call-by-value, of which some are familiar, some are new and some were known but previously appeared mysterious.

(Levy 1999)

CALL-BY-PUSH-VALUE

- No such problem with two kinds of types:
  $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \mathbf{F}\sigma \mid \sigma \to \underline{\tau}$

value types

bc-domains/continuous lattices

- This is the type structure of Paul B. Levy's **call-by-push-value** (except for the **V** construction)

Call-By-Push-Value: A Subsuming Paradigm
(extended abstract)

Paul Blain Levy*

Department of Computer Science, Queen Mary and Westfield College
LONDON E1 4NS pbl@dcs.qmw.ac.uk

Abstract. Call-by-push-value is a new paradigm that subsumes the call-by-name and call-by-value paradigms, in the following sense: both operational and denotational semantics for those paradigms can be seen as arising, via translations that we will provide, from similar semantics for call-by-push-value.
To explain call-by-push-value, we first discuss general operational ideas, especially the distinction between values and computations, using the principle that "a value is, a computation does". Using an example program, we see that the lambda-calculus primitives can be understood as push/pop commands for an operand-stack.
We provide operational and denotational semantics for a range of computational effects and show their agreement. We hence obtain semantics for call-by-name and call-by-value, of which some are familiar, some are new and some were known but previously appeared mysterious.

(Levy 1999)

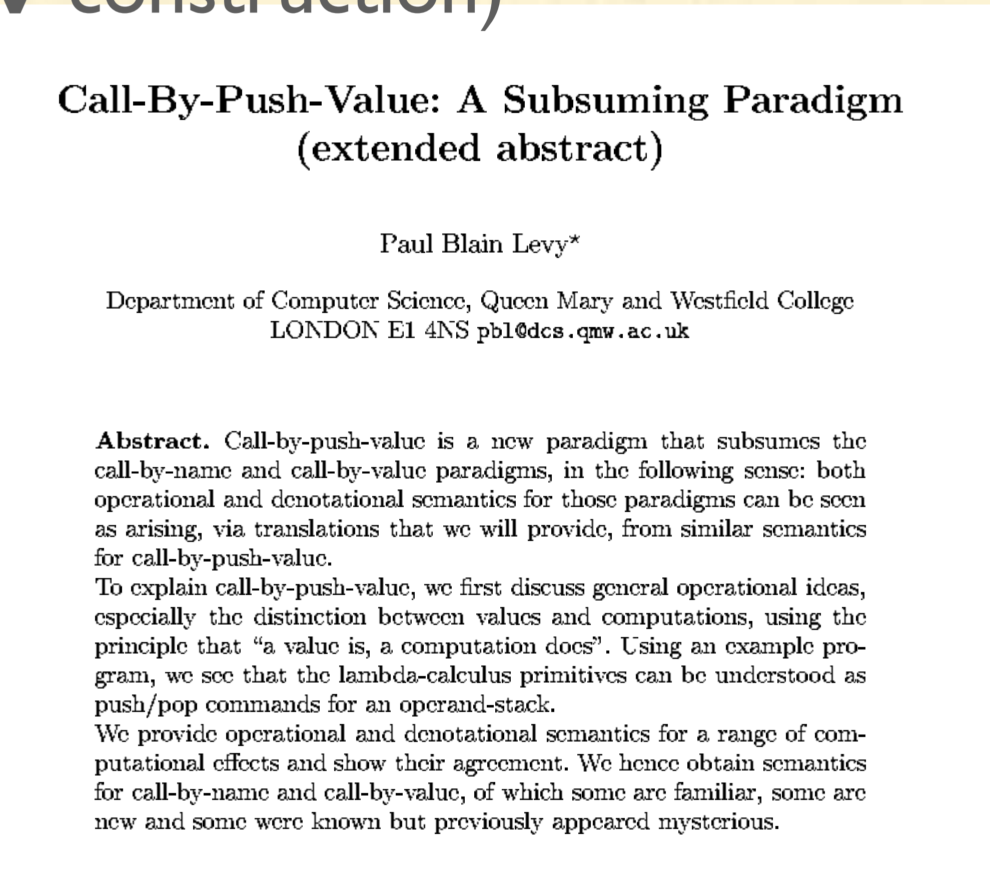
CALL-BY-PUSH-VALUE

- No such problem with two kinds of types:
  $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\underline{\tau}$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \mathbf{F}\sigma \mid \sigma \to \underline{\tau}$

value types

computation types

- This is the type structure of Paul B. Levy's **call-by-push-value** (except for the **V** construction)



Call-By-Push-Value: A Subsuming Paradigm
(extended abstract)

Paul Blain Levy*

Department of Computer Science, Queen Mary and Westfield College
LONDON E1 4NS pbl@dcs.qmw.ac.uk

Abstract. Call-by-push-value is a new paradigm that subsumes the call-by-name and call-by-value paradigms, in the following sense: both operational and denotational semantics for those paradigms can be seen as arising, via translations that we will provide, from similar semantics for call-by-push-value.
To explain call-by-push-value, we first discuss general operational ideas, especially the distinction between values and computations, using the principle that "a value is, a computation does". Using an example program, we see that the lambda-calculus primitives can be understood as push/pop commands for an operand-stack.
We provide operational and denotational semantics for a range of computational effects and show their agreement. We hence obtain semantics for call-by-name and call-by-value, of which some are familiar, some are new and some were known but previously appeared mysterious.

(Levy 1999)

U AND F

- $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \quad \sigma \times \tau \mid \mathbf{V}\tau$

  $\underline{\sigma}, \underline{\tau}, \ldots ::= \quad \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

U AND F

- $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- **U** converts from bc-domains to continuous coherent dcpos
  … semantically the identity: $[\![\mathbf{U}\underline{\sigma}]\!] = [\![\underline{\sigma}]\!]$

U AND F

- $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \quad \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- **U** converts from bc-domains to continuous coherent dcpos
  … semantically the identity: $[\![\mathbf{U}\underline{\sigma}]\!]=[\![\underline{\sigma}]\!]$

  - $M, N, \ldots ::= \ldots$
    $\mid$ **force** M $\quad (\mathbf{U}\underline{\sigma} \to \underline{\sigma})$
    $\mid$ **thunk** M $\quad (\underline{\sigma} \to \mathbf{U}\underline{\sigma})$

U AND **F**

- $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \quad \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- **U** converts from bc-domains to continuous coherent dcpos
  … semantically the identity: $[\![\mathbf{U}\underline{\sigma}]\!]=[\![\underline{\sigma}]\!]$

  - $M, N, \ldots ::= \ldots$
    $\mid \mathbf{force}\ M \quad (\mathbf{U}\underline{\sigma} \to \underline{\sigma})$
    $\mid \mathbf{thunk}\ M \quad (\underline{\sigma} \to \mathbf{U}\underline{\sigma})$

- $[\![\mathbf{force}\ M]\!] = [\![M]\!]$
  $[\![\mathbf{thunk}\ M]\!] = [\![M]\!]$

- $\mathbf{force\ thunk}\ M \to M$

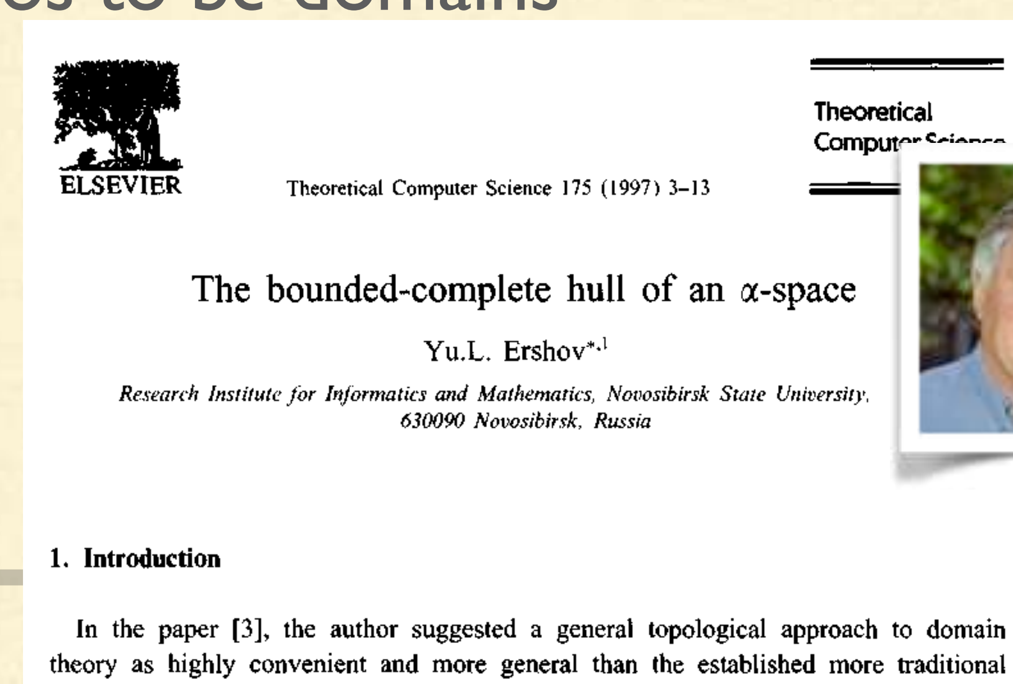
U AND F

- $\sigma, \tau, \ldots ::=$ **int** | **unit** | $\mathbf{U}\underline{\sigma}$ | $\sigma \times \tau$ | $\mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \mathbf{F}\sigma \mid \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- **U** converts from bc-domains to continuous coherent dcpos
  … semantically the identity: $[\![\mathbf{U}\underline{\sigma}]\!]=[\![\underline{\sigma}]\!]$
- **F** converts from continuous coherent dcpos to bc-domains
  … Ershov's bounded complete hull
  would be the canonical choice
- (but is too intricate for our purposes.)


ELSEVIER

Theoretical Computer Science 175 (1997) 3–13

Theoretical Computer Science

The bounded-complete hull of an α-space

Yu.L. Ershov*,[1]

Research Institute for Informatics and Mathematics, Novosibirsk State University, 630090 Novosibirsk, Russia

1. Introduction

In the paper [3], the author suggested a general topological approach to domain theory as highly convenient and more general than the established more traditional

THE SMYTH POWERDOMAIN

- **Q**X = {compact saturated subsets of X}, reverse inclusion $\supseteq$
- **Fact.** For X continuous coherent dcpo,
  Ershov's bc-hull of X is a subspace of **Q**X.
- **Q**X is itself a bc-domain (even a continuous complete lattice), and is much easier to use.
- Serves as a model of **demonic non-determinism**.

THE SMYTH POWERDOMAIN

- $\mathbf{Q}X$ = {compact saturated subsets of X}, reverse inclusion $\supseteq$ defines a(nother) **monad** on the cat. of cont. coh. dcpos.

- **Unit:** $\eta : X \to \mathbf{Q}X : x \mapsto \uparrow x$ (continuous)

- **Extension:** for $f : X \to L$ where L continuous complete lattice,

$$\text{let } f^* : \mathbf{Q}X \to L : Q \mapsto \inf \{f(x) \mid x \in Q\}$$

— if f is continuous then f^* is continuous
— $f^* \circ \eta = f$
— $f^* \circ g^* = (f^* \circ g)^*$

THE SMYTH$_\perp$ POWERDOMAIN

- $\mathbf{Q}_\perp X = \mathbf{Q}X$ plus a fresh bottom $\perp$
  defines a(nother) **monad** on the cat. of cont. coh. dcpos.
- **Unit:** $\eta : X \to \mathbf{Q}_\perp X : x \mapsto \uparrow x$ (continuous)
- **Extension:** for $f : X \to L$ where L continuous complete lattice,
  let $f^* : \mathbf{Q}_\perp X \to L : Q \mapsto \inf \{f(x) \mid x \in Q\}, \perp \mapsto \perp$
  — if f is continuous then f^* is continuous — and f^* is **strict** now
  — $f^* \circ \eta = f$
  — $f^* \circ g^* = (f^* \circ g)^*$

U AND **F**

- $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \mathbf{F}\sigma \mid \sigma \rightarrow \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- **U** converts from bc-domains to continuous coherent dcpos: $[\![\mathbf{U}\underline{\sigma}]\!]=[\![\underline{\sigma}]\!]$
- **F** converts from continuous coherent dcpos to bc-domains: $[\![\mathbf{F}\sigma]\!]=\mathbf{Q}_\perp[\![\sigma]\!]$

U AND F

- $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \mathbf{F}\sigma \mid \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- **U** converts from bc-domains to continuous coherent dcpos: $[\![\mathbf{U}\underline{\sigma}]\!]=[\![\underline{\sigma}]\!]$
- **F** converts from continuous coherent dcpos to bc-domains: $[\![\mathbf{F}\sigma]\!]=\mathbf{Q}_{\perp}[\![\sigma]\!]$
  - $M, N, \ldots ::= \ldots$
    $\mid \mathbf{abort}_{\mathbf{F}\sigma}$
    $\mid M \oslash N$
    $\mid \mathbf{produce}\ M \qquad (\sigma \to \mathbf{F}\sigma)$
    $\mid M\ \mathbf{to}\ x_\sigma\ \mathbf{in}\ N$

choice

monad

U AND **F**

- $\sigma, \tau, \ldots ::= \mathbf{int} \mid \mathbf{unit} \mid \mathbf{U}\underline{\sigma} \mid \sigma \times \tau \mid \mathbf{V}\tau$
  $\underline{\sigma}, \underline{\tau}, \ldots ::= \mathbf{F}\sigma \mid \sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- **U** converts from bc-domains to continuous coherent dcpos: $[\![\mathbf{U}\underline{\sigma}]\!]=[\![\underline{\sigma}]\!]$
- **F** converts from continuous coherent dcpos to bc-domains: $[\![\mathbf{F}\sigma]\!]=\mathbf{Q}_\perp[\![\sigma]\!]$
  - $M, N, \ldots ::= \ldots$
    $\mid \mathbf{abort}_{\mathbf{F}\sigma}$
    $\mid M \varowedge N$
    $\mid \mathbf{produce}\ M \qquad (\sigma \to \mathbf{F}\sigma)$
    $\mid M\ \mathbf{to}\ x_\sigma\ \mathbf{in}\ N$
  - $[\![\mathbf{abort}_{\mathbf{F}\sigma}]\!] = \varnothing$
    $[\![M \varowedge N]\!] = [\![M]\!] \wedge [\![N]\!]$
    $[\![\mathbf{produce}\ M]\!] = \eta([\![M]\!])$
    $[\![M\ \mathbf{to}\ x_\sigma\ \mathbf{in}\ N]\!] =$
    $(V \mapsto [\![N]\!][x_\sigma{:=}V])^* ([\![M]\!])$

choice

monad

U AND F

- $\underline{\sigma}, \underline{\tau}, \ldots ::=$ **int** | **unit** | **U**$\underline{\sigma}$ | $\sigma \times \tau$ | **V**τ

  $\underline{\sigma}, \underline{\tau}, \ldots ::=$ **F**σ | $\sigma \to \underline{\tau}$

continuous (coherent) dcpos

bc-domains/continuous lattices

- **U** converts from bc-domains to continuous coherent dcpos: $[\![\mathbf{U}\underline{\sigma}]\!]=[\![\underline{\sigma}]\!]$
- **F** converts from continuous coherent dcpos to bc-domains: $[\![\mathbf{F}\sigma]\!]=\mathbf{Q}_\perp[\![\sigma]\!]$
  - $M, N, \ldots ::= \ldots$
    | $\mathbf{abort}_{\mathbf{F}\sigma}$
    | $M \oslash N$
    | **produce** M $\quad (\sigma \to \mathbf{F}\sigma)$
    | M **to** x_σ **in** N

choice

monad

  - $[\![\mathbf{abort}_{\mathbf{F}\sigma}]\!] = \varnothing$
    $[\![M \oslash N]\!] = [\![M]\!] \wedge [\![N]\!]$
    $[\![\mathbf{produce}\ M]\!] = \eta([\![M]\!])$
    $[\![M\ \mathbf{to}\ x_\sigma\ \mathbf{in}\ N]\!] =$
    $(V \mapsto [\![N]\!][x_\sigma:=V])^* ([\![M]\!])$
  - $(\mathbf{produce}\ M)\ \mathbf{to}\ x_\sigma\ \mathbf{in}\ N \to N[x_\sigma:=M]$ + etc.

- PCF, probabilistic choice, and the trouble with $\mathbf{V}$
- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction

- PCF, probabilistic choice, and the trouble with **V**
- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction

OPERATIONAL SEMANTICS

- A Krivine machine for deterministic operations, working on configurations

  $C\,.\,M$

$$
\begin{array}{rr}
C \cdot E[M] \to CE \cdot M & C[\_N] \cdot \lambda x_\sigma.M \to C \cdot M[x_\sigma := N] \\
C[\_\ \mathtt{to}\ x_\sigma\ \mathtt{in}\ N] \cdot \mathtt{produce}\, M \to C \cdot N[x_\sigma := M] & C[\mathtt{force}\ \_] \cdot \mathtt{thunk}\, M \to C \cdot M \\
[\_] \cdot \mathtt{produce}\, M \to [\mathtt{produce}\ \_] \cdot M & \\
C[\mathtt{pred}\ \_] \cdot \underline{n} \to C \cdot \underline{n-1} & C[\mathtt{succ}\ \_] \cdot \underline{n} \to C \cdot \underline{n+1} \\
C[\mathtt{ifz}\ \_\ N\ P] \cdot \underline{0} \to C \cdot N & C[\mathtt{ifz}\ \_\ N\ P] \cdot \underline{n} \to C \cdot P \quad (n \neq 0) \\
C[\_; N] \cdot \underline{*} \to C \cdot N & \\
C[\pi_1 \_] \cdot \langle M, N \rangle \to C \cdot M & C[\pi_2 \_] \cdot \langle M, N \rangle \to C \cdot N \\
C[\mathtt{do}\, x_\sigma \leftarrow \_; N] \cdot \mathtt{ret}\, M \to C \cdot N[x_\sigma := M] & [\mathtt{produce}\ \_] \cdot \mathtt{ret}\, M \to [\mathtt{produce}\,\mathtt{ret}\ \_] \cdot M \\
C \cdot \mathtt{rec}\, x_\sigma.M \to C \cdot M[x_\sigma := \mathtt{rec}\, x_\sigma.M] &
\end{array}
$$

OPERATIONAL SEMANTICS

- A Krivine machine for deterministic operations, working on configurations
  $C \,.\, M$

$$
\begin{array}{cc}
C \cdot E[M] \to CE \cdot M & C[\_N] \cdot \lambda x_\sigma.M \to C \cdot M[x_\sigma := N] \\
C[\_\ \mathbf{to}\ x_\sigma\ \mathbf{in}\ N] \cdot \mathbf{produce}\, M \to C \cdot N[x_\sigma := M] & C[\mathbf{force}\ \_] \cdot \mathbf{thunk}\, M \to C \cdot M \\
[\_] \cdot \mathbf{produce}\, M \to [\mathbf{produce}\ \_] \cdot M & \\
C[\mathbf{pred}\ \_] \cdot \underline{n} \to C \cdot \underline{n-1} & C[\mathbf{succ}\ \_] \cdot \underline{n} \to C \cdot \underline{n+1} \\
C[\mathbf{ifz}\ \_\ N\ P] \cdot \underline{0} \to C \cdot N & C[\mathbf{ifz}\ \_\ N\ P] \cdot \underline{n} \to C \cdot P \quad (n \neq 0) \\
C[\_; N] \cdot \underline{*} \to C \cdot N & \\
C[\pi_1 \_] \cdot \langle M, N \rangle \to C \cdot M & C[\pi_2 \_] \cdot \langle M, N \rangle \to C \cdot N \\
C[\mathbf{do}\ x_\sigma \leftarrow \_; N] \cdot \mathbf{ret}\, M \to C \cdot N[x_\sigma := M] & [\mathbf{produce}\ \_] \cdot \mathbf{ret}\, M \to [\mathbf{produce}\,\mathbf{ret}\ \_] \cdot M \\
C \cdot \mathbf{rec}\, x_\sigma.M \to C \cdot M[x_\sigma := \mathbf{rec}\, x_\sigma.M] &
\end{array}
$$

- Prob. must-termination judgments
  $C \,.\, M \downarrow a$
  (« whichever way you resolve the demonic non-deterministic choices, the probability that $C \,.\, M$ terminates is $>a$. »)

$$
\dfrac{}{[\mathbf{produce}\,\mathbf{ret}\ \_] \cdot \underline{*} \downarrow a}\ (a \in \mathbb{Q} \cap [0,1)) \qquad
\dfrac{}{C \cdot M \downarrow 0} \qquad
\dfrac{}{C \cdot \mathbf{abort}_{\mathbf{F}\tau} \downarrow a}\ (a \in \mathbb{Q} \cap [0,1))
$$

$$
\dfrac{C' \cdot M' \downarrow a}{C \cdot M \downarrow a}\ (\text{if } C \cdot M \to C' \cdot M') \qquad
\dfrac{C \cdot M \downarrow a \quad C \cdot N \downarrow b}{C \cdot M \oplus N \downarrow (a+b)/2} \qquad
\dfrac{C \cdot M \downarrow a \quad C \cdot N \downarrow a}{C \cdot M \owedge N \downarrow a}
$$

$$
\dfrac{[\_] \cdot M \downarrow b \quad C \cdot \underline{*} \downarrow a}{C \cdot \bigcirc_{>b} M \downarrow a} \qquad
\dfrac{C \cdot \mathbf{ifz}\ M\ N\ P \downarrow a}{C \cdot \mathbf{pifz}\ M\ N\ P \downarrow a} \qquad
\dfrac{C \cdot N \downarrow a \quad C \cdot P \downarrow a}{C \cdot \mathbf{pifz}\ M\ N\ P \downarrow a}
$$

OPERATIONAL SEMANTICS

- A Krivine machine for deterministic operations, working on configurations
  $C \,.\, M$

$$
\begin{array}{cc}
C \cdot E[M] \to CE \cdot M & C[\_N] \cdot \lambda x_\sigma.M \to C \cdot M[x_\sigma := N] \\
C[\_\ \mathbf{to}\ x_\sigma\ \mathbf{in}\ N] \cdot \mathbf{produce}\, M \to C \cdot N[x_\sigma := M] & C[\mathbf{force}\ \_] \cdot \mathbf{thunk}\, M \to C \cdot M \\
[\_] \cdot \mathbf{produce}\, M \to [\mathbf{produce}\ \_] \cdot M & \\
C[\mathbf{pred}\ \_] \cdot \underline{n} \to C \cdot \underline{n-1} & C[\mathbf{succ}\ \_] \cdot \underline{n} \to C \cdot \underline{n+1} \\
C[\mathbf{ifz}\ \_\ N\ P] \cdot \underline{0} \to C \cdot N & C[\mathbf{ifz}\ \_\ N\ P] \cdot \underline{n} \to C \cdot P \quad (n \neq 0) \\
C[\_; N] \cdot \underline{*} \to C \cdot N & \\
C[\pi_1 \_] \cdot \langle M, N\rangle \to C \cdot M & C[\pi_2 \_] \cdot \langle M, N\rangle \to C \cdot N \\
C[\mathbf{do}\ x_\sigma \leftarrow \_; N] \cdot \mathbf{ret}\, M \to C \cdot N[x_\sigma := M] & [\mathbf{produce}\ \_] \cdot \mathbf{ret}\, M \to [\mathbf{produce}\,\mathbf{ret}\ \_] \cdot M \\
C \cdot \mathbf{rec}\, x_\sigma.M \to C \cdot M[x_\sigma := \mathbf{rec}\, x_\sigma.M] &
\end{array}
$$

- Prob. must-termination judgments
  $C \,.\, M \downarrow a$
  (« whichever way you resolve the demonic non-deterministic choices, the probability that $C \,.\, M$ terminates is $>a$. »)

$$
\frac{}{[\mathbf{produce}\,\mathbf{ret}\ \_] \cdot \underline{*} \downarrow a}\ (a \in \mathbb{Q} \cap [0,1)) \qquad \frac{}{C \cdot M \downarrow 0} \qquad \frac{}{C \cdot \mathbf{abort}_{\mathbf{F}\tau} \downarrow a}\ (a \in \mathbb{Q} \cap [0,1))
$$

$$
\frac{C' \cdot M' \downarrow a}{C \cdot M \downarrow a}\ (\text{if } C \cdot M \to C' \cdot M') \qquad \frac{C \cdot M \downarrow a \quad C \cdot N \downarrow b}{C \cdot M \oplus N \downarrow (a+b)/2} \qquad \frac{C \cdot M \downarrow a \quad C \cdot N \downarrow a}{C \cdot M \owedge N \downarrow a}
$$

$$
\frac{[\_] \cdot M \downarrow b \quad C \cdot \underline{*} \downarrow a}{C \cdot \bigcirc_{>b} M \downarrow a} \qquad \frac{C \cdot \mathbf{ifz}\ M\ N\ P \downarrow a}{C \cdot \mathbf{pifz}\ M\ N\ P \downarrow a} \qquad \frac{C \cdot N \downarrow a \quad C \cdot P \downarrow a}{C \cdot \mathbf{pifz}\ M\ N\ P \downarrow a}
$$

- Let $\Pr(C \,.\, M\!\downarrow) = \sup \{a \mid C \,.\, M \downarrow a\}$, $\Pr(M\!\downarrow) = \Pr([\_] \,.\, M\!\downarrow)$

ADEQUACY

- **Prop (adequacy).**
  For every M : **FVunit**,
  — $[\![M]\!]=\bot$ and $\Pr(M\downarrow)=0$, or
  — $[\![M]\!]=\varnothing$ and $\Pr(M\downarrow)=1$, or else
  — $\Pr(M\downarrow)=\min\{\nu(\{\top\}) \mid \nu \in [\![M]\!]\}$

$$
\begin{aligned}
[\![x_\sigma]\!]\,\rho &= \rho(x_\sigma) \\
[\![\lambda x_\sigma.M]\!]\,\rho &= V \in [\![\sigma]\!] \mapsto [\![M]\!]\,(\rho[x_\sigma \mapsto V]) \qquad [\![M N]\!]\,\rho = [\![M]\!]\,\rho([\![N]\!]\,\rho) \\
[\![\mathtt{produce}\, M]\!]\,\rho &= \eta^{\mathcal{Q}}([\![M]\!]\,\rho) \\
[\![M \;\mathtt{to}\; x_\sigma \;\mathtt{in}\; N]\!]\,\rho &= (V \in [\![\sigma]\!] \mapsto [\![N]\!]\,\rho[x_\sigma \mapsto V])^*([\![M]\!]\,\rho) \\
[\![\mathtt{thunk}\, M]\!]\,\rho &= [\![M]\!]\,\rho \qquad [\![\mathtt{force}\, M]\!]\,\rho = [\![M]\!]\,\rho \\
[\![\underline{*}]\!]\,\rho &= \top \qquad [\![\underline{n}]\!]\,\rho = n \\
[\![\mathtt{succ}\, M]\!]\,\rho &= \begin{cases} n+1 & \text{if } n = [\![M]\!]\,\rho \neq \bot \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathtt{pred}\, M]\!]\,\rho &= \begin{cases} n-1 & \text{if } n = [\![M]\!]\,\rho \neq \bot \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathtt{ifz}\; M\; N\; P]\!]\,\rho &= \begin{cases} [\![N]\!]\,\rho & \text{if } [\![M]\!]\,\rho = 0 \\ [\![P]\!]\,\rho & \text{if } [\![M]\!]\,\rho \neq 0, \bot \\ \bot & \text{if } [\![M]\!]\,\rho = \bot \end{cases} \\
[\![M; N]\!]\,\rho &= \begin{cases} [\![N]\!]\,\rho & \text{if } [\![M]\!]\,\rho = \top \\ \bot & \text{otherwise} \end{cases} \\
[\![\pi_1 M]\!]\,\rho &= m, [\![\pi_2 M]\!]\,\rho = n \text{ where } [\![M]\!]\,\rho = (m, n) \\
[\![\langle M, N\rangle]\!]\,\rho &= ([\![M]\!]\,\rho, [\![N]\!]\,\rho) \\
[\![\mathtt{ret}\, M]\!]\,\rho &= \delta_{[\![M]\!]\rho} \\
[\![\mathtt{do}\, x_\sigma \leftarrow M; N]\!]\,\rho &= (V \in [\![\sigma]\!] \mapsto [\![N]\!]\,\rho[x_\sigma \mapsto V])^\dagger([\![M]\!]\,\rho) \\
[\![M \oplus N]\!]\,\rho &= \frac{1}{2}([\![M]\!]\,\rho + [\![N]\!]\,\rho) \\
[\![M \owedge N]\!]\,\rho &= [\![M]\!]\,\rho \wedge [\![N]\!]\,\rho \qquad [\![\mathtt{abort}_{\mathtt{F}\tau}]\!]\,\rho = \emptyset \\
[\![\mathtt{rec}\, x_\sigma.M]\!]\,\rho &= \mathrm{lfp}(V \in [\![\sigma]\!] \mapsto [\![M]\!]\,\rho[x_\sigma \mapsto V])
\end{aligned}
$$

ADEQUACY

- **Prop (adequacy).**
  For every M : **FVunit**,
  — $[\![M]\!]=\bot$ and $\Pr(M\downarrow)=0$, or
  — $[\![M]\!]=\varnothing$ and $\Pr(M\downarrow)=1$, or else
  — $\Pr(M\downarrow)=\min\,\{\nu(\{\top\}) \mid \nu \in [\![M]\!]\}$

- I.e., $\Pr(M\downarrow)=h^*([\![M]\!])$
  where $h(\nu) = \nu(\{\top\})$

$$
\begin{aligned}
[\![x_\sigma]\!]\,\rho &= \rho(x_\sigma)\\
[\![\lambda x_\sigma.M]\!]\,\rho &= V \in [\![\sigma]\!] \mapsto [\![M]\!]\,(\rho[x_\sigma \mapsto V]) \qquad [\![MN]\!]\,\rho = [\![M]\!]\,\rho([\![N]\!]\,\rho)\\
[\![\mathtt{produce}\,M]\!]\,\rho &= \eta^{\mathcal{Q}}([\![M]\!]\,\rho)\\
[\![M\ \mathtt{to}\ x_\sigma\ \mathtt{in}\ N]\!]\,\rho &= (V \in [\![\sigma]\!] \mapsto [\![N]\!]\,\rho[x_\sigma \mapsto V])^*([\![M]\!]\,\rho)\\
[\![\mathtt{thunk}\,M]\!]\,\rho &= [\![M]\!]\,\rho \qquad [\![\mathtt{force}\,M]\!]\,\rho = [\![M]\!]\,\rho\\
[\![\underline{*}]\!]\,\rho &= \top \qquad [\![\underline{n}]\!]\,\rho = n\\
[\![\mathtt{succ}\,M]\!]\,\rho &= \begin{cases} n+1 & \text{if } n = [\![M]\!]\,\rho \neq \bot\\ \bot & \text{otherwise}\end{cases}\\
[\![\mathtt{pred}\,M]\!]\,\rho &= \begin{cases} n-1 & \text{if } n = [\![M]\!]\,\rho \neq \bot\\ \bot & \text{otherwise}\end{cases}\\
[\![\mathtt{ifz}\ M\ N\ P]\!]\,\rho &= \begin{cases} [\![N]\!]\,\rho & \text{if } [\![M]\!]\,\rho = 0\\ [\![P]\!]\,\rho & \text{if } [\![M]\!]\,\rho \neq 0, \bot\\ \bot & \text{if } [\![M]\!]\,\rho = \bot\end{cases}\\
[\![M;N]\!]\,\rho &= \begin{cases} [\![N]\!]\,\rho & \text{if } [\![M]\!]\,\rho = \top\\ \bot & \text{otherwise}\end{cases}\\
[\![\pi_1 M]\!]\,\rho &= m, [\![\pi_2 M]\!]\,\rho = n \text{ where } [\![M]\!]\,\rho = (m,n)\\
[\![\langle M,N\rangle]\!]\,\rho &= ([\![M]\!]\,\rho, [\![N]\!]\,\rho)\\
[\![\mathtt{ret}\,M]\!]\,\rho &= \delta_{[\![M]\!]\rho}\\
[\![\mathtt{do}\,x_\sigma \leftarrow M;N]\!]\,\rho &= (V \in [\![\sigma]\!] \mapsto [\![N]\!]\,\rho[x_\sigma \mapsto V])^\dagger([\![M]\!]\,\rho)\\
[\![M \oplus N]\!]\,\rho &= \frac{1}{2}([\![M]\!]\,\rho + [\![N]\!]\,\rho)\\
[\![M \owedge N]\!]\,\rho &= [\![M]\!]\,\rho \wedge [\![N]\!]\,\rho \qquad [\![\mathtt{abort}_{\mathtt{F}\tau}]\!]\,\rho = \emptyset\\
[\![\mathtt{rec}\,x_\sigma.M]\!]\,\rho &= \mathrm{lfp}(V \in [\![\sigma]\!] \mapsto [\![M]\!]\,\rho[x_\sigma \mapsto V])
\end{aligned}
$$

ADEQUACY

- **Prop (adequacy).**
  For every M : **FVunit**,
  — $[\![M]\!]=\bot$ and $\Pr(M\downarrow)=0$, or
  — $[\![M]\!]=\varnothing$ and $\Pr(M\downarrow)=1$, or else
  — $\Pr(M\downarrow)=\min\,\{\nu(\{\top\}) \mid \nu \in [\![M]\!]\}$

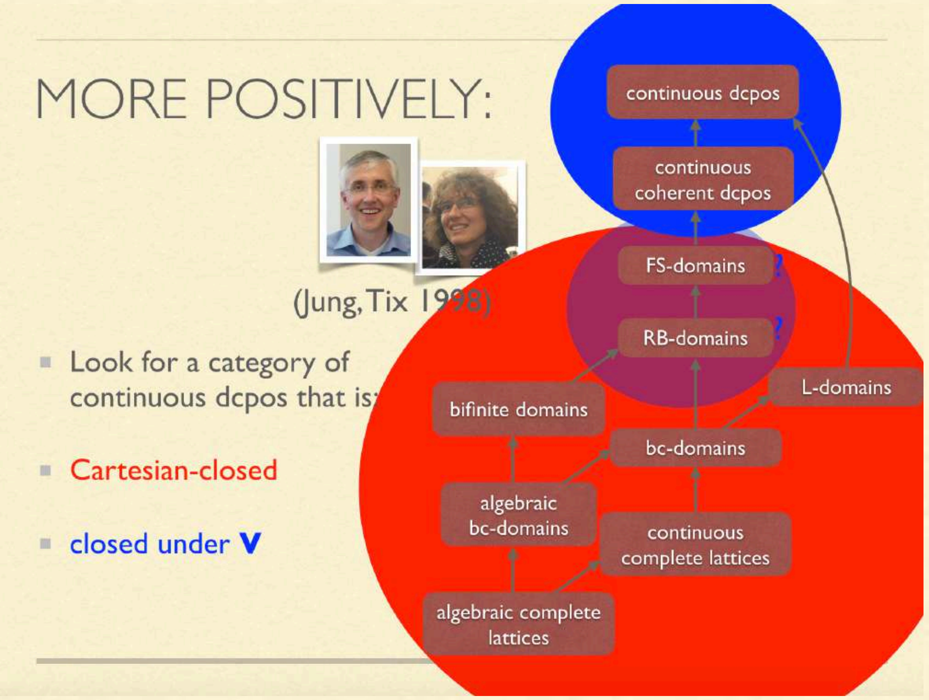
- I.e., $\Pr(M\downarrow)=h^*([\![M]\!])$
  where $h(\nu) = \nu(\{\top\})$

- Proof: by suitable logical relations.

$$
\begin{aligned}
[\![x_\sigma]\!]\,\rho &= \rho(x_\sigma) \\
[\![\lambda x_\sigma.M]\!]\,\rho &= V \in [\![\sigma]\!] \mapsto [\![M]\!]\,(\rho[x_\sigma \mapsto V]) \qquad [\![MN]\!]\,\rho = [\![M]\!]\,\rho([\![N]\!]\,\rho) \\
[\![\mathbf{produce}\,M]\!]\,\rho &= \eta^{\mathcal{Q}}([\![M]\!]\,\rho) \\
[\![M\ \mathbf{to}\ x_\sigma\ \mathbf{in}\ N]\!]\,\rho &= (V \in [\![\sigma]\!] \mapsto [\![N]\!]\,\rho[x_\sigma \mapsto V])^*([\![M]\!]\,\rho) \\
[\![\mathbf{thunk}\,M]\!]\,\rho &= [\![M]\!]\,\rho \qquad [\![\mathbf{force}\,M]\!]\,\rho = [\![M]\!]\,\rho \\
[\![\underline{*}]\!]\,\rho &= \top \qquad [\![\underline{n}]\!]\,\rho = n \\
[\![\mathbf{succ}\,M]\!]\,\rho &= \begin{cases} n+1 & \text{if } n = [\![M]\!]\,\rho \neq \bot \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathbf{pred}\,M]\!]\,\rho &= \begin{cases} n-1 & \text{if } n = [\![M]\!]\,\rho \neq \bot \\ \bot & \text{otherwise} \end{cases} \\
[\![\mathbf{ifz}\ M\ N\ P]\!]\,\rho &= \begin{cases} [\![N]\!]\,\rho & \text{if } [\![M]\!]\,\rho = 0 \\ [\![P]\!]\,\rho & \text{if } [\![M]\!]\,\rho \neq 0, \bot \\ \bot & \text{if } [\![M]\!]\,\rho = \bot \end{cases} \\
[\![M;N]\!]\,\rho &= \begin{cases} [\![N]\!]\,\rho & \text{if } [\![M]\!]\,\rho = \top \\ \bot & \text{otherwise} \end{cases} \\
[\![\pi_1 M]\!]\,\rho &= m, [\![\pi_2 M]\!]\,\rho = n \text{ where } [\![M]\!]\,\rho = (m,n) \\
[\![\langle M, N\rangle]\!]\,\rho &= ([\![M]\!]\,\rho, [\![N]\!]\,\rho) \\
[\![\mathbf{ret}\,M]\!]\,\rho &= \delta_{[\![M]\!]\rho} \\
[\![\mathbf{do}\,x_\sigma \leftarrow M;N]\!]\,\rho &= (V \in [\![\sigma]\!] \mapsto [\![N]\!]\,\rho[x_\sigma \mapsto V])^\dagger([\![M]\!]\,\rho) \\
[\![M \oplus N]\!]\,\rho &= \frac{1}{2}([\![M]\!]\,\rho + [\![N]\!]\,\rho) \\
[\![M \owedge N]\!]\,\rho &= [\![M]\!]\,\rho \wedge [\![N]\!]\,\rho \qquad [\![\mathbf{abort}_{\mathbf{F}\tau}]\!]\,\rho = \emptyset \\
[\![\mathbf{rec}\,x_\sigma.M]\!]\,\rho &= \mathrm{lfp}(V \in [\![\sigma]\!] \mapsto [\![M]\!]\,\rho[x_\sigma \mapsto V])
\end{aligned}
$$

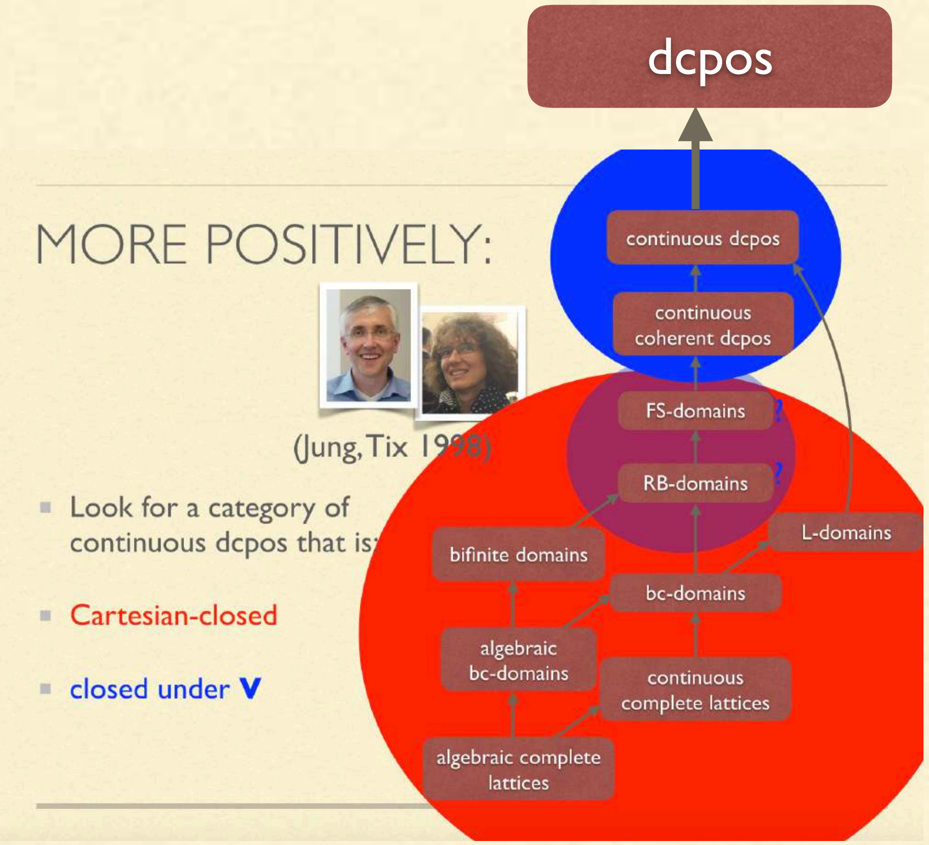
NOTE

- None of that yet requires CCCs of **continuous** (or algebraic) domains



NOTE

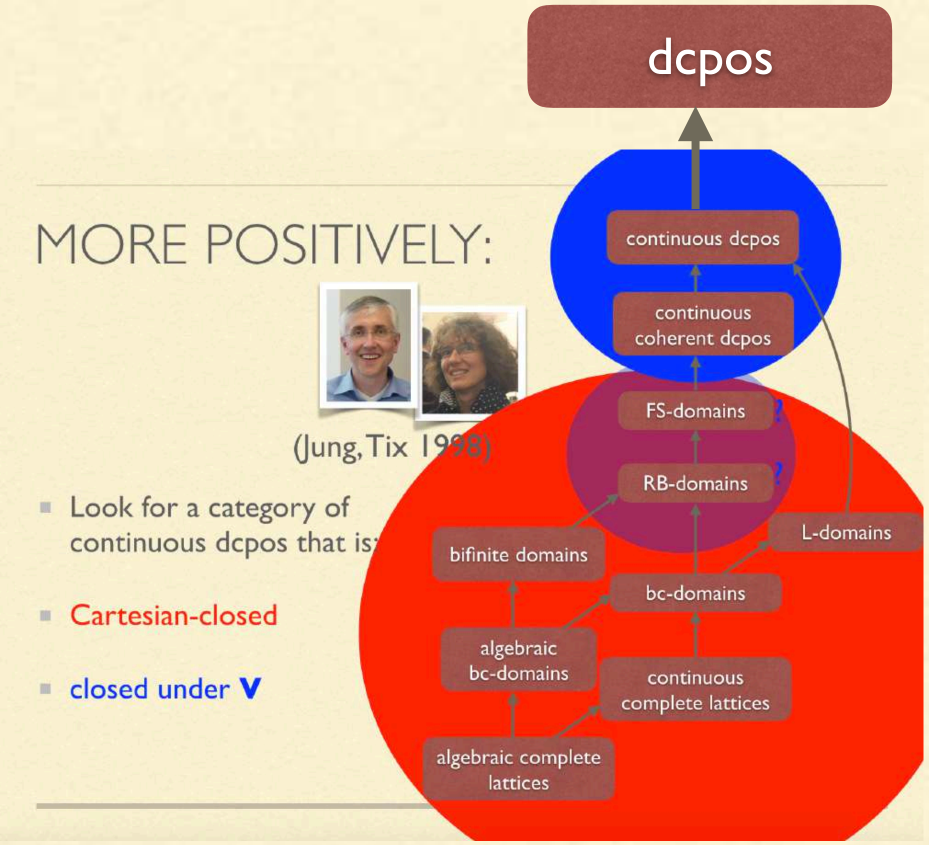
- None of that yet requires CCCs of **continuous** (or algebraic) domains
- Soundness/adequacy works even for non-call-by-push-value probabilistic languages, working in the CCC **Dcpo**



NOTE

- None of that yet requires CCCs of **continuous** (or algebraic) domains

- Soundness/adequacy works
even for non-call-by-push-value
probabilistic languages,
working in the CCC **Dcpo**



- Continuity is only needed for more advanced applications:
— full abstraction (next)
— commutativity of the **V** monad (Fubini) at higher types

THE CONTEXTUAL PREORDER

- Let $M \lesssim N$ iff for every context C of output type **FVunit**,
  $$\Pr(C\,.\,M{\downarrow}) \leq \Pr(C\,.\,N{\downarrow})$$

THE CONTEXTUAL PREORDER

- Let $M \lesssim N$ iff for every context C of output type **FVunit**,

$$\Pr(C\,.\,M\!\downarrow) \leq \Pr(C\,.\,N\!\downarrow)$$

- $M \lesssim N$ iff for every context C of output type **FVunit**,

$$h^*([\![C[M]]\!]) \leq h^*([\![C[N]]\!]) \qquad \text{(adequacy)}$$

THE CONTEXTUAL PREORDER

- Let $M \lesssim N$ iff for every context C of output type **FVunit**,
$$\Pr(C\,.\,M\!\downarrow) \leq \Pr(C\,.\,N\!\downarrow)$$

- $M \lesssim N$ iff for every context C of output type **FVunit**,
$$h^*([\![C[M]]\!]) \leq h^*([\![C[N]]\!]) \qquad \text{(adequacy)}$$

- **Corollary.** If $[\![M]\!] \leq [\![N]\!]$ then $M \lesssim N$.

THE CONTEXTUAL PREORDER

- Let $M \lesssim N$ iff for every context C of output type **FVunit**,
  $$\Pr(C\,.\,M\downarrow) \leq \Pr(C\,.\,N\downarrow)$$
- $M \lesssim N$ iff for every context C of output type **FVunit**,
  $$h^*([\![C[M]]\!]) \leq h^*([\![C[N]]\!]) \qquad \text{(adequacy)}$$
- **Corollary.** If $[\![M]\!] \leq [\![N]\!]$ then $M \lesssim N$.
- Proof. $[\![C[M]]\!] = [\![C]\!]\,([\![M]\!]) \leq [\![C]\!]\,([\![N]\!]) = [\![C[N]]\!]$
  since $[\![C]\!]$ ($= [\![\lambda x\,.\,C[x]]\!]$) is Scott-continuous hence monotonic.
  Then apply h^*, which is monotonic as well. $\square$

THE APPLICATIVE PREORDER

- Let $M \leq N$ iff for every context C of output type **FVunit**,
  $$\Pr(C\,.\,M{\downarrow}) \leq \Pr(C\,.\,N{\downarrow})$$
- Let $M \leq^{\mathrm{app}} N$ iff for every term $P : \tau \to$ **FVunit**,
  $$\Pr(PM{\downarrow}) \leq \Pr(PN{\downarrow})$$

THE APPLICATIVE PREORDER

- Let $M \leq N$ iff for every context C of output type **FVunit**,
  $$\Pr(C\,.\,M\!\downarrow) \leq \Pr(C\,.\,N\!\downarrow)$$
- Let $M \leq^{app} N$ iff for every term $P : \tau \to$ **FVunit**,
  $$\Pr(PM\!\downarrow) \leq \Pr(PN\!\downarrow)$$
- **Proposition** (« Milner's context lemma » in PCF):
  $M \leq N$ iff $M \leq^{app} N$.

THE APPLICATIVE PREORDER

- Let $M \lesssim N$ iff for every context C of output type **FVunit**,

  $\Pr(C\,.\,M\downarrow) \leq \Pr(C\,.\,N\downarrow)$

- Let $M \lesssim^{app} N$ iff for every term $P : \tau \to$ **FVunit**,

  $\Pr(PM\downarrow) \leq \Pr(PN\downarrow)$

- **Proposition** (« Milner's context lemma » in PCF):

  $M \lesssim N$ iff $M \lesssim^{app} N$.

- Proof: based on an idea of A. Jung (Streicher 06), reusing our previous logical relation.

FULL ABSTRACTION?

- **Conjecture (full abstraction):** $[\![M]\!] \leq [\![N]\!]$ iff $M \leq N$.

FULL ABSTRACTION?

- **Conjecture (full abstraction):** $[\![M]\!] \leq [\![N]\!]$ iff $M \leq N$.
- **Wrong.** (As in (Plotkin 77).) Using another logical relation, for every P : **int** → **int** → **Fint**, if $[\![P]\!](\perp)(0)=[\![P]\!](0)(\perp)=\{0\}$ then $[\![P]\!](\perp)(\perp)=\{0\}$

FULL ABSTRACTION?

- **Conjecture (full abstraction):** $[\![M]\!] \leq [\![N]\!]$ iff $M \lesssim N$.
- **Wrong.** (As in (Plotkin 77).) Using another logical relation, for every P : **int** → **int** → **Fint**, if $[\![P]\!](\bot)(0)=[\![P]\!](0)(\bot)=\{0\}$
  then $[\![P]\!](\bot)(\bot)=\{0\}$
- Hence with $M = \lambda P\,.\,P\Omega 0{=}{=}0\ \&\&\ P0\Omega{=}{=}0$
  $N = \lambda P\,.\,MP\ \&\&\ P\Omega\Omega{=}{=}0$ (and some syntactic sugar)
  we have $M \lesssim N$.

FULL ABSTRACTION?

- **Conjecture (full abstraction):** $[\![M]\!] \leq [\![N]\!]$ iff $M \leq N$.
- **Wrong.** (As in (Plotkin 77).) Using another logical relation, for every P : **int** → **int** → **Fint**, if $[\![P]\!](\bot)(0)=[\![P]\!](0)(\bot)=\{0\}$ then $[\![P]\!](\bot)(\bot)=\{0\}$
- Hence with $M = \lambda P\ .\ P\Omega 0{=}{=}0\ \&\&\ P0\Omega{=}{=}0$
  $N = \lambda P\ .\ MP\ \&\&\ P\Omega\Omega{=}{=}0$ (and some syntactic sugar)
  we have $M \leq N$.
- But $[\![M]\!] \not\leq [\![N]\!]$ since $[\![M]\!](\text{por})=\top$, $[\![N]\!](\text{por})=\bot$.

FULL ABSTRACTION?

- Add parallel if **pifz**:
  $[\![\mathbf{pifz}\ M\ N\ P]\!] = [\![N]\!]$ if $[\![M]\!]=0$
  $[\![P]\!]$ if $[\![M]\!]\neq 0,\perp$
  $[\![N]\!]\wedge[\![P]\!]$ (not $\perp$!) if $[\![M]\!]=\perp$

FULL ABSTRACTION?

- Add parallel if **pifz**:
  $[\![\textbf{pifz}\ M\ N\ P]\!] = [\![N]\!]$ if $[\![M]\!]=0$
  $[\![P]\!]$ if $[\![M]\!]\neq 0, \perp$
  $[\![N]\!] \wedge [\![P]\!]$ (not $\perp$!) if $[\![M]\!]=\perp$

- **Conjecture (full abstraction):** $[\![M]\!] \leq [\![N]\!]$ iff $M \leq N$.

FULL ABSTRACTION?

- Add parallel if **pifz**:
  $[\![\mathbf{pifz}\ M\ N\ P]\!] = [\![N]\!]$ if $[\![M]\!]=0$
  $[\![P]\!]$ if $[\![M]\!]\neq 0, \bot$
  $[\![N]\!] \wedge [\![P]\!]$ (not $\bot$!) if $[\![M]\!]=\bot$

- **Conjecture (full abstraction):** $[\![M]\!] \leq [\![N]\!]$ iff $M \leq N$.

- **Still wrong.** As in (GL 15), missing statistical termination testers.

STATISTICAL TERMINATION TESTERS

- Let $M = \lambda P \,.\, P(\Omega \oplus \textbf{ret}\ *)$
  $N = \lambda P \,.\, P(\Omega) \oplus \textbf{ret}\ *$ (modulo some missing **force**, **produce**, etc.)
  Then $M \leq N$, even with **pifz**

STATISTICAL TERMINATION TESTERS

- Let $M = \lambda P \,.\, P(\Omega \oplus \textbf{ret}\ *)$
  $N = \lambda P \,.\, P(\Omega) \oplus \textbf{ret}\ *$ (modulo some missing **force**, **produce**, etc.)
  Then $M \leq N$, even with **pifz**

- But $[\![M]\!] \not\leq [\![N]\!]$ since $[\![M]\!]([>b]) = \uparrow\delta_\top$, $[\![N]\!]([>b]) = \bot$ for all $b<1/2$,
  where $[>b] : [\![\textbf{Vunit}]\!] \to [\![\textbf{FVunit}]\!]$
  maps every ν to 'termination' ($\uparrow\delta_\top$) if $\nu(\{\top\})>b$,
  to $\bot$ otherwise.

STATISTICAL TERMINATION TESTERS

- Let $M = \lambda P\,.\,P(\Omega \oplus \mathbf{ret}\ {*})$
  $N = \lambda P\,.\,P(\Omega) \oplus \mathbf{ret}\ {*}$ (modulo some missing **force**, **produce**, etc.)
  Then $M \leq N$, even with **pifz**

- But $[\![M]\!] \not\leq [\![N]\!]$ since $[\![M]\!]([>b]) = \uparrow\delta_\top$, $[\![N]\!]([>b]) = \perp$ for all $b<1/2$,
  where $[>b] : [\![\mathbf{Vunit}]\!] \to [\![\mathbf{FVunit}]\!]$
  maps every ν to 'termination' ($\uparrow\delta_\top$) if $\nu(\{\top\})>b$,
  to $\perp$ otherwise.

- $[>b]$ tests whether the prob. that its argument terminates is $> b$.

FULL ABSTRACTION

- Add **pifz** + $\bigcirc_{>b}$ (with the semantics of $[>b]$, $0<b<1$ dyadic)

FULL ABSTRACTION

- Add **pifz** + $\bigcirc_{>b}$ (with the semantics of $[>b]$, $0<b<1$ dyadic)
- **Theorem (full abstraction):** with **pifz** and $\bigcirc_{>b}$,
  $[\![M]\!] \leq [\![N]\!]$ iff $M \leq N$.

FULL ABSTRACTION

- Add **pifz** + $\bigcirc_{>b}$ (with the semantics of $[>b]$, $0<b<1$ dyadic)
- **Theorem (full abstraction):** with **pifz** and $\bigcirc_{>b}$,
  $[\![M]\!] \leq [\![N]\!]$ iff $M \leq N$.
- And now a glimpse of the argument…

FULL ABSTRACTION: PROOF

- We assume $[\![M]\!] \not\leq [\![N]\!]$, and we wish to prove not ($M \leq N$).

FULL ABSTRACTION: PROOF

- We assume $[\![M]\!] \not\leq [\![N]\!]$, and we wish to prove not ($M \leq N$).
- There is a subbasic open set U / $[\![M]\!] \in U$, $[\![N]\!] \notin U$

  (because $\leq$ is the specialization ordering of the Scott topology)

FULL ABSTRACTION: PROOF

- We assume $[\![M]\!] \not\leq [\![N]\!]$, and we wish to prove not ($M \leq N$).
- There is a subbasic open set U / $[\![M]\!] \in U$, $[\![N]\!] \notin U$
  (because $\leq$ is the specialization ordering of the Scott topology)
- If U is **definable** by a term P
  (i.e., $[\![P]\!]$ maps each $x \in U$ to $\uparrow\delta_\top$ and each $x \notin U$ to $\perp$)
  then $[\![PM]\!] = \uparrow\delta_\top$, $[\![PN]\!] = \perp$, so not ($M \leq^{app} N$).
  Conclude by Milner's context lemma.

FULL ABSTRACTION: PROOF

- We assume $[\![M]\!] \not\leq [\![N]\!]$, and we wish to prove not $(M \leq N)$.
- There is a subbasic open set U / $[\![M]\!] \in U$, $[\![N]\!] \notin U$

  (because $\leq$ is the specialization ordering of the Scott topology)
- If U is **definable** by a term P
  (i.e., $[\![P]\!]$ maps each $x \in U$ to $\uparrow\delta_\top$ and each $x \notin U$ to $\perp$)
  then $[\![PM]\!] = \uparrow\delta_\top$, $[\![PN]\!] = \perp$, so not $(M \leq^{app} N)$.
  Conclude by Milner's context lemma.
- Challenge: show that each $[\![\tau]\!]$ has a subbase of definable opens.

PROBABILISTIC TYPES

- $[\![\mathbf{V}\sigma]\!]$: subbasic opens $[U>b]$
  where U is a basic open of $[\![\sigma]\!]$, b dyadic in $(0,1)$

PROBABILISTIC TYPES

- $[\![\mathbf{V}\sigma]\!]$: subbasic opens $[U>b]$
  where U is a basic open of $[\![\sigma]\!]$, b dyadic in $(0,1)$
- I.e. (requires $[\![\sigma]\!]$ **continuous**!) Scott=weak topology (Kirch 93)

PROBABILISTIC TYPES

- $[\![\mathbf{V}\sigma]\!]$: subbasic opens $[U>b]$

  where U is a basic open of $[\![\sigma]\!]$, b dyadic in $(0,1)$

- I.e. (requires $[\![\sigma]\!]$ **continuous**!) Scott=weak topology (Kirch 93)

- test $\nu \in [U>b]$: iff $\nu(U)>b$, implemented through

  '$\bigcirc_{>b}$ (**do** $x_\sigma \leftarrow \nu$; $\langle$test $x_\sigma \in U\rangle$)'

PROBABILISTIC TYPES

- $[\![\mathbf{V}\sigma]\!]$: subbasic opens $[U>b]$
  where U is a basic open of $[\![\sigma]\!]$, b dyadic in $(0,1)$
- I.e. (requires $[\![\sigma]\!]$ **continuous**!) Scott=weak topology (Kirch 93)
- test $\nu\in[U>b]$: iff $\nu(U)>b$, implemented through
  '$\bigcirc_{>b}$ (**do** $x_\sigma \leftarrow \nu$; $\langle$test $x_\sigma \in U\rangle$)'
- Note: $[\![\mathbf{V}\sigma]\!]$ also has a basis of
  $\sum_x a_x\, \delta_x$, a_x dyadic, each x from a basis of $[\![\sigma]\!]$
  implementable using **ret** and $\oplus$

F TYPES

- $[\![\mathbf{F}\sigma]\!]$: subbasic opens $\Box U$
  where U is a basic open of $[\![\sigma]\!]$

F TYPES

- $[\![\mathbf{F}\sigma]\!]$: subbasic opens $\Box U$
  where U is a basic open of $[\![\sigma]\!]$

- I.e. (requires $[\![\sigma]\!]$ **continuous**!) Scott=upper Vietoris topology

F TYPES

- $[\![\mathbf{F}\sigma]\!]$: subbasic opens $\Box U$
  where U is a basic open of $[\![\sigma]\!]$
- I.e. (requires $[\![\sigma]\!]$ **continuous**!) Scott=upper Vietoris topology
- $\mathtt{test}\ Q \in \Box U$: iff $Q \subseteq U$, implemented through
  ‘Q **to** x_σ **in** $\langle \mathtt{test}\ x_\sigma \in U \rangle$’

F TYPES

- $[\![\mathbf{F}\sigma]\!]$: subbasic opens $\Box U$
  where U is a basic open of $[\![\sigma]\!]$

- I.e. (requires $[\![\sigma]\!]$ **continuous**!) Scott=upper Vietoris topology

- $\mathtt{test}\ Q \in \Box U$: iff $Q \subseteq U$, implemented through
  'Q **to** x_σ **in** $\langle \mathtt{test}\ x_\sigma \in U \rangle$'

- Note: $[\![\mathbf{F}\sigma]\!]$ also has a basis of $\uparrow\{x_1, \ldots, x_n\}$,
  (each x_i from a basis of $[\![\sigma]\!]$), plus $\perp$,
  implementable using **produce**, **abort**, and $\circledwedge$

FUNCTION TYPES (1/2)

- $[\![\sigma \to \tau]\!]$: subbasic opens $[x \mapsto V]$,
  where x is from some basis of $[\![\sigma]\!]$,
  V is a subbasic open of $[\![\tau]\!]$

FUNCTION TYPES (1/2)

- $[\![\sigma \to \tau]\!]$: subbasic opens $[x \mapsto V]$,
  where x is from some basis of $[\![\sigma]\!]$,
  V is a subbasic open of $[\![\tau]\!]$
- I.e. (uses $[\![\sigma]\!]$ **bc-domain**!) Scott=top. of pointwise convergence

FUNCTION TYPES (1/2)

- $[\![\sigma \to \tau]\!]$: subbasic opens $[x \mapsto V]$,
  where x is from some basis of $[\![\sigma]\!]$,
  V is a subbasic open of $[\![\tau]\!]$
- I.e. (uses $[\![\sigma]\!]$ **bc-domain**!) Scott=top. of pointwise convergence
- test f in $[x \mapsto V]$: iff $f(x) \in V$, implemented straightforwardly

FUNCTION TYPES (1/2)

- $[\![\sigma \to \tau]\!]$: subbasic opens $[x \mapsto V]$,
  where x is from some basis of $[\![\sigma]\!]$,
  V is a subbasic open of $[\![\tau]\!]$
- I.e. (uses $[\![\sigma]\!]$ **bc-domain**!) Scott=top. of pointwise convergence
- test f in $[x \mapsto V]$: iff $f(x) \in V$, implemented straightforwardly
- Note we need to also define a **basis** of each type $[\![\sigma]\!]$ now.
  We have them for **V** and **F** types, and the difficult case is for $\to$.

FUNCTION TYPES (2/2)

- Basis for $[\![\sigma \to \tau]\!]$: step functions $\vee_{1 \leq i \leq n} U_i \searrow y_i$
  mapping each x to $\vee\{y_i \mid x \in U_i\}$…
  but that sup is hard to implement — we only have infs.

FUNCTION TYPES (2/2)

- Basis for $[\![\sigma \to \tau]\!]$: step functions $\vee_{1 \leq i \leq n} U_i \searrow y_i$
  mapping each x to $\vee\{y_i \mid x \in U_i\}$…
  but that sup is hard to implement — we only have infs.

- Trick: for each subset I of $\{1, \ldots, n\}$,
  let U_I = intersection of U_i, i in I,
  y_I = sup of y_i, i in I.

  Then $\vee_{1 \leq i \leq n} U_i \searrow y_i\,(x) = \inf\{y_I \mid I \supseteq I_x\}$ where $I_x = \{i \mid x \in U_i\}$

FUNCTION TYPES (2/2)

- Basis for $[\![\sigma \to \tau]\!]$: step functions $\vee_{1 \leq i \leq n} U_i \searrow y_i$
  mapping each x to $\vee\{y_i \mid x \in U_i\}$…
  but that sup is hard to implement — we only have infs.

- Trick: for each subset I of $\{1, \ldots, n\}$,
  let U_I = intersection of U_i, i in I,
  y_I = sup of y_i, i in I.
  Then $\vee_{1 \leq i \leq n} U_i \searrow y_i\,(x) = \inf\,\{y_I \mid I \supseteq I_x\}$ where $I_x = \{i \mid x \in U_i\}$

- Additional difficulties (need for **pifz** notably)… omitted. Done! $\square$

SUMMARY

- Circumventing the trouble with **V** by using two classes of types, as provided by call-by-push-value
- We obtain (inequational) full abstraction with prob. choice + demonic non-determinism
- Questions?