

Probabilistic Programming Languages

M2 MPRI 2021-2022

Guillaume BAUDART (guillaume.baudart@ens.fr)

Christine TASSON (christine.tasson@lip6.fr)

Semantics of Probabilistic Programming

Discrete probability (Part II)

Probabilistic PCF - Denotational Semantics

A sound model

Pcoh is a cartesian closed category, CPO-enriched, hence a model of PCF, which is also a model of probabilistic PCF.

Interpretation of terms

$$\llbracket \overline{G \vdash \text{bernoulli}(p) : \text{bool dist}} \rrbracket : \begin{cases} (\gamma, \text{true}) \mapsto p \\ (\gamma, \text{false}) \mapsto 1 - p \end{cases}$$

$$\llbracket \frac{G \vdash d : t_1 \text{ dist} \quad G, x : t_1 \vdash e : t_2}{G \vdash \text{let } x = d \text{ in } e : t_2} \rrbracket : (\gamma, a_2) \mapsto \sum_a \llbracket d \rrbracket(\gamma, a) \llbracket e \rrbracket(\gamma, a, a_2)$$

$$\llbracket \overline{G \vdash \text{true} : \text{bool}} \rrbracket : \begin{cases} (\gamma, \text{true}) \mapsto 1 \\ (\gamma, \text{false}) \mapsto 0 \end{cases}$$

$$\llbracket \text{if } M \text{ then } N \text{ else } P \rrbracket(\gamma, a) = \llbracket M \rrbracket(\gamma, \text{true}) \llbracket N \rrbracket(\gamma, a) + \llbracket M \rrbracket(\gamma, \text{false}) \llbracket P \rrbracket(\gamma, a)$$

Compute the denotational semantics

$\vdash \text{bern } 0,5 : \text{bool dist}$

bern 0,5	
t	1/2
f	1/2

$a, b, c : \text{bool} \vdash a+b+c : \text{nat}$

a	b	c	(a+b+c)	\mathbb{R}^+
t	t	t	3	1
t	t	f	2	1
t	f	t	2	0
...				

$\vdash \underbrace{\text{let } a = \text{sample bern } 0,5 \text{ in let } b = \dots \text{ in let } c = \dots \text{ in } a+b+c}_{\mathcal{R}}$: nat

\mathcal{R}

Bernoulli :

- let a = sample(bernoulli 0.5) in
- let b = sample(bernoulli 0.5) in
- let c = sample(bernoulli 0.5) in
- a+b+c

$M : \text{nat}$	\mathbb{R}_+
n	$\sum_{x \in \{\text{bool}\}} \mathbb{P}(a=x) \sum_{y \in \{\text{bool}\}} \mathbb{P}(a=y) \sum_{z \in \{\text{bool}\}} \mathbb{P}(c=z) \mathbb{1}_{a+b+c=n}$
0	$\mathbb{P}(a=f) \mathbb{P}(b=f) \mathbb{P}(c=f) = \frac{1}{8}$
1	$\mathbb{P}(a=f) \mathbb{P}(b=f) \mathbb{P}(c=t) + \mathbb{P}(a=f) \mathbb{P}(b=t) \mathbb{P}(c=f) + \mathbb{P}(a=t) \mathbb{P}(b=f) \mathbb{P}(c=f)$ $= \frac{3}{8}$
...	

Examples - Compute the semantics

- ② $a, b, c : \text{bool} \vdash a = 0 \text{ or } b = 0 : \text{bool}$ ③ $f : \text{nat}, a, b, c : \text{bool} \vdash a + b + c : \text{nat}$ ④ $f : \text{nat}, a, b, c : \text{bool} \vdash f : \text{nat}$ ⑤
- ① $\vdash \text{bern } 0.5 : \text{dist bool}$ $f : \text{nat}, a, b, c : \text{bool} \vdash \text{if } a = 0 \text{ or } b = 0 \text{ then } a + b + c \text{ else } f$ ⑥
- ⑦ $f : \text{nat} \vdash \text{let } a = \text{sample bern } 0.5; \text{ let } b = \dots \text{ let } c = \dots \text{ in if } a = 0 \text{ or } b = 0 \text{ then } a + b + c \text{ else } f$
 $\vdash \text{fix } \lambda f. \dots$

Funny Bernoulli by rejection sampling :

```

1  fix lambda f : bool
2    let a = sample(bernoulli 0.5) in
3    let b = sample(bernoulli 0.5) in
4    let c = sample(bernoulli 0.5) in
5    if (a==0 or b==0) then a+b+c
6    else f
    
```

③ \cup ④ \Rightarrow ⑤

①

bool	\mathbb{R}^+	Probab Table
t	0,5	Factor Σ
f	0,5	

② Relation $\rightarrow 0/1$

a, b, c	$a=0 \vee b=0$	\mathbb{R}^+
$[t][t][t]$	f	1
$[t][t][f]$	f	0
$[t][f][t]$	t	1
\dots	\dots	\dots

③

a, b, c	$a+b+c$	$\mathbb{R}^+ (0,1)$
$[t][t][t]$	3	1
$[t][t][f]$	2	1
$[t][f][t]$	2	0

④

a, b, c, f	$\mathbb{R}^+ (0,1)$
$[m][t][t][t]$ m	1
otherwise	0

4

Examples - Compute the semantics

⑥

f	π_b	\mathbb{R}^+
$[]$	a	$\sum_{x,y,z} P(a=x)P(b=y)P(c=z) \mathbb{1}_{a=0 \text{ or } b=0}(x,y) \mathbb{1}_{a+b+c=k}(x,y,z)$
$[m]$	k	$\sum_{x,y,z} \pi(a=x)\pi(b=y)\pi(c=z) \mathbb{1}_{a=1 \text{ and } b=1}(x,y) \mathbb{1}_{k=m} = \frac{1}{4} \mathbb{1}_{k=m}$

Funny Bernoulli by rejection sampling :

```

1  fix lambda f:bool
2    let a = sample(bernoulli 0.5) in
3    let b = sample(bernoulli 0.5) in
4    let c = sample(bernoulli 0.5) in
5    if (a==0 or b==0) then a+b+c
6                                else f

```

⑦ F^0 : does not use f : $0 \mapsto 1/8$
 F^1 : uses f at most 1 : $0 \mapsto 1/8 + \frac{1}{4} \cdot F^0 = 1/8 (1 + 1/4)$
 F^m : uses f at most m times : $0 \mapsto 1/8 (1 + 1/4 + \dots + 1/4^m)$

$$F = \bigvee F^m : 0 \mapsto \frac{1}{8} \frac{1}{3/4} = \frac{1}{6}$$

PCF - Denotational semantics

CPO-enriched :

CPO, REL, MREL

$\mathcal{C}(A, B)$ is a complete partial order and \circ is continuous.

$$\llbracket \frac{G \vdash e : t \rightarrow t}{G \vdash \text{fixe} : t} \rrbracket = \sup F^n \text{ with } \begin{cases} F^0 = \perp \in \mathcal{C}(G, t) \\ F^{n+1} = \text{ev} \circ \langle \llbracket G \vdash e : t \rightarrow t \rrbracket, F^n \rangle \end{cases}$$

Probabilistic Language - Type and Semantics

$$\llbracket G \vdash e : t \rrbracket : \llbracket G \rrbracket \times \llbracket t \rrbracket \rightarrow \mathbb{R}^+ \text{ and } \llbracket G \vdash e : t \text{ dist} \rrbracket(\gamma, a) \in P(G \rightarrow t)$$

$$\llbracket \overline{G \vdash \text{bernoulli}(p) : \text{bool dist}} \rrbracket : \begin{cases} (\gamma, \text{true}) \mapsto p \\ (\gamma, \text{false}) \mapsto 1 - p \end{cases}$$

$$\llbracket \frac{G \vdash d : t_1 \text{ dist} \quad G, x : t_1 \vdash e : t_2}{G \vdash \text{let } x = d \text{ in } e : t_2} \rrbracket : (\gamma, a_2) \mapsto \sum_a \llbracket d \rrbracket(\gamma_1, a) \llbracket e \rrbracket(\gamma_2, a, a_2)$$

$$\llbracket \frac{G \vdash e_1 : \text{bool} \quad G \vdash e_2 : t}{G \vdash \text{assume}(e_1) ; e_2 : t} \rrbracket(\gamma, a) = \begin{cases} \llbracket e_2 \rrbracket(\gamma, a) & \text{if } \llbracket e_1 \rrbracket(g, \text{true}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\llbracket \frac{G \vdash e : t}{G \vdash \text{infer}(e) : t \text{ dist}} \rrbracket(\gamma, a) = \frac{\llbracket G \vdash e : t \rrbracket(\gamma, a)}{\sum_{a \in |t|} \llbracket G \vdash e : t \rrbracket(\gamma, a)}$$

Examples

$$\frac{\text{mat}}{x \in \{1, \dots, 6\}} \mid \mathbb{R}^+$$

$$\frac{a \quad b \quad (\text{assume } |a-b|=1)}{x \quad x+1 \quad x \quad x-1} \mid \mathbb{R}^+$$

otherwise

$$x \in \{1, \dots, 5\}$$

$$x \in \{2, \dots, 6\}$$

$$\frac{a \quad b \quad \pi}{x \quad x+1 \quad 2x+1 \quad x \quad x-1 \quad 2x-1} \mid \mathbb{R}^+$$

otherwise

Dice :

$$\vdash \text{unif}(1,6)$$

$$a, b \vdash \text{assume } |a-b|=1 ; a+b ; \text{mat}$$

$$\vdash \text{Dice} ; \text{mat}$$

```

1 let Dice =
2   let a = sample(unif(1,6)) in
3   let b = sample(unif(1,6)) in
4   assume(|a - b| == 1) ;
5   a+b
6 infer Dice

```

$$\frac{\text{Dice}}{k} \mid \mathbb{R}^+$$

$$\sum_x \mathbb{P}(a=x) \sum_y \mathbb{P}(a=y) \frac{\mathbb{1}_{(x,y)} \mathbb{1}_{(x,y)}}{|a-b|=1 \quad a+b=k}$$

Cannabis :

```

1 let Cannabis =
2   let b = sample(bernoulli 0.5) in
3   if b then let c = sample(bernoulli 0.6) in
4     assume c ; c
5   else true
6 infer Cannabis

```

Examples - Compute the semantics

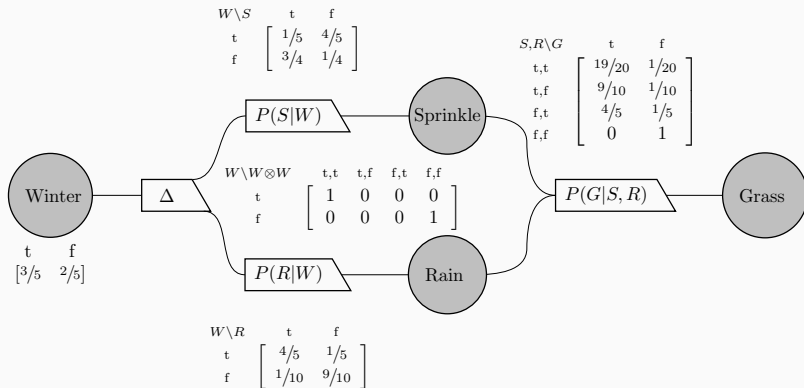
Funny Bernoulli :

```
1  let FB =  
2    let a = sample(bernoulli 0.5) in  
3    let b = sample(bernoulli 0.5) in  
4    let c = sample(bernoulli 0.5) in  
5    assume(a==0 or b==0) ;  
6    a+b+c  
7  infer FB
```

Semantics of Probabilistic Programming

Graphical Models

Bayesian Network - Example



Compute $\mathbb{P}(G)$ using :

Joint dist. $\mathbb{P}(G, R, S, W) = \mathbb{P}(G|S, R) \mathbb{P}(S|W) \mathbb{P}(R|W) \mathbb{P}(W)$

Marginal $\mathbb{P}(G) = \sum_{(r,s,w) \in |R| \times |S| \times |W|} \mathbb{P}(G, R, S, W)$

Bayesian Network

Definition

Directed Acyclic Graph with Conditional Probability Tables

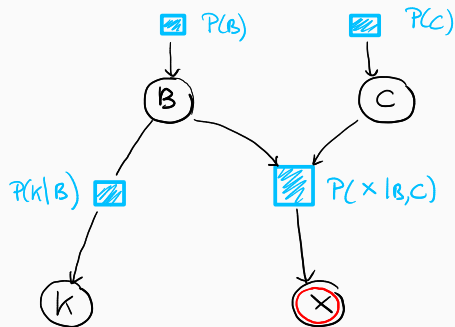
Factors are functions from states to \mathbb{R}^+

Computing

Exact inference by variable elimination

Implemented by message passing

Message Passing - Example



Joint distribution

$$P(X, K, B, C) = P(X|B, C)P(C)P(K|B)P(B)$$

Message Passing - Example

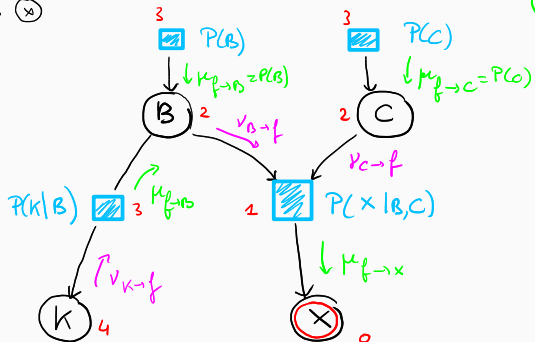
Factors \square

Variables \circ

① distance from root x

② Messages from factors

③ Messages from variables



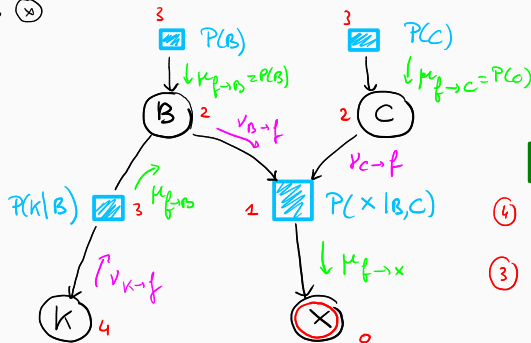
Joint distribution

$$P(x, k, b, c) = P(x|b, c) P(c) P(k|b) P(b)$$

Message Passing - Example

Factors (■)

Variables (○)



① distance from root X

② Messages from factors

③ Messages from variables

Compute From Leaves

④ $\psi_{K \rightarrow f} = 1$ (leaf)

③ $\psi_B = \sum_K P(B) \cdot P(K|B)$
 $\psi_C = P(C)$

② $\varphi = \psi_B \cdot \psi_C \cdot P(X|B,C)$

Joint distribution & Marginalisation

$$P(x, k, b, c) = P(x|b, c) P(c) P(k|b) P(b)$$

$$P(x) = \sum_{k, b, c} P(x, k, b, c)$$

① $\psi_x = \sum_{b, c} \varphi = \sum_{b, c} P(c) \cdot \sum_k P(k|b) P(k|b) P(x|b, c)$
 \downarrow
 $P(x)$

Message Passing - Algorithm

Initialisation of leaf nodes :

Messages from leaf node variable are set to 1

Messages from leaf node factors are set to the factor

Propagation

Starting from the further nodes

Variable to factor message : $\nu_{x \rightarrow f} = \prod_{g \in \text{ne}(x) \setminus f} \mu_{g \rightarrow x}$

Factor to variable message : $\mu_{f \rightarrow x} = \sum_{X_f \setminus x} \prod_{y \in \text{ne}(f) \setminus x} \nu_{y \rightarrow f}$

Implementation

Dag as an adjacent matrix

Message passing by memoisation

Numerical stability needs logs

Semantics of Probabilistic Programming

Verification of probabilistic Programs

Mc Iver, Morgan, Katoen

Hoare Logic - Principles

Language

$e ::= \text{true} \mid \text{false} \mid n \mid x \mid e \text{ op } e$ states $\Sigma : x \mapsto e$

$\text{op} ::= + \mid - \mid * \mid = \mid \leq \mid \text{and} \mid \text{not}$

$s ::= \text{skip} \mid x := e \mid s; s \mid \text{if } e \text{ then } s \text{ else } s \mid \text{while } e \text{ do } s$

Operational semantics

$$\overline{\Sigma, x := e \rightsquigarrow \Sigma\{x \leftarrow \text{ev}_\Sigma(e)\}, \text{skip}}$$

$$\frac{\Sigma, s_1 \rightsquigarrow \Sigma', s'_1}{\overline{\Sigma, (s_1; s_2) \rightsquigarrow \Sigma', (s'_1; s_2)}} \quad \frac{}{\overline{\Sigma, (\text{skip}; s) \rightsquigarrow \Sigma, s}}$$

$$\frac{\text{ev}_\Sigma(e) = \text{true}}{\overline{\Sigma, \text{if } e \text{ then } s_1 \text{ else } s_2 \rightsquigarrow \Sigma, s_1}} \quad \frac{\text{ev}_\Sigma(e) = \text{false}}{\overline{\Sigma, \text{if } e \text{ then } s_1 \text{ else } s_2 \rightsquigarrow \Sigma, s_2}}$$

$$\frac{\text{ev}_\Sigma(e) = \text{true}}{\overline{\Sigma, \text{while } e \text{ do } s \rightsquigarrow \Sigma, (s; \text{while } e \text{ do } s)}} \quad \frac{\text{ev}_\Sigma(e) = \text{false}}{\overline{\Sigma, \text{while } e \text{ do } s \rightsquigarrow \Sigma, \text{skip}}}$$

Hoare Logic - Rules

Hoare triple $\{P\}s\{Q\}$ is **valid** if for any state Σ and Σ' such that $\Sigma, s \rightsquigarrow \Sigma'$, skip, $[P]_{\Sigma}$ valid implies $[Q]_{\Sigma'}$ valid.

Rules

$$\begin{array}{c} \frac{}{\{P[x := E]\} x := E \{P\}} \quad \frac{\{P\} C \{Q\} \quad \{Q\} D \{R\}}{\{P\} C; D \{R\}} \\ \frac{\{E = \text{true} \wedge P\} C \{Q\} \quad \{E = \text{false} \wedge P\} D \{Q\}}{\{P\} \text{ if } E \text{ then } C \text{ else } D \{Q\}} \\ \frac{\{E = \text{true} \wedge I\} C \{I\}}{\{I\} \text{ while } E \text{ do } C \text{ done } \{E = \text{false} \wedge I\}} \\ \frac{P' \Rightarrow P \quad \{P\} C \{Q\} \quad Q \Rightarrow Q'}{\{P'\} C \{Q'\}} \end{array}$$

Soundness

If there is a proof tree with a hoare triple as root, then the program is correct with respect to pre and post conditions.

Weakest Precondition

Definition

Let Q be a predicate on the memory state and C be a program

$WP(C, \cdot)$ is a predicate transformer, that is

$WP(C, Q)$ is the **weakest precondition** P such that $\langle P \rangle C \langle Q \rangle$

Inductive definition

$$WP(x := E, Q(x)) = Q(E)$$

$$WP(S_1; S_2, Q) = WP(S_1, WP(S_2, Q))$$

$$WP(\text{if } E \text{ then } C \text{ else } D, Q) = (E \wedge WP(C, Q)) \vee (\neg E \wedge WP(D, Q))$$

$$WP(\text{while } E \text{ do } C \text{ done}\{\text{inv } I, \text{var } V\}, Q) = I$$

$$\text{with } \begin{cases} E = \text{true} \wedge I \wedge V = z \Rightarrow WP(C, I \wedge V < z) \\ I \Rightarrow V \geq 0 \\ (E = \text{false} \wedge I) \Rightarrow Q \end{cases}$$

Examples

$WP(x:=y+3, x > 3) = x > 3 [x := y+3] = y+3 > 3 = y > 0.$

$WP(n:=n+1, n > 4)$

$WP(y:=x+2; y:=y-2, y > 5)$

$WP(\text{if } x > 2 \text{ then } y:=1 \text{ else } y:=-1, y > 0)$

$WP(\text{while } n \neq 0 \text{ do } n:=n-1 \text{ done } \{I = n \geq 0, V = n\}, n = 0)$

Verifying Probabilistic Programs

Language

$e ::= \text{true} \mid \text{false} \mid n \mid x \mid e \text{ op } e \quad \text{states } \Sigma : e \rightarrow \text{val}$
 $\text{op} ::= + \mid - \mid * \mid = \mid \leq \mid \text{and} \mid \text{not}$
 $s ::= \text{skip} \mid x := e \mid s; s \mid \text{if } e \text{ then } s \text{ else } s \mid \text{while } e \text{ do } s \mid \text{bernoulli}$

Quantitative predicates : factors associate weights to states

$$\{f \mid f : |X| \rightarrow \mathbb{R}^+ \cup \{\infty\}\}$$

Weakest weighted precondition : $\text{WP}[P]$ transform factors to factors.

Weakest Pre-expectation

Weakest weighted precondition

if f associates scores to states, then $g = \text{WP}[P](f)$ maps a state s to the expected value of f after executing P on s . transform factors to factors.

Inductive definition

$$\text{WP}[x:=E](f) = f(x:=E)$$

$$\text{WP}[S_1; S_2] = \text{WP}[S_1](\text{WP}[S_2](f))$$

$$\text{WP}[\text{if } E \text{ then } C \text{ else } D](f) = [E] \cdot \text{WP}[C](f) + [\neg E] \cdot \text{WP}[D](f)$$

$$\text{WP}(\text{let } x = \text{sample bernoulli } p \text{ in } C)(f) =$$

$$p \cdot \text{WP}[C](f[x := \text{true}]) + (1 - p) \cdot \text{WP}[C](f[x := \text{false}])$$

$$\text{WP}[\text{while } E \text{ do } C \text{ done}](f) = \text{lfp } X \cdot [E] \cdot \text{WP}[C](X) + [\neg E] \cdot f$$

Soundness

if f associates scores to states, then

$$\text{WP}[P](f)(s) = \sum_{t \in |S|} \text{Proba}(s \xrightarrow{P} t) f(t).$$

Weakest Pre-expectation - Example

$$\begin{aligned} & \text{WP} \left[\begin{array}{l} a=1, b=1 \\ \text{while } (a=1 \text{ and } b=1) \text{ do } \text{let } a = \text{bern } 0,5 \text{ in } \text{let } b = \text{bern } 0,5 \text{ in } . a+b+c \\ \text{let } c = \text{bern } 0,5 \text{ in} \end{array} \right] (r=0) \\ & \text{WP} [\text{while } (y=1) \{ \text{let } a = \text{sample bern } 0,3 \text{ in if } a \text{ then } y := 0 \text{ else } c := c+1 \}] (f=1) \\ & \text{WP} [\text{let } a = \text{bern } 4/5 \text{ in if } a \text{ then } x := 5 \text{ else } x := 0] (x) \\ & \text{WP} [\text{let } a = \text{bern } 4/5 \text{ in if } a \text{ then } x := x+5 \text{ else } x := 0] (x) \\ & \text{WP} [\text{let } a = \text{bern } 4/5 \text{ in if } a \text{ then } x := x+5 \text{ else } x := 0] (x=10) \end{aligned}$$

Take home

Semantics for (discrete) probabilistic programs

Probabilistic coherent spaces

Bayesian Networks and message passing

Weakest Predicate Transformer for probabilistic programs

References

Semantics of probabilistic programs by Dexter Kozen

An Introduction to Probabilistic Programming. Jan-Willem van de Meent, Brooks Paige, Hongseok Yang and Frank Wood

Probabilistic Coherent Spaces by Vincent Danos and Thomas Ehrhard

Bayesian Reasoning and Machine Learning by David Barber

Deriving probabilistic semantics via the 'weakest completion' by McIver and Morgan

Probabilistic Weakest Precondition by Katoen & al.