Full Abstraction

40 years of research

The Gandy Colloquium

Christine Tasson (tasson@irif.fr)

February 2020

Institut de Recherche en Informatique Fondamentale

Road map

Computability at higher types

Church's Thesis (1930) $(\mathtt{nat} \to \mathtt{nat})$

A partial function on natural numbers with *first order* type is 'computable' or 'effectively decidable',
if and only if it is computable by a Turing machine,
if and only if it belongs to the general recursive functions (Gödel),
if and only if it is definable in λ-calculus (Church).

Kleene's Problem (1960-1980) $(\sigma, \tau ::= \mathtt{nat} \mid \sigma \to \tau)$

Characterize all partial functions with *higher* types that are 'computable' or 'effectively decidable'.

- Realizability, used to extract concrete programs from math. proofs
- An effective version of higher-type computability in term of dialogue
- Gandy and Pani explored this direction introducing concepts such as no dangling questions, visibility with (counter-)examples, in a framework tracing the route to *game theory*

Programming Computable Functions (PCF) (Scott 69-Plotkin 77)

Another approach to the higher type functions computability characterization

Syntax

$(\sigma, \tau ::= \mathtt{nat} \mid \sigma \to \tau)$

$$M, N, P := \underbrace{x \mid \lambda x\, M \mid (M)\, N}_{\lambda\text{-calculus}} \mid \underbrace{0 \mid \mathtt{succ}\, M}_{\text{Arithmetics}} \mid \underbrace{\mathtt{if}\, M = 0\, \mathtt{then}\, N\, \mathtt{else}\, P}_{\text{Conditionnal}} \mid \underbrace{\mathtt{fix}\, M}_{\text{Recursion}}$$

Operational Semantics (Call-by-Name)

(selected rules)

$$\frac{}{(\lambda x\, M)N \to M\,[N/x]} \qquad \frac{}{\mathtt{fix}\ M \to (M)(\mathtt{fix} M)} \qquad \frac{M \to M'}{(M)N \to (M')N}$$

Denotational Semantics

Complete Partial Orders with continuous functions

- `nat` interpreted as flat domains with partiality with arithmetic
- `fix` induces partiality

Results relating operational and denotational semantics

Soundeness: *Semantics is an invariant of computation*

$$\text{If } M \to M', \text{ then } [\![M]\!] = [\![M']\!].$$

Computational Adequacy: *A perfect match at observational types*

For all closed term M of type `nat`, $M \to^* n$ if and only if $[\![M]\!] = n$.

Observational equivalence:

$$M \simeq_{\mathrm{o}bs} N \text{ if } \forall C \; C[M] \to^* n \iff C[N] \to^* n$$

Full abstraction: *Correspondence at any type*

$$M \simeq_{\mathrm{o}bs} N \text{ if and only if } [\![M]\!] = [\![N]\!]$$

Milner proved that there is a (unique) Fully Abstract standard model of PCF based on the syntax.

Definability criteria for Full Abstraction

A *definable* element of the model is the semantics of a program of PCF.

Separable definability: if $f \neq g$ definable of type τ, then there is h definable of type $\tau \to \mathtt{nat}$ such that $h \circ f \neq h \circ g$.

Propostion Separable definability implies Full Abstraction

Proof. By contradiction, if $[\![M]\!] \neq [\![N]\!]$, then there is P such that

$$[\![(P)M]\!] = [\![P]\!] \circ [\![M]\!] \neq [\![P]\!] \circ [\![N]\!] = [\![(P)N]\!]$$

Conclude by adequacy, as $(P)M$ is of type $\mathtt{nat}$.

Milner characterisation: Full Abstraction results from

- Compact definability (all compact elements at all types are definable)
- Extensionality (elements of the model at function types are functions)

Quest for a syntax free description of the FA model

Examples:

- Scott Model is not FA for PCF, but for PCF+POR
  (Plotkin)
- Sequential algorithm is not FA for PCF, but for PCF+catch
  (Curien-Berry)
- Strongly Stable model is extensional but not compact definable
  (Bucciarelli-Ehrhard)

Game semantics are compact definable but not extensional

- Games and history free strategies (Abramsky-Jagadesen-Malacaria)
- Games and innocent, well-bracketed strategies (Hyland-Ong and Nickau)

The extensional quotient is a Fully Abstract model

at any type τ, $f \simeq g$ whenever $h \circ f = h \circ g$ for all $h : \tau \to \mathtt{nat}$.

Gandy's inspiration

Game semantics took is root in Kleene and Gandy's work on higher-order computability where game ingredients appeared (dialogue, views, non dangling questions).

It resulted in a wide range of Full Abstraction results for various languages and effects and new insight in denotational semantics.

Open questions

- Does PCF answers the question of higher type computability ?
  (There is no hope for an effective characterization of the FA model of PCF as the observational order is undecidable for finitary PCF as shown by Loader.)
- What is the Church thesis for Higher Type ?

Probabilistic Programming

Bayesian Inference

Probabilistic programming languages

Every programmer can perform data analysis by describing models as programs and key operations (inference computations are delegated to compiler).

BUGS (Spiegelhalter et al. 1995), BLOG (Milch et al. 2005), Church (Goodman et al. 2008), WebPPL (Goodman et al. 2014), Venture (Mansinghka et al. 2014), Anglican (Wood et al. 2015), Stan (Stan Development Team 2014), Hakaru (Narayanan et al., 2016) BayesDB (Mansinghka et al. 2017), Edward (Tran et al.Tran et al. 2017), Birch (Murray et al. 2018), Turing (Ge et al. 2018), Gen (Cusumano-Towner et al. 2019), Pyro (Bingham et al. 2019), . . .

Sampling

Idea: How to model probability distributions by programs

```
def plinko(n):
  if (n==0):
    return 0
  else:
    if coin():
      return plinko(n-1)+1
    else:
      return plinko(n-1)-1
```

By Matemateca (IME USP)

Sampling

Idea: How to model probability distributions by programs

```
def plinko(n):
  if (n==0):
    return 0
  else:
    if coin():
      return plinko(n-1)+1
    else:
      return plinko(n-1)-1
```

```
sample(plinko(4))
> 2
```

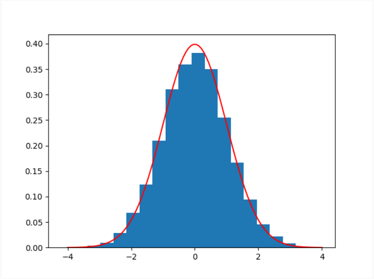
Sampling

Idea: How to model probability distributions by programs

```
def plinko(n):
  if (n==0):
    return 0
  else:
    if coin():
      return plinko(n-1)+1
    else:
      return plinko(n-1)-1
```

```
sample(plinko(4))
> 2

nSample(plinko(4), 1000)
plot(gaussian(0,1))
```



What is Bayesian Inference

Gender Bias (Laplace): Paris, from 1745 to 1770
$f0 = 241\,945$ females out of $B0 = 493\,472$ births (49%).

What is Bayesian Inference

Gender Bias (Laplace): Paris, from 1745 to 1770
$f0 = 241\,945$ females out of $B0 = 493\,472$ births (49%).

What is the probability to be born female ?

- female births are independent and follow the same law with bias θ
- the probability to get f females out of B births is

$$P(f|\theta, B) = \binom{B}{f} \theta^f (1-\theta)^{B-f}$$

Novelty: the bias θ to be born female follows a probabilistic distribution.

What is Bayesian Inference

Gender Bias (Laplace): Paris, from 1745 to 1770
$f0 = 241\,945$ females out of $B0 = 493\,472$ births (49%).

What is the probability to be born female ?

- female births are independent and follow the same law with bias θ
- the probability to get f females out of B births is

$$P(f|\theta, B) = \binom{B}{f}\theta^f(1-\theta)^{B-f}$$

Novelty: the bias θ to be born female follows a probabilistic distribution.

Inference paradigm: what is the law of θ conditioned on f and B?

- Sample θ from a postulated distribution π (prior)
- Simulate data f from the outcome θ (likelihood)
- Infer the distribution of θ (posterior) by **Bayes Law**

$$P(\theta \mid f, B) = \frac{P(f \mid \theta, B)\ \pi(\theta)}{\int_\theta P(f \mid \theta, B)\ \pi(\theta)} = \alpha \cdot P(f \mid \theta, B)\ \pi(\theta)$$

Conditioning and inference

```
# model
def fBirth(theta, B):
    if (B == 0):
        return 0
    else:
        f = flip(theta)
        return f + fBirth(theta, B-1)

# parameter (prior)
theta = uniform(0,1)

# data 1747 - 1783
f0 = 241 945
B0 = 493 472

# inference (posterior)
infer(fBirth, theta, f0, B0)
```

Idea: adjust `theta` distribution by comparison to data by simulation.

Inference by rejection sampling

```
# prior: Unit -> S
def guesser() :
    sample(uniform(0,1))

# predicate: int x int -> (S -> Boolean)
def checker(f0, B0) :
    lambda theta: gBirth(theta, B0) == f0

# infer: (Unit -> S) -> (S -> Boolean) -> S
def rejection(guesser, checker(f0, B0)):
    theta = guesser()
    if checker(f0, B0)(theta):
        return theta
    else:
        rejection(guesser, checker(f0, B0))
```

Problem: inefficient, hence other approximated methods

Inference by Metropolis-Hasting

Infer θ by **Bayes Law**: $P(\theta \mid f, B) = \alpha \cdot P(f \mid \theta, B)\ \pi(\theta)$

```
# proportion: S x S -> float
def proportion(x, y):
    return P(f | x, B0) / P(f | y, B0)

# Metropolis-Hasting: int * int * int -> S
def metropolis(n, f0, B0):
    if (n=0):
        return f0/B0
    else:
        x = metropolis(n-1, f0, B0)
        y = gaussian(x, 1)
        z = bernouilli(proportion(x, y))
        if (z == 0):
            return x
        else:
            return y
```

Probabilistic Programming

Semantics

Problems in semantics

- Prove formally the correspondence between algorithms, **implementations** and mathematics.
- Prove that two programs have **equivalent behavior**

Operational Semantics describes **how** probabilistic programs compute.

Denotational Semantics describes **what** probabilistic programs compute

Problems in semantics

- Prove formally the correspondence between algorithms, **implementations** and mathematics.
- Prove that two programs have **equivalent behavior**

Operational Semantics describes **how** probabilistic programs compute.

Proba(M, N) is the probability p that M reduces to N in one step, $M \xrightarrow{p} N$ defined by induction on the structure of M:

- $(\lambda x.M)N \xrightarrow{1} M[N/x]$
- $\text{coin} \xrightarrow{1/2} \underline{0}$
- $\text{coin} \xrightarrow{1/2} \underline{1}$

$\dots$

Denotational Semantics describes **what** probabilistic programs compute

Problems in semantics

- Prove formally the correspondence between algorithms, **implementations** and mathematics.
- Prove that two programs have **equivalent behavior**

Operational Semantics describes **how** probabilistic programs compute.

Proba(M, N) is the probability p that M reduces to N in one step, $M \xrightarrow{p} N$ defined by induction on the structure of M:

- $(\lambda x.M)N \xrightarrow{1} M[N/x]$
- $\text{coin} \xrightarrow{1/2} \underline{0}$
- $\text{coin} \xrightarrow{1/2} \underline{1}$

$\dots$

Denotational Semantics describes **what** probabilistic programs compute

$[\![M]\!]$ is a probabilistic distribution, if M is a closed ground type program.

- If M has type `nat`, then $[\![M]\!]$ a *discrete* distribution over integers
- If M has type `real`, then $[\![M]\!]$ a *continuous* distribution over reals

Operational Semantics on an example

(Borgström-Dal Lago-Gordon-Szymczak ICFP'16)

```
def addCoins():
  a = coin
  b = coin
  c = coin
  return (a + b + c)
```

- $(\lambda x.M)N \xrightarrow{1} M[N/x]$
- coin $\xrightarrow{1/2} \underline{0}$
- coin $\xrightarrow{1/2} \underline{1}$ ...

$$\text{addCoins()} \xrightarrow{1} \begin{array}{l}\text{a = coin}\\ \text{b = coin}\\ \text{c = coin}\\ \text{(a + b + c)}\end{array} \xrightarrow{1/2} \begin{array}{l}\text{a = 0}\\ \text{b = coin}\\ \text{c = coin}\\ \text{(a + b + c)}\end{array} \xrightarrow{1/2} \begin{array}{l}\text{a = 0}\\ \text{b = 1}\\ \text{c = coin}\\ \text{(a + b + c)}\end{array} \xrightarrow{1/2} \begin{array}{l}\text{a = 0}\\ \text{b = 1}\\ \text{c = 1}\\ \text{(a + b + c)}\end{array}$$

$$\xrightarrow{1} \begin{array}{l}\text{b = 1}\\ \text{c = 1}\\ \text{(0 + b + c)}\end{array} \xrightarrow{1} \begin{array}{l}\text{c = 1}\\ \text{(0 + 1 + c)}\end{array} \xrightarrow{1} \text{(0 + 1 + 1)} \xrightarrow{1} 2$$

Operational Semantics on an example

(Borgström-Dal Lago-Gordon-Szymczak ICFP'16)

```
def addCoins():
  a = coin
  b = coin
  c = coin
  return (a + b + c)
```

- $(\lambda x.M)N \xrightarrow{1} M[N/x]$
- coin $\xrightarrow{1/2} \underline{0}$
- coin $\xrightarrow{1/2} \underline{1}$ $\ldots$

$$\texttt{addCoins()} \xrightarrow{1} \begin{array}{l}\texttt{a = coin}\\ \texttt{b = coin}\\ \texttt{c = coin}\\ \texttt{(a + b + c)}\end{array} \xrightarrow[a=0]{1/2} \begin{array}{l}\texttt{a = }0\\ \texttt{b = coin}\\ \texttt{c = coin}\\ \texttt{(a + b + c)}\end{array} \xrightarrow[b=1]{1/2} \xrightarrow[c=1]{1/2} \xrightarrow{1\ *} 2$$

$$\begin{tikzcd}
\texttt{addCoins()} \arrow[r, bend left=50, "a{=}1\ b{=}1\ c{=}0", "1/8"', "*" near end] \arrow[r, "1/8", "a{=}1\ b{=}0\ c{=}1"', "*" near end] \arrow[r, bend right=50, "1/8", "a{=}0\ b{=}1\ c{=}1"', "*"' near end] & 2
\end{tikzcd}$$

Operational Semantics on an example

(Borgström-Dal Lago-Gordon-Szymczak ICFP'16)

```
def addCoins():
  a = coin
  b = coin
  c = coin
  return (a + b + c)
```

- $(\lambda x.M)N \xrightarrow{1} M[N/x]$
- coin $\xrightarrow{1/2} \underline{0}$
- coin $\xrightarrow{1/2} \underline{1}$ $\quad\ldots$

$$\texttt{addCoins()} \xrightarrow{1} \begin{array}{l}\texttt{a = coin}\\ \texttt{b = coin}\\ \texttt{c = coin}\\ \texttt{(a + b + c)}\end{array} \xrightarrow[a=0]{1/2} \begin{array}{l}\texttt{a = 0}\\ \texttt{b = coin}\\ \texttt{c = coin}\\ \texttt{(a + b + c)}\end{array} \xrightarrow[b=1]{1/2} \xrightarrow[c=1]{1/2} \xrightarrow{1\ *} 2$$

$$\begin{tikzcd}
\texttt{addCoins()} \arrow[r, bend left=50, "a{=}1\ b{=}1\ c{=}0", "1/8"', "*" near end] \arrow[r, "1/8", "a{=}1\ b{=}0\ c{=}1"', "*" near end] \arrow[r, bend right=50, "1/8", "a{=}0\ b{=}1\ c{=}1"', "*"' near end] & 2
\end{tikzcd}$$

$$\mathbf{Proba}^{\infty}(\texttt{addCoins()}, 2) = \frac{3}{8}$$

Operational Semantics

$\mathbf{Proba}^{\infty}(M, N)$ is the proba. that M reduces to N in *any* number of steps

Behavioral equivalence:

$$M_1 \simeq M_2 \quad \textbf{iff} \quad \forall C[\,],\ \mathbf{Proba}^{\infty}(C[M_1], \underline{0}) = \mathbf{Proba}^{\infty}(C[M_2], \underline{0})$$

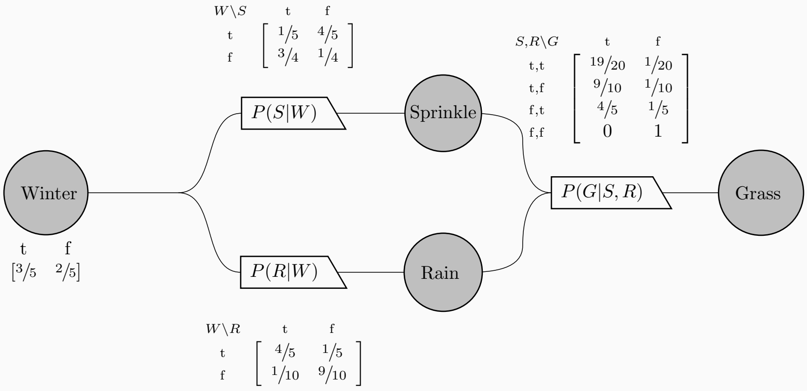
```
def addCoins1():
  a = coin
  b = coin
  c = coin
  return (a + b + c)
```

```
def addCoins2():
  b = coin
  a = coin
  c = coin
  return (a + b + c)
```

```
def infer1(f0, B0):
    rejection(guesser, checker(f0, B0)):

def infer2(f0, B0):
    metropolis(f0, B0, 1000)
```

Semantics of a Bayesian Network (Jacobs-Kissinger-Zanasi FOSSACS'19)



$$\begin{array}{c|cc} & \mathrm{t} & \mathrm{f} \\ \hline & 3/5 & 2/5 \end{array}$$

$$\begin{array}{c|cc} W\backslash S & \mathrm{t} & \mathrm{f} \\ \hline \mathrm{t} & 1/5 & 4/5 \\ \mathrm{f} & 3/4 & 1/4 \end{array}$$

$$\begin{array}{c|cc} S,R\backslash G & \mathrm{t} & \mathrm{f} \\ \hline \mathrm{t,t} & 19/20 & 1/20 \\ \mathrm{t,f} & 9/10 & 1/10 \\ \mathrm{f,t} & 4/5 & 1/5 \\ \mathrm{f,f} & 0 & 1 \end{array}$$

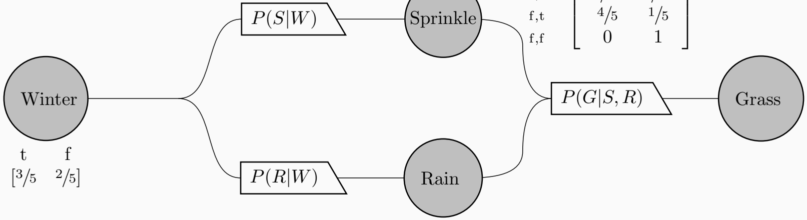
$$\begin{array}{c|cc} W\backslash R & \mathrm{t} & \mathrm{f} \\ \hline \mathrm{t} & 4/5 & 1/5 \\ \mathrm{f} & 1/10 & 9/10 \end{array}$$

$$p(S) = \left(\sum_{a \in \{\mathrm{t},\mathrm{f}\}} P(S|W)_{a,b} \cdot p(W)_a \right)_{b \in \{\mathrm{t},\mathrm{f}\}}$$

Semantics of a Bayesian Network (Jacobs-Kissinger-Zanasi FOSSACS'19)

$$\begin{array}{c|cc} W\backslash S & \text{t} & \text{f} \\ \hline \text{t} & 1/5 & 4/5 \\ \text{f} & 3/4 & 1/4 \end{array}$$

$$\begin{array}{c|cc} S,R\backslash G & \text{t} & \text{f} \\ \hline \text{t,t} & 19/20 & 1/20 \\ \text{t,f} & 9/10 & 1/10 \\ \text{f,t} & 4/5 & 1/5 \\ \text{f,f} & 0 & 1 \end{array}$$

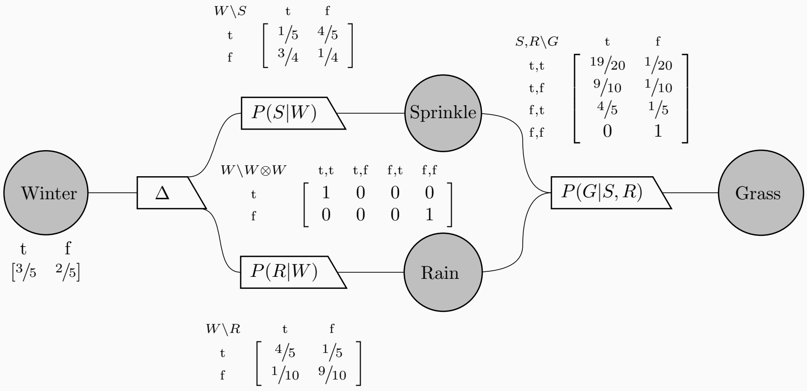


$$\begin{array}{cc} \text{t} & \text{f} \\ [3/5 & 2/5] \end{array}$$

$$\begin{array}{c|cc} W\backslash R & \text{t} & \text{f} \\ \hline \text{t} & 4/5 & 1/5 \\ \text{f} & 1/10 & 9/10 \end{array}$$

$$\begin{aligned} p(W)\ P(S|W) &= p(S) \\ p(W)\ P(R|W) &= p(R) \end{aligned} \quad \text{and} \quad (p(S) \otimes p(R))\ P(G|S,R) = p(G)$$

Semantics of a Bayesian Network (Jacobs-Kissinger-Zanasi FOSSACS'19)



$$\begin{array}{c|cc} & t & f \\ \hline & 3/5 & 2/5 \end{array}$$

$$\begin{array}{c|cc} W\backslash S & t & f \\ \hline t & 1/5 & 4/5 \\ f & 3/4 & 1/4 \end{array}$$

$$\begin{array}{c|cccc} W\backslash W\otimes W & t,t & t,f & f,t & f,f \\ \hline t & 1 & 0 & 0 & 0 \\ f & 0 & 0 & 0 & 1 \end{array}$$

$$\begin{array}{c|cc} W\backslash R & t & f \\ \hline t & 4/5 & 1/5 \\ f & 1/10 & 9/10 \end{array}$$

$$\begin{array}{c|cc} S,R\backslash G & t & f \\ \hline t,t & 19/20 & 1/20 \\ t,f & 9/10 & 1/10 \\ f,t & 4/5 & 1/5 \\ f,f & 0 & 1 \end{array}$$

$$p(W)\ \Delta\ (P(S|W) \otimes P(R|W))\ P(G|S,R)\ =\ p(G)$$

Denotational Semantics:

(Danos-Ehrhard 2011)

Probabilistic Coherent Spaces (**Pcoh**) an adequate model of probabilistic functional programming with discrete probability.

Object $(|X|, \mathrm{P}(X))$

- the universe $|X|$ is a (potentially infinite) set of final states
- a set of vectors $\mathrm{P}(X) \subseteq (\mathbb{R}^+)^{|X|}$

closure: $\mathbf{P}(\mathbf{X})^{\perp\perp} = \mathbf{P}(\mathbf{X})$ where
$\forall P \subseteq (\mathbb{R}^+)^{|X|},\ P^\perp = \{v \in (\mathbb{R}^+)^{|X|} \ ;\ \forall u \in P,\ \sum_{a\in|X|} u_a v_a \leq 1\}$

bounded covering:

$$\forall a \in |X|,\ \exists v \in P(X)\ ;\ v_a \neq 0 \quad \text{and} \quad \exists p > 0,\ ;\ \forall v \in \mathrm{P}(X),\ v_a \leq p.$$

Denotational Semantics:

(Danos-Ehrhard 2011)

Probabilistic Coherent Spaces (**Pcoh**) an adequate model of probabilistic functional programming with discrete probability.

Object $(|X|, \mathrm{P}(X))$

- the universe $|X|$ is a (potentially infinite) set of final states
- a set of vectors $\mathrm{P}(X) \subseteq (\mathbb{R}^+)^{|X|}$

Type $A, B ::= \mathtt{nat} \mid A \to B \mid \dots$ are interpreted by objects $[\![A]\!] = (|A|, \mathrm{P}(A))$ defined by induction on A.

Denotational Semantics:

(Danos-Ehrhard 2011)

Probabilistic Coherent Spaces (**Pcoh**) an adequate model of probabilistic functional programming with discrete probability.

Object $(|X|, \mathrm{P}(X))$

- the universe $|X|$ is a (potentially infinite) set of final states
- a set of vectors $\mathrm{P}(X) \subseteq (\mathbb{R}^+)^{|X|}$

Type $A, B ::= \texttt{nat} \mid A \to B \mid \dots$ are interpreted by objects $[\![A]\!] = (|A|, \mathrm{P}(A))$ defined by induction on A.

- unit type 1: $|1| = \{()\}$ and $\mathrm{P}(1) = [0, 1]$

Denotational Semantics:

(Danos-Ehrhard 2011)

Probabilistic Coherent Spaces (**Pcoh**) an adequate model of probabilistic functional programming with discrete probability.

Object $(|X|, \mathrm{P}(X))$

- the universe $|X|$ is a (potentially infinite) set of final states
- a set of vectors $\mathrm{P}(X) \subseteq (\mathbb{R}^+)^{|X|}$

Type $A, B ::= \texttt{nat} \mid A \to B \mid \ldots$ are interpreted by objects $[\![A]\!] = (|A|, \mathrm{P}(A))$ defined by induction on A.

- unit type 1: $|1| = \{()\}$ and $\mathrm{P}(1) = [0, 1]$
- $\mathcal{B} = 1 \oplus 1$: $|\mathcal{B}| = \{\mathrm{t}, \mathrm{f}\}$ and $\mathrm{P}(\mathcal{B}) = \{x \cdot \mathrm{t} + y \cdot \mathrm{f} \mid x + y \leq 1\}$

$p(W) = \left[{}^{3}/_{5}, {}^{2}/_{5}\right] \in \mathrm{P}(\mathcal{B})$.

Denotational Semantics:

(Danos-Ehrhard 2011)

Probabilistic Coherent Spaces (**Pcoh**) an adequate model of probabilistic functional programming with discrete probability.

Object $(|X|, \mathrm{P}(X))$

- the universe $|X|$ is a (potentially infinite) set of final states
- a set of vectors $\mathrm{P}(X) \subseteq (\mathbb{R}^+)^{|X|}$

Type $A, B ::= \mathtt{nat} \mid A \to B \mid \ldots$ are interpreted by objects $[\![A]\!] = (|A|, \mathrm{P}(A))$ defined by induction on A.

- unit type 1: $|1| = \{()\}$ and $\mathrm{P}(1) = [0,1]$
- $\mathcal{B} = 1 \oplus 1$: $|\mathcal{B}| = \{\mathrm{t}, \mathrm{f}\}$ and $\mathrm{P}(\mathcal{B}) = \{x \cdot \mathrm{t} + y \cdot \mathrm{f} \mid x + y \leq 1\}$

  $p(W) = \left[{}^{3}\!/_{5}, {}^{2}\!/_{5}\right] \in \mathrm{P}(\mathcal{B})$.
- $\mathtt{nat} = 1 \oplus \mathtt{nat}$: $|\mathtt{nat}| = \mathbb{N}$ and $\mathrm{P}(\mathtt{nat})$ sub-proba distrib. over $\mathbb{N}$

Denotational Semantics: (Danos-Ehrhard 2011)

Probabilistic Coherent Spaces (**Pcoh**) an adequate model of probabilistic functional programming with discrete probability.

Object $(|X|, \mathrm{P}(X))$

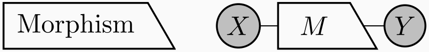
- the universe $|X|$ is a (potentially infinite) set of final states
- a set of vectors $\mathrm{P}(X) \subseteq (\mathbb{R}^+)^{|X|}$

Type $A, B ::= \mathtt{nat} \mid A \to B \mid \dots$ are interpreted by objects $[\![A]\!] = (|A|, \mathrm{P}(A))$ defined by induction on A.

- unit type 1: $|1| = \{()\}$ and $\mathrm{P}(1) = [0,1]$
- $\mathcal{B} = 1 \oplus 1$: $|\mathcal{B}| = \{\mathrm{t}, \mathrm{f}\}$ and $\mathrm{P}(\mathcal{B}) = \{x \cdot \mathrm{t} + y \cdot \mathrm{f} \mid x + y \leq 1\}$

  $p(W) = \left[{}^{3}/{}_{5}, {}^{2}/{}_{5}\right] \in \mathrm{P}(\mathcal{B})$.
- $\mathtt{nat} = 1 \oplus \mathtt{nat}$: $|\mathtt{nat}| = \mathbb{N}$ and $\mathrm{P}(\mathtt{nat})$ sub-proba distrib. over $\mathbb{N}$
- $\mathcal{B}^* = 1 \oplus (\mathcal{B} \otimes \mathcal{B}^*)$: $|\mathcal{B}^*| = \{\epsilon\} \cup \{b_1 \cdot \dots \cdot b_n \mid n \in \mathbb{N},\ b_i \in |\mathcal{B}|\}$ and $\mathrm{P}(\mathcal{B}^*)$ sub-probability distribution over words of booleans.

Semantics: Probabilistic Coherent Spaces (Danos-Ehrhard 2011)

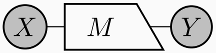


$\in (\mathbb{R}^+)^{|X| \times |Y|}$ is a matrix

$$\forall x \in \underset{\subseteq (\mathbb{R}^+)^{|X|}}{\mathrm{P}(X)},\ M\cdot x = \left(\sum_{a\in|X|} M_{a,b}\, x_a\right)_{b\in|Y|} \in \underset{\subseteq (\mathbb{R}^+)^{|Y|}}{\mathrm{P}(Y)}$$

Semantics: Probabilistic Coherent Spaces (Danos-Ehrhard 2011)

Morphism



$\in (\mathbb{R}^+)^{\mathcal{M}_{\mathsf{fin}}(|X|)\times|Y|}$ is a matrix

$$\forall x \in \underset{\subseteq(\mathbb{R}^+)^{|X|}}{\mathrm{P}(X)},\ M(x) = \left(\sum_{m\in\mathcal{M}_{\mathsf{fin}}(|X|)} M_{m,b} \prod_{a\in m} x_a^{m(a)}\right)_{b\in|Y|} \in \underset{\subseteq(\mathbb{R}^+)^{|Y|}}{\mathrm{P}(Y)}$$

Semantics: Probabilistic Coherent Spaces

(Danos-Ehrhard 2011)

Morphism

$\in (\mathbb{R}^+)^{\mathcal{M}_{\mathsf{fin}}(|X|)\times|Y|}$ is a matrix

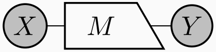
$$\forall x \in \underset{\subseteq(\mathbb{R}^+)^{|X|}}{\mathrm{P}(X)}\,,\; M(x) = \left(\sum_{m\in\mathcal{M}_{\mathsf{fin}}(|X|)} M_{m,b}\prod_{a\in m} x_a^{m(a)}\right)_{b\in|Y|} \in \underset{\subseteq(\mathbb{R}^+)^{|Y|}}{\mathrm{P}(Y)}$$

Program

$M, N ::= x \mid \lambda x^A.M \mid (M)N \mid \mathbf{fix}(M) \mid \underline{n} \mid \mathsf{coin} \mid \ldots$
are interpreted by morphisms, by induction on M

Semantics: Probabilistic Coherent Spaces (Danos-Ehrhard 2011)

Morphism



$\in (\mathbb{R}^+)^{\mathcal{M}_{\mathsf{fin}}(|X|)\times|Y|}$ is a matrix

$$\forall x \in \underset{\subseteq(\mathbb{R}^+)^{|X|}}{\mathrm{P}(X)},\ M(x) = \left(\sum_{m\in\mathcal{M}_{\mathsf{fin}}(|X|)} M_{m,b} \prod_{a\in m} x_a^{m(a)}\right)_{b\in|Y|} \in \underset{\subseteq(\mathbb{R}^+)^{|Y|}}{\mathrm{P}(Y)}$$

Program

$M, N ::= x \mid \lambda x^A.M \mid (M)N \mid \mathbf{fix}(M) \mid \underline{n} \mid \mathsf{coin} \mid \dots$
are interpreted by morphisms, by induction on M

- if $M : A$, then $[\![M]\!] \in \mathrm{P}(A)$

$$[\![\underline{n}]\!] = (0, \dots, \underset{n}{1}, 0, \dots) \qquad [\![\mathsf{coin}]\!] = (\underset{0}{\tfrac{1}{2}}, \underset{1}{\tfrac{1}{2}}, 0, \dots)$$

- if $M : A \to B$, then $[\![M]\!] : \mathrm{P}(A) \to \mathrm{P}(B)$ is a Taylor series

Semantics: Probabilistic Coherent Spaces (Danos-Ehrhard 2011)

Morphism

$\in (\mathbb{R}^+)^{\mathcal{M}_{\mathsf{fin}}(|X|)\times|Y|}$ is a matrix

$$\forall x \in \underset{\subseteq(\mathbb{R}^+)^{|X|}}{\mathrm{P}\,(X)}\,,\ M(x) = \left(\sum_{m\in\mathcal{M}_{\mathsf{fin}}(|X|)} M_{m,b} \prod_{a\in m} x_a^{m(a)}\right)_{b\in|Y|} \in \underset{\subseteq(\mathbb{R}^+)^{|Y|}}{\mathrm{P}\,(Y)}$$

Program

$M, N ::= x \mid \lambda x^A.M \mid (M)N \mid \mathbf{fix}(M) \mid \underline{n} \mid \mathsf{coin} \mid \dots$
are interpreted by morphisms, by induction on M

- if $M : A$, then $[\![M]\!] \in \mathrm{P}\,(A)$

$$[\![\underline{n}]\!] = (0,\dots,\underset{n}{1},0,\dots) \qquad [\![\mathsf{coin}]\!] = (\underset{0}{\tfrac{1}{2}},\underset{1}{\tfrac{1}{2}},0,\dots)$$

- if $M : A \to B$, then $[\![M]\!] : \mathrm{P}\,(A) \to \mathrm{P}\,(B)$ is a Taylor series
  if $M : 1 \to 1$, then $[\![M]\!]$ is smooth real function from $[0,1]$ to $[0,1]$
  if $M : \mathtt{nat} \multimap \mathtt{nat}$, then $[\![M]\!]$ is a sub-stochastic matrix

Probabilistic Coherent Spaces

Sound: Deterministic case: if $M \to N$, then $[\![M]\!] = [\![N]\!]$.

$$[\![M]\!] = \sum_N \mathbf{Proba}(M, N)[\![N]\!]$$

Probabilistic Coherent Spaces

Sound: Deterministic case: if $M \to N$, then $[\![M]\!] = [\![N]\!]$.

$$[\![M]\!] = \sum_N \mathbf{Proba}(M, N)[\![N]\!]$$

Adequate: If M close term of type `nat`, then $[\![M]\!]_n = \mathbf{Proba}^\infty(M, \underline{n})$
(Danos-Ehrhard 2011) $[\![M]\!]$ sub-proba distrib. on $\mathbb{N}$.

Probabilistic Coherent Spaces

Sound: Deterministic case: if $M \to N$, then $[\![M]\!] = [\![N]\!]$.

$$[\![M]\!] = \sum_N \mathbf{Proba}(M, N)[\![N]\!]$$

Adequate: If M close term of type `nat`, then $[\![M]\!]_n = \mathbf{Proba}^\infty(M, \underline{n})$
(Danos-Ehrhard 2011) $[\![M]\!]$ sub-proba distrib. on $\mathbb{N}$.

Fully abstract: $[\![M]\!] = [\![N]\!]$ iff $M \simeq N$
(Ehrhard-Pagani-T. POPL'14) Based on Taylor series

This Full Abstraction result generalizes to:

- Probabilistic Games (Castellan-Clairambault-Paquet-Winskel LICS'18)
- Call-By-Push-Value (Ehrhard-T. JACM 2019)
- Quantum Programming (Clairambault-De Visme POPL'20)

Zoom on analytic Full Abstraction

Key idea: Prove separate definability

Assume $f \neq g$ of type σ. Then, there is $\alpha \in |\sigma|$ such that $f_\alpha \neq g_\alpha$.

Zoom on analytic Full Abstraction

Key idea: Prove separate definability

Assume $f \neq g$ of type σ. Then, there is $\alpha \in |\sigma|$ such that $f_\alpha \neq g_\alpha$.

1. Define $T_\alpha(\zeta_\alpha)$ a testing term with finitely many parameters ζ_α

Zoom on analytic Full Abstraction

Key idea: Prove separate definability

Assume $f \neq g$ of type σ. Then, there is $\alpha \in |\sigma|$ such that $f_\alpha \neq g_\alpha$.

1. Define $T_\alpha(\zeta_\alpha)$ a testing term with finitely many parameters ζ_α
2. if ζ_α are replaced by small enough positive reals, then T_α is in pPCF

Zoom on analytic Full Abstraction

Key idea: Prove separate definability

Assume $f \neq g$ of type σ. Then, there is $\alpha \in |\sigma|$ such that $f_\alpha \neq g_\alpha$.

1. Define $T_\alpha(\zeta_\alpha)$ a testing term with finitely many parameters ζ_α
2. if ζ_α are replaced by small enough positive reals, then T_α is in pPCF
3. $[\![T_\alpha(\zeta_\alpha)]\!]f$ is a formal series with finitely many parameters such that

$$[\![T_\alpha(\zeta_\alpha)]\!]f = \cdots + c \cdot f_\alpha \prod_{i \in \alpha} \zeta_i + \ldots$$

Zoom on analytic Full Abstraction

Key idea: Prove separate definability

Assume $f \neq g$ of type σ. Then, there is $\alpha \in |\sigma|$ such that $f_\alpha \neq g_\alpha$.

1. Define $T_\alpha(\zeta_\alpha)$ a testing term with finitely many parameters ζ_α
2. if ζ_α are replaced by small enough positive reals, then T_α is in pPCF
3. $[\![T_\alpha(\zeta_\alpha)]\!]f$ is a formal series with finitely many parameters such that

$$[\![T_\alpha(\zeta_\alpha)]\!]f = \cdots + c \cdot f_\alpha \prod_{i \in \alpha} \zeta_i + \ldots$$

4. If two formal series have different coefficients, then there are reals small enough on which they differ

Zoom on analytic Full Abstraction

Key idea: Prove separate definability

Assume $f \neq g$ of type σ. Then, there is $\alpha \in |\sigma|$ such that $f_\alpha \neq g_\alpha$.

1. Define $T_\alpha(\zeta_\alpha)$ a testing term with finitely many parameters ζ_α
2. if ζ_α are replaced by small enough positive reals, then T_α is in pPCF
3. $[\![T_\alpha(\zeta_\alpha)]\!]f$ is a formal series with finitely many parameters such that

$$[\![T_\alpha(\zeta_\alpha)]\!]f = \dots + c \cdot f_\alpha \prod_{i \in \alpha} \zeta_i + \dots$$

4. If two formal series have different coefficients, then there are reals small enough on which they differ
5. T_α is the program that separates f and g

Definability: **Pcoh** morphisms are far from being definable

Next Full Abstraction Challenges

Continuous probability semantics

Denotational Semantics

" The developers of probabilistic programming languages need to ensure that the implementation of compilers, optimizers, and inference algorithms do not have bugs." (van de Meent-Paige-Yang-Wood 2018)

Denotational semantics allows to define the mathematical meaning of every probabilistic program.

Problem: Measurable sets and measurable functions are not suitable to interpret higher order functional probabilistic programming languages.

The evaluation map $\mathrm{ev} : \mathcal{F}(\mathbb{R}, \mathbb{R}) \times \mathbb{R} \to \mathbb{R}$ with $\mathrm{ev}(f, r) = f(r)$ is not measurable whatever measurable sets we put on the set $\mathcal{F}(\mathbb{R}, \mathbb{R})$ of measurable functions between reals endowed with borel sets.

(Aumann 1961)

Towards FA for continuous probability

Models for Higher-order languages with continuous probabilities

- Quasi Borel Spaces
  (Kammar-Staton+Heunen-Yang LICS'17, +Vakar POPL'19)
- Measurable postive Cones and Stable maps
  (Ehrhard-Pagani-T. POPL'18)
- Ordered Banach Spaces and Regular maps
  (Dahlqvist-Kozen POPL'20)

Towards FA for continuous probability

Models for Higher-order languages with continuous probabilities

- Quasi Borel Spaces
  (Kammar-Staton+Heunen-Yang LICS'17, +Vakar POPL'19)
- Measurable postive Cones and Stable maps
  (Ehrhard-Pagani-T. POPL'18)
- Ordered Banach Spaces and Regular maps
  (Dahlqvist-Kozen POPL'20)

Towards full abstraction for the measurable positive cones model

- This model is a conservative extension of **Pcoh** (Crubillé LICS18)
- The linear and non-linear structures of cones (Ehrhard 2020)
- Wanted: prove that morphisms are power series

Next Full Abstraction Challenges

An effective observational equivalence

Distance and Observational equivalence

(Ehrhard FSCD19)

Observational equivalence: is not effective

$$M_1 \simeq M_2 \quad \textbf{iff} \quad \forall C[\,],\ \textbf{Proba}^{\infty}(C[M_1], \underline{0}) = \textbf{Proba}^{\infty}(C[M_2], \underline{0})$$

Observational distance:

$$\mathrm{d}_{\mathrm{obs}}(M, N) = \sup_{C} |\,\textbf{Proba}^{\infty}(C\ \ [M_1], \underline{0}) - \textbf{Proba}^{\infty}(C\ \ [M_2], \underline{0})|$$

Distance and Observational equivalence

(Ehrhard FSCD19)

Observational equivalence: is not effective

$$M_1 \simeq M_2 \quad \textbf{iff} \quad \forall C[\,],\ \textbf{Proba}^\infty(C[M_1], \underline{0}) = \textbf{Proba}^\infty(C[M_2], \underline{0})$$

Observational distance:

$$\mathrm{d}_{\mathrm{obs}}(M, N) = \sup_{C} |\,\textbf{Proba}^\infty(C\ \ [M_1], \underline{0}) - \textbf{Proba}^\infty(C\ \ [M_2], \underline{0})|$$

Semantical distance: $\mathrm{d}_{\textbf{Pcoh}}(x, y) = \| x - x \wedge y \| + \| x - x \wedge y \|$

Distance and Observational equivalence

(Ehrhard FSCD19)

Observational equivalence: is not effective

$$M_1 \simeq M_2 \quad \textbf{iff} \quad \forall C[\ ],\ \textbf{Proba}^{\infty}(C[M_1], \underline{0}) = \textbf{Proba}^{\infty}(C[M_2], \underline{0})$$

Observational distance:

$$\mathrm{d}_{\mathrm{obs}}(M, N) = \sup_{C} |\,\textbf{Proba}^{\infty}(C\ \ [M_1], \underline{0}) - \textbf{Proba}^{\infty}(C\ \ [M_2], \underline{0})|$$

Semantical distance: $\mathrm{d}_{\mathbf{Pcoh}}(x, y) = \| x - x \wedge y \| + \| x - x \wedge y \|$

Amplification Pb: $\mathrm{d}_{\mathrm{obs}}(\textbf{coin}\, 0, \textbf{coin}\, \epsilon) = 1$ due to $C = \lambda x.\mathrm{if}(x, C\, x, 0)$

Distance and Observational equivalence

(Ehrhard FSCD19)

Observational equivalence: is not effective

$$M_1 \simeq M_2 \quad \textbf{iff} \quad \forall C[\ ],\ \textbf{Proba}^\infty(C[M_1], \underline{0}) = \textbf{Proba}^\infty(C[M_2], \underline{0})$$

Observational distance:

$$\mathrm{d}^{\mathbf{p}}_{\mathrm{obs}}(M, N) = \sup_{C} |\, \textbf{Proba}^\infty(C^{\mathbf{p}}[M_1], \underline{0}) - \textbf{Proba}^\infty(C^{\mathbf{p}}[M_2], \underline{0})|$$

Semantical distance: $\mathrm{d}_{\mathbf{Pcoh}}(x, y) = \| x - x \wedge y \| + \| x - x \wedge y \|$

Amplification Pb: $\mathrm{d}_{\mathrm{obs}}(\mathbf{coin}\, 0, \mathbf{coin}\, \epsilon) = 1$ due to $C = \lambda x.\mathrm{if}(x, C\, x, 0)$

Calm down contexts: $C^p = \lambda z. C[\mathrm{if}(\mathbf{coin}\, p, z, \omega)]$

Distance and Observational equivalence

(Ehrhard FSCD19)

Observational equivalence: is not effective

$$M_1 \simeq M_2 \quad \textbf{iff} \quad \forall C[\,],\ \textbf{Proba}^\infty(C[M_1], \underline{0}) = \textbf{Proba}^\infty(C[M_2], \underline{0})$$

Observational distance:

$$\mathrm{d}^{\mathbf{p}}_{\mathrm{obs}}(M, N) = \sup_C |\,\textbf{Proba}^\infty(C^{\mathbf{p}}[M_1], \underline{0}) - \textbf{Proba}^\infty(C^{\mathbf{p}}[M_2], \underline{0})|$$

Semantical distance: $\mathrm{d}_{\mathbf{Pcoh}}(x, y) = \| x - x \wedge y \| + \| x - x \wedge y \|$

Amplification Pb: $\mathrm{d}_{\mathrm{obs}}(\textbf{coin}\,0, \textbf{coin}\,\epsilon) = 1$ due to $C = \lambda x.\mathrm{if}(x, C\,x, 0)$

Calm down contexts: $C^p = \lambda z.C[\mathrm{if}(\textbf{coin}\,p, z, \omega)]$

Theorem: correspondence between syntax and semantics

$$\mathrm{d}^{p}_{\mathrm{obs}}(M, N) = 0 \Rightarrow [\![M]\!] = [\![N]\!]$$

$$\mathrm{d}^{p}_{\mathrm{obs}}(M, N) \leq \frac{p}{1-p}\mathrm{d}_{\mathbf{Pcoh}}([\![M]\!], [\![N]\!])$$

Conclusion

Higher-order computability and the related Full Abstraction are fascinating questions developed by Gandy with inspiring insights.

They generated theoretical development that had practical applications in the programming language community.

I believe that it is not the end.