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

Jean Goubault-Larrecq
PCF, probabilistic choice, and the trouble with $\mathbf{V}$

- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction
PLOTKIN’S PCF (1977)

- Types $\sigma, \tau, \ldots ::= \text{int} \mid \sigma \rightarrow \tau$

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

(All terms are typed. Call by name.)
PLOTKIN’S PCF (1977)

- **Types** \( \sigma, \tau, \ldots ::= \text{int} \mid \sigma \to \tau \)

- **Terms** \( M, N, \ldots ::= x_\tau \mid MN \mid \lambda x_\sigma . M \mid \text{rec } x_\sigma . M \mid n \mid \text{succ } M \mid \text{pred } M \mid \text{ifz } M N P \)

- (All terms are typed. Call by name.)

- **An operational semantics:** \( M \to^* N \)

- **A denotational semantics:** \( \llbracket M \rrbracket = n \) iff \( M \to^* n \)

- **Adequacy:**
  for every ground \( M : \text{int}, \)
  \( \llbracket M \rrbracket = n \) iff \( M \to^* n \)
PLOTKIN’S PCF (1977)

- An **operational** semantics:
  \[ M \rightarrow^* N \]

- A **denotational** semantics:
  \[ \llbracket M \rrbracket \]

- **Adequacy:**
  for every ground \( M : \text{int} \),
  \( \llbracket M \rrbracket = n \) iff \( M \rightarrow^* n \)

- **Contextual preordering:**
  \( M \preceq N \) iff
  for every context \( C : \text{int} \),
  \( C[M] \rightarrow^* n \Rightarrow C[N] \rightarrow^* n \)

- **Fact:** if \( \llbracket M \rrbracket \preceq \llbracket N \rrbracket \) then \( M \preceq N \)

- Converse is **full abstraction**.
  Fails for PCF, works for PCF+por
Every type $\tau$ interpreted as a dcpo $\llbracket \tau \rrbracket$... = poset in which every directed family $D$ has a supremum $\bigvee D$
Every type $T$ interpreted as a **dcpo** $[[T]]$... = poset in which every directed family $D$ has a supremum $\bigvee D$

$[[\text{int}]] = \mathbb{Z}_\perp (\perp \leq n, \text{all } n \text{ incomparable})$
**DCPOs**

- Every type $\tau$ interpreted as a **dcpo** $\llbracket \tau \rrbracket$...
  
  $= \text{poset in which every directed family } D$

  has a supremum $\bigvee D$

- $\llbracket \text{int} \rrbracket = \mathbb{Z}_\bot$ ($\bot \leq n, \text{all } n \text{ incomparable}$)

- $\llbracket \sigma \to \tau \rrbracket = \llbracket \sigma \rrbracket \to \llbracket \tau \rrbracket$
  
  dcpo of Scott-continuous maps $: \llbracket \sigma \rrbracket \to \llbracket \tau \rrbracket$

  (monotonic + preserves directed sups)
THE SEMANTICS OF PCF

- Types  \( \sigma, \tau, \ldots ::= \text{int} | \sigma \rightarrow \tau \)

- Terms  \( M, N, \ldots ::= x_\tau \)
  \[
  | MN \\
  | \lambda x_\sigma.M \\
  | \text{rec } x_\sigma.M \\
  | n \\
  | \text{succ } M \\
  | \text{pred } M \\
  | \text{ifz } M \, N \, P
  \]

\[
\begin{align*}
\llbracket MN \rrbracket &= \llbracket M \rrbracket(\llbracket N \rrbracket) \\
\llbracket \lambda x_\sigma.M \rrbracket &= (V \mapsto \llbracket M \rrbracket[x_\sigma := V])
\end{align*}
\]

- Meaningful since \textbf{Dcpo} is a Cartesian-closed category
CARTESIAN-CLOSEDNESS

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

- Meaningful since $\mathbf{Dcpo}$ is a Cartesian-closed category

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

- In the case of PCF, each $\llbracket \tau \rrbracket$ is an algebraic bc-domain, making that possible.

- Cartesian-closed… good.
In order to prove full abstraction (with por), we require to be able to approximate elements of $⟦T⟧$ by definable elements $⟦M⟧$.

In the case of PCF, each $⟦T⟧$ is an algebraic bc-domain, making that possible.

Cartesian-closed… good.
In order to prove full abstraction (with por), we require to be able to approximate elements of $\llbracket T \rrbracket$ by definable elements $\llbracket M \rrbracket$.

In the case of PCF, each $\llbracket T \rrbracket$ 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):**
  \[ x \ll y \text{ iff for every directed } D \text{ such that } y \leq \bigvee D, \]
  \[ x \text{ is already below some element of } D \]
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):**
  \[ x \ll y \text{ iff for every directed } D \text{ such that } y \leq \bigvee D, \]
  \[ x \text{ is already below some element of } D \]

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

- Ex: the finite subsets of \( A \) form a basis of \( P(A) \) 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)
ADDITION PROBABILITIES

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

- Terms
  \( M, N, \ldots ::= \ldots \)
  \( \mid M \oplus N \)
  \( \mid \text{ret } M \)
  \( \mid \text{do } x_\sigma \leftarrow M; N \)
ADDING PROBABILITIES

- Types
  \( \sigma, \tau, \ldots ::= \text{int} \mid \sigma \rightarrow \tau \mid V_\tau \)

- Terms
  \( M, N, \ldots ::= \ldots \)
  \( \mid M \oplus N \)
  \( \mid \text{ret} \ M \)
  \( \mid \text{do} \ x_\sigma \leftarrow M; N \)

Monadic type of subprobability valuations over \( \tau \)
ADDING PROBABILITIES

- Types
  \( \sigma, \tau, \ldots ::= \text{int} | \sigma \rightarrow \tau | \mathbf{V}\tau \)

- Terms
  \( M, N, \ldots ::= \ldots \)
  \( | M \oplus N \)
  \( | \text{ret } M \)
  \( | \text{do } x_\sigma \leftarrow M; N \)

Monadic type of subprobability valuations over \( \tau \)

with \( M, N : \mathbf{V}\tau \),
choose between \( M \) and \( N \)
with probability \( 1/2 \)
ADDING PROBABILITIES

Types
\[ \sigma, \tau, \ldots ::= \text{int} \mid \sigma \to \tau \mid \mathbf{V}_\tau \]

Terms
\[ M, N, \ldots ::= \ldots \mid M \oplus N \mid \text{ret } M \mid \text{do } x_\sigma \leftarrow M; N \]

Monadic type of subprobability valuations over \( \tau \)

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

monadic constructions:
\[ M : \tau \Rightarrow \text{ret } M : \mathbf{V}_\tau \]
\[ M : \mathbf{V}_\sigma \quad N : \mathbf{V}_\tau \Rightarrow \text{do } x_\sigma \leftarrow M; N : \mathbf{V}_\tau \]

(Moggi 1991)
THE TROUBLE WITH \textbf{V}

(Jung, Tix 1998)

- Look for a category of continuous dcpos that is...
THE TROUBLE WITH $\forall$

(Jung, Tix 1998)

- Look for a category of continuous dcpos that is...
- Cartesian-closed
THE TROUBLE WITH \( \mathbf{V} \)

(Jung, Tix 1998)

- Look for a category of continuous dcpos that is...
  - Cartesian-closed
  - closed under \( \mathbf{V} \)
MORE POSITIVELY:

- Look for a category of continuous dcpos that is:
  - Cartesian-closed
  - closed under $\mathcal{V}$

(Jung, Tix 1998)
MORE POSITIVELY:

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

(Jung, Tix 1998)
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

MORE POSITIVELY:

- Look for a category of continuous dpos that is
  - Cartesian-closed
  - closed under $\forall$

(Jung, Tix 1996)
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 $\nabla$

- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction
TWO KINDS OF TYPES?

- No such problem with two kinds of types:
  \[ \sigma, \tau, \ldots ::= \textbf{int} \mid \ldots \mid \sigma \times \tau \mid \bigvee \tau \]
  \[ \sigma, \tau, \ldots ::= \ldots \mid \sigma \to \tau \]

  - continuous (coherent) dcpos
  - bc-domains/continuous lattices
CALL-BY-PUSH-VALUE

- No such problem with two kinds of types:
  \[ \sigma, \tau, \ldots ::= \text{int} \mid \text{unit} \mid U\sigma \mid \sigma \times \tau \mid V\tau \]
  \[ \sigma, \tau, \ldots ::= F\sigma \mid \sigma \to \tau \]
  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 blaine@eecs.qmul.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 these paradigms can be seen as existing, 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 done”. Using an example program, we see that the lambda-calculus primitives can be understood as push/pop commands for an operational 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)
No such problem with two kinds of types:

\[ \sigma, \tau, \ldots ::= \text{int} | \text{unit} | \text{unit} \sigma | \sigma \times \tau | \nu \tau \]

\[ \sigma, \tau, \ldots ::= \text{false} \sigma | \sigma \rightarrow \tau \]

This is the type structure of Paul B. Levy's call-by-push-value (except for the \( \nu \) 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 statistics.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 section, both operational and denotational semantics for these paradigms can be seen as existing, 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 done". Using an example program, we show that the lambda-calculus primitives can be understood as push/pull commands for an operadetic.

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, or 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:
  \[ \sigma, \tau, \ldots ::= \text{int} \mid \text{unit} \mid U\sigma \mid \sigma \times \tau \mid V\tau \]
  \[ \sigma, \tau, \ldots ::= F\sigma \mid \sigma \to \tau \]

- 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 statistics.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 these paradigms can be seen as existing, 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 done”. 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 now and some were known but previously unappreciated mysteries.

(Levy 1999)
\( \sigma, \tau, \ldots ::= \textbf{int} \mid \textbf{unit} \mid \sigma \times \tau \mid \mathbf{V}\tau \)

\( \sigma, \tau, \ldots ::= \sigma \rightarrow \tau \)

- \( \sigma \times \tau \mid \mathbf{V}\tau \) : continuous (coherent) dcpos
- \( \sigma \rightarrow \tau \) : bc-domains/continuous lattices
U AND F

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

- $\sigma, \tau, \ldots ::= \sigma \to \tau$

- **U** converts from bc-domains to continuous coherent dcpos
  - ... semantically the identity: $\llbracket U\sigma \rrbracket = \llbracket \sigma \rrbracket$

**Notes**
- Continuous (coherent) dcpos
- Bc-domains/continuous lattices
\[ U \text{ AND } F \]

- \( \sigma, \tau, \ldots ::= \text{int} | \text{unit} | U\sigma | \sigma \times \tau | V\tau \)
- \( \sigma, \tau, \ldots ::= \sigma \rightarrow \tau \)

- **U** converts from bc-domains to continuous coherent dcpos
  
  ... semantically the identity: \([U\sigma] = [\sigma]\)

- \( M, N, \ldots ::= \ldots \)
  
  \| \text{force } M \) \((U\sigma \rightarrow \sigma)\)

  \| \text{thunk } M \) \((\sigma \rightarrow U\sigma)\)

- continuous (coherent) dcpos

- bc-domains/continuous lattices
\[ U \text{ AND } F \]

- \( \sigma, \tau, \ldots ::= \text{int} | \text{unit} | U\sigma | \sigma \times \tau | V\tau \)
- \( \sigma, \tau, \ldots ::= \sigma \rightarrow \tau \)

- \( U \) converts from bc-domains to continuous coherent dcpos
  ... semantically the identity: \([U\sigma]=\sigma]\)

- \( M, N, \ldots ::= \ldots \)
  - \( | \text{force } M \quad (U\sigma \rightarrow \sigma) \)
  - \( | \text{thunk } M \quad (\sigma \rightarrow U\sigma) \)
  - \([\text{force } M] = [M]\)
  - \([\text{thunk } M] = [M]\)

- \( \text{force thunk } M \rightarrow M \)
**U AND F**

- \( \sigma, \tau, \ldots \) := \text{int} \mid \text{unit} \mid U\sigma \mid \sigma \times \tau \mid V\tau
- \( \sigma, \tau, \ldots \) := F\sigma \mid \sigma \rightarrow \tau

- **U** converts from bc-domains to continuous coherent dcpos
  ... semantically the identity: \([U\sigma] = [\sigma]\)

- **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.)
THE SMYTH POWERDOMAIN

- \( Q_X = \{ \text{compact saturated subsets of } X \} \), 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

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

- **Unit**: \( \eta : X \rightarrow \mathcal{Q}X : x \mapsto \uparrow x \) (continuous)

- **Extension**: for \( f : X \rightarrow L \) where \( L \) continuous complete lattice,
  \( f^* : \mathcal{Q}X \rightarrow L : Q \mapsto \inf \{ f(x) \mid x \in Q \} \)
  - if \( f \) is continuous then \( f^* \) is continuous
  - \( f^* \circ \eta = f \)
  - \( f^* \circ g^* = (f^* \circ g)^* \)
THE SMYTH\textsubscript{⊥} POWERDOMAIN

- \(Q_{\bot}X = QX\) plus a fresh bottom \(\bot\) defines a(nother) **monad** on the cat. of cont. coh. dcpos.

- **Unit**: \(\eta : X \to Q_{\bot}X : x \mapsto \uparrow x\) (continuous)

- **Extension**: for \(f : X \to L\) where \(L\) continuous complete lattice,
  let \(f^* : Q_{\bot}X \to L : Q \mapsto \inf\{f(x) \mid x \in Q\}, \bot \mapsto \bot\)
  — 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**

- \( \sigma, \tau, \ldots ::= \text{int} | \text{unit} | U\sigma | \sigma \times \tau | V\tau \)
- \( \sigma, \tau, \ldots ::= F\sigma | \sigma \to \tau \)

**Continuous (coherent) dcpo**

- \( U \) converts from bc-domains to continuous coherent dcpo: \( [U\sigma] = [\pi] \)

**bc-domains/continuous lattices**

- \( F \) converts from continuous coherent dcpo to bc-domains: \( [F\sigma] = \bot [\sigma] \)
**U AND F**

- \(\sigma, \tau, \ldots ::= \text{int} \mid \text{unit} \mid U\sigma \mid \sigma \times \tau \mid V\tau\)
- \(\sigma, \tau, \ldots ::= F\sigma \mid \sigma \rightarrow \tau\)

- **U** converts from bc-domains to continuous coherent dcpos: \([U\sigma] = [\sigma]\)

- **F** converts from continuous coherent dcpos to bc-domains: \([F\sigma] = Q_\bot [\sigma]\)

### choice
- \(\text{abort}_{F\sigma}\)
- \(M \triangleleft N\)
- \(\text{produce} M \quad (\sigma \rightarrow F\sigma)\)

### monad
- \(M \text{ to } x_\sigma \text{ in } N\)
## U and F

- \( \sigma, \tau, \ldots ::= \text{int} \mid \text{unit} \mid U\sigma \mid \sigma \times \tau \mid \nu \tau \)
- \( \sigma, \tau, \ldots ::= F\sigma \mid \sigma \rightarrow \tau \)

- **U** converts from bc-domains to continuous coherent dcpos: \( [U\sigma] = [\sigma] \)

- **F** converts from continuous coherent dcpos to bc-domains: \( [F\sigma] = Q_\bot [\sigma] \)

### bc-domains/continuous lattices

- \( M, N, \ldots ::= \ldots \)
  - \( \text{abort}_{F\sigma} \)
  - \( M \odot N \)
  - \( \text{produce} \ M \ (\sigma \rightarrow F\sigma) \)
  - \( M \text{ to } x_\sigma \text{ in } N \)

### continuous (coherent) dcpos

- \( [\text{abort}_{F\sigma}] = \emptyset \)
- \( [M \odot N] = [M] \land [N] \)
- \( [\text{produce} \ M] = \eta([M]) \)
- \( [M \text{ to } x_\sigma \text{ in } N] = \left( V \mapsto [N][x_\sigma := V]^* [M] \right) \)
U AND F

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

- $\sigma, \tau, \ldots ::= F\sigma \mid \sigma \to \tau$

- **U** converts from bc-domains to continuous coherent dcpos: $[U\sigma] = [\sigma]$

- **F** converts from continuous coherent dcpos to bc-domains: $[F\sigma] = Q_\bot [\sigma]$

- $M, N, \ldots ::= \ldots$
  - $\text{abort}_{F\sigma}$
  - $M \land N$
  - $\text{produce } M \ (\sigma \to F\sigma)$
  - $M \ \text{to } x_\sigma \ \text{in } N$

- $[\text{abort}_{F\sigma}] = \emptyset$
  - $[M \land N] = [M] \land [N]$
  - $[\text{produce } M] = \eta([M])$
  - $[M \ \text{to } x_\sigma \ \text{in } N] =$
    $$ (V \mapsto [N][x_\sigma:=V])^* ([M]) $$

- $(\text{produce } M) \ \text{to } x_\sigma \ \text{in } N \to N[x_\sigma:=M] + \text{etc.}$
PCF, probabilistic choice, and the trouble with $\forall$

- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction
- PCF, probabilistic choice, and the trouble with \( \nu \)
- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction
A Krivine machine for deterministic operations, working on configurations $C \cdot M$
OPERATIONAL SEMANTICS

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

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

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

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

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

- **Prop (adequacy).**
  For every $M : \text{FVunit}$,
  - $\llbracket M \rrbracket = \bot$ and $\Pr(M \downarrow) = 0$, or
  - $\llbracket M \rrbracket = \emptyset$ and $\Pr(M \downarrow) = 1$, or else
  - $\Pr(M \downarrow) = \min \{ \forall (\{ \top \}) \mid \forall \in \llbracket M \rrbracket \}$
ADEQUACY

- **Prop (adequacy).**
  For every $M : FVunit$,
  - $\llbracket M \rrbracket = \bot$ and $Pr(M \downarrow) = 0$, or
  - $\llbracket M \rrbracket = \emptyset$ and $Pr(M \downarrow) = 1$, or else
  - $Pr(M \downarrow) = \min \{ \forall \{ \top \} \mid \forall \in \llbracket M \rrbracket \}$

- I.e., $Pr(M \downarrow) = h^*(\llbracket M \rrbracket)$
  where $h(\forall) = \forall(\{ \top \})$

\[\begin{align*}
[x_\tau] \rho &= \rho(x_\tau) \\
[\lambda x_\tau. M] \rho &= V \in [x] \mapsto [M] \rho(x_\tau \mapsto V) \quad [MN] \rho = [M] \rho[N] \rho \\
[\text{produce } M] \rho &= \eta^0([M] \rho) \\
[M \to x, \text{in } N] \rho &= (V \in [x] \mapsto [N] \rho(x_\tau \mapsto V))^*([M] \rho) \\
[\text{thunk } M] \rho &= [M] \rho \\
[\text{force } M] \rho &= [M] \rho \\
[\text{succ } M] \rho &= \{ n + 1 \quad \text{if } n = [M] \rho \neq \bot \\
&\quad \bot \quad \text{otherwise} \} \\
[\text{pred } M] \rho &= \{ n - 1 \quad \text{if } n = [M] \rho \neq \bot \\
&\quad \bot \quad \text{otherwise} \} \\
[\text{ifz } M \ N \ P] \rho &= \{ [N] \rho \quad \text{if } [M] \rho = 0 \\
&\quad [P] \rho \quad \text{if } [M] \rho \neq 0, \bot \\
&\quad \bot \quad \text{otherwise} \} \\
[M; N] \rho &= \{ [N] \rho \quad \text{if } [M] \rho = \top \\
&\quad \bot \quad \text{otherwise} \} \\
[\text{force } M; N] \rho &= m, [\tau_\gamma M] \rho = n \text{ where } [M] \rho = (m, n) \\
([M; N]) \rho &= ([M]) \rho, ([N]) \rho \\
[\text{ret } M] \rho &= \delta(x_\tau)_\rho \\
[\text{do } x_\tau \leftarrow M; N] \rho &= (V \in [x] \mapsto [N] \rho(x_\tau \mapsto V))^!(([M] \rho) \\
[M \oplus N] \rho &= \frac{1}{2}([M] \rho \uplus [N] \rho) \\
[M \otimes N] \rho &= [M] \rho \land [N] \rho \\
[M \oplus N] \rho &= [M] \rho \land [N] \rho \\
[\text{abort}_p] \rho &= \emptyset \\
[\text{rec } x_\tau. M] \rho &= \text{ifp}(V \in [x] \mapsto [M] \rho(x_\tau \mapsto V))
\end{align*}\]
ADEQUACY

- **Prop (adequacy).**
  For every $M : \text{FVunit}$,
  — $⟦M⟧ = \perp$ and $\Pr(M \downarrow) = 0$, or
  — $⟦M⟧ = \emptyset$ and $\Pr(M \downarrow) = 1$, or else
  — $\Pr(M \downarrow) = \min \{ \forall (\{ \top \}) \mid \forall \in ⟦M⟧ \}$
  
  - I.e., $\Pr(M \downarrow) = h^*(⟦M⟧)$
    
    where $h(V) = V(\{ \top \})$

- **Proof:** by suitable logical relations.

---

\[
[x_x] \rho = \rho(x_x) \\
[λx_x.M] \rho = V ∈ [x_x] → [M] (ρ(x_x → V)) \\
[M N] \rho = [M] \rho([N] \rho)
\]

\[
\text{produce } M \rho = ρ^0([M] ρ) \\
\text{to } x, \text{in } N \rho = (V ∈ [x] → [N] \rho(x → V))^∗([M] \rho) \\
\text{think } M \rho = [M] \rho \\
\text{force } M \rho = [M] \rho
\]

\[
\text{succ } M \rho = \begin{cases} 
  n + 1 & \text{if } n = [M] \rho \neq \perp \\
  \perp & \text{otherwise}
\end{cases}
\]

\[
\text{pred } M \rho = \begin{cases} 
  n - 1 & \text{if } n = [M] \rho \neq \perp \\
  \perp & \text{otherwise}
\end{cases}
\]

\[
\text{if } M N P \rho = \begin{cases} 
  [N] \rho & \text{if } [M] \rho = 0 \\
  [P] \rho & \text{if } [M] \rho \neq 0, \perp
\end{cases}
\]

\[
[M ; N ] \rho = \begin{cases} 
  [M] \rho & \text{if } [M] \rho = \top \\
  \perp & \text{otherwise}
\end{cases}
\]

\[
[M ; N ] \rho = m, \text{if } M \rho = m, n \text{ where } [M] \rho = (m, n)
\]

\[
\text{if } M ; N \rho = [M] \rho, [N] \rho
\]

\[
\text{ret } M \rho = σ_{[x_x] ρ}
\]

\[
\text{do } x_x \leftarrow M ; N \rho = (V ∈ [x] → [N] \rho(x → V))^∗([M] \rho)
\]

\[
[M ⊕ N ] \rho = \frac{1}{2}([M] \rho + [N] \rho)
\]

\[
[M ⊓ N ] \rho = [M] \rho ∧ [N] \rho
\]

\[
[M ⊓ N ] \rho = [M] \rho ∧ [N] \rho
\]

\[
[M; N ] \rho = [M] \rho ∧ [N] \rho
\]

\[
\text{rec } x_x.M \rho = \text{if } V ∈ [x] → [M] \rho(x → V)
\]
NOTE

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

MORE POSITIVELY:

- Look for a category of continuous dcpoś that is
  - Cartesian-closed
  - closed under $\forall$

(jung, tix 1991)
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**

MORE POSITIVELY:

- Look for a category of continuous dcpos that is **Cartesian-closed**
- closed under \( \land \)
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 $M \preceq N$ iff for every context $C$ of output type $\text{FVunit}$,
$\Pr(C . M \downarrow) \leq \Pr(C . N \downarrow)$
The Contextual Preorder

- Let $M \preceq N$ iff for every context $C$ of output type $\text{FVunit}$,
  \[ \Pr(C \cdot M \downarrow) \leq \Pr(C \cdot N \downarrow) \]

- $M \preceq N$ iff for every context $C$ of output type $\text{FVunit}$,
  \[ h^*(\llbracket C[M] \rrbracket) \leq h^*(\llbracket C[N] \rrbracket) \] (adequacy)
THE CONTEXTUAL PREORDER

- Let $M \preceq N$ iff for every context $C$ of output type $\text{FVunit}$,
  \[ \Pr(C \cdot M \downarrow) \leq \Pr(C \cdot N \downarrow) \]

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

- **Corollary.** If $\llbracket M \rrbracket \leq \llbracket N \rrbracket$ then $M \preceq N$. 
THE CONTEXTUAL PREORDER

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

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

- **Corollary.** If $\llbracket M \rrbracket \leq \llbracket N \rrbracket$ then $M \leq N$.

- **Proof.** $\llbracket C[M] \rrbracket = \llbracket C \rrbracket (\llbracket M \rrbracket) \leq \llbracket C \rrbracket (\llbracket N \rrbracket) = \llbracket C[N] \rrbracket$
  since $\llbracket C \rrbracket (= \llbracket \lambda x . C[x] \rrbracket)$ is Scott-continuous hence monotonic. Then apply $h^*$, which is monotonic as well. □
THE APPLICATIVE PREORDER

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

- Let $M \leq^{app} N$ iff for every term $P : \tau \rightarrow FV_{unit}$,
  $Pr(PM\downarrow) \leq Pr(PN\downarrow)$
THE APPLICATIVE PREORDER

- Let $M \preceq N$ iff for every context $C$ of output type $\mathbf{FV}_{\text{unit}}$, 
  $\Pr(C . M \downarrow) \leq \Pr(C . N \downarrow)$

- Let $M \preceq_{\text{app}} N$ iff for every term $P : \tau \to \mathbf{FV}_{\text{unit}}$, 
  $\Pr(PM \downarrow) \leq \Pr(PN \downarrow)$

- **Proposition** (« Milner’s context lemma » in PCF): 
  $M \preceq N$ iff $M \preceq_{\text{app}} N$. 
THE APPLICATIVE PREORDER

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

- Let $M \leq_{app} N$ iff for every term $P : \tau \to \text{FVunit}$, $\Pr(PM\downarrow) \leq \Pr(PN\downarrow)$

- Proposition (« Milner’s context lemma » in PCF): $M \leq N$ iff $M \leq_{app} N$.

- 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): $\llbracket M \rrbracket \leq \llbracket N \rrbracket$ iff $M \leq N$.

- **Wrong.** (As in (Plotkin 77).) Using another logical relation, for every $P : \text{int} \to \text{int} \to \text{Fint}$, if $\llbracket P \rrbracket(\bot)(0) = \llbracket P \rrbracket(0)(\bot) = \{0\}$ then $\llbracket P \rrbracket(\bot)(\bot) = \{0\}$.
**FULL ABSTRACTION?**

- **Conjecture (full abstraction):** \([M] \leq [N] \text{ iff } M \preceq N\).

- **Wrong.** (As in (Plotkin 77).) Using another logical relation, for every \(P : \text{int} \to \text{int} \to \text{Fint}\), 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 \(M \preceq N\).
FULL ABSTRACTION?

- **Conjecture (full abstraction):** \([M] \leq [N] \text{ iff } M \preceq N.\)

- **Wrong.** (As in (Plotkin 77).) Using another logical relation, for every \(P : \text{int} \rightarrow \text{int} \rightarrow \text{Fint, if } [P](\bot)(0)=[P](0)(\bot)=[0] \text{ then } [P](\bot)(\bot)=[0]\)

- 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 \(M \preceq N.\)

- But \([M]\npreceq[N] \text{ since } [M](\text{por})=\top, [N](\text{por})=\bot.\)
FULL ABSTRACTION?

- Add parallel if \textbf{pifz}: \[ \begin{align*}
\llbracket \text{pifz } M N P \rrbracket &= \llbracket N \rrbracket \text{ if } \llbracket M \rrbracket = 0 \\
\llbracket P \rrbracket &\text{ if } \llbracket M \rrbracket \neq 0, \bot \\
\llbracket N \rrbracket \land \llbracket P \rrbracket \text{ (not } \bot !\text{) if } \llbracket M \rrbracket = \bot
\end{align*} \]
FULL ABSTRACTION?

- Add parallel if $\textbf{pifz}$:
  $$\llbracket \textbf{pifz } M \ N \ P \rrbracket = \llbracket N \rrbracket \text{ if } \llbracket M \rrbracket = 0$$
  $$\llbracket P \rrbracket \text{ if } \llbracket M \rrbracket \neq 0, \bot$$
  $$\llbracket N \rrbracket \land \llbracket P \rrbracket \text{ (not } \bot!) \text{ if } \llbracket M \rrbracket = \bot$$

- **Conjecture (full abstraction):** $\llbracket M \rrbracket \leq \llbracket N \rrbracket \text{ iff } M \preceq N$. 
FULL ABSTRACTION?

- Add parallel if \( pifz \):
  \[
  [pifz\ M\ N\ P] = [N] \text{ if } [M] = 0
  \]
  \[
  [P] \text{ if } [M] \neq 0, \bot
  \]
  \[
  [N] \land [P] \text{ (not } \bot!) \text{ if } [M] = \bot
  \]

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

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

- Let $M = \lambda P . P(\Omega \oplus \text{ret} *)$
  
  $N = \lambda P . P(\Omega) \oplus \text{ret} *$

  Then $M \preceq N$, even with $\text{pifz}$

  (modulo some missing $\text{force, produce, etc.}$)
STATISTICAL TERMINATION TESTERS

- Let $M = \lambda P . P(\Omega \oplus \text{ret} \ast)$
  $N = \lambda P . P(\Omega) \oplus \text{ret} \ast$

  Then $M \preceq N$, even with pifz

- But $\llbracket M \rrbracket \not\equiv \llbracket N \rrbracket$ since $\llbracket M \rrbracket([>b]) = \uparrow \delta_{\top}$, $\llbracket N \rrbracket([>b]) = \bot$ for all $b < 1/2$,
  where $[>b] : \llbracket \text{Vunit} \rrbracket \rightarrow \llbracket \text{FVunit} \rrbracket$
    maps every $v$ to ‘termination’ ($\uparrow \delta_{\top}$) if $v(\{\top\}) > b$,
    to $\bot$ otherwise.
Let $M = \lambda P . P(\Omega \oplus \text{ret} * )$

$N = \lambda P . P(\Omega) \oplus \text{ret} *$  

Then $M \preceq N$, even with petit.

But $[M] \not= [N]$ since $[M]([>b]) = \uparrow \delta_{\top}$, $[N]([>b]) = \bot$ for all $b < 1/2$, where $[>b] : [\text{Vunit}] \rightarrow [\text{FVunit}]$

maps every $\nu$ to ‘termination’ ($\uparrow \delta_{\top}$) if $\nu(\{\top\}) > b$,

to $\bot$ otherwise.

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

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

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

- **Theorem (full abstraction):** with $\text{pifz}$ and $\bigcirc_{>b}$,
  $\llbracket M \rrbracket \leq \llbracket N \rrbracket$ iff $M \leq N$. 
FULL ABSTRACTION

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

- \textbf{Theorem (full abstraction):} with \texttt{pifz} and \bigcirc_{>b},\\ \[ \llbracket M \rrbracket \leq \llbracket N \rrbracket \text{ iff } M \preceq N. \]

- And now a glimpse of the argument…
FULL ABSTRACTION: PROOF

- We assume $\llbracket M \rrbracket \not\approx \llbracket N \rrbracket$, and we wish to prove not $(M \leq N)$. 
FULL ABSTRACTION: PROOF

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

- There is a subbasic open set $U / [M] \in U, [N] \not\in U$
  (because $\leq$ is the specialization ordering of the Scott topology)
FULL ABSTRACTION: PROOF

- We assume $[M] \not\approx [N]$, 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 $\uparrow \delta_T$ and each $x \notin U$ to $\bot$)
  then $[PM] = \uparrow \delta_T, [PN] = \bot$, so not $(M \leq_{\text{app}} N)$.
  Conclude by Milner’s context lemma.
FULL ABSTRACTION: PROOF

- We assume $\llbracket M \rrbracket \not\subseteq \llbracket N \rrbracket$, and we wish to prove not $(M \leq N)$.

- There is a subbasic open set $U / \llbracket M \rrbracket \in U, \llbracket N \rrbracket \not\in U$
  (because $\leq$ is the specialization ordering of the Scott topology)

- If $U$ is **definable** by a term $P$
  (i.e., $\llbracket P \rrbracket$ maps each $x \in U$ to $\uparrow \delta_T$ and each $x \not\in U$ to $\perp$)
  then $\llbracket PM \rrbracket = \uparrow \delta_T, \llbracket PN \rrbracket = \perp$, so not $(M \leq_{\text{app}} N)$.
  Conclude by Milner’s context lemma.

- Challenge: show that each $\llbracket T \rrbracket$ has a subbase of definable opens.
PROBABILITY TYPES

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

- $\mathcal{V}\sigma$: 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

- \([V\sigma]\): 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 \(v \in [U>b]\): iff \(v(U)>b\), implemented through
  \(\bigcirc>_b (\text{do } x_\sigma \leftarrow v; \langle\text{test } x_\sigma \in U\rangle)\)
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
  \['\circlearrowright>b \ (\textbf{do} \ x_\sigma \leftarrow \nu; \langle\text{test } x_\sigma \in U\rangle)'\]

- 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 $\textbf{ret}$ and $\oplus$
F TYPES

- \([F_\sigma] \): subbasic opens \( \square U \)
  where \( U \) is a basic open of \( [\sigma] \)
TYPES

- $\llbracket F \sigma \rrbracket$: subbasic opens $\Box U$
  where $U$ is a basic open of $\llbracket \sigma \rrbracket$

- I.e. (requires $\llbracket \sigma \rrbracket$ continuous!) Scott=upper Vietoris topology
F TYPES

- \( [F\sigma]: \) subbasic opens \( \Box U \)
  where \( U \) is a basic open of \( [\sigma] \)

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

- test \( Q \in \Box U \): if \( Q \subseteq U \), implemented through
  \( 'Q \texttt{ to } x_\sigma \texttt{ in } \langle \text{test } x_\sigma \in U \rangle' \)
**F TYPES**

- $\langle F \sigma \rangle$: subbasic opens $\Box U$
  
  where $U$ is a basic open of $\langle \sigma \rangle$

- I.e. (requires $\langle \sigma \rangle$ **continuous**)! Scott=upper Vietorisoris topology

- $test \ Q \in \Box U$: iff $Q \subseteq U$, implemented through
  
  \[ \langle Q \ \text{to} \ x_\sigma \ \text{in} \ \langle \text{test} \ x_\sigma \in U \rangle \rangle \]

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

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

- I.e. (uses $\llbracket \sigma \rrbracket$ bc-domain!) Scott=top. of pointwise convergence
FUNCTION TYPES (1/2)

- $[\sigma \rightarrow \tau]$: 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 $\\llbracket \sigma \rrbracket$, $V$ is a subbasic open of $\\llbracket \tau \rrbracket$

- I.e. (uses $\\llbracket \sigma \rrbracket$ 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 $\\llbracket \sigma \rrbracket$ now. We have them for $V$ and $F$ types, and the difficult case is for $\to$. 
FUNCTION TYPES (2/2)

- Basis for \([\sigma \rightarrow \tau]\): step functions \(\bigvee_{1 \leq i \leq n} U_i \downarrow y_i\)
  mapping each \(x\) to \(\bigvee \{y_i \mid x \in U_i\}\)...
  but that sup is hard to implement — we only have infs.
FUNCTION TYPES (2/2)

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

- Trick: for each subset $I$ of $\{1, \ldots, n\}$,
  let $U_I = \text{intersection of } U_i, i \in I$,
  $y_I = \text{sup of } y_i, i \in I$.
  Then $\bigvee_{1 \leq i \leq n} U_i \searrow y_i (x) = \inf \{ y_I \mid I \supseteq I_x \}$ where $I_x = \{ i \mid x \in U_i \}$
FUNCTION TYPES (2/2)

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

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

- Additional difficulties (need for $\text{pifz}$ notably)… omitted. Done! ☐
SUMMARY

- Circumventing the trouble with \( \mathbb{V} \) 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?