Formally Reasoning about (In)dependencies in Probabilistic Programs

Jialu Bao, Aug. 26th, 2022   A-Exam

Committee Members: Justin Hsu (Chair), Joseph Halpern, Dexter Kozen, Alexandra Silva

Formally Reasoning about (In)dependencies in Probabilistic Programs

Probabilistic Independence
Conditional Independence
Negative Dependence

Formally Reasoning about (In)dependencies in Probabilistic Programs

Programs that may sample from distributions

Syntax

Imp

$c ::= \textbf{skip}$
$\mid x := a$
$\mid c_1; c_2$
$\mid \textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2$
$\mid \textbf{while } b \textbf{ do } c$

PWhile

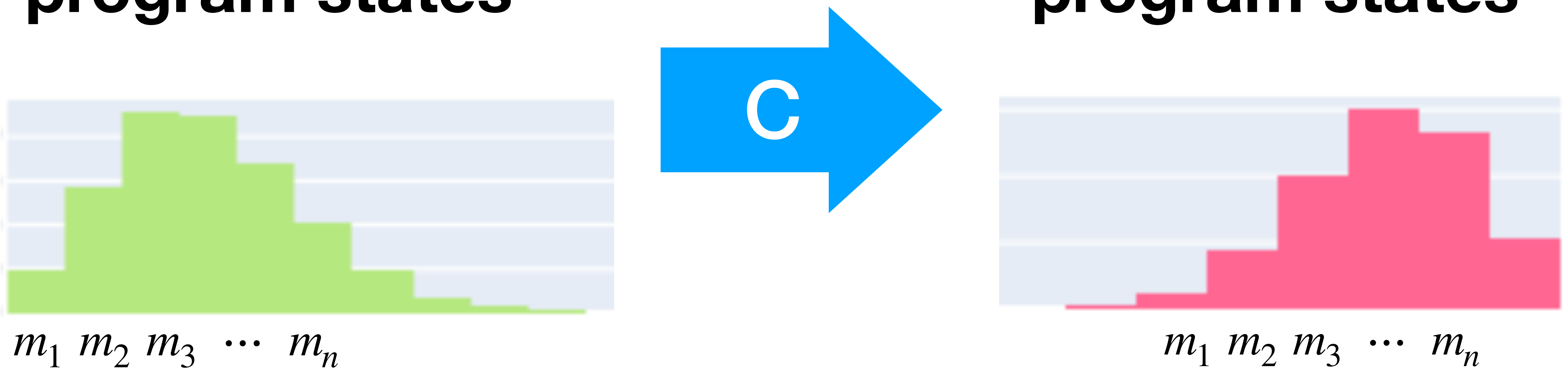
$\mid x := coin()$
$\cancel{\mid observe(b)}$

Semantics

Distribution Transformer

A distribution over program states

A distribution over program states



Formally Reasoning about (In)dependencies in Probabilistic Programs

Motivating Example

How do we ensure the security of an encryption algorithm?

Motivating Example

How do we ensure the security of an encryption algorithm?

Check  does not give information about .

Probabilistic Independence

Definition: random variables X, Y independent iff,

$$\mathbb{P}(X, Y) = \mathbb{P}(X) \cdot \mathbb{P}(Y).$$

Intuition: the value of one variable does not give information about the other.

Example:

$X \sim$ 

$Y \sim$ 

Motivating Example

How to we ensure the security of an encryption algorithm?

Encode 

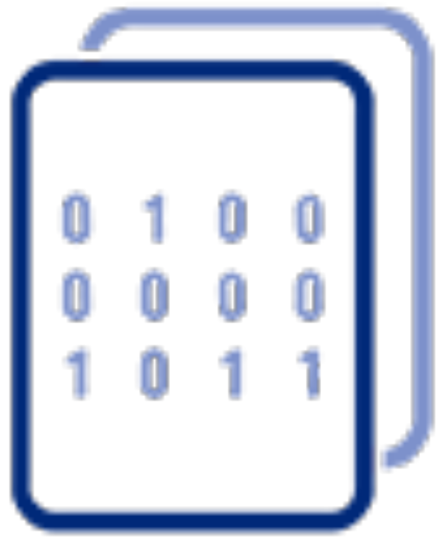
Encryption

as probabilistic programs;

However, reasoning about probabilities can be hard.

Verify 

Plain text

, 

Encrypted text

are probabilistically independent.

My Goal

Design formal methods to reason about independence and dependencies in the distribution constructed by probabilistic programs.

Why Formal Methods and Which Kind?

Rigorous: unlike documentations in natural languages, formal specifications have **no vagueness** and can **capture target properties exactly**.

Axiomatic: a set of axioms and rules that a computer can follow, e.g. program logic, type systems.

Want relative simplicity:

Require less human ingenuity or human time.

Match better with pen-and-paper proofs.

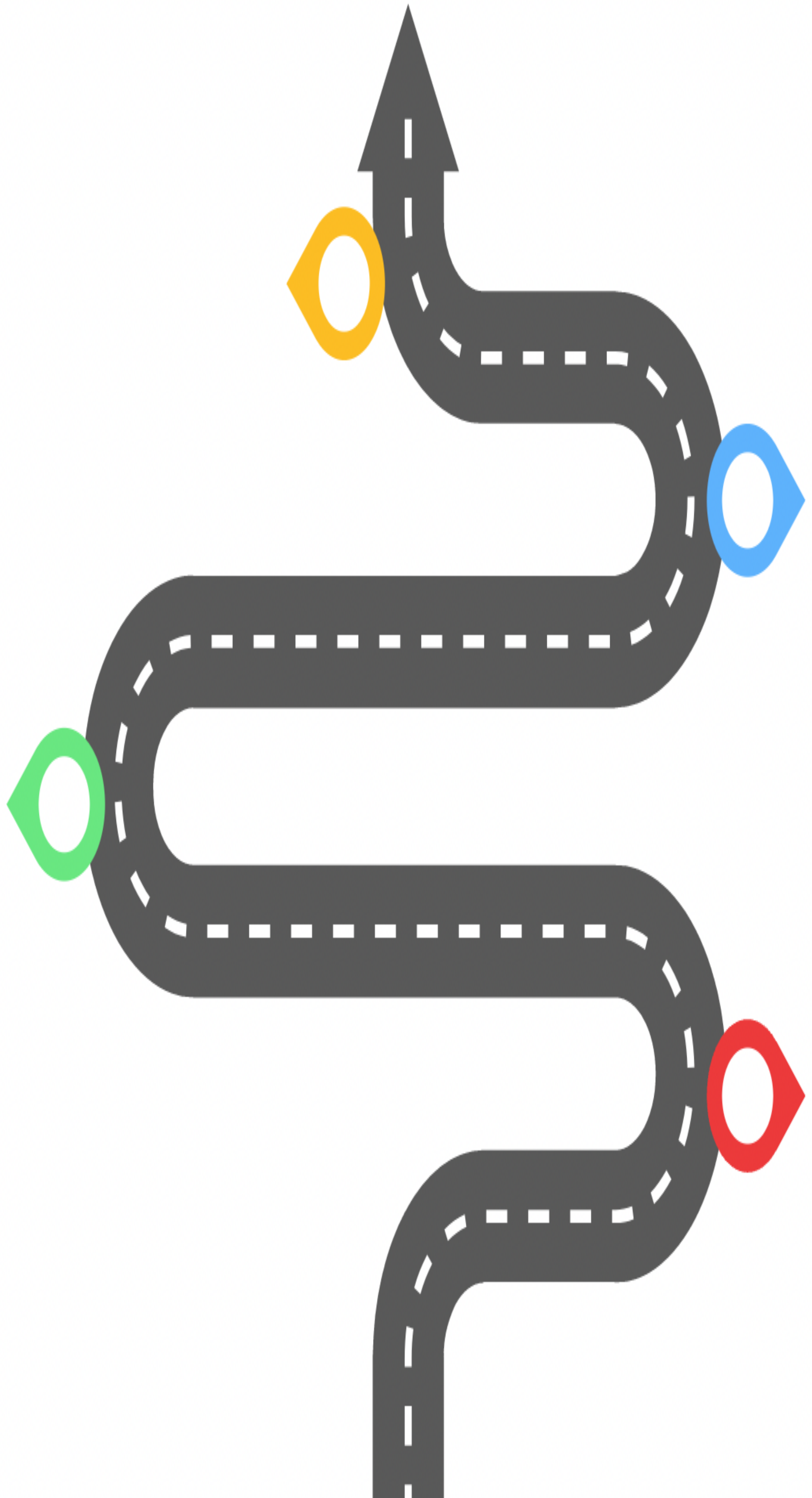
My Existing Work

A Separation Logic for **Negative Dependence**. POPL 2022

Jialu Bao, Marco Gaboardi, Justin Hsu, Joseph Tassarotti.

A Bunched Logic for **Conditional Independence**. LICS 2021

Jialu Bao, Simon Docherty, Justin Hsu, Alexandra Silva.



Related Work

Formally Reasoning about Conditional Independence

Conditional Independence

Intuition: the value of X does not give information about the value of Y if we already know the value of Z.

Definition: Variables X, Y are conditionally independent given Z iff,

$$\mathbb{P}(X, Y \mid Z) = \mathbb{P}(X \mid Z) \cdot \mathbb{P}(Y \mid Z).$$

Example: ice cream sales and sunglasses sales

sunny $\sim$ 

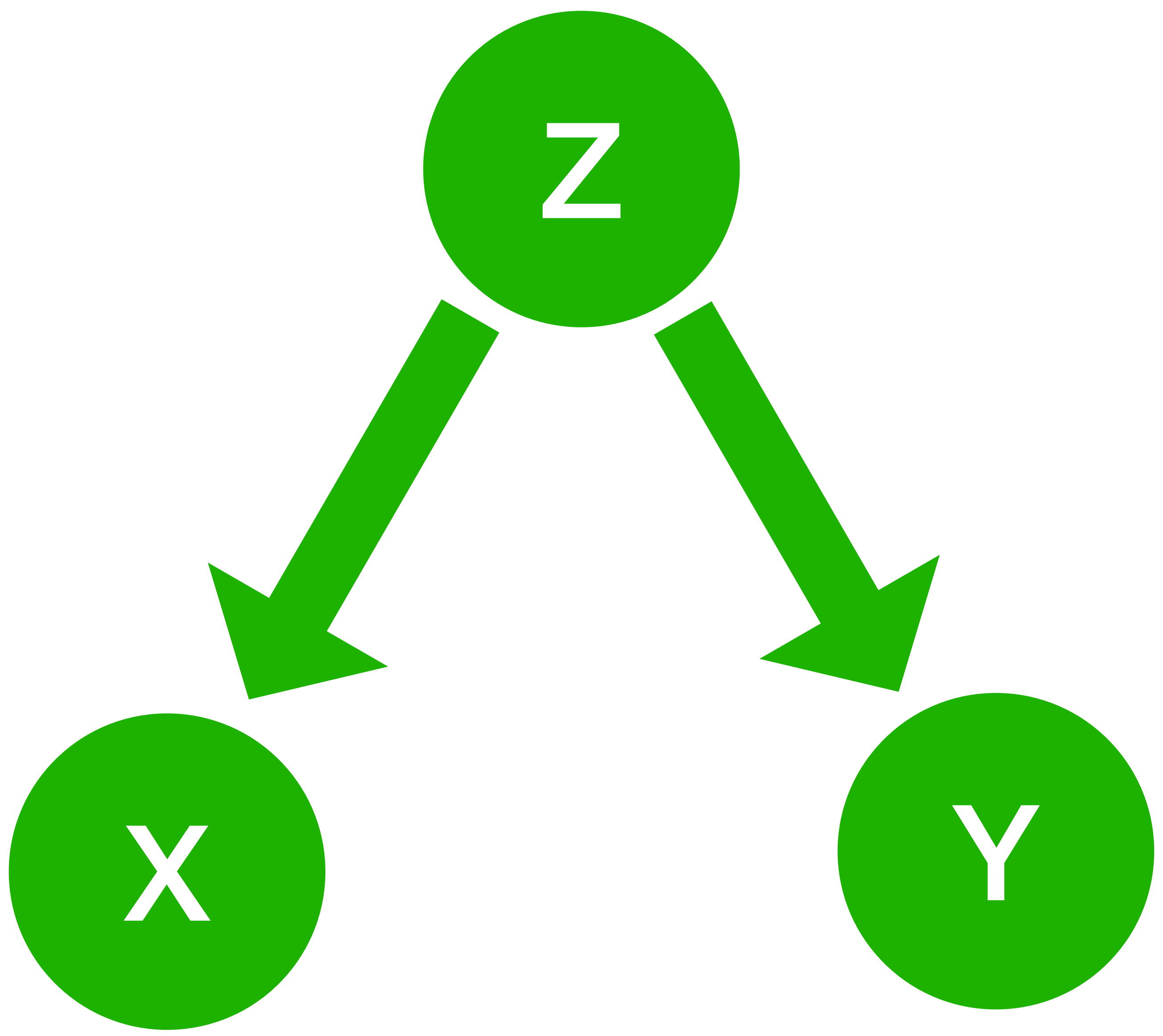
if sunny:

buy  $\sim$ 

buy  $\sim$ 

Applications of Conditional Independence

- Represent and transform a joint distribution more efficiently.



Our Goal

Design a program logic for proving conditional independence (CI)

Example: **Precondition:** $\{\ \top\ \}$

```
sunny := coin()
if sunny:
   icecream := coin();
   sunglasses := coin();
else:
   icecream := False;
   sunglasses := False;
```

- How to express CI as assertions?
- How to prove CI in programs?

Post-condition: $\{\text{icecream, sunglasses}$ **are CI given** $\text{sunny}\}$

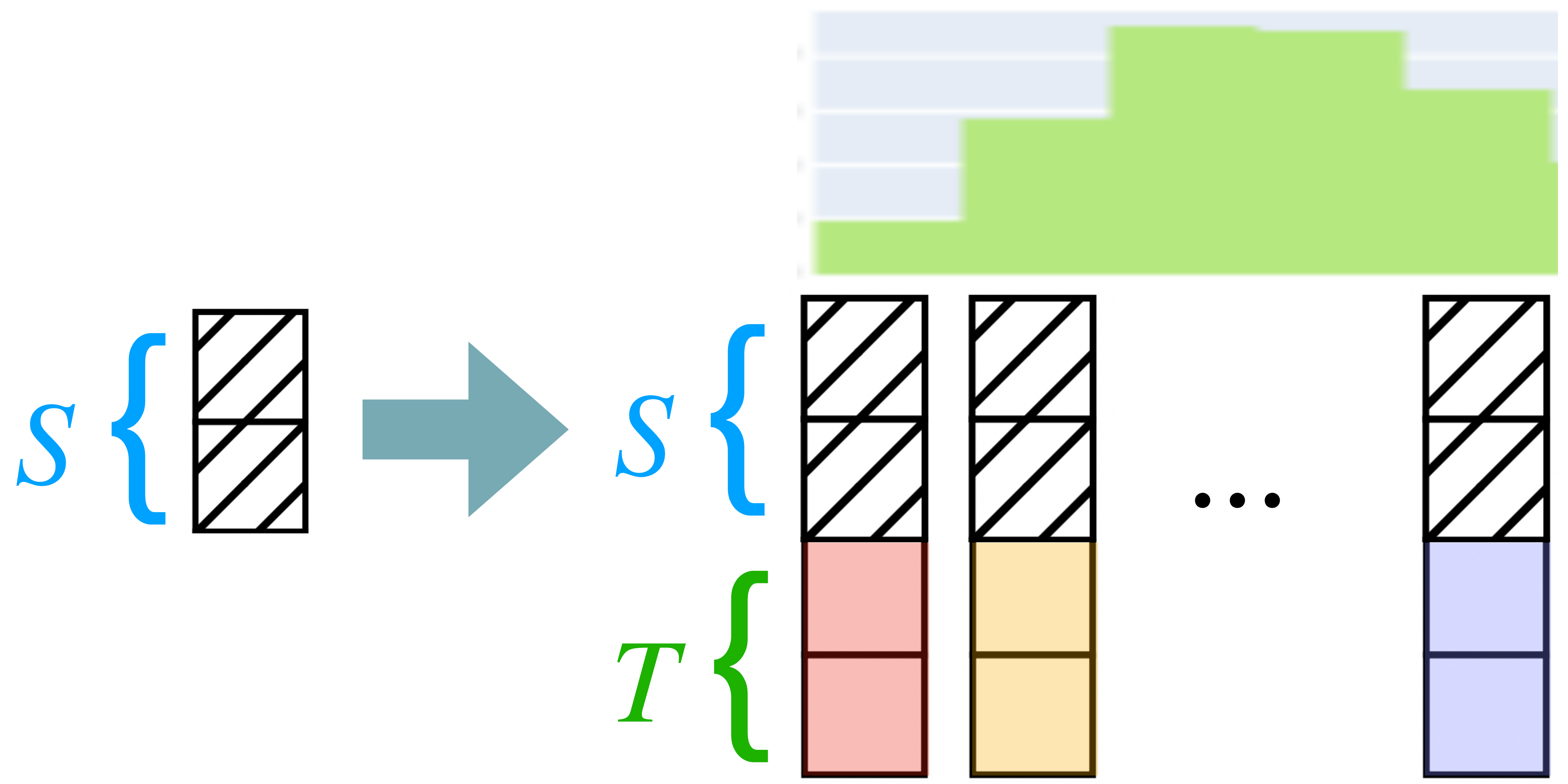
Notations

Var	Set of all program variables
Val	Set of possible values
$[S]$	Set of program memories on a finite $S \subseteq \mathsf{Var}$,
	where a program memory on S is a map of type $S \to \mathsf{Val}$
$\mathscr{D}W$	Set of discrete distributions over a set W $\quad \boxed{\mathscr{D}[S]}$
$\mathscr{D}\mathsf{Mem}$	$\bigcup_{T \subseteq \mathsf{Var}} \mathscr{D}[T]$

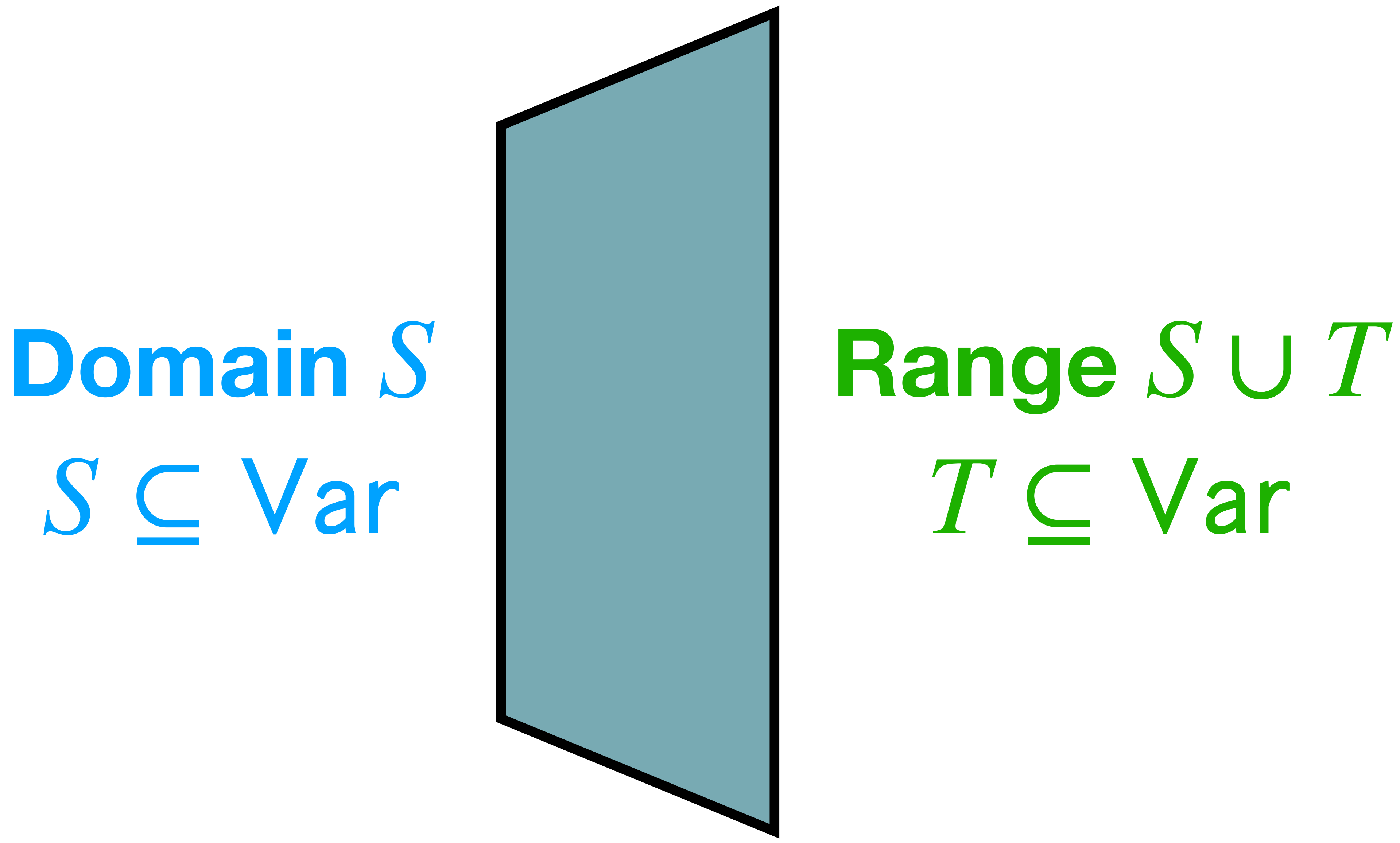
Visual Representation

Conditional Probability Distribution

Input-preserving maps of type
$[S] \to \mathscr{D}[S \cup T]$



a.k.a., Kernels

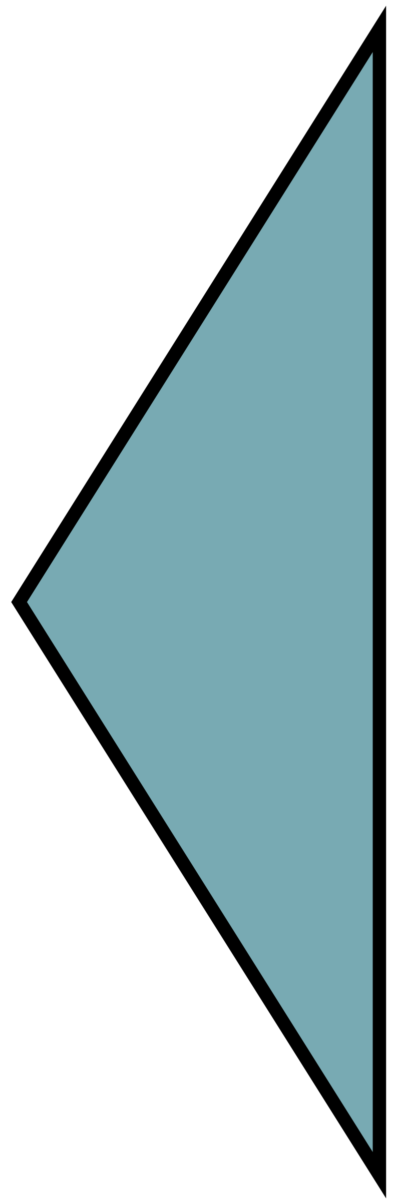


Visual Representation

Input-preserving maps of type
$[\varnothing] \to \mathscr{D}[T]$

Kernels

Domain $\varnothing$



Range T
$T \subseteq \mathsf{Var}$

Intuition

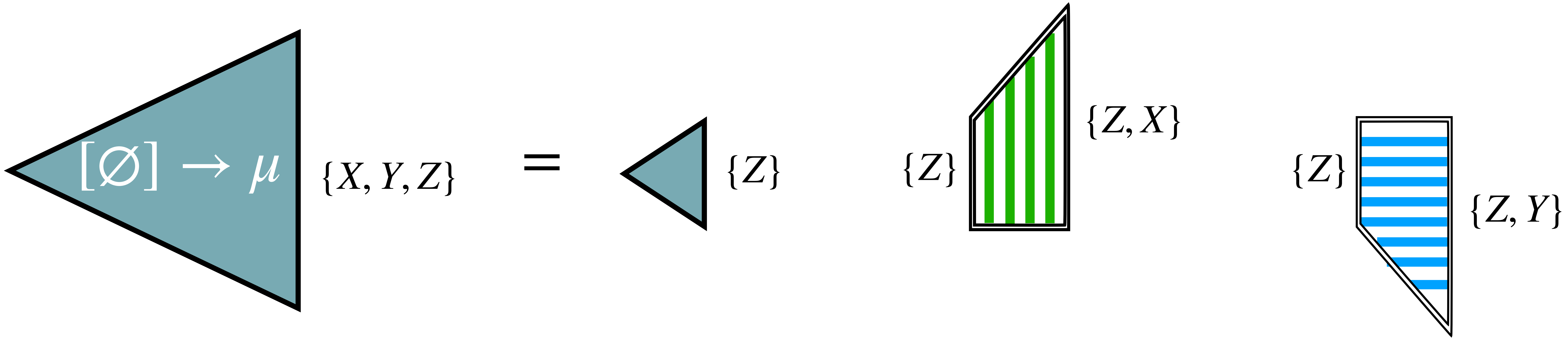
X, Y are conditionally independent given Z in a distribution μ iff

Sample X, Y, Z from μ

1. Sample Z

2a. Sample X given Z

2b. **Separately** sample Y given Z



Bunched Logic [O'Hearn and Pym 1999]

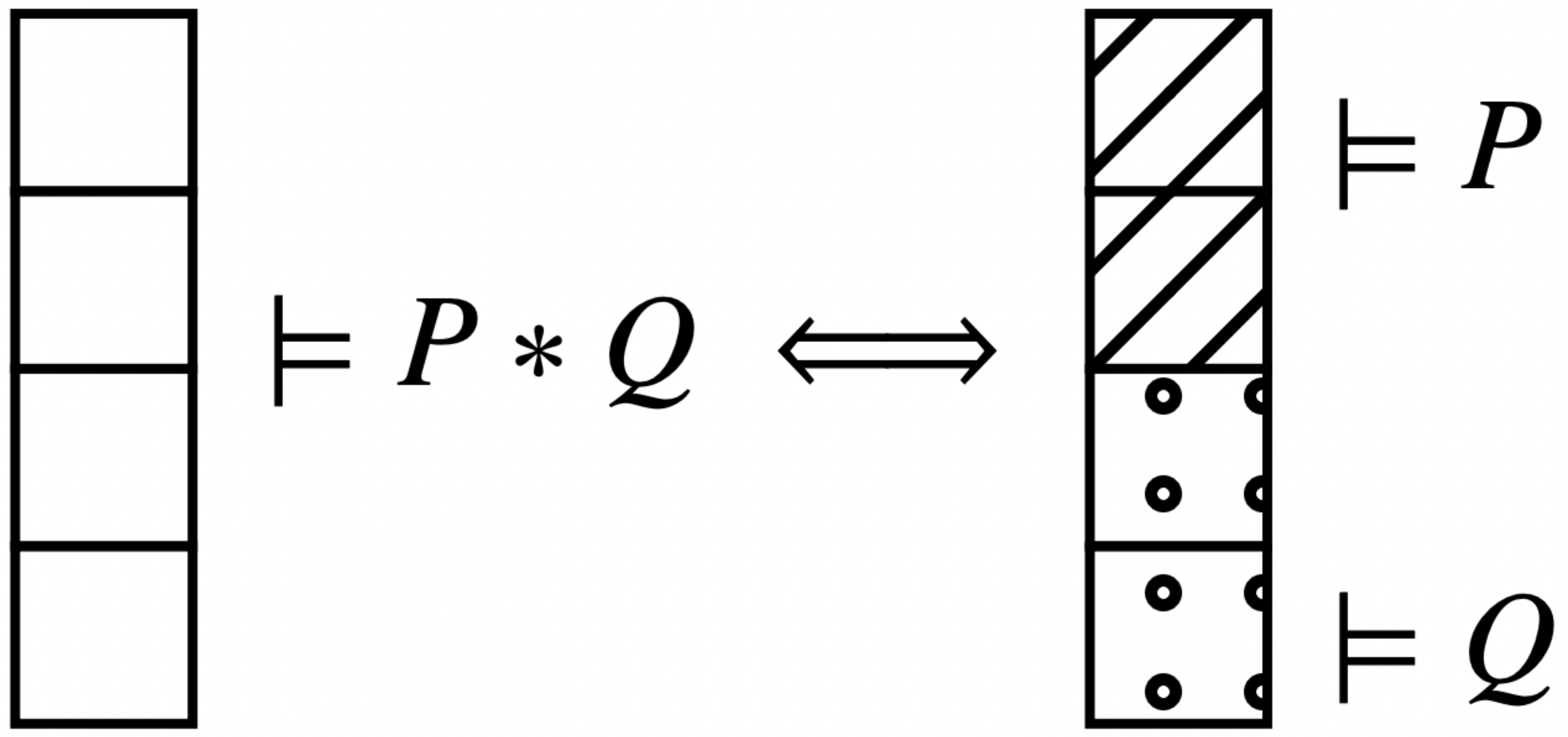
A flexible framework to reason about separation

The logic of bunched implications (BI)

$$P, Q ::= p \in \mathscr{AP} \mid \top \mid \bot \mid P \land Q \mid P \lor Q \mid P \Rightarrow Q \mid P * Q$$

The conjunction $*$ is substructural (no weakening or contraction)

Resource interpretation:



A BI model for Independence [Barthe et al. 2020]

Kripke Semantics Definition

Let states be $\mathcal{D}$Mem.

$\mu \models P * Q$ iff there exist μ_1, μ_2 such that $\mu_1 \models P, \mu_2 \models Q$ and $\mu_1 \oplus \mu_2 = \mu$, where $\oplus$ takes the **independent product** of two distributions.

$\mu \models \langle X \rangle$ iff there exists S such that $\mu \in \mathcal{D}[S]$ and $X \in S$.

Theorem

$\mu \models \langle X \rangle * \langle Y \rangle$ iff X, Y are independent in μ.

How do we adapt this logic for capturing conditional independence?

DIBI: Dependence and Independence BI [Bao et al. 2021]

$P, Q ::= p \in \mathscr{AP} \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid P * Q \mid P \fatsemi Q$

A new non-commutative conjunction for modeling dependence: $\fatsemi$

read "$P \fatsemi Q$" as "Q may depend on P"

Sample proof rules for $\fatsemi$

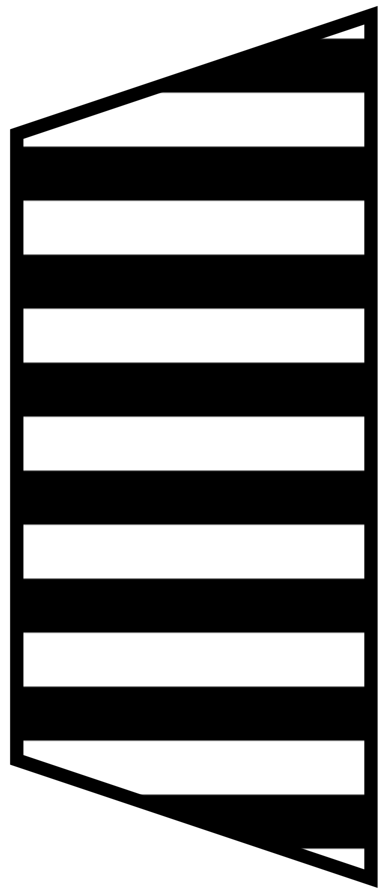
$$\frac{}{(P \fatsemi Q) \fatsemi R \dashv\vdash P \fatsemi (Q \fatsemi R)}\ \fatsemi\text{-}\textsc{Assoc} \qquad \frac{P \vdash R \qquad Q \vdash S}{P \fatsemi Q \vdash R \fatsemi S}\ \fatsemi\text{-}\textsc{Conj}$$

DIBI Model for Conditional Independence

Kripke Semantics Definition

Let states be the set of kernels:

Domain S



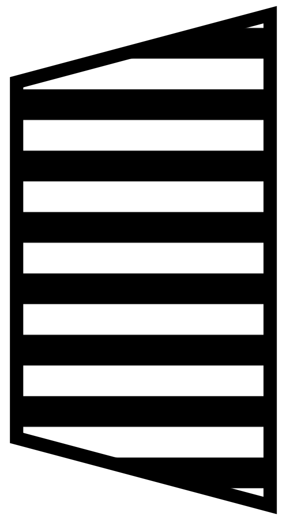
Range $S \cup T$

for finite $S, T \subseteq \mathsf{Var}$

We can lift any distribution μ to a kernel f by defining $f = [\emptyset] \mapsto \mu$

Semantics

Domain S



Range
$S \cup T \cup U$

$\models S \triangleright T$

Atomic Proposition

$f \models S \triangleright T$ if the domain of f is **exactly** S and the range of f **includes** T.

Satisfaction Rules

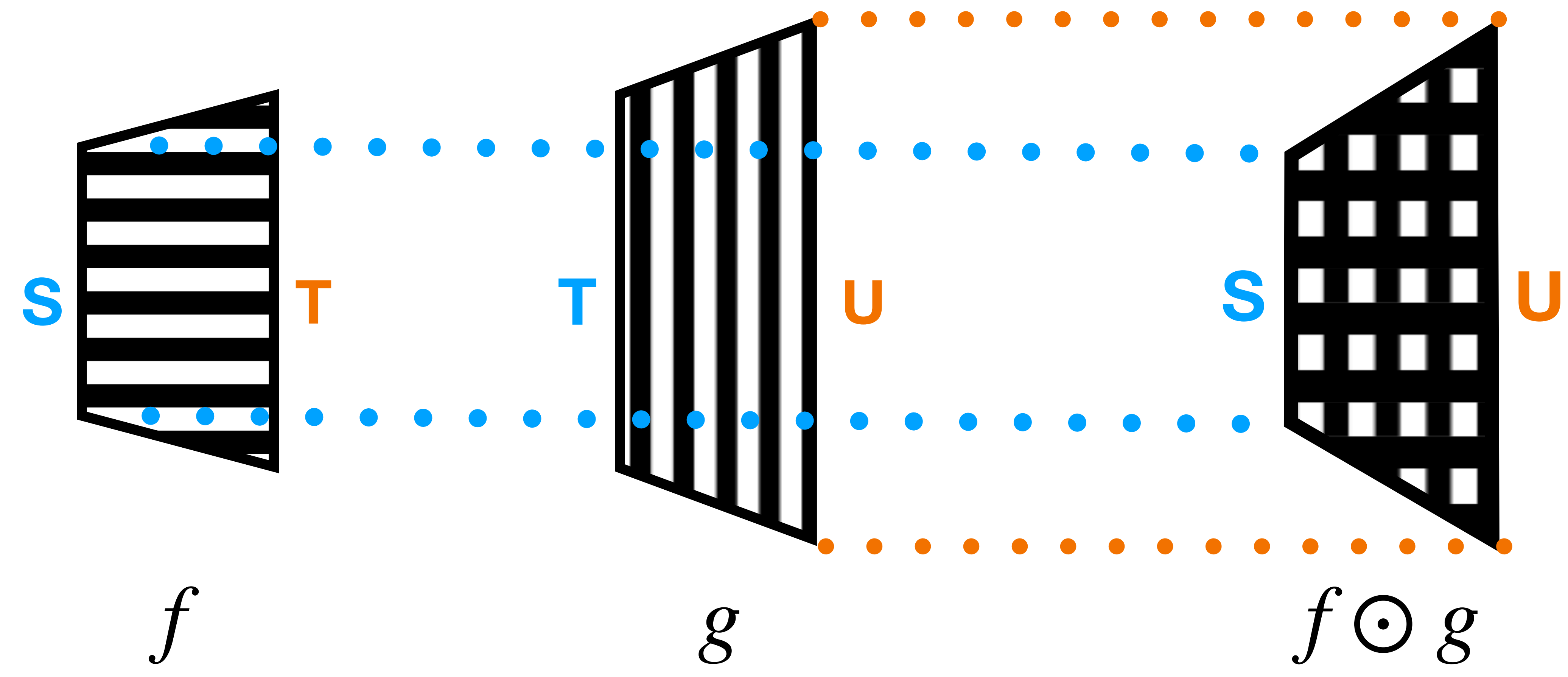
$f \models P * Q$ iff there exist f_1, f_2 such that $f_1 \models P, f_2 \models Q$ and $f_1 \oplus f_2 = \mu$.

$f \models P \fatsemi Q$ iff there exist f_1, f_2 such that $f_1 \models P, f_2 \models Q$ and $f_1 \odot f_2 = f$.

Binary Operator $\odot$ for Interpreting $\fatsemi$

Let $\odot$ sequence two kernels together

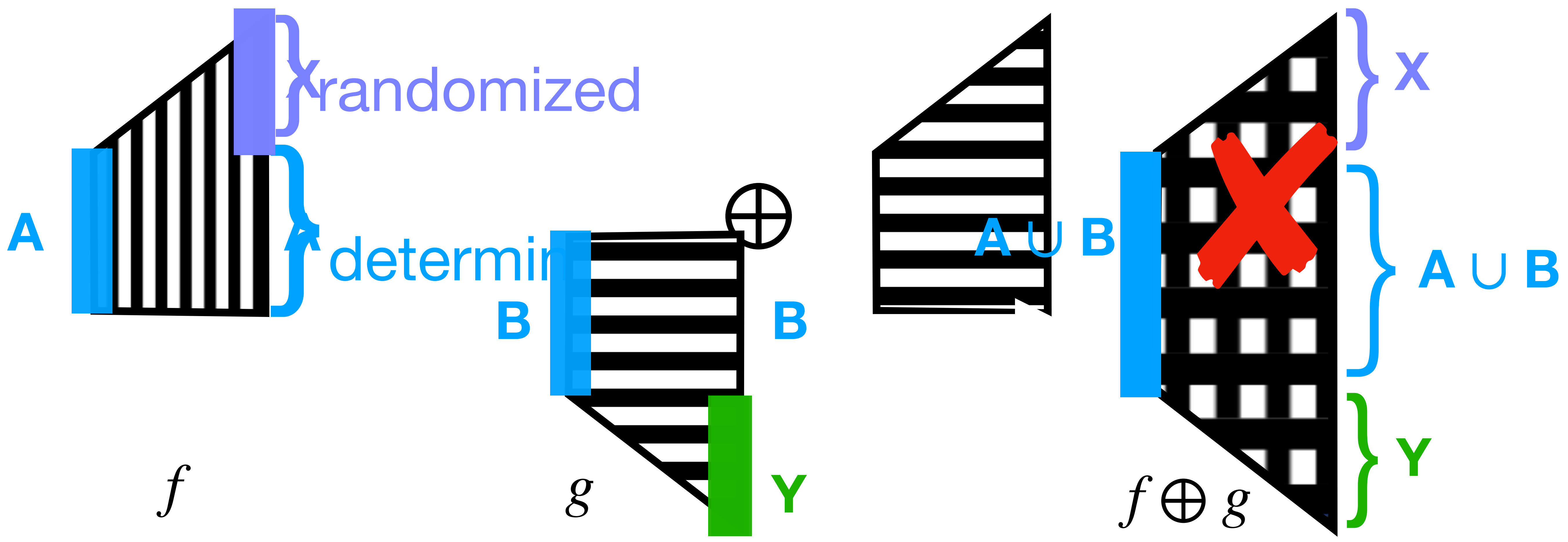
$f \odot g$ is defined if the range of the f equals the domain of g.



$$(f \odot g)(s)(u) := \sum_t f(s)(t) \cdot g(t)(u)$$

Binary Operator ⊕ for Interpreting $*$

Let ⊕ take the product of two kernels.



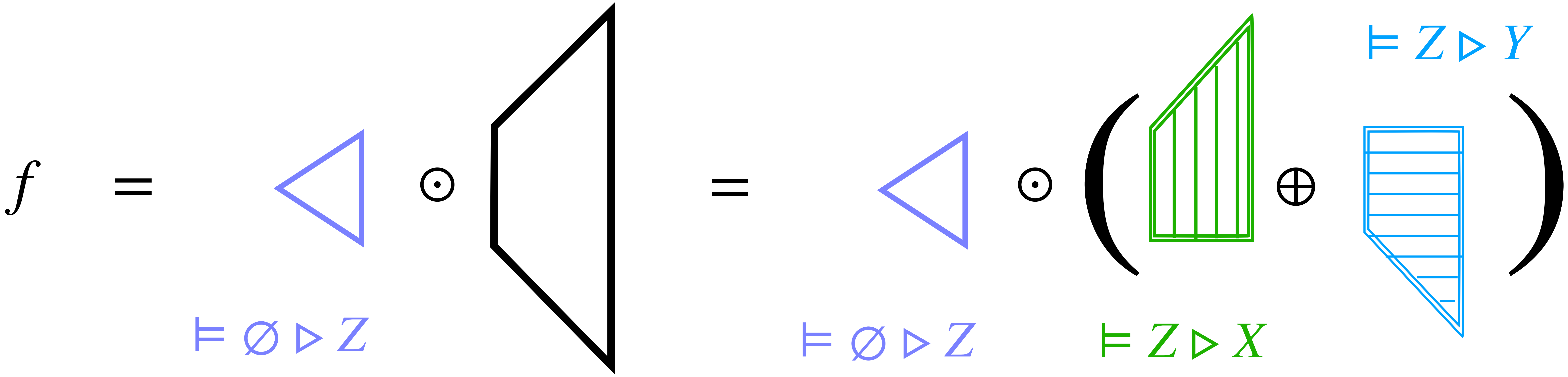
$f \oplus g$ is defined iff $X \cap Y = \varnothing$.

Assert Conditional Independence

Theorem. **A sound and complete assertion logic for CI**

In the probabilistic DIBI, X, Y are CI given Z in distribution μ iff

$$f \models (\emptyset \triangleright Z) \fatsemi (Z \triangleright X * Z \triangleright Y), \text{ where } f = [\emptyset] \mapsto \mu$$



Program Logic

Judgement: $\{\phi\}C\{\psi\}$

where $C \in$ PWhile, and

ϕ, ψ are formulas in the probabilistic model of DIBI.

Proof system: Sound though incomplete; decidability unknown.

A program logic for proving CI

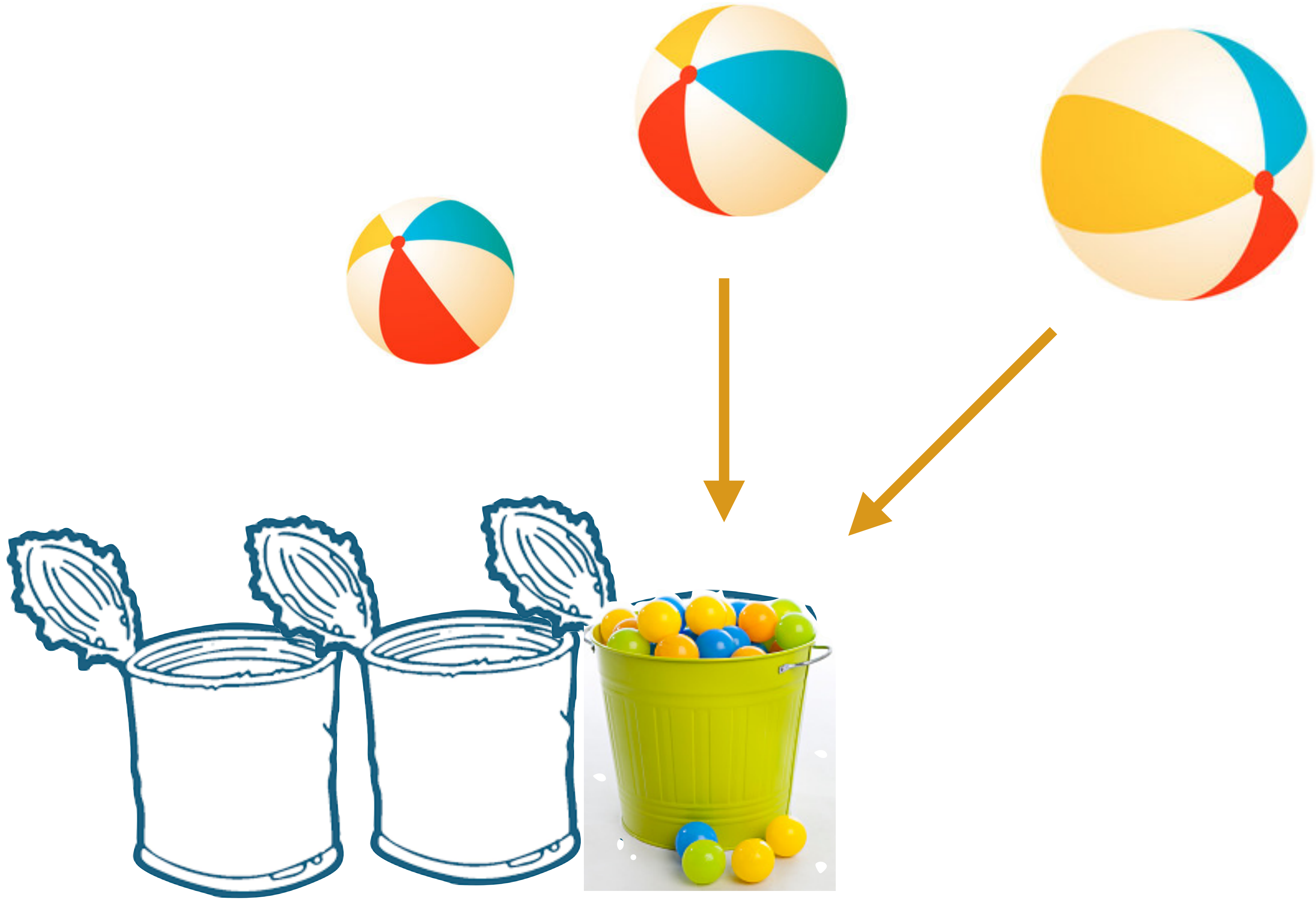
Our Contributions

1. **a new bunched logic (DIBI) with a sound and complete proof system.**
2. **a probabilistic DIBI model that can capture CI.**
3. a Hoare-style program logic to verify CI.
4. a powerset DIBI model that can capture join dependencies.

https://arxiv.org/pdf/2008.09231

Formally Reasoning about
Negative Dependence

Motivating Example: Balls-into-Bins

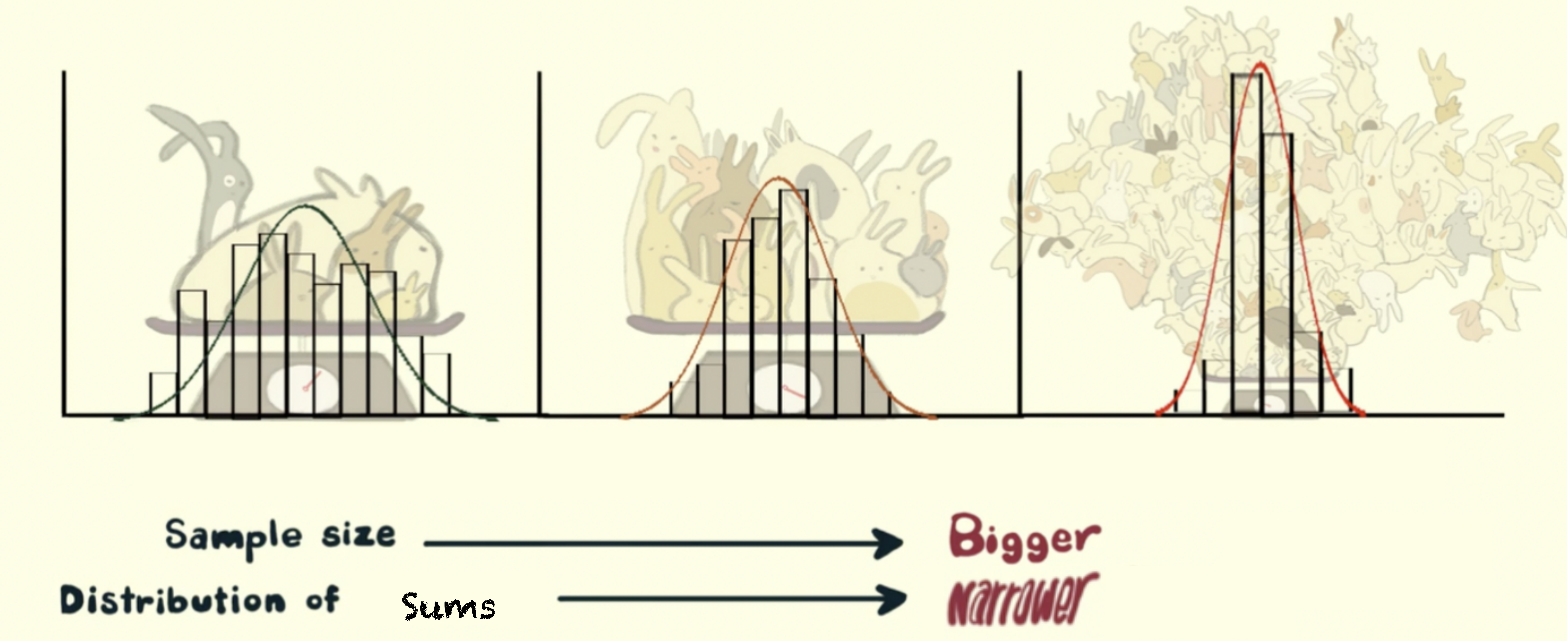


Intuition: unlikely for many bins to overflow together

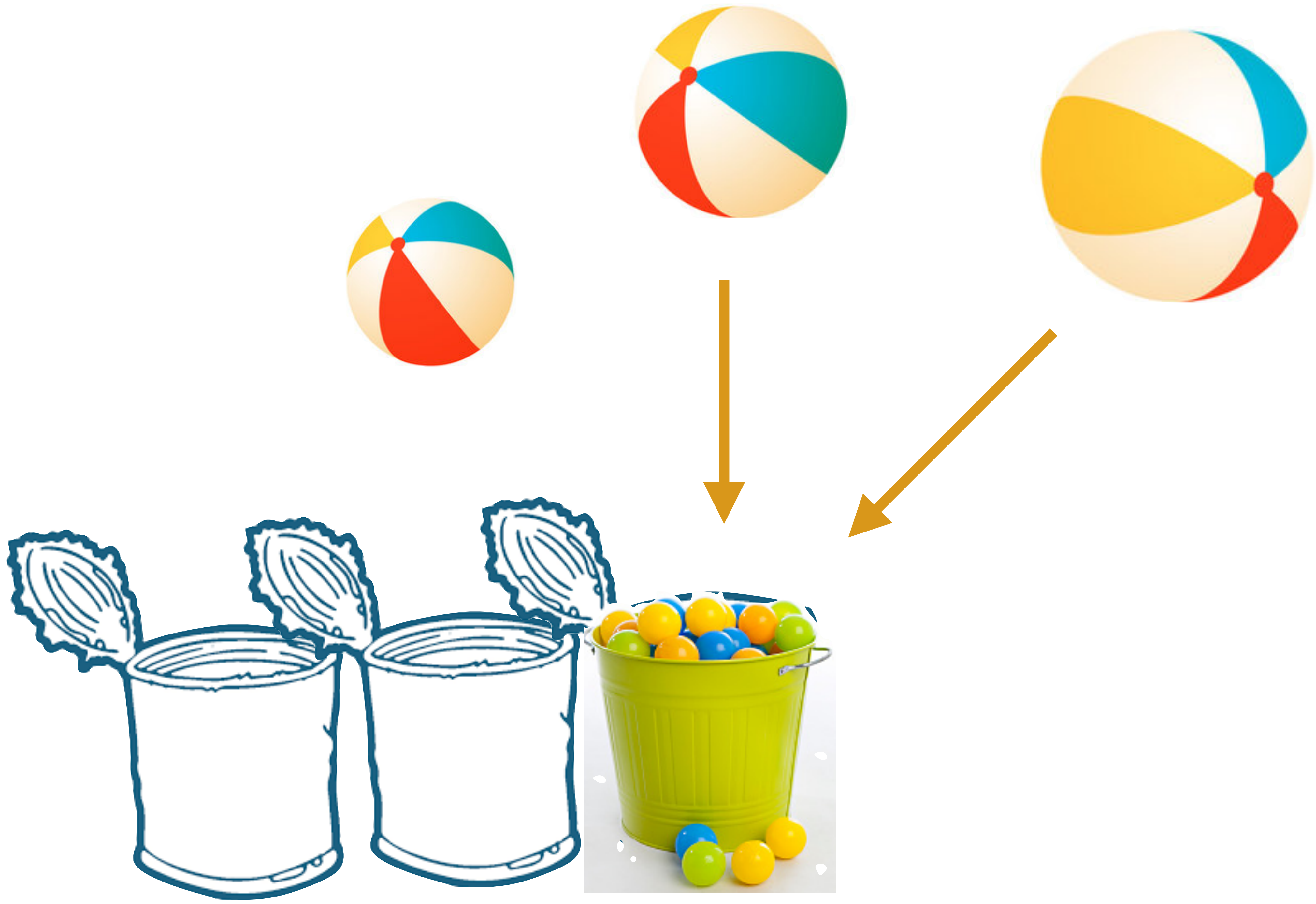
$$\mathbb{P}\left[\sum_{bin} \text{overflow}[bin] \geq 2\right] \leq ?$$

Concentration Bounds

$Y = \sum_{i}^{n} Y_i$, where Y_i are independent, then $\mathbb{P}[\,|\,Y - \mathbb{E}[Y]\,|\, \geq M] \leq g(n, M)$



Motivating Example: Balls-into-Bins



$$\mathbb{P}\left[\sum_{bin} \text{overflow}[bin] \geq 2\right] \leq ?$$

Not Independent!

Concentration bounds:

$$Y = \sum_{i}^{n} Y_i \text{, where } Y_i \text{ are}$$ **negatively dependent**

$$\mathbb{P}[\,|Y - \mathbb{E}[Y]| \geq M] \leq f(n, M)$$

The number of balls in each bin is negatively dependent.

Our goal: Prove negative dependence formally.

Negative Dependence

Negative Covariance

Negative Regression

Negative Association (NA)

Negative Right Orthant Dependence

Negative Quadrant Dependence

Negative Association (NA) [Joag-Dev and Proschan 1983]

Definition

Real-valued random variables $X_1, \ldots, X_n$ satisfy NA **iff**

for any disjoint $Y, Z \subseteq \{X_1, \ldots, X_n\}$,

for any monotone functions $f : \mathbb{R}^{|Y|} \to \mathbb{R}_{\geq 0}$ and $g : \mathbb{R}^{|Z|} \to \mathbb{R}_{\geq 0}$,

$$\mathbb{E}[f(Y) \cdot g(Z)] \leq \mathbb{E}[f(Y)] \cdot \mathbb{E}[g(Z)] \ .$$

Examples of NA [Joag-Dev and Proschan 1983]

Independent random variables

Deterministic variables

Bernoulli random variables that sum to 1

Uniformly random permutations

one-hot vectors

	X_1	X_2	X_3
p	1	0	0
q	0	1	0
1-p-q	0	0	1

shuffle(cards);
Y_i = cards[i]

Closure of NA [Joag-Dev and Proschan 1983]

Subsets of NA variables are NA

Union of independent NA sets is also NA

Monotonically increasing map preserves NA

X_1	X_2
1	0
0	1
0	0

shuffle(cards);
Y_i = cards[i]

if two processes independent,
$\{X_1, X_2, Y_1, \ldots, Y_n\}$ satisfies NA

$Z_1 = X_1 + Y_1$ $\quad$ $Z_2 = X_2 + Y_n$

$\{Z_1, Z_2\}$ satisfies NA

A Bunched Logic for NA [Bao et al. 2022]

We introduce a negative association conjunction $\circledast$:

$$P, Q ::= p \in \mathcal{AP} \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid P * Q \mid P \circledast Q$$

Challenge: semantics for $\circledast$

Let states be $\mathscr{D}$Mem.

$\mu \vDash P \circledast Q$ iff there exist μ_1, μ_2 such that $\mu_1 \vDash P, \mu_2 \vDash Q$ and $\mu \in \mu_1 \otimes \mu_2$?

Subtle to define $\otimes$ for NA star $\circledast$

$\mu_1 \otimes \mu_2$ has to be a set of distributions.

Natural choices don't validate required axioms:

$$\frac{}{P \dashv\vdash P \circledast I}\ \circledast\text{-}\textsc{Unit} \qquad \frac{}{(P \circledast Q) \circledast R \dashv\vdash P \circledast (Q \circledast R)}\ \circledast\text{-}\textsc{Assoc}$$

Partition Negative Association (PNA) [Bao et al. 2022]

NA is a relation on a set of random variables.
PNA is a relation on a partition of random variables.

PNA satisfies similar closure properties as NA.

Lemma

$X_1, \ldots, X_n$ satisfies NA iff partition $\{X_1\}, \ldots, \{X_n\}$ satisfies PNA.

Semantics for $\circledast$

Let states be $\mathcal{D}$Mem.

$\mu \in \mu_1 \otimes \mu_2$ iff μ is a joint distribution of μ_1, μ_2 and partition $\{\text{dom}(\mu_1), \text{dom}(\mu_2)\}$satisfies PNA in μ.

$\mu \vDash P \circledast Q$ iff there exist μ_1, μ_2 such that $\mu_1 \vDash P, \mu_2 \vDash Q$, $\mu \in \mu_1 \otimes \mu_2$.

A Bunched Logic for NA [Bao et al. 2022]

Theorem

$\mu \models \langle X_1 \rangle \circledast \langle X_2 \rangle \circledast \cdots \circledast \langle X_n \rangle$ **iff** $X_1, X_2, \ldots, X_n$ **are negatively associated in** μ**.**

Theorem

Properties of PNA can be encoded as valid axioms in our logic.

Our Contributions

Assertion Logic (with a sound and complete proof system)

Separating conjunction for asserting negative association

Program Logic (with a sound proof system)

LINA: a probabilistic Separation Logic for Independence and NA

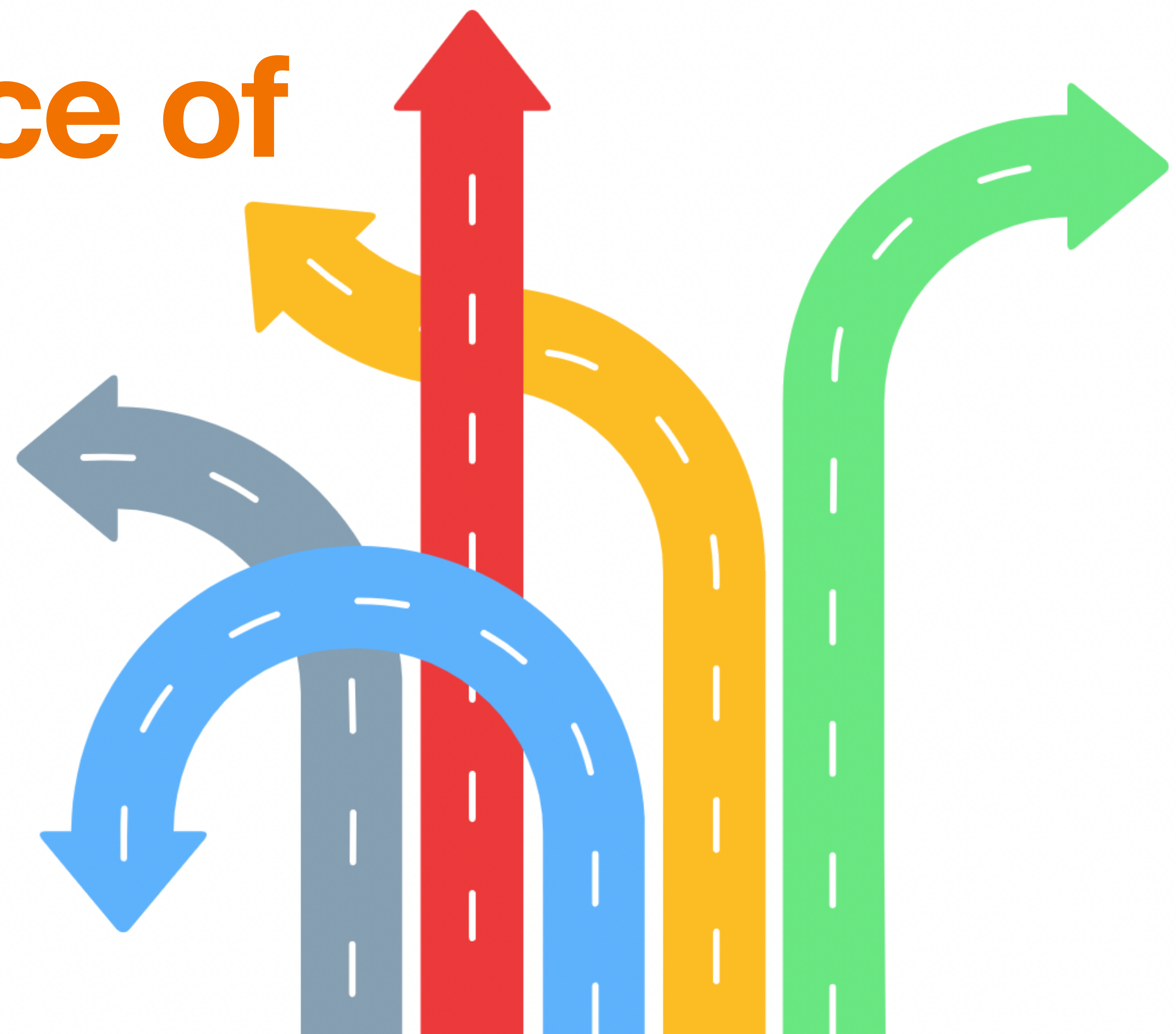
Applications

Verify Bloom filter and other probabilistic data structures

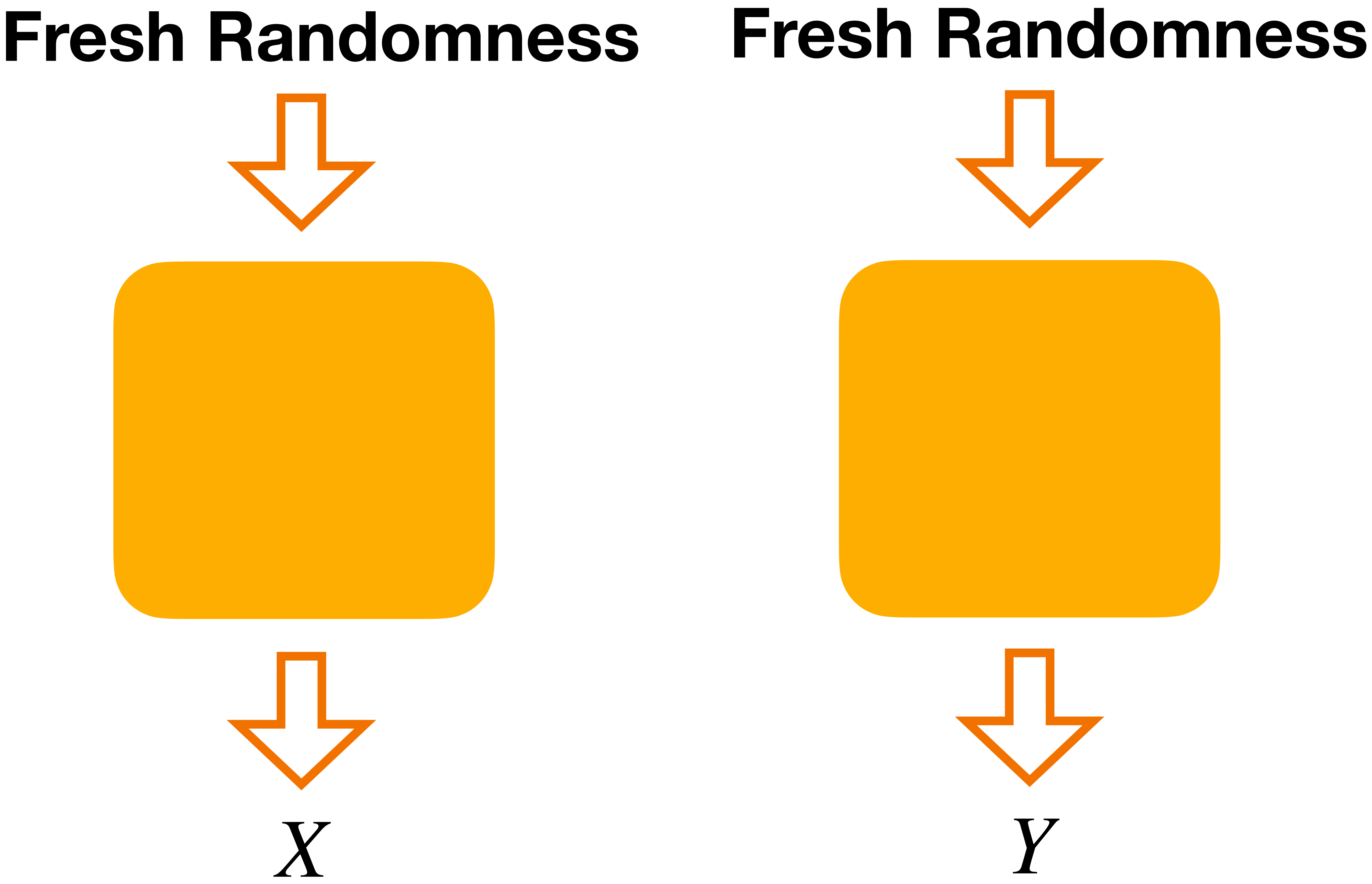
https://arxiv.org/abs/2111.14917

Future Work

Verifying Independence of Variables with Shared Randomness



Existing Methods for Independence



Toy Example 1

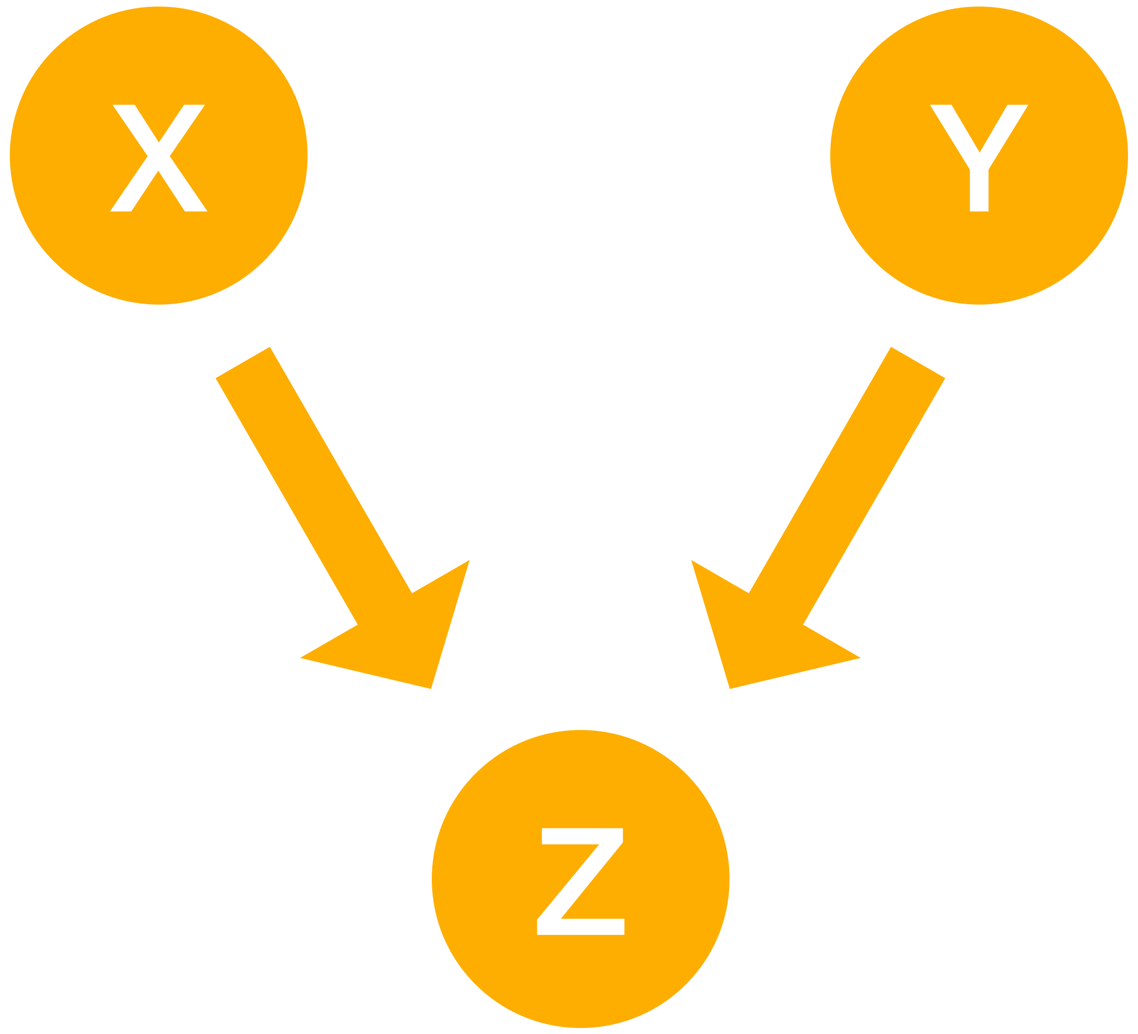
Fair coin

$X \sim$ 

Fair coin

$Y \sim$ 

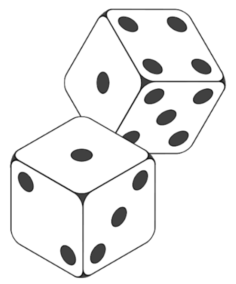
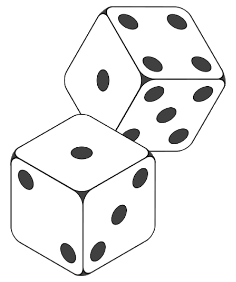
$Z = X \text{ xor } Y$



Program logic can not prove X, Z independent, or Y, Z independent.

$$(X * Y) \land (X * Z) \land (Y * Z)$$

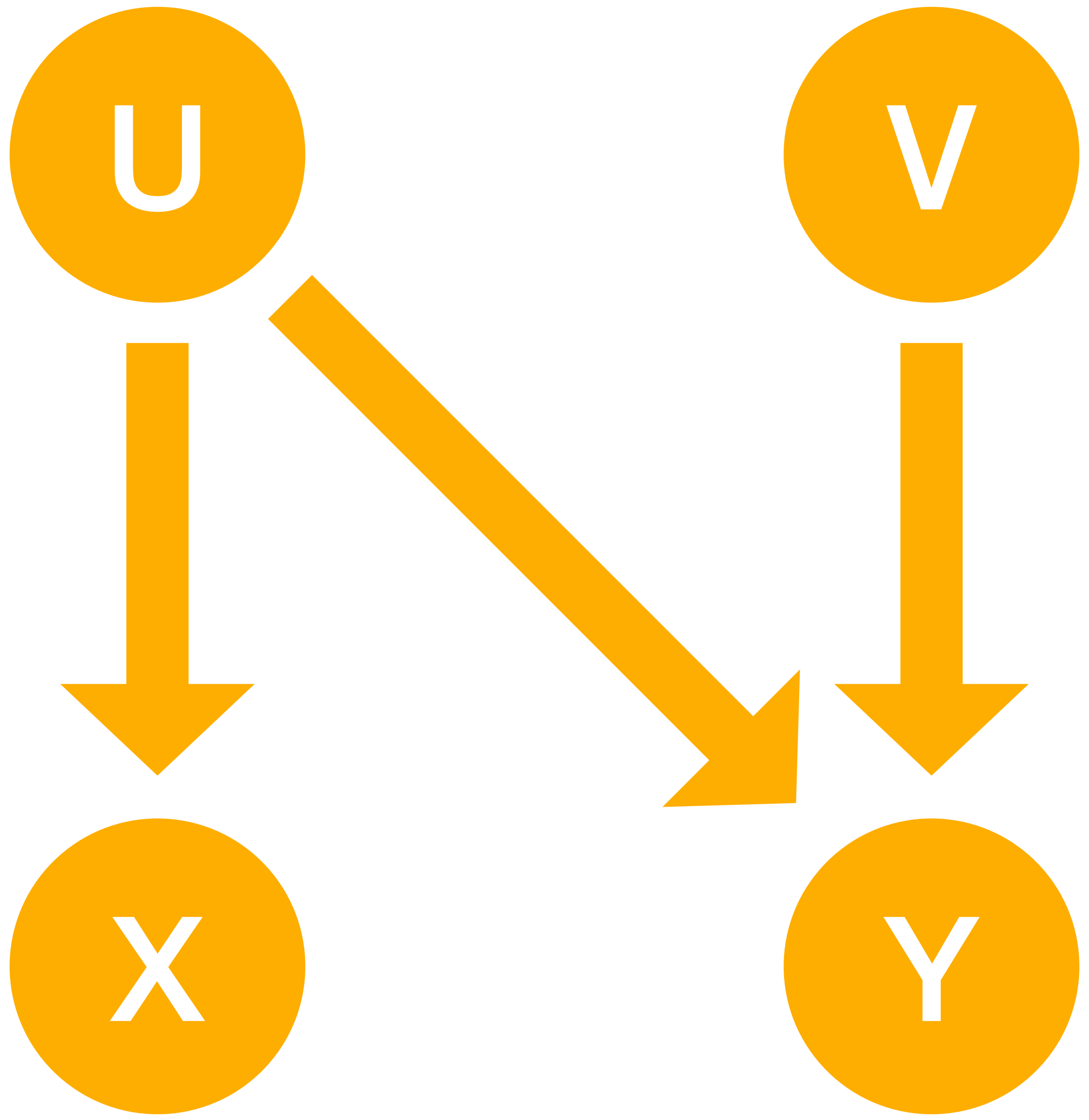
Toy Example 2

$U \sim$  $V \sim$ 

$X = (U = 3)$

$\cancel{Y = (U + V = 8)}$

$Y = (U + V = 7)$



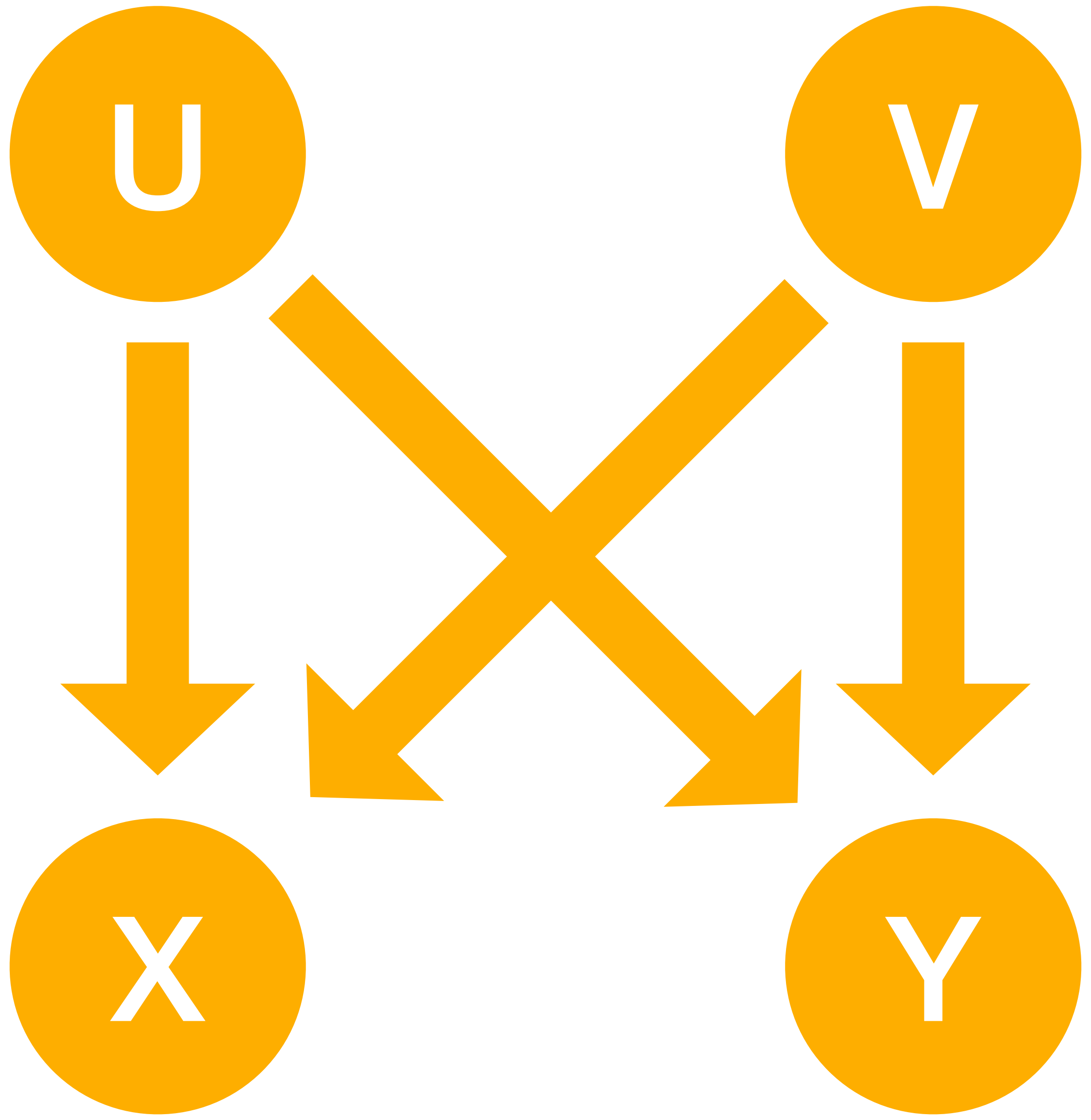
X, Y are not independent. Program logic can not prove them independent.

X, Y are independent

Box-Muller Transform

Use: two independent uniformly distributed variables U, V.

Output: two independent normally distributed variables X, Y.




```
U = uniform(0,1)

V = uniform(0,1)
```

X = $\sqrt{-2\log_2 U}\cos(2\pi V)$

Y = $\sqrt{-2\log_2 U}\sin(2\pi V)$

X, Y are independent but we cannot prove it

Observation

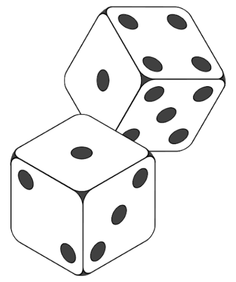
Fair coin

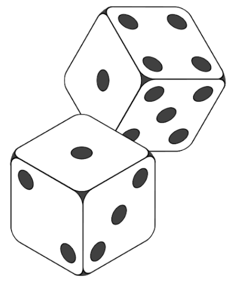
$X \sim$ 

Fair coin

$Y \sim$ 

$Z = X \text{ xor } Y$

$U \sim$ 

$V \sim$ 

$X = (U = 3)$

$Y = (U + V = 7)$

`U = uniform(0,1)`

`V = uniform(0,1)`

`X =` $\sqrt{-2 \log_2 U} \cos(2\pi V)$

`Y =` $\sqrt{-2 \log_2 U} \sin(2\pi V)$

Uniform distributions seems to be special!

Naive Solution: Add Axioms!

Fact:

Given a finite group G with binary operation $+$.

If $U \sim \text{uniform}(G)$, random variable X takes value in Val and is independent from U, and $f, h : \mathsf{Val} \to G$.

Then variables $f(X)$ and $h(X) + U$ are independent no matter what f, h are.

If we add this fact as an axiom:

We can prove independence in toy example 1.

Still cannot prove independence in toy example 2 and Box-Muller.

Add more axioms?

Desired Solution

An assertion logic to capture the interactions between uniformity and independence so that

we can derive more axioms about uniformity and independence using its proof system;

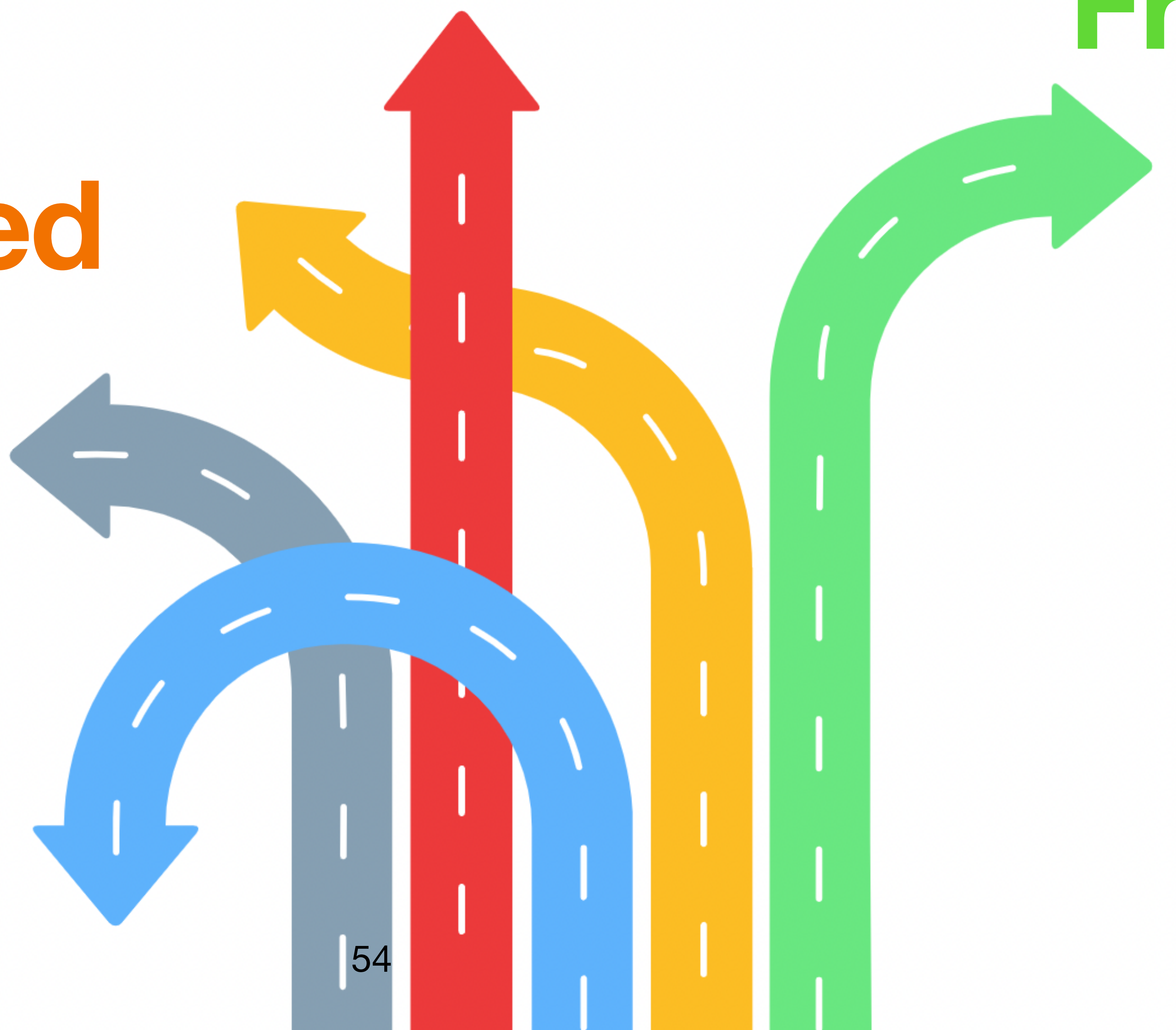
and prove independence of variables that possibly share source of randomness.

Other Thoughts

NA Arisen from Sampling

Conditional Independence From d-separation

Independence of Variables with Shared Randomness



Questions?