Bluebell

An alliance of Relational Lifting and Independence for Probabilistic Reasoning

EMANUELE D'OSUALDO • Uni Konstanz
TIALU BAO • Cornell
AZADEH FARZAN • Uni Toronto

[DRAFT ON arXiv]
THE JOINT CONDITIONING-MODALITY
GOAL

Unify and generalize Proof principles for Unary & Relational Probabilistic Reasoning

Long Term:

Build an "Iris Core Logic" for Probabilistic Reasoning
Probabilistic Programs

We consider a simple programming language:

- Sequential & First Order
- Imperative with mutable variable store (no heap)
- Bounded Coops: everything terminates

Normal assignments $x := e$

Sampling assignments $x := \nu \mu$

\[
\textbf{Big Step Semantics}
\]

\[
\llbracket t \rrbracket : \mathbb{D}(\text{Store}) \rightarrow \mathbb{D}(\text{Store})
\]

Program term

$D(\text{Val}) = \text{Probability distribution over values}$
Probabilistic Programs

We consider a simple programming language:

- Sequential & First Order
- Imperative with mutable variable store (no heap)
- Bounded Coops: everything terminates
- Normal assignments $x := e$
- Sampling assignments $x := \nu_\text{ID}(\text{Val})$

Simple? Yes, but already hard enough to keep us busy for a while!
REASONING STYLES

UNARY

• Goal involves one program \( t \)
• Example properties:
  - Output distribution of \( x \) is \( \frac{1}{4} \)
  - Probability of \( x \geq 10 \) is \( \frac{1}{2} \)
  - Expected value of \( x \) is \( \frac{1}{3} \)
  - By the end, \( m \) and \( c \) are probabilistically independent
  - \( m \) could be a plaintext message
  - \( c \) its cyphertext

RELATIONAL

• Goal involves two programs \([t_1, t_2]\)
• Example properties:
  - \( t_1 \) and \( t_2 \) induce the same
distribution on \( x \)
  - \( t_2 \) could be an optimization of \( t_1 \)
  - \( t_1 \) could be a cryptographic protocol
  - \( t_2 \) its idealized perfect version
  - Starting from similar input,
  \( t_1 \) and \( t_2 \) will produce “similar”
distributions
  - \( t_2 \) differential privacy
// Encryption of 1 bit
\( k \approx \text{Ber}(\frac{1}{2}) \)  // New random key (1 bit)
\( m \approx \text{Ber}(\rho) \)  // Message to encrypt (arbitrary bias \( \rho \))
\( c := m \oplus k \)  // Compute cyphertext
// Encryption of 1 bit
k \sim \text{Ber}(1/2)
m \sim \text{Ber}(\rho)
c := m \oplus k
\{ c \sim \text{Ber}(1/2) \}

Reasoning (informally)

1. K and m are independent:
   \[ P(k=v, m=w) = P(k=v) \cdot P(m=w) \]

2. Conditioning on m:
   - if m=0 then c = k so c \sim \text{Ber}(1/2)
   - if m=1 then c = \overline{k} so c \sim \text{Ber}(1/2)

\[ c \sim \rho \cdot \text{Ber}(1/2) + (1-\rho) \cdot \text{Ber}(1/2) \]
\[ = \text{Ber}(1/2) \]
// Encryption of 1 bit

\[ k \sim \text{Ber}(\frac{1}{2}) \]
\[ m \sim \text{Ber}(\rho) \]
\[ c := m \oplus k \]
\[
\{ c \sim \text{Ber}(\frac{1}{2}) \} \\
\wedge c \text{ and } m \text{ are independent!}
\]

**Reasoning (informally)**

1. \( k \) and \( m \) are independent:
   \[ P(k=v, m=w) = P(k=v) \cdot P(m=w) \]
2. Conditioning on \( m \):
   - if \( m=0 \) then \( c = k \) so \( c \sim \text{Ber}(\frac{1}{2}) \)
   - if \( m=1 \) then \( c = \neg k \) so \( c \sim \text{Ber}(\frac{1}{2}) \)

\[ \implies c \sim p \cdot \text{Ber}(\frac{1}{2}) + (1-p) \text{Ber}(\frac{1}{2}) \]
\[ = \text{Ber}(\frac{1}{2}) \]
// Encryption of 1 bit
k \sim \text{Ber}(\frac{1}{2})
\{ k \sim \text{Ber}(\frac{1}{2}) \}
\begin{align*}
m \sim \text{Ber}(p) \\
\{ k \sim \text{Ber}(\frac{1}{2}) \} \land \{ m \sim \text{Ber}(p) \}
\end{align*}
c \coloneqq m \text{ XOR } k
\{ c \sim \text{Ber}(\frac{1}{2}) \} \land \{ m \sim \text{Ber}(p) \}

\text{Reasoning (informally)}
\begin{enumerate}
\item K and m are independent:
P(k=0, m=0) = P(k=0) \cdot P(m=0)
\item Conditioning on m:
\begin{align*}
&\text{if } m=0 \text{ then } c = k \text{ so } c \sim \text{Ber}(\frac{1}{2}) \\
&\text{if } m=1 \text{ then } c = \bar{k} \text{ so } c \sim \text{Ber}(\frac{1}{2})
\end{align*}
\Rightarrow c \sim p \cdot \text{Ber}(\frac{1}{2}) + (1-p) \cdot \text{Ber}(\frac{1}{2})
\begin{align*}
&= \text{Ber}(\frac{1}{2})
\end{align*}
\end{enumerate}

\text{IDEA (2)} : \text{Separation } = \text{Independence} \ [\text{PSL}][\text{LILAC}]
UNARY EXAMPLE

// Encryption of 1 bit
k \sim \text{Ber}(1/2)
\{ k \sim \text{Ber}(1/2) \}
\{ k \sim \text{Ber}(1/2) \} \ast m \sim \text{Ber}(p)
\{ k \sim \text{Ber}(1/2) \} \ast m \sim \text{Ber}(p)
\{ c \sim \text{Ber}(1/2) \} \ast m \sim \text{Ber}(p)
\{ c \sim \text{Ber}(1/2) \} \ast m \sim \text{Ber}(p)

Reasoning (informally)

(2) Conditioning on m:
\{ \{ k \sim \text{Ber}(1/2) \} \ast \{ c = k \text{? if } v = 0 \} \}
\uparrow
\{ c \sim \text{Ber}(p) \}
\{ c = k \} \text{ if } v = 1
\text{Deterministic value}
\text{Predicate over stores holds with probability 1}

IDEA① : Separation = Independence [P3L] [LILAC]
IDEA② : Conditioning via a modality [LILAC]
REASONING TOOLS

- UNARY TRIPLES: \{P\} \rightarrow \{Q\}  Assertions over ID(Store)
- PROBABILISTIC INDEPENDENCE: Separation *
- CONDITIONING: via a modality \Box_x [\cdot]
RELATIONAL REASONING

1: $x \sim \mu$
   $d \sim \text{unif}(-1,1)$
   $y := x - d$

2: $x \sim \mu$
   $d \sim \text{unif}(-1,1)$
   $y := x + d$

GOAL: $y^{(1)}$ is distributed like $y^{(2)}$

UNARY PROOF STRATEGY: Characterize the exact distribution of $y$
   in the two programs, then compare.

\[ \rightarrow \] Can be prohibitively hard to do!

RELATIONAL STRATEGY: Execute programs in lockstep showing that
   whatever the steps might be computing, the two sides remain the same.
RELATIONAL REASONING

1: \( x \sim \mu \)
\( d \sim \text{unif}(-1,1) \)
\( y := x - d \)

2: \( x \sim \mu \)
\( d \sim \text{unif}(-1,1) \)
\( y := x + d \)

A world of pure imagination

GOAL: \( y^{(1)} \) is distributed like \( y^{(2)} \)
RELATIONAL REASONING

1: \( x := a \)
   \( d \sim \text{unif}(-1,1) \)
   \( y := x - d \)

2: \( x := a \)
   \( d \sim \text{unif}(-1,1) \)
   \( y := x + d \)

GOAL: \( y^{(1)} \) is distributed like \( y^{(2)} \)
RELATIONAL REASONING

1: \( x := a \)

\( d := b \)

\( y := x - d \)

2: \( x := a \)

\( d := -b \)

\( y := x + d \)

**Goal:** \( y^{(1)} \) is distributed like \( y^{(2)} \)
RELATIONAL REASONING

1: $x \sim \mu$
   $d \sim \text{unif}(-1, 1)$
   $y := x - d$

2: $x \sim \mu$
   $d \sim \text{unif}(-1, 1)$
   $y := x + d$

$\lbrack \text{pRHL} \rbrack$

Relation over Store x Store
Holding with probability 1
in some "fictional" joint distribution
RELATIONAL REASONING

1: $x \sim \mu$
   
   $d \sim \text{unif}(-1, 1)$
   
   $y := x - d$

2: $x \sim \mu$
   
   $d \sim \text{unif}(-1, 1)$
   
   $d \leftarrow -d$

$y := x + d$

$\uparrow$

Relation over Store \times Store

Holding with probability 1 in some "fictional" joint distribution

$[pRHL]$ = Relational Lifting $[LR]$
RELATIONAL REASONING

1: \[ x \sim \mu \]
\[ d \sim \text{unif}(-1,1) \]
\[ y := x - d \]

2: \[ x \sim \mu \]
\[ d \sim \text{unif}(-1,1) \]
\[ y := x + d \]

\[ [p_{\text{RHL}}] \]

Relation over Store x Store
Holding with probability 1
in some "fictional" joint distribution

= Relational lifting \[ LR \]

FUNDAMENTAL THEOREM OF RELATIONAL LIFTING: (Meta)

If \[ y^{(1)} = y^{(2)} \] then \[ y^{(1)} \] is distributed like \[ y^{(2)} \]
RELATIONAL REASONING (LIMITATIONS)

1. $x \sim \mu$  
   $d \sim \text{unif}(-1, 1)$  
   $y := x - d$

2. $d \sim \text{unif}(-1, 1)$  
   $x \sim \mu$
   $y := x + d$

Only asserting via Relational lifting is too limiting!

**Goal:** Improve expressivity while retaining the relational "spirit"
**REASONING TOOLS**

- **UNARY TRIPLES**: \( \{ p \} \triangleright \{ q \} \)  
  *Assertions over D(Store)*

- **PROBABILISTIC INDEPENDENCE**: Separation *

- **CONDITIONING**: via a modality \( x \mapsto v \)
REASONING TOOLS

- UNARY TRIPLES: \( \{ p \} t \{ q \} \) Assertions over ID(Store)

- PROBABILISTIC INDEPENDENCE: Separation

- CONDITIONING: via a modality \( \Box \)

- RELATIONAL TRIPLES: \( [1:t_1, 2:t_2] \) \( LR_1 \) \( LR_2 \) Relations over Store

\( R_1, R_2 \subseteq \text{Store} \times \text{Store} \)
REASONING TOOLS

- **UNARY TRIPLES**: $\{ p \} \triangleleft \{ q \}$  
  **Assertions over $D(\text{Store})$**

- **PROBABILISTIC INDEPENDENCE**: Separation *

- **CONDITIONING**: via a modality $\Box_x$

- **RELATIONAL TRIPLES**: $\left[ R_1 \right] [1:t_1, 2:t_2] \left[ R_2 \right]$  
  **Relations over $\text{Store}$**
  $R_1, R_2 \subseteq \text{Store} \times \text{Store}$

- **RELATIONAL LIFTING**: $\left[ L \right]$
REASONING TOOLS

- **UNARY TRIPLES**: \( \{ \mathcal{P}_i \} \triangleright \{ \mathcal{Q}_j \} \)  
  Assertions over \( \mathcal{D}(\text{Store}) \)

- **PROBABILISTIC INDEPENDENCE**: Separation \( \ast \)

- **CONDITIONING**: via a modality \( \triangleright \)

- **RELATIONAL TRIPLES**: \( [\mathcal{L}_1 : [1 : \mathcal{t}_1, 2 : \mathcal{t}_2] \mathcal{L}_2] \)  
  Relations over \( \text{Store} \)
  \( \mathcal{R}_1, \mathcal{R}_2 \subseteq \text{Store} \times \text{Store} \)

- **RELATIONAL LIFTING**: \( \mathcal{L} \)

---

Can we unify and generalize?

*spoiler: YES*
BLUEBELL

First observation: We can harmonize all these features by:

- Using $\text{Assrt} := \text{ID}(\text{Store}) \times \text{ID}(\text{Store}) \rightarrow \text{Prop}$
  - Unary assertions just ignore one of the two distributions $\times \langle 1 \rangle \sim \text{Ber}(\frac{1}{2})$
  - Relational lifting as a construct
    \[ R \subseteq \text{Store} \times \text{Store} \Rightarrow [R] : \text{ID}(\text{Store}) \times \text{ID}(\text{Store}) \rightarrow \text{Prop} \]

- Multi-ary $\text{wp from}[\text{LHC}] : \forall t \in \{Q\} \rightarrow \text{partial map Indices} \rightarrow \text{Terms}$

\[ \text{wp} [1 : t_1, 2 : t_2] \{Q\} \equiv \text{wp} [1 : t_1] \{\text{wp}[2 : t_2] \{Q\}\} \]

Can have unary triples, binary triples, switch back & forth.
SMALL EXAMPLE

1: \( x \sim \mu \)
   \( d \sim \text{unif}(-1,1) \)
   \( y := x - d \)

2: \( d \sim \text{unif}(-1,1) \)
   \( x \sim \mu \)
   \( y := x + d \)
**Small Example**

1: \( x \sim \mu \)
   \( d \sim \text{unif}(-1,1) \)
   \( x \sim \mu \)

\[ \exists x \langle 1 \rangle \sim \mu * x \langle 2 \rangle \sim \mu * a \langle 1 \rangle \sim \text{unif}(1,1) * a \langle 2 \rangle \sim \text{unif}(-1,1) \]

\[ \exists y \langle 1 \rangle = x \langle 2 \rangle \wedge d \langle 1 \rangle = -d \langle 2 \rangle \]

\[ y := x - d \]

\[ \exists y \langle 1 \rangle = y \langle 2 \rangle \]

**Questions:**

1) Can entailment \( \otimes \) be proven in the logic?
2) Are there useful interactions between \( \ast \), \( \Theta \), and \( LR \)?
BLUEBELL'S KEY INSIGHT

Questions:
1) Can entailment be proven in the logic?
2) Are there useful interactions between $\$, $\cdot$ and $\mathrm{LR}$?
BLUEBELL'S KEY INSIGHT

QUESTIONS:
1) Can entailment be proven in the logic?
2) Are there useful interactions between *, I and LR?

Bluebell says YES!

UNARY CONDITIONING

RELATIONAL LIFTING

THE JOINT CONDITIONING-

MODELL AND *
BLUEBELL'S KEY INSIGHT

**Questions:**

1. Can entailment be proven in the logic?
2. Are there useful interactions between $\star$, $\cdot$, and $LR$?

Bluebell says YES!

Their definitions and laws can be derived.

Rich set of core laws

THE JOINT CONDITIONING MODALITY AND $\star$
RELATIONAL LIFTING AS CONDITIONING

Usual picture:
\[ \text{D(Store)} \times \text{D(Store)} \xrightarrow{\text{Lift}} \text{LR} \]
\[ \text{Store} \times \text{Store} \xrightarrow{\text{Lift}} \text{R} \]

Bluebell’s view:
\[ \text{D(Store)} \times \text{D(Store)} \xrightarrow{\text{Conditioning}} \text{LR} \]
\[ \text{Store} \times \text{Store} \xrightarrow{\text{Conditioning}} \text{R} \]

(Analog of)
\[ \text{D(Store)} \xrightarrow{\text{IP}(4) = 1} \text{Store} \xrightarrow{\text{A}} \]

If you condition jointly on the two distributions, you get a pair of stores satisfying R.

So, what is “joint conditioning”?
**Joint Conditioning**

**Def** Given \( M : \mathbb{D}(A) \) and \( k : A \rightarrow \mathbb{D}(\text{Store}) \) define

\[
\text{bind}(\mu, k) := \lambda s. \sum_{a \in A} \mu(a) k(a)(s)
\]

**Example** \( A = \{0, 1\} \) \( \mu = \text{Ber}(\frac{1}{3}) \)

\[
\text{bind}(\mu, k) = \frac{1}{3} k(0) + \frac{2}{3} k(1)
\]

[This is actually the bind of the monad \( \mathbb{D}(\cdot) \)!

]
**Joint Conditioning**

**Def.** Given $\mu : D(A)$ and $P : A \to \text{Assrt}$

define $C_\mu v. P(v) : \text{Assrt}$ by

\[
(\mu_1, \mu_2) \vdash C_\mu v. P(v) \quad \iff \quad \exists K_1, K_2 : A \to D(\text{store}).
\]

\[
\quad \quad \quad M_1 = \text{bind}(\mu_1, K_1) \land
\]

\[
\quad \quad \quad M_2 = \text{bind}(\mu_1, K_2) \land
\]

\[
\quad \quad \quad \forall a \in \text{supp}(\mu).
\]

\[
\quad \quad \quad (k_1(a), k_2(a)) \vdash P(a)
\]
JOINT CONDITIONING

$(M_1, M_2) \models C \iff \exists K_1, K_2 : A \rightarrow D(\text{store}) .
\begin{align*}
M_1 &= \text{bind}(M_1, K_1) \\
M_2 &= \text{bind}(M_1, K_2) \\
\forall a \in \text{supp}(M) . \quad (K_1(a), K_2(a)) \models P(a)
\end{align*}
**Joint Conditioning**

\[ (M_1, M_2) \models C \text{ if } P(v) \text{ iff } \exists K_1, K_2 : A \rightarrow D(\text{store}). \]
\[ M_1 = \text{bind}(M, K_1) \land M_2 = \text{bind}(M, K_2) \land \forall a \in \text{supp}(M). \]
\[ (K_1(a), K_2(a)) \models P(a) \]

**Example:**
\[ A = \{0, 1, 2\} \quad M = \text{Ber}(\frac{1}{3}) \]

\[ M_1 = \frac{1}{3} K_1(0) + \frac{2}{3} K_1(1) \]
\[ M_2 = \frac{1}{3} K_2(0) + \frac{2}{3} K_2(1) \]

\[ P(0) \quad P(1) \]

\[ x \leftarrow \text{Ber}(\frac{1}{3}) \quad \Gamma[x] = 0 \quad \Gamma[x] = 1 \]
J
O
I
T
C
O
N
D
I
T
I
O
N
I
G

\((M_1, M_2) = C_M \triangleright P(\alpha) \iff \exists K_1, K_2 : A \rightarrow \text{D(store)}.\)

\begin{align*}
M_1 &= \text{bind}(M_1, K_1) \land \\
M_2 &= \text{bind}(M_1, K_2) \land \\
\forall \text{assmp}(\mu), \\
(K_1(\alpha), K_2(\alpha)) &\vdash P(\alpha)
\end{align*}

Example: \(A = \{0, 1, 2\}, M = \text{Ber}(\frac{1}{3})\)

\begin{align*}
M_1 &= \frac{1}{3} K_1(0) + \frac{2}{3} K_1(1) \\
M_2 &= \frac{1}{3} K_2(0) + \frac{2}{3} K_2(1)
\end{align*}

\begin{align*}
P(0) &\quad P(1)
\end{align*}

\begin{align*}
\Gamma x\triangleleft \text{Ber}(\frac{1}{3}) &\quad \Gamma x\triangleleft = 0 \quad \Gamma x\triangleleft = 1
\end{align*}

\(C_M \triangleright (\Gamma x\triangleleft = \alpha \neq P(\alpha))\)

\([C\_\text{UNIT-R}] \quad X\triangleleft \sim M \quad \vdash C_M \triangleright [X\triangleleft = \alpha]\)

[This reflects the right unit law of the underlying monad!]

\([C\_\text{UNIT-R}] \quad X\triangleleft \sim M \quad \vdash C_M \triangleright [X\triangleleft = \alpha]\)
**Encoding Lifting as Conditioning**

**Unary Conditioning:** \[ C_x v \cdot (\exists x = v \cdot P(v)) \]

**Relational Lifting:**

\[ \mathbf{LR}(\langle x_1 \rangle, \langle x_2 \rangle) \] :=

\[ \exists f : \mathcal{D}(\text{Val} \times \text{Val}). \ C_x (v_1, v_2). \ (\exists x_1 = v_1 \land \exists x_2 = v_2 \land R(v_1, v_2)) \]

(pure)
JOINT COND. RULES

[C-UNIT-R]
\[ x \langle i \rangle \sim \mu \vdash C_\mu \nu. [x \langle i \rangle = \nu] \]

[C-FRAME]
\[ P \ast C_\mu \nu. Q(\nu) \vdash C_\mu \nu. (P \ast Q(\nu)) \]

[C-CONS]
\[ \forall u \in supp(\mu). P(u) \vdash P'(u) \]
\[ C_\mu \nu. P(\nu) \vdash C_\mu \nu. P'(\nu) \]

\[ x \langle i \rangle \sim \mu \ast y \langle i \rangle \sim \mu' \ast [z = x + y] \]
\[ \vdash (C_\mu \nu. [x \langle i \rangle = \nu]) \ast y \langle i \rangle \sim \mu' \ast [z = x + y] \]
\[ \vdash C_\mu \nu. (x \langle i \rangle = \nu \ast y \langle i \rangle \sim \mu' \ast [z = x + y]) \]
\[ \vdash C_\mu \nu. (C_{\mu_1} \nu_1. (x \langle i \rangle = \nu \ast y \langle i \rangle = \nu' \ast [z = x + y])) \]
\[ \vdash C_\mu \nu. C_{\mu_1} \nu_1. [x \langle i \rangle = \nu \land y \langle i \rangle = \nu' \land z = x + y] \]
[c- Assoc]

\[ C_\mu \nu \cdot C_{k(\nu)} \nu' \cdot P(\nu, \nu') \vdash C_{\text{bind}^1(\mu, k)} \cdot P(\nu, \nu') \]

\[ \text{bind}^1(\mu, k) = \text{do } \nu \leftarrow \mu \; \text{j} \; \nu' \leftarrow k(\nu) \; \text{j} \; \text{return } (\nu, \nu') \]

[c- unassoc]

\[ C_{\text{bind}(\mu, k)} \nu' P(\nu') \vdash C_\mu \nu \cdot C_{k(\nu)} \nu' P(\nu') \]
Some Derivable Rules

\( C_n \models [R] \vdash [LR] \quad \text{(Convexity of Rel Liftings)} \)

\[ [R_1] \ast [R_2] \vdash [R_1 \land R_2] \]

Note:
\[ [R_1] \land [R_2] \not\vdash [R_1 \land R_2] \]
CHALLENGES

- Generalization to Iris-style user-defined ghost resources
- \([c-\text{wp}\text{-swap}]\)

\[
\text{ownVars} \land C_{\mu.\nu}. \text{wp t } \{ Q(u) \} \vdash \text{wp t } \{ C_{\mu.\nu}. Q(u) \}
\]

\[\uparrow\]
Bluebell needs this for soundness

OPEN QUESTION: Can we find a model that validates the rule without ownVars?
Thanks