Bluebell

An alliance of Relational Lifting and Independence for Probabilistic Reasoning

EMANUELE D'OSUALDO · Uni Konstanz
JIALU 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 loops: everything terminates
- Normal assignments $x := e$

  Sampling assignments $x :\approx \mu$

  $\mathbb{D}(Val)$ = Probability distribution over values

BIG STEP SEMANTICS

$$[\![t]\!] : \mathbb{D}(Store) \longrightarrow \mathbb{D}(Store)$$

Program term

PROBABILISTIC PROGRAMS

We consider a simple programming language:

- Sequential & First Order
- Imperative with mutable variable store (no heap)
- Bounded loops: everything terminates
- Normal assignments $x := e$

  Sampling assignments $x :\approx \mu$

  $\mathbb{D}(Val)$ = Probability distribution over values

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 μ
  - Probability of $x \geq 10$ is $1/2$
  - Expected value of x is $1/3$
  - By the end, m and c are probabilistically independent

    m could be a plaintext message
    c its cyphertext

Relational

- Goal involves two programs $[1{:}t_1, 2{:}t_2]$
- 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 and t_2 its idealized perfect version
  - Starting from similar input, t_1 and t_2 will produce "similar" distributions

    ↳ differential privacy

UNARY EXAMPLE

// Encryption of 1 bit

$k :\approx \mathrm{Ber}(1/2)$ // New random key (1 bit)

$m :\approx \mathrm{Ber}(p)$ // Message to encrypt (arbitrary bias p)

$c := m \text{ XOR } k$ // Compute cyphertext

UNARY EXAMPLE

// Encryption of 1 bit

$k :\approx \mathrm{Ber}(1/2)$

$m :\approx \mathrm{Ber}(p)$

$c := m \text{ XOR } k$

$\{c \sim \mathrm{Ber}(1/2)\}$

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 $c \sim \mathrm{Ber}(1/2)$
- if $m=1$ then $c=\neg k$ so $c \sim \mathrm{Ber}(1/2)$

$$\Rightarrow c \sim p \cdot \mathrm{Ber}(1/2) + (1-p)\,\mathrm{Ber}(1/2)$$
$$= \mathrm{Ber}(1/2)$$

UNARY EXAMPLE

// Encryption of 1 bit

$k :\approx \mathrm{Ber}(1/2)$

$m :\approx \mathrm{Ber}(p)$

$c := m \text{ XOR } k$

$\{c \sim \mathrm{Ber}(1/2)\}$

$\wedge$ c and m 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 $c \sim \mathrm{Ber}(1/2)$
- if $m=1$ then $c = \neg k$ so $c \sim \mathrm{Ber}(1/2)$

$$\Rightarrow c \sim p \cdot \mathrm{Ber}(1/2) + (1-p)\,\mathrm{Ber}(1/2)$$
$$= \mathrm{Ber}(1/2)$$

UNARY EXAMPLE

// Encryption of 1 bit

$k :\approx \mathrm{Ber}(1/2)$

$\{k \sim \mathrm{Ber}(1/2)\}$

$m :\approx \mathrm{Ber}(p)$

$\{k \sim \mathrm{Ber}(1/2) * m \sim \mathrm{Ber}(p)\}$

$c := m \text{ XOR } k$

$\{c \sim \mathrm{Ber}(1/2) * m \sim \mathrm{Ber}(p)\}$

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 $c \sim \mathrm{Ber}(1/2)$
   - if $m=1$ then $c = \neg k$ so $c \sim \mathrm{Ber}(1/2)$

   $\implies c \sim p \cdot \mathrm{Ber}(1/2) + (1-p)\,\mathrm{Ber}(1/2)$

   $= \mathrm{Ber}(1/2)$

IDEA 1 : Separation = Independence [PSL] [LILAC]

UNARY EXAMPLE

// Encryption of 1 bit

$K :\approx \mathrm{Ber}(1/2)$

$\{K \sim \mathrm{Ber}(1/2)\}$

$m :\approx \mathrm{Ber}(p)$

$\{K \sim \mathrm{Ber}(1/2) * m \sim \mathrm{Ber}(p)\}$

$c := m \text{ XOR } K$

$\{c \sim \mathrm{Ber}(1/2) * m \sim \mathrm{Ber}(p)\}$

Reasoning (informally)

② Conditioning on m:

case analysis

$$\underset{\substack{m \to v \\ \uparrow}}{\mathbb{C}} \left(K \sim \mathrm{Ber}(1/2) * \begin{cases} \lceil c = K \rceil & \text{if } v = 0 \\ \underbrace{\lceil c = \neg K \rceil}_{} & \text{if } v = 1 \end{cases} \right)$$

Deterministic value

Predicate over stores holds with probability 1

IDEA ①: Separation = Independence [PSL] [LILAC]

IDEA ②: Conditioning via a modality [LILAC]

REASONING TOOLS

- UNARY TRIPLES: $\{P\}\ t\ \{Q\}$ Assertions over $\mathbb{D}$(Store)
- PROBABILISTIC INDEPENDENCE: Separation $*$
- CONDITIONING: via a modality $\mathbb{C}_{X \to v}$

RELATIONAL REASONING

1: $x :\approx \mu$	2: $x :\approx \mu$
$d :\approx unif(-1,1)$	$d :\approx unif(-1,1)$
$y := x - d$	$y := x + d$

GOAL: $y\langle 1 \rangle$ is distributed like $y\langle 2 \rangle$

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

↳ Can be prohibitively hard to do!

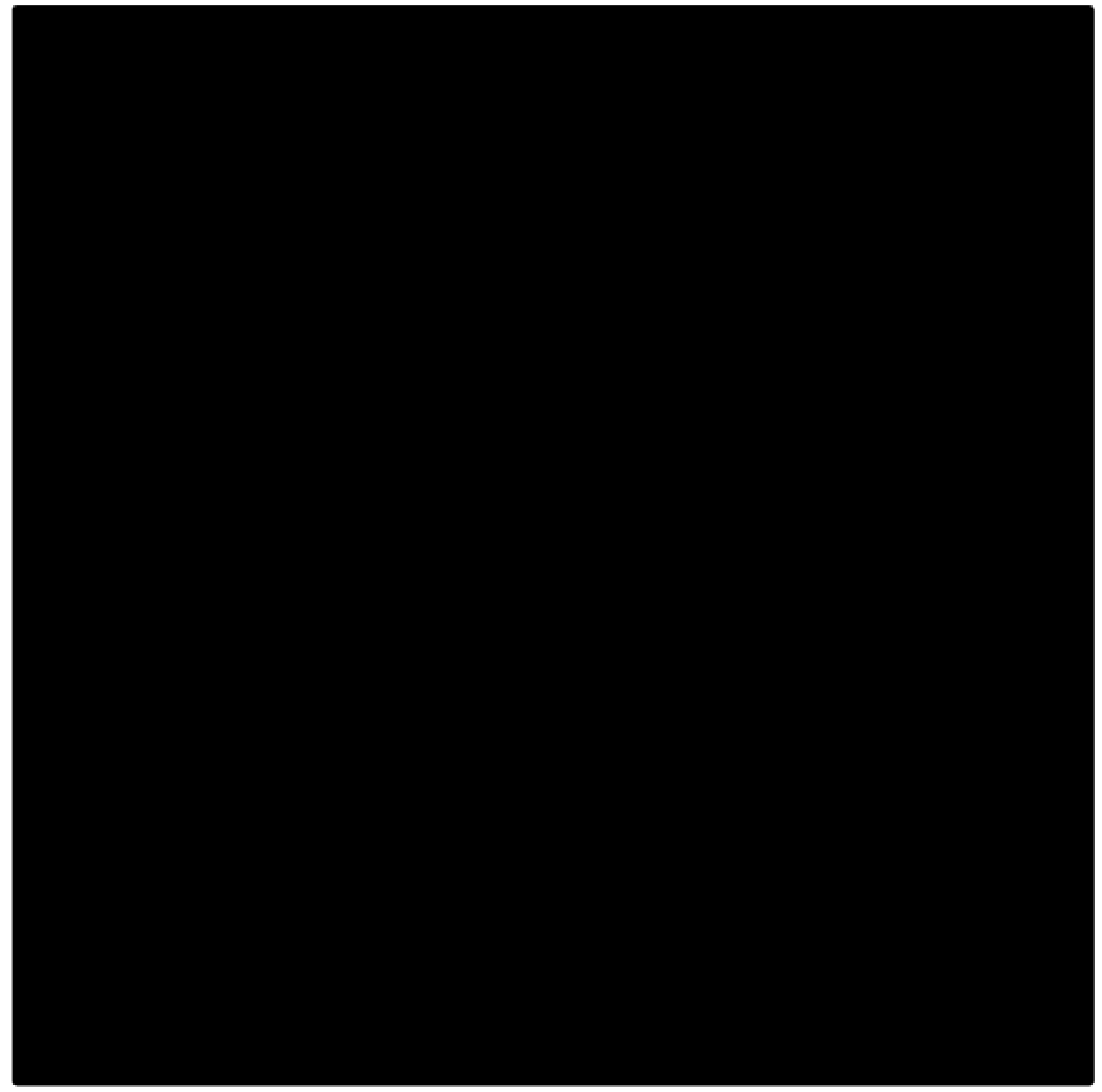
RELATIONAL STRATEGY: Execute programs in lockstep showing that whatever the steps might be computing, the two sides remain the same

RELATIONAL REASONING

1: $x :\approx \mu$

$d :\approx \mathrm{unif}(-1,1)$

$y := x - d$



A world of pure imagination

2: $x :\approx \mu$

$d :\approx \mathrm{unif}(-1,1)$

$y := x + d$

GOAL: $y\langle 1\rangle$ is distributed like $y\langle 2\rangle$

RELATIONAL REASONING

1: $x := a$

$d :\approx \mathrm{unif}(-1,1)$

$y := x - d$

$a \sim \mu$

"coupling"

2: $x := a$

$d :\approx \mathrm{unif}(-1,1)$

$y := x + d$

GOAL: $y\langle 1\rangle$ is distributed like $y\langle 2\rangle$

RELATIONAL REASONING

1: $x := a$

$d := b$

$y := x - d$

$a \sim \mu$

$b \sim \mathrm{unif}(-1, 1)$

2: $x := a$

$d := -b$

$y := x + d$

GOAL: $y\langle 1\rangle$ is distributed like $y\langle 2\rangle$

RELATIONAL REASONING

1:		2:
$x :\approx \mu$		$x :\approx \mu$
	$\lfloor x\langle 1\rangle = x\langle 2\rangle \rfloor$	
$d :\approx \mathrm{unif}(-1,1)$		$d :\approx \mathrm{unif}(-1,1)$
	$\left\lfloor \begin{array}{l} x\langle 1\rangle = x\langle 2\rangle \\ d\langle 1\rangle = -\, d\langle 2\rangle \end{array} \right\rfloor$	
$y := x - d$		$y := x + d$
	$\lfloor y\langle 1\rangle = y\langle 2\rangle \rfloor$	

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

[pRHL]

RELATIONAL REASONING

1: $x :\approx \mu$

$\lfloor x\langle 1\rangle = x\langle 2\rangle \rfloor$

2: $x :\approx \mu$

$d :\approx \mathrm{unif}(-1,1)$

$\left\lfloor \begin{array}{l} x\langle 1\rangle = x\langle 2\rangle \\ d\langle 1\rangle = -\,d\langle 2\rangle \end{array} \right\rfloor$

$d :\approx \mathrm{unif}(-1,1)$

$y := x - d$

$y := x + d$

$\lfloor y\langle 1\rangle = y\langle 2\rangle \rfloor$

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

[pRHL]

= Relational lifting $\lfloor R \rfloor$

RELATIONAL REASONING

1: $x :\approx \mu$ ———— $\lfloor x\langle 1\rangle = x\langle 2\rangle \rfloor$ ———— 2: $x :\approx \mu$

$d :\approx \mathrm{unif}(-1,1)$ ———— $\left\lfloor \begin{array}{l} x\langle 1\rangle = x\langle 2\rangle \\ d\langle 1\rangle = -d\langle 2\rangle \end{array} \right\rfloor$ ———— $d :\approx \mathrm{unif}(-1,1)$

$y := x - d$ ———— $\lfloor y\langle 1\rangle = y\langle 2\rangle \rfloor$ ———— $y := x + d$

[pRHL]

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

= Relational lifting $\lfloor R \rfloor$

FUNDAMENTAL THEOREM OF RELATIONAL LIFTING: (Meta)

If $\lfloor y\langle 1\rangle = y\langle 2\rangle \rfloor$ then $y\langle 1\rangle$ is distributed like $y\langle 2\rangle$

RELATIONAL REASONING (LIMITATIONS)

1: $x :\approx \mu$

$d :\approx \mathrm{unif}(-1,1)$

$y := x - d$

$\lfloor$???? $\rfloor$

2: $d :\approx \mathrm{unif}(-1,1)$

$x :\approx \mu$

$y := x + d$

Only asserting via Relational Lifting is too limiting!

GOAL: Improve expressivity while retaining the relational "spirit"

REASONING TOOLS

- UNARY TRIPLES: $\{P\}\ t\ \{Q\}$ Assertions over $\mathbb{D}(\mathsf{Store})$
- PROBABILISTIC INDEPENDENCE: Separation $*$
- CONDITIONING: via a modality $\underset{X \to v}{\mathbb{C}}$

REASONING TOOLS

- UNARY TRIPLES: $\{P\}\ t\ \{Q\}$ Assertions over $\mathbb{D}(\text{Store})$
- PROBABILISTIC INDEPENDENCE: Separation $*$
- CONDITIONING: via a modality $\mathbb{C}_{x \to v}$
- RELATIONAL TRIPLES: $\lfloor R_1 \rfloor\ [1{:}t_1, 2{:}t_2]\ \lfloor R_2 \rfloor$ Relations over Store
  $R_1, R_2 \subseteq \text{Store} \times \text{Store}$

REASONING TOOLS

- UNARY TRIPLES: $\{P\}\ t\ \{Q\}$ Assertions over $\mathbb{D}(\text{Store})$
- PROBABILISTIC INDEPENDENCE: Separation $*$
- CONDITIONING: via a modality $\underset{X \to v}{\mathbb{C}}$
- RELATIONAL TRIPLES: $\lfloor R_1 \rfloor\ [1{:}t_1, 2{:}t_2]\ \lfloor R_2 \rfloor$ Relations over Store $R_1, R_2 \subseteq \text{Store} \times \text{Store}$
- RELATIONAL LIFTING: $\lfloor R \rfloor$

REASONING TOOLS

- UNARY TRIPLES: $\{P\}\ t\ \{Q\}$ Assertions over $\mathbb{D}(\mathrm{Store})$
- PROBABILISTIC INDEPENDENCE: Separation $*$
- CONDITIONING: via a modality $\underset{X \leftarrow v}{\mathbb{C}}$
- RELATIONAL TRIPLES: $\lfloor R_1 \rfloor\ [1{:}t_1, 2{:}t_2]\ \lfloor R_2 \rfloor$ Relations over Store $R_1, R_2 \subseteq \mathrm{Store} \times \mathrm{Store}$
- RELATIONAL LIFTING: $\lfloor R \rfloor$

Can we unify and generalize?

(spoiler: YES)

BLUEBELL

First observation We can harmonize all these features by:

- Using $\mathrm{Assrt} := \mathbb{D}(\mathrm{Store}) \times \mathbb{D}(\mathrm{Store}) \to \mathrm{Prop}$
  - Unary assertions just ignore one of the two distributions $x\langle 1\rangle \sim \mathrm{Ber}(1/2)$
  - Relational lifting as a construct

$$R \subseteq \mathrm{Store} \times \mathrm{Store} \Rightarrow \lfloor R \rfloor : \mathbb{D}(\mathrm{Store}) \times \mathbb{D}(\mathrm{Store}) \to \mathrm{Prop}$$

- Multi-ary wp from [LHC]: $\mathrm{wp}\ \mathbf{t}\ \{Q\}$ partial map Indices → Terms

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

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

SMALL EXAMPLE

1: $x :\approx \mu$

$d :\approx \mathrm{unif}(-1,1)$

$y := x - d$

2: $d :\approx \mathrm{unif}(-1,1)$

$x :\approx \mu$

$y := x + d$

SMALL EXAMPLE

1: $x :\approx \mu$ 2: $d :\approx \mathrm{unif}(-1,1)$

$d :\approx \mathrm{unif}(-1,1)$ $x :\approx \mu$

$$\{x\langle 1\rangle \sim \mu * x\langle 2\rangle \sim \mu * d\langle 1\rangle \sim \mathrm{unif}(-1,1) * d\langle 2\rangle \sim \mathrm{unif}(-1,1)\} \circledast$$

$$\{ \lfloor x\langle 1\rangle = x\langle 2\rangle \wedge d\langle 1\rangle = -d\langle 2\rangle \rfloor \}$$

$y := x - d$ $y := x + d$

$$\{ \lfloor y\langle 1\rangle = y\langle 2\rangle \rfloor \}$$

QUESTIONS :

1) Can entailment $\circledast$ be proven in the logic?
2) Are there useful interactions between $*$, $\mathbb{C}$ and $\lfloor R \rfloor$?

BLUEBELL'S KEY INSIGHT

QUESTIONS :

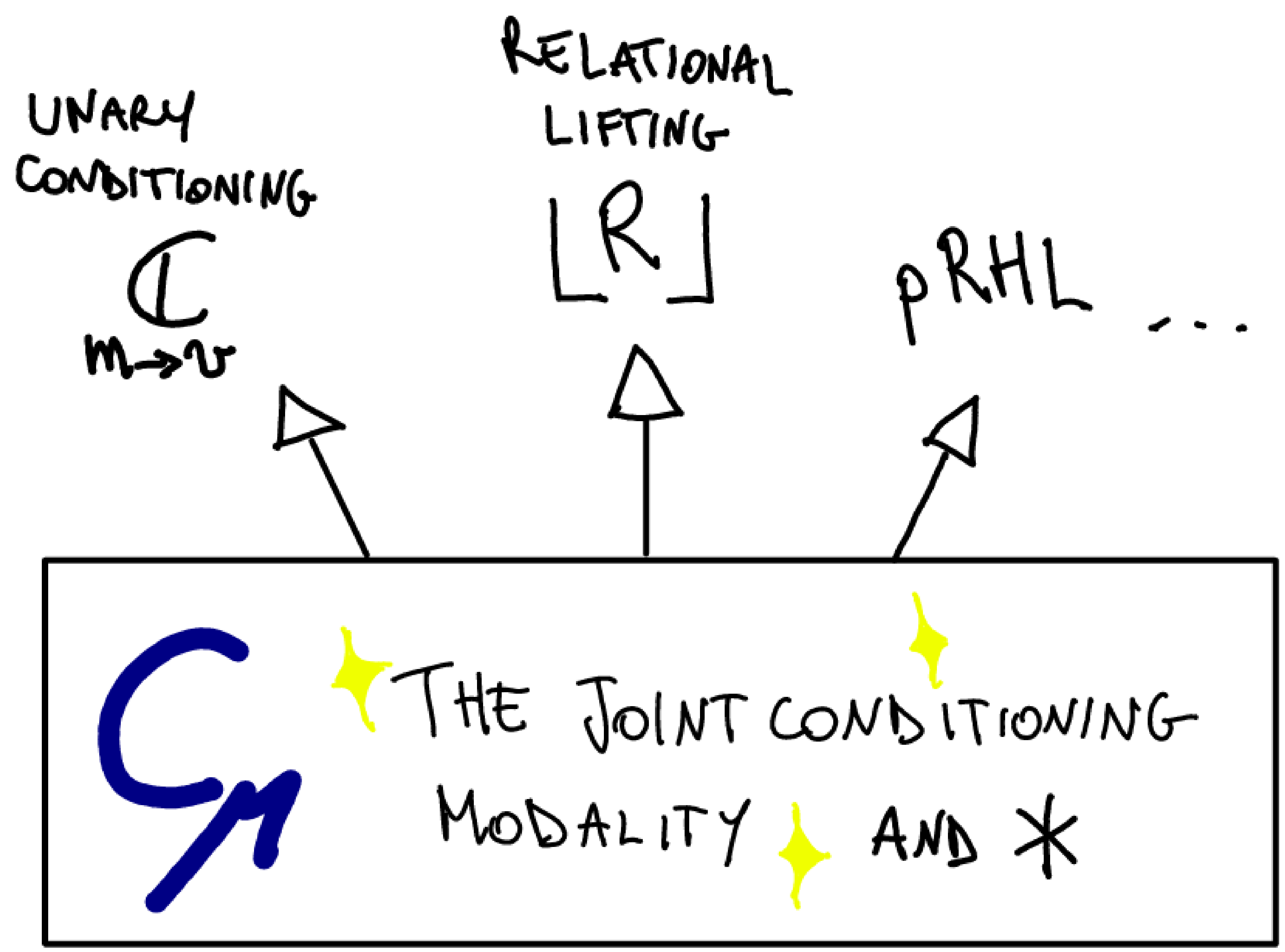
1) Can entailment ⊛ be proven in the logic?
2) Are there useful interactions between $*$, $\mathbb{C}$ and $\lfloor R \rfloor$?

BLUEBELL'S KEY INSIGHT

QUESTIONS :

1) Can entailment ⊛ be proven in the logic?
2) Are there useful interactions between $*$, $\mathbb{C}$ and $\lfloor R \rfloor$?

Bluebell says YES!



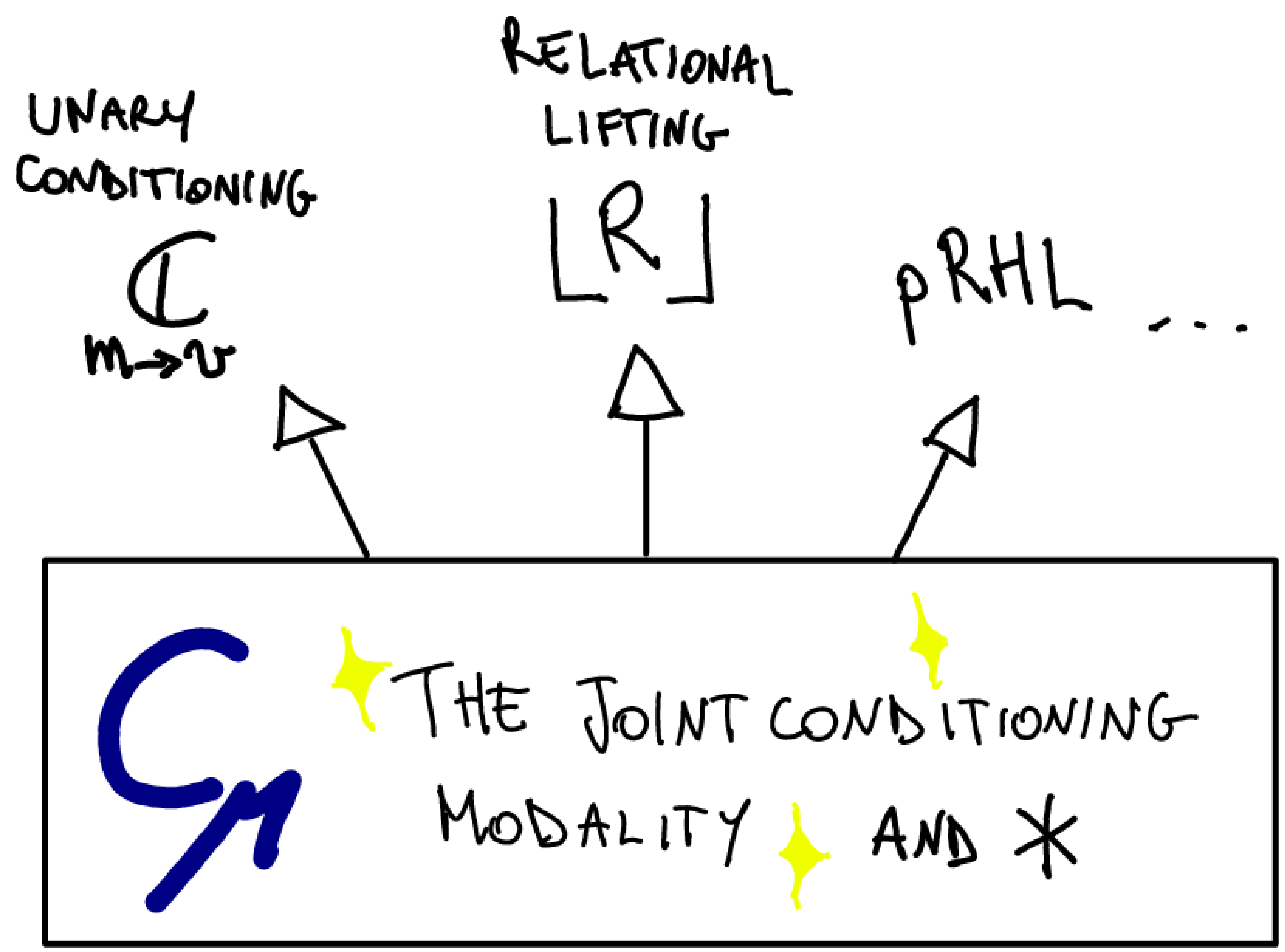
BLUEBELL'S KEY INSIGHT

QUESTIONS :

1) Can entailment ⊛ be proven in the logic?
2) Are there useful interactions between $*$, $\mathbb{C}$ and $\lfloor R \rfloor$?

Bluebell says YES!

Their definitions and laws can be derived



Rich set of core laws

RELATIONAL LIFTING AS CONDITIONING

Usual picture:

$$\begin{array}{ccc} \mathbb{D}(\mathrm{Store}) \times \mathbb{D}(\mathrm{Store}) & \supseteq & \lfloor R \rfloor \\ \uparrow \text{Lift} & & \uparrow \\ \mathrm{Store} \times \mathrm{Store} & \supseteq & R \end{array}$$

(Analog of

$$\begin{array}{ccc} \mathbb{D}(\mathrm{Store}) & \supseteq & \mathbb{P}(A) = 1 \\ \uparrow & & \uparrow \\ \mathrm{Store} & \supseteq & A \end{array}$$
)

Bluebell's view:

$$\begin{array}{ccc} \mathbb{D}(\mathrm{Store}) \times \mathbb{D}(\mathrm{Store}) & \supseteq & \lfloor R \rfloor \\ \downarrow \text{Conditioning} & & \downarrow \\ \mathrm{Store} \times \mathrm{Store} & \supseteq & R \end{array}$$

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 $\mu : \mathbb{D}(A)$ and $\kappa : A \to \mathbb{D}(\mathrm{Store})$ define

$$\mathrm{bind}(\mu, \kappa) := \lambda s.\ \sum_{a \in A} \mu(a)\, \kappa(a)(s)$$

Example $A = \{0, 1\}$ $\mu = \mathrm{Ber}(1/3)$

$$\mathrm{bind}(\mu, \kappa) = 1/3\, \kappa(0) + 2/3\, \kappa(1)$$

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

JOINT CONDITIONING

Def Given $\mu : \mathbb{D}(A)$ and $P : A \to \mathrm{Asrt}$
define $C_\mu v. P(v) : \mathrm{Asrt}$ by

$$(\mu_1, \mu_2) \models C_\mu v. P(v) \quad \text{iff} \quad \left(\begin{array}{l} \exists \kappa_1, \kappa_2 : A \to \mathbb{D}(\mathrm{Store}). \\ \qquad \mu_1 = \mathrm{bind}(\mu, \kappa_1) \;\wedge \\ \qquad \mu_2 = \mathrm{bind}(\mu, \kappa_2) \;\wedge \\ \quad \forall a \in \mathrm{supp}(\mu). \\ \qquad\quad (\kappa_1(a), \kappa_2(a)) \models P(a) \end{array} \right)$$

JOINT CONDITIONING

$$(\mu_1, \mu_2) \models C_\mu v. P(v) \quad \text{iff} \quad \left(\begin{array}{l} \exists k_1, k_2 : A \to \mathbb{D}(\mathrm{store}). \\ \qquad \mu_1 = \mathrm{bind}(\mu, k_1) \;\wedge \\ \qquad \mu_2 = \mathrm{bind}(\mu, k_2) \;\wedge \\ \quad \forall a \in \mathrm{supp}(\mu). \\ \qquad\quad (k_1(a), k_2(a)) \models P(a) \end{array} \right)$$

JOINT CONDITIONING

$$(\mu_1, \mu_2) \models C_\mu v. P(v) \quad \text{iff} \quad \left(\begin{array}{l} \exists \kappa_1, \kappa_2 : A \to \mathbb{D}(\mathrm{Store}). \\ \quad \mu_1 = \mathrm{bind}(\mu, \kappa_1) \wedge \\ \quad \mu_2 = \mathrm{bind}(\mu, \kappa_2) \wedge \\ \forall a \in \mathrm{supp}(\mu). \\ \quad (\kappa_1(a), \kappa_2(a)) \models P(a) \end{array} \right)$$

Example: $A = \{0,1\}$ $\quad \mu = \mathrm{Ber}(1/3)$

$$\mu_1 = 1/3\, \kappa_1(0) + 2/3\, \kappa_1(1)$$

$$\mu_2 = 1/3\, \kappa_2(0) + 2/3\, \kappa_2(1)$$

$$P(0) \qquad P(1)$$

$x\langle 1\rangle \sim \mathrm{Ber}(1/3)$ $\quad \lceil x\langle 1\rangle = 0 \rceil$ $\quad \lceil x\langle 1\rangle = 1 \rceil$

$$C_\mu v. \left(\lceil x\langle 1\rangle = v \rceil * P(v) \right)$$

JOINT CONDITIONING

$$(\mu_1, \mu_2) \models C_\mu v. P(v) \quad \text{iff} \quad \left[\begin{array}{l} \exists \kappa_1, \kappa_2 : A \to \mathbb{D}(\mathsf{Store}). \\ \quad \mu_1 = \mathrm{bind}(\mu, \kappa_1) \wedge \\ \quad \mu_2 = \mathrm{bind}(\mu, \kappa_2) \wedge \\ \forall a \in \mathrm{supp}(\mu). \\ \quad (\kappa_1(a), \kappa_2(a)) \models P(a) \end{array}\right]$$

Example: $A = \{0,1\}$ $\quad \mu = \mathrm{Ber}(1/3)$

$$\mu_1 = 1/3\, \kappa_1(0) + 2/3\, \kappa_1(1)$$
$$\mu_2 = 1/3\, \kappa_2(0) + 2/3\, \kappa_2(1)$$

$$P(0) \qquad P(1)$$

$x\langle 1\rangle \sim \mathrm{Ber}(1/3)$ $\quad \lceil x\langle 1\rangle = 0 \rceil$ $\quad \lceil x\langle 1\rangle = 1 \rceil$

$$C_\mu v. \left(\lceil x\langle 1\rangle = v \rceil * P(v) \right)$$

[C-UNIT-R]

$$x\langle 1\rangle \sim \mu \dashv\vdash C_\mu v. \lceil x\langle 1\rangle = v \rceil$$

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

ENCODING LIFTING AS CONDITIONING

Unary Conditioning: $C_\mu\, v.\left(\lceil x = v \rceil * P(v)\right)$

Relational Lifting:

$$\lfloor R(x\langle 1\rangle, x\langle 2\rangle)\rfloor :=$$

$$\exists \mu : \mathbb{D}(Val \times Val).\; C_\mu\,(v_1, v_2).\left(\lceil x\langle 1\rangle = v_1 \rceil * \lceil x\langle 2\rangle = v_2 \rceil * \overset{\text{pure}}{R(v_1, v_2)}\right)$$

JOINT COND. RULES

[C-UNIT-R]

$$x\langle i\rangle \sim \mu \dashv\vdash \mathbf{C}_\mu\, v. \lceil x\langle i\rangle = v \rceil$$

[C-FRAME]

$$P * \mathbf{C}_\mu\, v.\, Q(v) \vdash \mathbf{C}_\mu\, v.\, (P * Q(v))$$

[C-CONS]

$$\frac{\forall v \in \mathrm{supp}(\mu).\ P(v) \vdash P'(v)}{\mathbf{C}_\mu\, v.\, P(v) \vdash \mathbf{C}_\mu\, v.\, P'(v)}$$

$$
\begin{aligned}
& x\langle 1\rangle \sim \mu * y\langle 1\rangle \sim \mu' * \lceil z = x+y \rceil \\
\vdash\ & \left(\mathbf{C}_\mu\, v.\, \lceil x\langle 1\rangle = v \rceil\right) * y\langle 1\rangle \sim \mu' * \lceil z = x+y \rceil \\
\vdash\ & \mathbf{C}_\mu\, v.\left(\lceil x\langle 1\rangle = v \rceil * y\langle 1\rangle \sim \mu' * \lceil z = x+y \rceil\right) \\
\vdash\ & \mathbf{C}_\mu\, v.\left(\mathbf{C}_{\mu'}\, v'.\left(\lceil x\langle 1\rangle = v \rceil * \lceil y\langle 1\rangle = v' \rceil * \lceil z = x+y \rceil\right)\right) \\
\vdash\ & \mathbf{C}_\mu\, v.\, \mathbf{C}_{\mu'}\, v'.\, \lceil x\langle 1\rangle = v \wedge y\langle 1\rangle = v' \wedge z = x+y \rceil
\end{aligned}
$$

[C-ASSOC]

$$\mathcal{C}_{\mu}\, v.\ \mathcal{C}_{\kappa(v)}\, v'.\ P(v, v') \vdash \mathcal{C}_{\mathrm{bind}'(\mu, \kappa)}\, .\ P(v, v')$$

$$\mathrm{bind}'(\mu, \kappa) = \mathrm{do}\ v \leftarrow \mu\ ;\ v' \leftarrow \kappa(v)\ ;\ \mathrm{return}\ (v, v')$$

[C-UNASSOC]

$$\mathcal{C}_{\mathrm{bind}(\mu, \kappa)}\, v'.\ P(v') \vdash \mathcal{C}_{\mu}\, v.\ \mathcal{C}_{\kappa(v)}\, v'.\ P(v')$$

SOME DERIVABLE RULES

$$C_{\mu}\, v.\, \lfloor R \rfloor \vdash \lfloor R \rfloor$$ (Convexity of Rel Liftings)

$$\lfloor R_1 \rfloor * \lfloor R_2 \rfloor \vdash \lfloor R_1 \wedge R_2 \rfloor$$

NOTE

$$\lfloor R_1 \rfloor \wedge \lfloor R_2 \rfloor \nvdash \lfloor R_1 \wedge R_2 \rfloor$$

CHALLENGES

- Generalization to Iris-style user-defined ghost resources
- [C-wp-SWAP]

$$\text{ownVars} \land C_\mu v.\ \mathbf{wp}\ \mathtt{t}\ \{Q(v)\} \vdash \mathbf{wp}\ \mathtt{t}\ \{C_\mu v.\ Q(v)\}$$

Bluebell needs this for soundness

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

Thanks