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

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

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

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

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

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

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

- A **denotational** semantics: 
  \( \llbracket M \rrbracket \)

- **Adequacy:**
  for every ground \( M : \text{int} \), 
  \( \llbracket M \rrbracket = n \) iff \( M \rightarrow^* n \)
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 \leq N \iff \text{for every context } C : \text{int}, \]
  \[ C[M] \rightarrow^* n \Rightarrow C[N] \rightarrow^* n \]

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

- Converse is **full abstraction.**
  Fails for PCF, works for PCF+por
Every type $\tau$ interpreted as a dcpo $\llbracket \tau \rrbracket$.
Every type $\tau$ interpreted as a dcpo $\llbracket \tau \rrbracket$

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

A directed family $D$. In a dcpo, every directed family $D$ has a supremum $\sqcup D$
Every type $\tau$ interpreted as a dcpo $\llbracket \tau \rrbracket$

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

$\llbracket \sigma \rightarrow \tau \rrbracket = \llbracket [\sigma] \rightarrow [\tau] \rrbracket$, dcpo of Scott-continuous maps $[\sigma] \rightarrow [\tau]$ (monotonic + preserves directed sups)

A directed family $D$.
In a dcpo, every directed family $D$ has a supremum $\bigvee D$
THE SEMANTICS OF PCF

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

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

\[ [MN] = [M][N] \]
\[ [\lambda x_\sigma . M] = (V \mapsto [M][x_\sigma := V]) \]

- Meaningful since \( \text{Dcpo} \) is a Cartesian-closed category

\[ \in [\sigma \rightarrow \tau] \quad \in [\sigma] \]

\[ \in [\sigma] \quad \in [\tau] \]
CARTESIAN-CLOSEDNESS

- $[MN] = [M](N)$
- $[\lambda x_\sigma . M] = (V \mapsto [M][x_\sigma := V])$
- 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.
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 $\llbracket T \rrbracket$ is an algebraic bc-domain, making that possible.

Cartesian-closed… good.
CCCS OF CONTINUOUS DCPPOS

- 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, provided they consist of continuous dcpos.
ADDING PROBABILITIES

- **Types**
  \[ \sigma, \tau, \ldots ::= \textbf{int} \mid \sigma \rightarrow \tau \mid \mathbf{V}\tau \]

- **Terms**
  \[ M, N, \ldots ::= \ldots \]
  \[ \mid M \oplus N \]
  \[ \mid \textbf{ret} \ M \]
  \[ \mid \textbf{do} \ x_\sigma \leftarrow M; N \]
Types
\( \sigma, \tau, \ldots ::= \text{int} \mid \sigma \rightarrow \tau \mid \mathbf{V}_\tau \)

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 \( \tau \)
ADDING PROBABILITIES

- **Types**
  \[ \sigma, \tau, \ldots ::= \text{int} \mid \sigma \rightarrow \tau \mid \nu_{\tau} \]

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

- **Monadics**
  - Type of subprobability valuations over \( \tau \) with \( M, N: \nu_{\tau} \), choose between \( M \) and \( N \) with probability \( 1/2 \)
ADDING PROBABILITIES

**Types**
\[ \sigma, \tau, \ldots ::= \text{int} \mid \sigma \rightarrow \tau \mid \Delta \]

**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 \( M, N: \Delta \), choose between \( M \) and \( N \) with probability 1/2

Monadic constructions:
\[ M: \tau \Rightarrow \text{ret} \; M: \Delta \]
\[ M: \Delta \sigma \quad N: \Delta \Rightarrow \text{do } x_\sigma \leftarrow M \; ; \; N : \Delta \]

(Moggi 1991)
THE TROUBLE WITH $\mathbb{V}$

(Jung, Tix 1998)

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

(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:

(Jung, Tix 1998)

- Look for a category of continuous dcpos that is:
  - Cartesian-closed
  - Closed under $\land$
MORE POSITIVELY:

- Look for a category of continuous dcpos that is:
  - Cartesian-closed
  - closed under $\land$

(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
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 dcpo's that is
  - **Cartesian-closed**
  - closed under $\bigvee$
- PCF, probabilistic choice, and the trouble with \( \mathbb{V} \)
- Curing the trouble using call-by-push-value
- Semantics, adequacy, full abstraction
PCF, probabilistic choice, and the trouble with \( \mathbf{V} \)

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

- No such problem with two kinds of types:
  \( \sigma, \tau, \ldots ::= \text{int} \mid \ldots \mid \sigma \times \tau \mid \bigvee \tau \)
  \( \sigma, \tau, \ldots ::= \ldots \mid \sigma \rightarrow \tau \)
CALL-BY-PUSH-VALUE

- No such problem with two kinds of types: 
  \( \sigma, \tau, \ldots ::= \text{int} | \text{unit} | \mathbf{U}\sigma | \sigma \times \tau | \mathbf{V}\tau \)
  
  \( \sigma, \tau, \ldots ::= \mathbf{F}\sigma | \sigma \to \tau \)

- This is the type structure of Paul B. Levy’s **call-by-push-value** (except for the \( \mathbf{V} \) 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, blain.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 senses: both
operational and denotational semantics for these paradigms can be seen
as existing, and 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 pro-
gram, 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 com-
putational 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 unexplored.

(Levy 1999)
CALL-BY-PUSH-VALUE

- No such problem with two kinds of types:
  \[ \sigma, \tau, \ldots ::= \text{int} \mid \text{unit} \mid U\sigma \mid \sigma \times \tau \mid V\tau \]
  \[ \sigma, \tau, \ldots ::= F\sigma \mid \sigma \rightarrow \tau \]

- 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 gb jeswq@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 object 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 unexplained mysteries.

(Levy 1999)
CALL-BY-PUSH-VALUE

- No such problem with two kinds of types:
  \( \sigma, \tau, \ldots ::= \text{int} \mid \text{unit} \mid U\sigma \mid \sigma \times \tau \mid V\tau \)
  \( \sigma, \tau, \ldots ::= F\sigma \mid \sigma \rightarrow \tau \)

- 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 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 see that the lambda-calculus primitives can be understood as push/pop commands for an operator 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 unproved mysterious.

(Levy 1999)
U AND F

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

\[ \alpha, \beta, \ldots ::= \sigma \rightarrow \tau \]

continuous (coherent) dcpos

bc-domains/continuous lattices
\( \sigma, \tau, \ldots ::= \text{int} \mid \text{unit} \mid U\sigma \mid \sigma \times \tau \mid V\tau \)

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

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

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

- continuous (coherent) dcpos

- 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: \( [U\sigma] = [\sigma] \)

- \( M, N, \ldots ::= \ldots \)
  | force \( M \) \( (U\sigma \to \sigma) \)
  | thunk \( M \) \( (\sigma \to U\sigma) \)
**U AND F**

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

**U** converts from bc-domains to continuous coherent dcpos

… semantically the identity: \( \llbracket U\sigma \rrbracket = \llbracket \sigma \rrbracket \)

- \( M, N, \ldots ::= \ldots \)
  - \( | \text{force } M \) \( (U\sigma \to \sigma) \)
  - \( | \text{thunk } M \) \( (\sigma \to U\sigma) \)
  - \( \llbracket \text{force } M \rrbracket = \llbracket M \rrbracket \)
  - \( \llbracket \text{thunk } M \rrbracket = \llbracket M \rrbracket \)
  - \( \text{force thunk } M \to M \)

**continuous (coherent) dcpos**

**bc-domains/continuous lattices**
\( U \) and \( F \)

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

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

- \( F \) converts from continuous coherent dcpos to bc-domains
  ... we take \( \llbracket F\sigma \rrbracket = (\text{lifted}) \) Smyth powerdomain of \( \llbracket \sigma \rrbracket \)
THE SMYTH POWERDOMAIN

- $\mathcal{Q}^X = \{\text{compact saturated subsets of } X\}$, reverse inclusion $\supseteq$
- $\mathcal{Q}^X$ is a continuous complete lattice for every continuous coherent dcpo $X$
- Serves as a model of **demonic non-determinism**.
THE SMYTH POWERDOMAIN

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

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

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

- Technically, we use $\mathbb{Q}_{\bot} X = \mathbb{Q} X$ plus a fresh bottom $\bot$
  … allows $f^*$ to be strict now (needed for adequacy)
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: \([F\sigma] = Q_\bot[\sigma]\)

**continuous (coherent) dcpos**

**continuous complete lattices**
**U AND F**

- **σ, τ, ... ::= int | unit | Uσ | σ × τ | Vτ**
- **σ, τ, ... ::= Fσ | σ → τ**

**continuous (coherent) dcpos**

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

- **F** converts from continuous coherent dcpos to bc-domains: \([Fσ] = Q_⊥[σ]\)

- **M, N, ... ::= ...**
  - \(\text{abort}_Fσ\)
  - \(M ⊗ N\)
  - **produce** \(M\) \((σ → Fσ)\)
  - \(M \text{ to } x_σ \text{ in } N\)

**choice**

**monad**
\section*{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 dcpo: \([U\sigma]=\lbrack \sigma \rbrack\)
- **F** converts from continuous coherent dcpo to bc-domains: \([F\sigma]=Q_\perp \lbrack \sigma \rbrack\)

\[ \begin{align*}
| & \text{abort}_{F\sigma} \\
| & M \triangleleft N \\
| & \text{produce } M \quad (\sigma \to F\sigma) \\
| & M \textbf{ to } x_\sigma \textbf{ in } N \\
\end{align*} \]

- \([\text{abort}_{F\sigma}] = \emptyset\)
- \([M \triangleleft N] = \lbrack M \rbrack \land \lbrack N \rbrack\)
- \([\text{produce } M] = \eta(\lbrack M \rbrack)\)
- \([M \textbf{ to } x_\sigma \textbf{ in } N] = \lbrack M \rbrack[S_\sigma=V] (\lbrack M \rbrack)\)
**U AND F**

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

**continuous (coherent) dcpos**

**continuous complete lattices**

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

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

- **choice**
- **Monad**

- $\llbracket \text{abort}_{F\sigma} \rrbracket = \emptyset$
- $[M \otimes N] = [M] \land [N]$
- $[\text{produce } M] = \eta([M])$
- $[M \text{ to } x_\sigma \text{ in } N] = (V \mapsto [N][x_\sigma:=V])^* ([M])$
- $(\text{produce } M) \text{ to } x_\sigma \text{ in } N \rightarrow N[x_\sigma:=M] + \text{ etc.}$
PCF, probabilistic choice, and the trouble with \( \odot \)

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

Curing the trouble using call-by-push-value

Semantics, adequacy, full abstraction
A Krivine machine for deterministic operations, working on configurations $C . M$
**OPERATIONAL SEMANTICS**

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

- Prob. must-termination judgments $C . M \downarrow a$

(« whichever way you resolve the demonic non-deterministic choices, the probability that $C . M$ terminates is $> a$. »)
A Krivine machine for deterministic operations, working on configurations \( C . M \)

Prob. must-termination judgments \( C . M \downarrow a \)

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

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

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

\[
\begin{align*}
[x] \rho &= \rho(x) \\
[\lambda x. M] \rho &= V \in [x] \rightarrow [M] (\rho(x) \mapsto V) \\
[M N] \rho &= [M] \rho \circ [N] \rho \\
[\text{produce } M] \rho &= \eta^0([M] \rho) \\
[M \text{ to } x, \text{ in } N] \rho &= (V \in [x] \mapsto [N] \rho(x) \mapsto V)’([M] \rho) \\
[\text{think } M] \rho &= [M] \rho \\
[\text{force } M] \rho &= [M] \rho \\
[\bot] \rho &= \top \\
[\top] \rho &= \bot \\
[n + 1] \rho &= n + 1 \quad \text{if } n = [M] \rho \neq \bot \\
[n - 1] \rho &= n - 1 \quad \text{if } n = [M] \rho \neq \bot \\
[n] \rho &= \bot \quad \text{otherwise} \\
[\text{succ } M] \rho &= \begin{cases} 
  n + 1 & \text{if } n = [M] \rho \neq \bot \\
  \bot & \text{otherwise}
\end{cases} \\
[\text{pred } M] \rho &= \begin{cases} 
  n - 1 & \text{if } n = [M] \rho \neq \bot \\
  \bot & \text{otherwise}
\end{cases} \\
[\text{if } M \text{ N P}] \rho &= \begin{cases} 
  [N] \rho & \text{if } [M] \rho = 0 \\
  [P] \rho & \text{if } [M] \rho = 1 \\
  [M] \rho & \text{otherwise}
\end{cases} \\
[M : N] \rho &= \begin{cases} 
  [M] \rho & \text{if } [M] \rho = \top \\
  [N] \rho & \text{if } [M] \rho = \bot \\
  \bot & \text{otherwise}
\end{cases} \\
[x₁, M] \rho &= m, [x₂, M] \rho = n \text{ where } [M] \rho = (m, n) \\
([M ; N]) \rho &= ([M] \rho, [N] \rho) \\
[\text{rec } x, M] \rho &= \lambda p (V \in [x] \mapsto [M] \rho(x) \mapsto V) \\
[M \oplus N] \rho &= \frac{1}{2} ([M] \rho \lor [N] \rho) \\
[M \ominus N] \rho &= [M] \rho \land [N] \rho \\
[\text{abort}_p] \rho &= \emptyset \\
[r] \rho &= \text{fp}(V \in [x] \mapsto [M] \rho(x) \mapsto V)
\end{align*}
\]
ADEQUACY

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

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

$\begin{align*}
[x_\ast]_\rho &= \rho(x_\ast) \\
[\lambda x_\ast. M]_\rho &= V \in [x] \rightarrow \llbracket M \rrbracket (\rho(x_\ast \mapsto V)) \\
[M N]_\rho &= \llbracket M \rrbracket \rho (\llbracket N \rrbracket \rho)
\end{align*}$

$\begin{align*}
[\text{produce } M]_\rho &= \eta \circ (M_\rho) \\
[M \text{ to } x_\ast, \text{ in } N]_\rho &= (V \in [x] \rightarrow \llbracket N \rrbracket (\rho(x_\ast \mapsto V))' (M_\rho) \\
[\text{thunk } M]_\rho &= \llbracket M \rrbracket_\rho \\
[\text{force } M]_\rho &= \llbracket M \rrbracket_\rho \\
[\text{if } M N P]_\rho &= \texttt{if } (\llbracket M \rrbracket_\rho = 0) \texttt{ then } \llbracket N \rrbracket_\rho \texttt{ else } \llbracket P \rrbracket_\rho \\
\&[\text{if } M N P]_\rho &= \texttt{if } (\llbracket M \rrbracket_\rho = 0) \texttt{ then } \perp \texttt{ else } \perp \\
[M : N]_\rho &= \texttt{if } (\llbracket M \rrbracket_\rho = \top) \texttt{ then } \perp \texttt{ else } \perp \\
[x_\ast, M]_\rho &= m, [x_\ast, M]_\rho = n \texttt{ where } (M_\rho, \llbracket N \rrbracket_\rho) \\
[M N]_\rho &= (M_\rho \cdot \llbracket N \rrbracket_\rho) \\
[\text{if } M N P]_\rho &= (\texttt{if } M_\rho \texttt{ then } N_\rho \texttt{ else } P_\rho) \\
[\text{reduce } M]_\rho &= \texttt{if } N_\rho \texttt{ then } \texttt{abort}_\rho \texttt{ else } \emptyset \\
[\text{do } x_\ast \leftarrow M : N]_\rho &= (V \in [x] \rightarrow \llbracket M \rrbracket (\rho(x_\ast \mapsto V))) (\llbracket N \rrbracket_\rho) \\
[M \odot N]_\rho &= \frac{1}{2}((M_\rho \cdot N_\rho) + (M_\rho \cdot N_\rho)) \\
[M \odot N]_\rho &= \texttt{if } (\llbracket M \rrbracket_\rho \cdot \llbracket N \rrbracket_\rho) \texttt{ then } \texttt{abort}_\rho \texttt{ else } \emptyset \\
[\text{reduce } M]_\rho &= \texttt{if } (\llbracket M \rrbracket (\rho(x_\ast \mapsto V))) \\
\end{align*}$
ADEQUACY

- **Prop (adequacy).**
  
  For every $M : \text{FVunit}$,
  
  - $[M] = \bot$ 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., $\Pr(M \downarrow) = h^*(\lbrack M \rbrack)$
  
  where $h(\forall) = \forall(\{\top\})$

- **Proof:** by suitable logical relations.
None of that yet requires CCCs of \textbf{continuous} (or algebraic) domains

MORE POSITIVELY:
- Look for a category of continuous dcpo\textsuperscript{s} that is \textbf{Cartesian-closed}
- \textbf{closed under }\forall
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**
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 $FV\text{unit}$,
  $\Pr(C \cdot M\downarrow) \leq \Pr(C \cdot N\downarrow)$
THE CONTEXTUAL PREORDER

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

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

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

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

- $M \preceq N$ 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 $M \preceq N$.

- **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. □
FULL ABSTRACTION?

- Conjecture (full abstraction): $[M] \leq [N]$ iff $M \preceq N$. 
\textbf{FULL ABSTRACTION?}

- **Conjecture (full abstraction):** $\sem{M} \leq \sem{N}$ iff $M \preceq N$.

- **Wrong.**
  - missing parallel if $\text{(pifz)}$, as in (Plotkin77)
  - even with $\text{pifz}$, missing \textit{statistical termination testers} $\bigcirc_{>b}$, as in (GL15):
    $\bigcirc_{>b} M$ terminates if $M$ terminates with prob. $>b$, otherwise does not terminate.
FULL ABSTRACTION

- Adding \textbf{pifz} + $\bigcirc_{>b}$,

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

- For the argument, see the paper. Uses the deep structure of continuous coherent dcpos and continuous complete lattices. Core: theorems on (effective) coincidence of topologies.
SUMMARY

- Circumventing the trouble with $\forall$ 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?