A SEPARATION LOGIC FOR NEGATIVE DEPENDENCE

Jialu Bao at PLDG, Oct. 6, 2021
Joint work with Marco Gaboardi, Justin Hsu, Joseph Tassarotti

Motivating Example

Motivating Example

Motivating Example

Bad events: collision, overflow, …

Motivating Example

Bad events: collision, overflow, …

```
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    bin = uniform([0,1,2])
    loads[bin] = loads[bin] + 1
overflow = [n >= 10 for n in loads]
```

Motivating Example

Bad events: collision, overflow, …

```
tasks = [A, …, Z]
loads = [0, 0, 0]
for task in tasks:
    bin = uniform([0,1,2])
    loads[bin] = loads[bin] + 1
overflow = [n >= 10 for n in loads]
```

$$Prob\left[\sum_i \text{overflow}[i] \geq 1\right] \leq ?$$

One standard recipe:

One standard recipe:

Concentration bound:

One standard recipe:

Concentration bound:

$Y = \sum_{i}^{n} Y_i$, where Y_i are independent

One standard recipe:

Concentration bound:

$Y = \sum_{i}^{n} Y_i$, where Y_i are independent

One standard recipe:

Concentration bound:

$Y = \sum_{i}^{n} Y_i$, where Y_i are independent

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

One standard recipe:

Concentration bound:

$Y = \sum_{i}^{n} Y_i$, where Y_i are independent

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

$Prob\left[\sum_i \text{overflow}[i] \geq 1\right] \leq ?$

One standard recipe:

Concentration bound:

$Y = \sum_{i}^{n} Y_i$, where Y_i are independent

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

$Prob\left[\sum_i \text{overflow}[i] \geq 1\right] \leq ?$

The bins' loads are not independent!

One standard recipe:

Concentration bound:

$Y = \sum_{i}^{n} Y_i$, where Y_i are independent

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

$Prob\left[\sum_i \text{overflow}[i] \geq 1\right] \leq ?$

The bins' loads have negative dependence!

One standard recipe:

Concentration bound:

$$Y = \sum_{i}^{n} Y_i \text{, where } Y_i \text{ are } \boxed{\text{negative dependence}}$$

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

$$Prob\left[\sum_{i} \text{overflow}[i] \geq 1\right] \leq ?$$

The bins' loads have negative dependence!

One standard recipe:

How to prove negative dependence formally?

Concentration bound:

$Y = \sum_{i}^{n} Y_i$, where Y_i are negative dependence

$$Prob\left[\sum_i \text{overflow}[i] \geq 1\right] \leq ?$$

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

The bins' loads have negative dependence!

Our Contribution

Our Contribution

- **A program logic for proving negative dependence**
  - Extending probabilistic separation logic [Barthe et al. 2020]

Our Contribution

- **A program logic for proving negative dependence**
  - Extending probabilistic separation logic [Barthe et al. 2020]
- **Show its applications to various probabilistic data structure**
  - Bloom filter [Bloom 1970]
  - Permutation Hashing [Ding and König 2011]
  - Fully-dynamic dictionary [Bercea and Even 2019]
  - Repeated balls-into-bins [Becchetti et al. 2019]

NEGATIVE DEPENDENCE

Probabilities 101

Probabilities 101

- A distribution over a finite set S is a function $\mu : S \to [0,1]$ such that $\sum_{s \in S} \mu(s) = 1$

Probabilities 101

- A distribution over a finite set S is a function $\mu : S \to [0,1]$ such that $\sum_{s \in S} \mu(s) = 1$
- Expected value of a (discrete) random variable X in distribution μ is $\sum_{v} \mu(X = v) \cdot v$

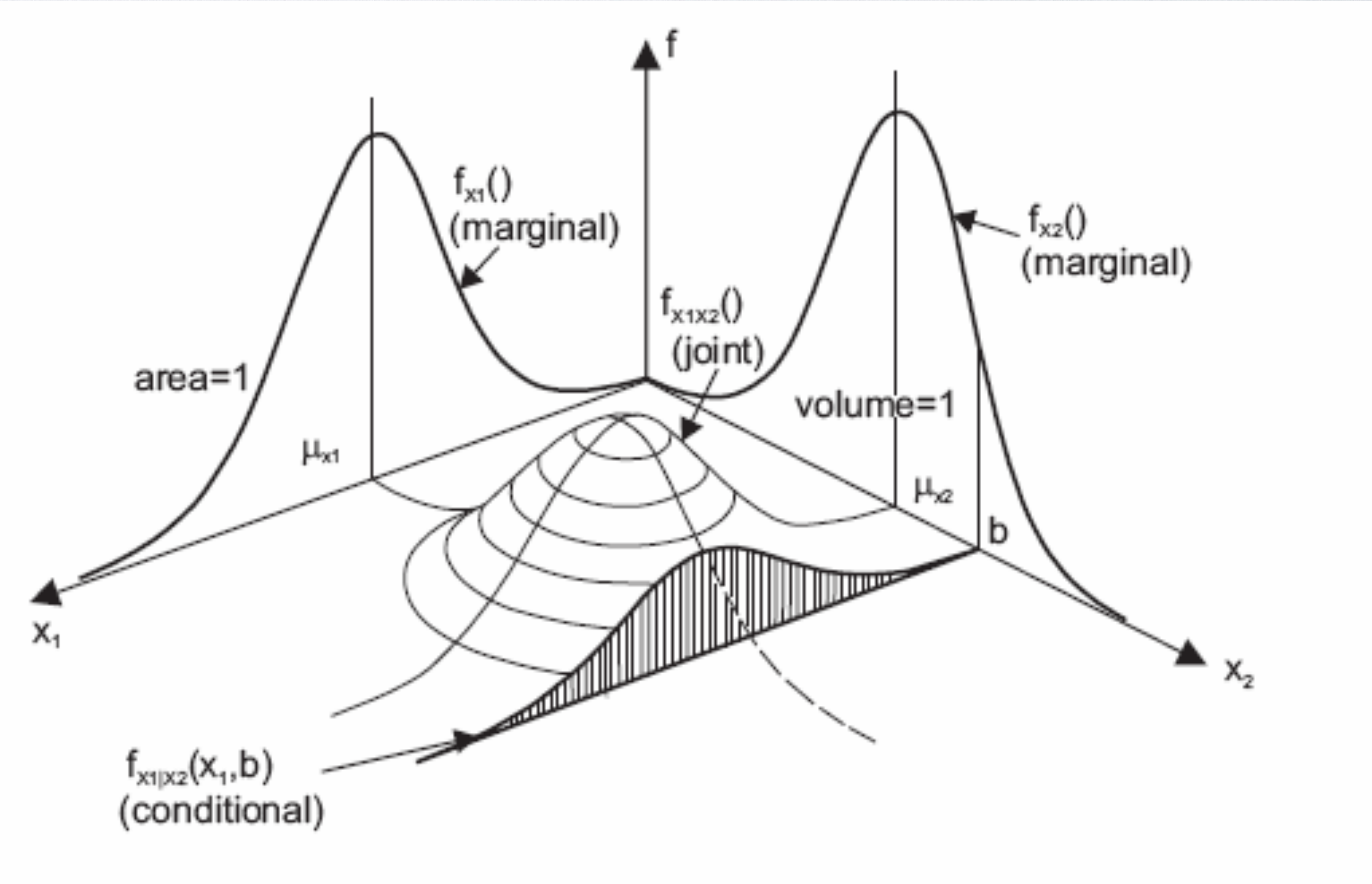
Probabilities 101

- A distribution over a finite set S is a function $\mu : S \to [0,1]$ such that $\sum_{s \in S} \mu(s) = 1$

- Expected value of a (discrete) random variable X in distribution μ is $\sum_{v} \mu(X = v) \cdot v$

- Marginal distribution $f_{X_1}(x_1) = \sum_{x_2 \in X_2} f_{X_1,X_2}(x_1, x_2)$

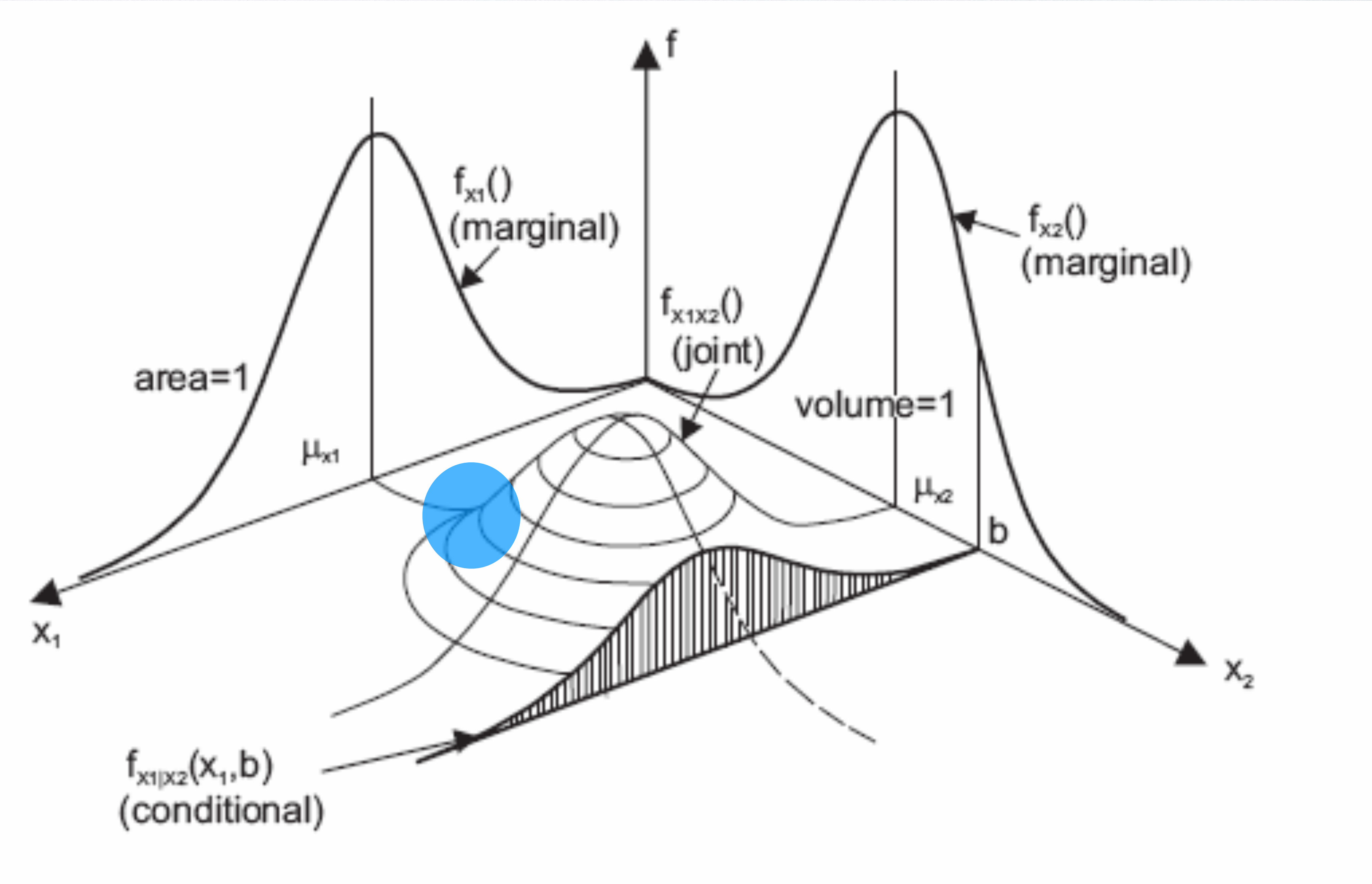
$$f_{X_2}(x_2) = \sum_{x_1 \in X_1} f_{X_1,X_2}(x_1, x_2)$$



Probabilities 101

- A distribution over a finite set S is a function $\mu : S \to [0,1]$ such that $\sum_{s \in S} \mu(s) = 1$
- Expected value of a (discrete) random variable X in distribution μ is $\sum_{v} \mu(X = v) \cdot v$
- Marginal distribution $f_{X_1}(x_1) = \sum_{x_2 \in X_2} f_{X_1,X_2}(x_1, x_2)$

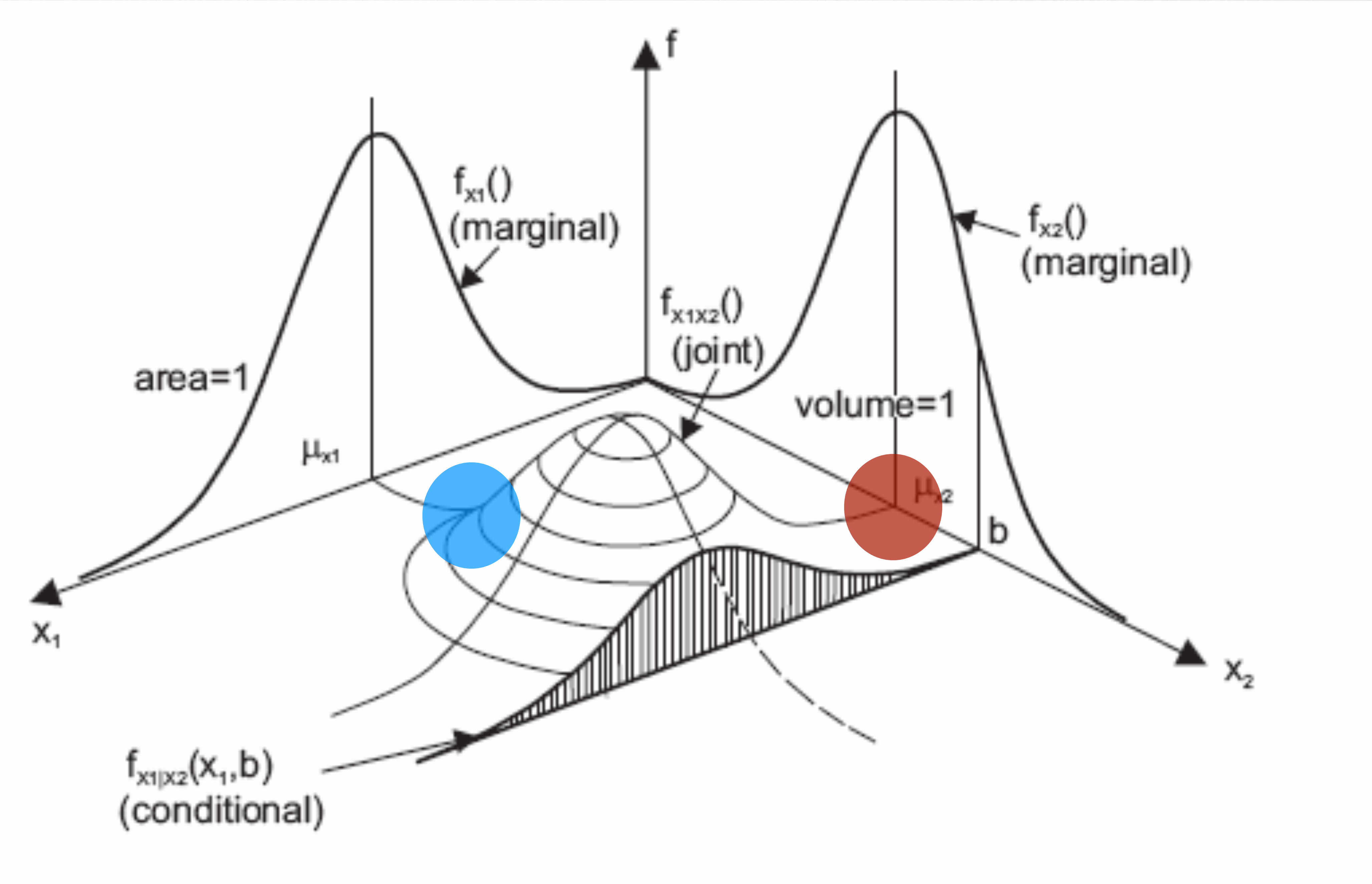
$$f_{X_2}(x_2) = \sum_{x_1 \in X_1} f_{X_1,X_2}(x_1, x_2)$$



Probabilities 101

- A distribution over a finite set S is a function $\mu : S \to [0,1]$ such that $\sum_{s \in S} \mu(s) = 1$
- Expected value of a (discrete) random variable X in distribution μ is $\sum_{v} \mu(X = v) \cdot v$
- Marginal distribution $f_{X_1}(x_1) = \sum_{x_2 \in X_2} f_{X_1,X_2}(x_1, x_2)$

$$f_{X_2}(x_2) = \sum_{x_1 \in X_1} f_{X_1,X_2}(x_1, x_2)$$



Negative Dependence

Negative Dependence

Negative Covariance

Negative Regression

Negative Association (NA)

Negative Right Orthant Dependence

Negative Quadrant Dependence

Negative Association

Negative Association

Let $X_1, \ldots, X_n$ be non-negative random variables.

A. Negative Covariance:

For any $I \subseteq \{1, \ldots, n\}$, $\mathbb{E}[\prod_{i \in I} X_i] \leq \prod_{i \in I} \mathbb{E}[X_i]$

Negative Association

Let $X_1, \ldots, X_n$ be non-negative random variables.

A. Negative Covariance:

For any $I \subseteq \{1,\ldots,n\}$, $\mathbb{E}[\prod_{i \in I} X_i] \leq \prod_{i \in I} \mathbb{E}[X_i]$

$$0.5 \cdot 0 + 0.5 \cdot 0 \leq 0.5 \cdot 0.5$$

		coin 1	
		0	**1**
coin 2	**0**	0	0.5
	1	0.5	0

Negative Association

Let $X_1, \ldots, X_n$ be non-negative random variables.

A. Negative Covariance:

For any $I \subseteq \{1,\ldots,n\}$, $\mathbb{E}[\prod_{i\in I} X_i] \leq \prod_{i\in I} \mathbb{E}[X_i]$

$0.5 \cdot 0 + 0.5 \cdot 0 \leq 0.5 \cdot 0.5$

		coin 1	
		0	**1**
coin 2	**0**	0	0.5
	1	0.5	0

B. For any $I \subseteq \{1,\ldots,n\}$, for any family of non-negative monoto
functions $\{f_i\}_i$ that are all decreasing or all increasing,
$\mathbb{E}[\prod_{i\in I} f_i(X_i)] \leq \prod_{i\in I} \mathbb{E}[f_i(X_i)]$

Negative Association

Let $X_1, \ldots, X_n$ be non-negative random variables.

A. Negative Covariance:

For any $I \subseteq \{1,\ldots,n\}$, $\mathbb{E}[\prod_{i\in I} X_i] \leq \prod_{i \in I} \mathbb{E}[X_i]$

$0.5 \cdot 0 + 0.5 \cdot 0 \leq 0.5 \cdot 0.5$

coin 1

coin 2	0	1
0	0	0.5
1	0.5	0

B. For any $I \subseteq \{1,\ldots,n\}$, for any family of non-negative monoto functions $\{f_i\}_i$ that are all decreasing or all increasing,

$\mathbb{E}[\prod_{i\in I} f_i(X_i)] \leq \prod_{i\in I} \mathbb{E}[f_i(X_i)]$

$\prod_{i \in J \subseteq I} f_i : \mathbb{R}^{|J|} \to \mathbb{R}$

Negative Association

Let $X_1, \ldots, X_n$ be non-negative random variables.

A. Negative Covariance:

For any $I \subseteq \{1, \ldots, n\}$, $\mathbb{E}[\prod_{i \in I} X_i] \leq \prod_{i \in I} \mathbb{E}[X_i]$

$0.5 \cdot 0 + 0.5 \cdot 0 \leq 0.5 \cdot 0.5$

coin 1

coin 2	0	1
0	0	0.5
1	0.5	0

B. For any $I \subseteq \{1, \ldots, n\}$, for any family of non-negative monoto functions $\{f_i\}_i$ that are all decreasing or all increasing,

$\mathbb{E}[\prod_{i \in I} f_i(X_i)] \leq \prod_{i \in I} \mathbb{E}[f_i(X_i)]$

$\prod_{i \in J \subseteq I} f_i : \mathbb{R}^{|J|} \to \mathbb{R}$

C. Negative Association:

For any disjoint $I, J \subseteq \{1, \ldots, n\}$, any non-negative functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq$$
$$\mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,

any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Examples of NA random variables:

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and
$g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Examples of NA random variables:

- Deterministic variables

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Examples of NA random variables:

- Deterministic variables
- Independent random variables

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1

one-hot vectors

X_1	X_2	X_3
1	0	0
0	1	0
0	0	1

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1
- Uniformly random permutations

one-hot vectors

X_1	X_2	X_3
1	0	0
0	1	0
0	0	1

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$, any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1
- Uniformly random permutations

one-hot vectors

X_1	X_2	X_3
1	0	0
0	1	0
0	0	1

shuffle(cards);
Y_i = cards[i]

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,

any non-negative monotone functions $f : \mathbb{R}^{|I|} \rightarrow \mathbb{R}$ and $g : \mathbb{R}^{|J|} \rightarrow \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq$$
$$\mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

- Subsets of NA variables are NA

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

- Subsets of NA variables are NA

X_1	X_2
1	0
0	1
0	0

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$, any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

- Subsets of NA variables are NA
- Union of independent NA sets is also NA

X_1	X_2
1	0
0	1
0	0

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

- Subsets of NA variables are NA
- Union of independent NA sets is also NA

X_1	X_2
1	0
0	1
0	0

shuffle(cards);
Y_i = cards[i]

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

- Subsets of NA variables are NA
- Union of independent NA sets is also 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

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

- 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

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and
$g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

- 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

$Z_1 = X_1 + Y_1$ $\qquad$ $Z_2 = X_2 \cdot Y_2$

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Closure of Negative Association:

- 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

$Z_1 = X_1 + Y_1$ $Z_2 = X_2 \cdot Y_2$

$\{Z_1, Z_2, Y_3, \ldots, Y_n\}$ satisfies NA

NA: For any disjoint $I, J \subseteq \{1, \ldots, n\}$,
any non-negative monotone functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both decreasing or both increasing,

$$\mathbb{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \leq \mathbb{E}[f(X_i, i \in I)] \cdot \mathbb{E}[g(X_j, j \in J)]$$

Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    bin = uniform([0,1,2])
    loads[bin] = loads[bin] + 1
overflow = [n >= 10 for n in loads]
```

Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    bin = uniform([0,1,2])            new_load = one-hot(3)
    loads[bin] = loads[bin] + 1       loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Deterministic NA

Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Deterministic NA

Inductive Hypothesis

Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Deterministic NA

Inductive Hypothesis

One-Hot NA

Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Deterministic NA

Inductive Hypothesis

One-Hot NA

loads	new_load

Proving NA for our example

loads	new_load

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Deterministic NA

Inductive Hypothesis
One-Hot NA
$\Big\}$ Independent union NA

Proving NA for our example

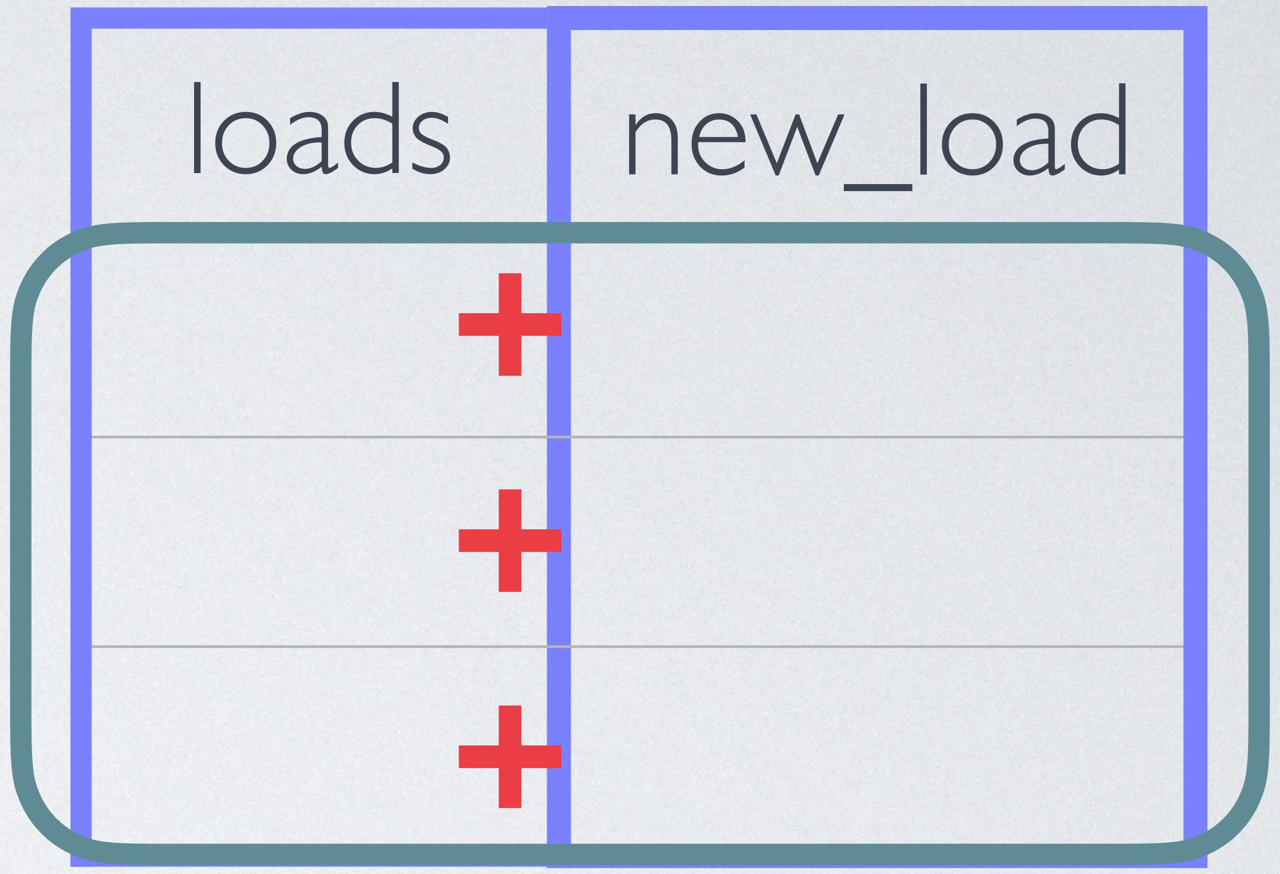
```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Deterministic NA

Inductive Hypothesis

One-Hot NA

Independent union NA



Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

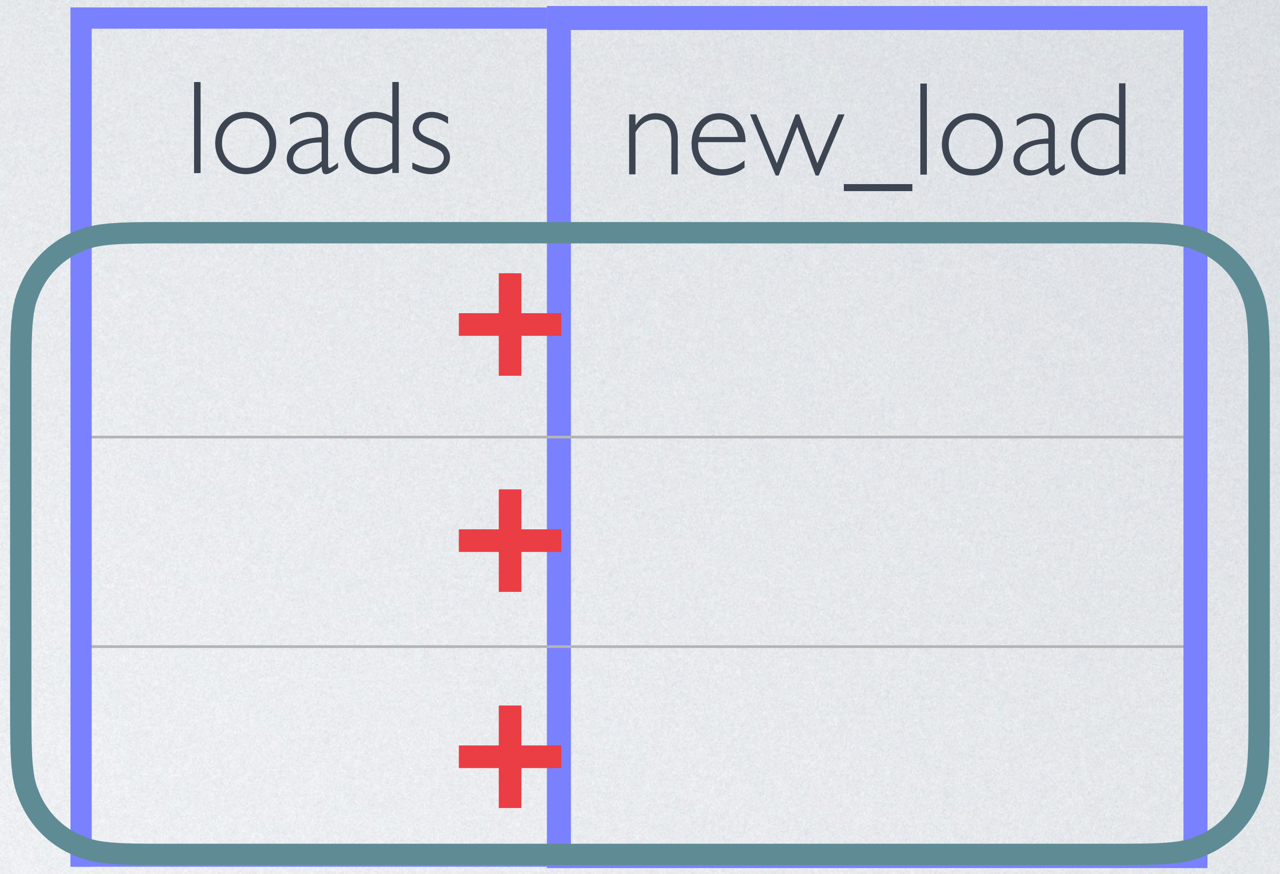
Deterministic NA

Inductive Hypothesis

One-Hot NA

Independent union NA

Monotone map NA



Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

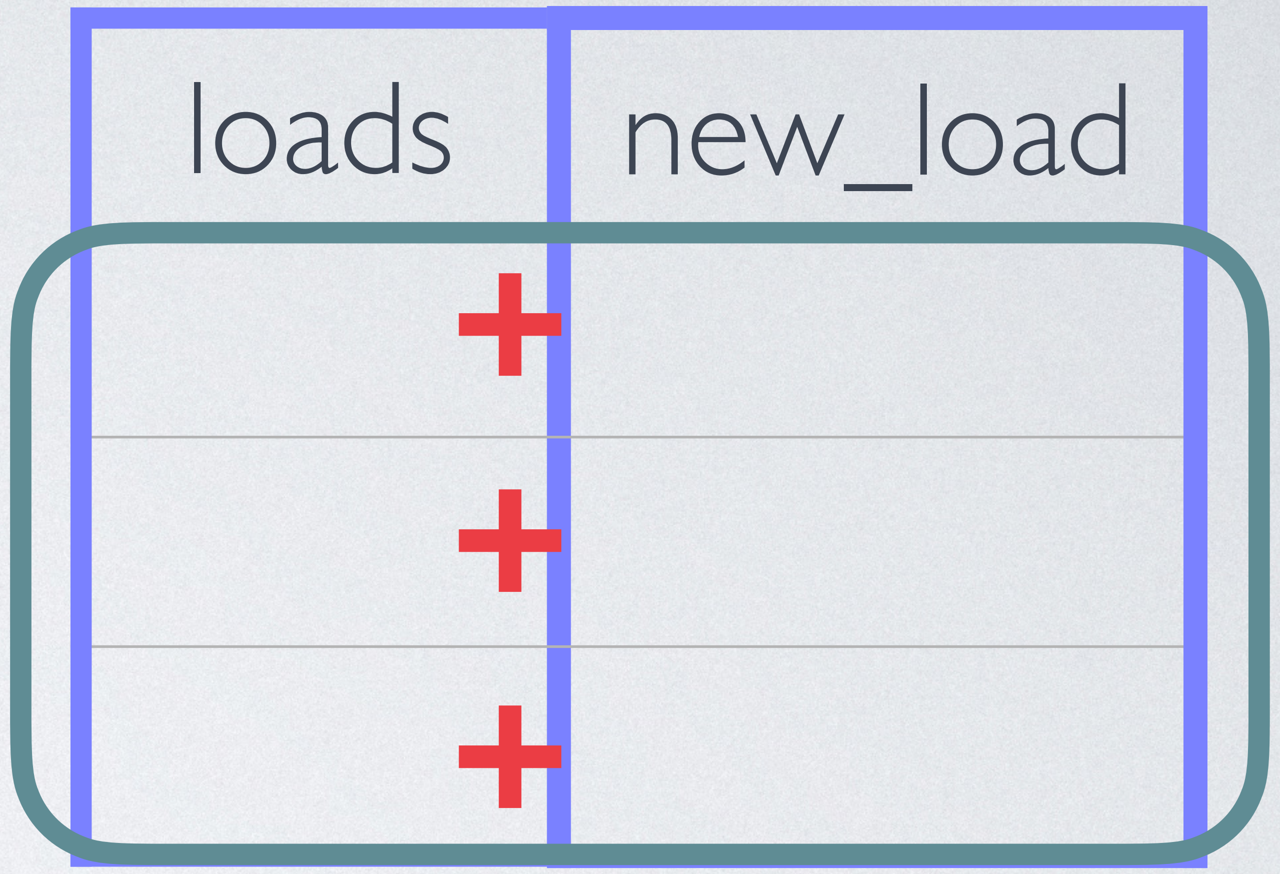
Deterministic NA

Inductive Hypothesis

One-Hot NA

Independent union NA

Monotone map NA



Proving NA for our example

```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

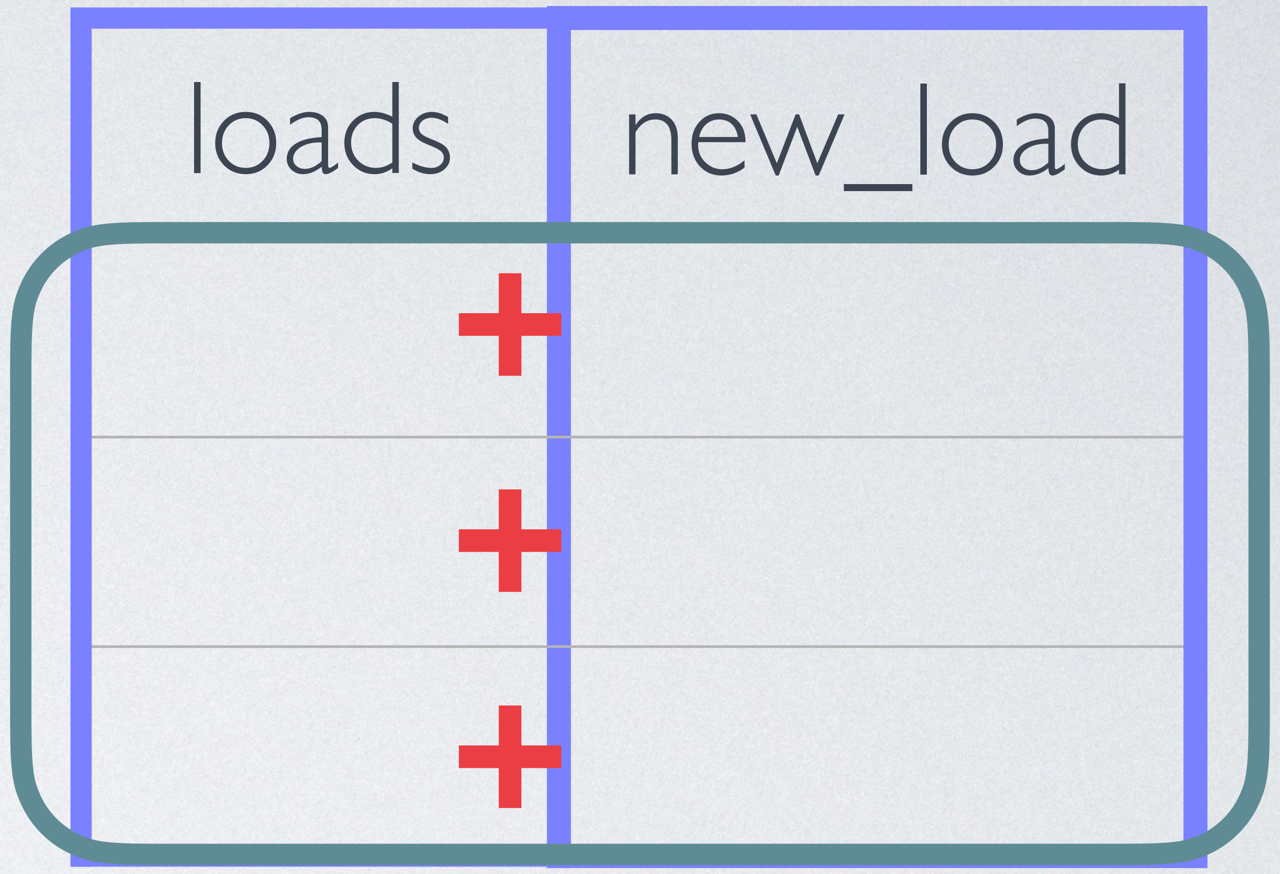
Deterministic NA

Inductive Hypothesis

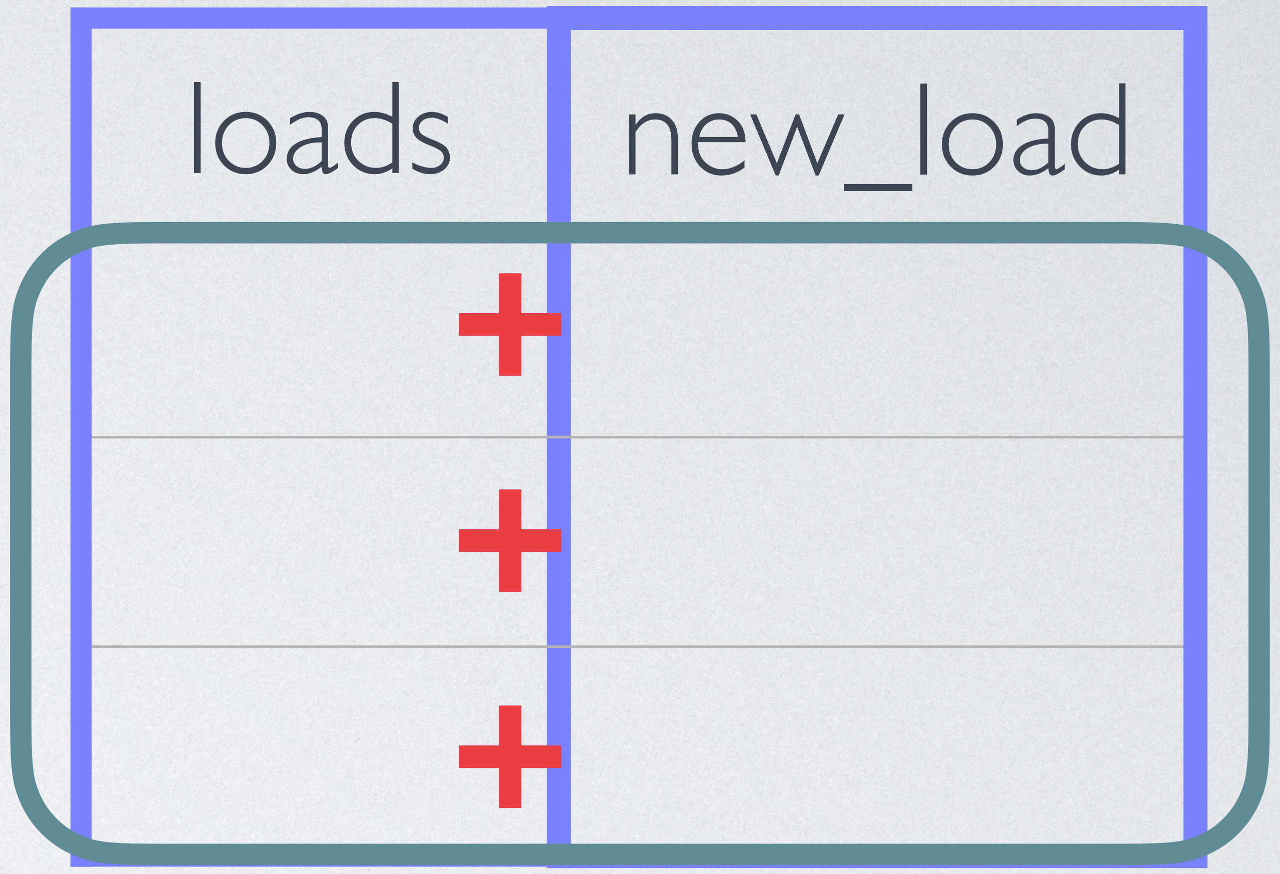
One-Hot NA

Independent union NA

Monotone map NA



Proving NA for our example




```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Deterministic NA

Inductive Hypothesis

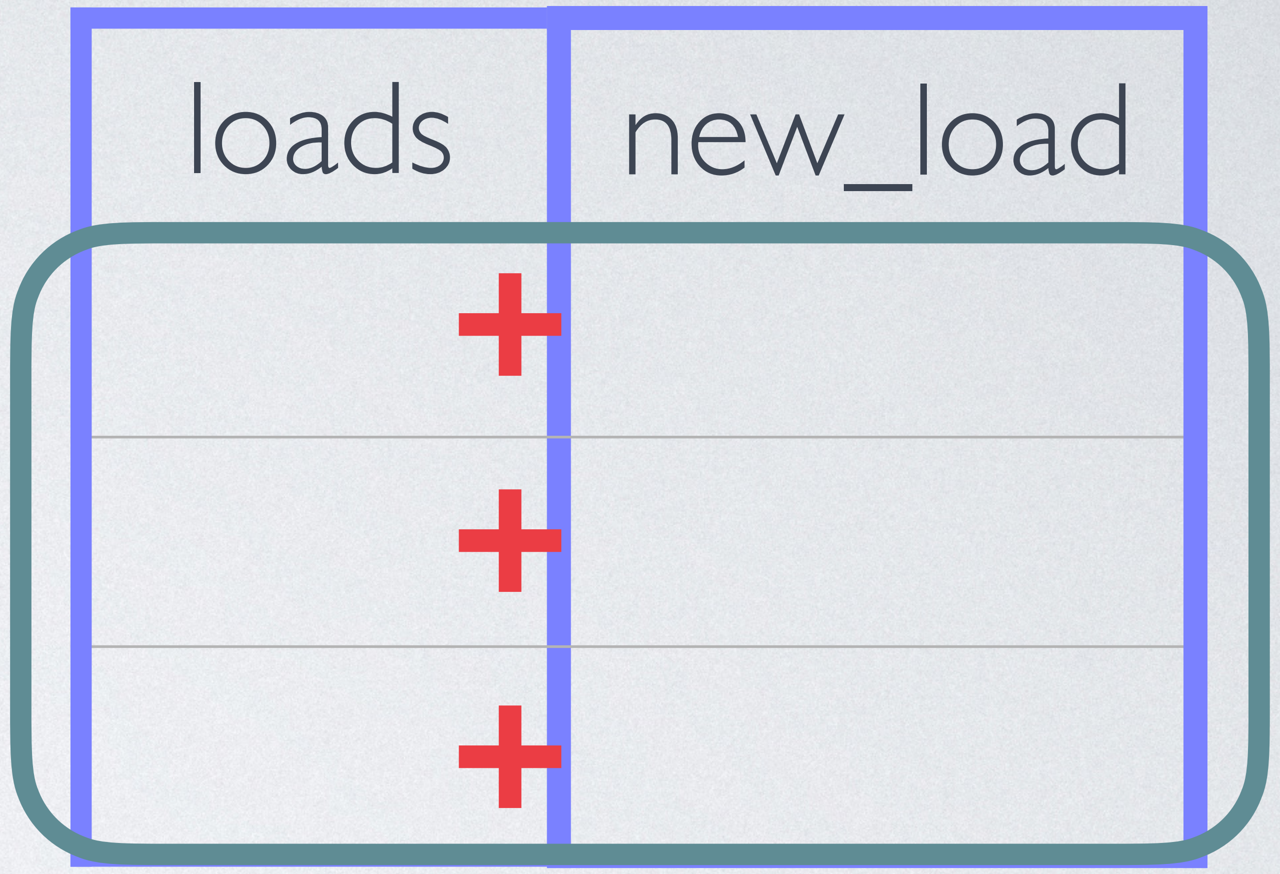
One-Hot NA

Independent union NA

Monotone map NA

Monotone map NA

Proving NA for our example




```
# Simple scheduler
tasks = [A, ..., Z]
loads = [0, 0, 0]
for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load
overflow = [n >= 10 for n in loads]
```

Deterministic NA

Inductive Hypothesis

One-Hot NA

Independent union NA

Monotone map NA

Monotone map NA

Entries in overflow are NA!

PROBABILISTIC SEPARATION LOGIC

Separation Logic

Separation Logic

- **A flexible framework to reason about sharing and separation**

Separation Logic

- **A flexible framework to reason about sharing and separation**
- **Program logic**
  - **Judgement:** $\{P\}C\{Q\}$

Separation Logic

- **A flexible framework to reason about sharing and separation**
- **Program logic**
  - **Judgement:** $\{P\}C\{Q\}$
- **Assertion logic (logic of Bunched Implications, BI)**
  - $P, Q ::= p \in \mathscr{AP} \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q \mid P * Q \mid P -\!\!* Q$

Separation Logic

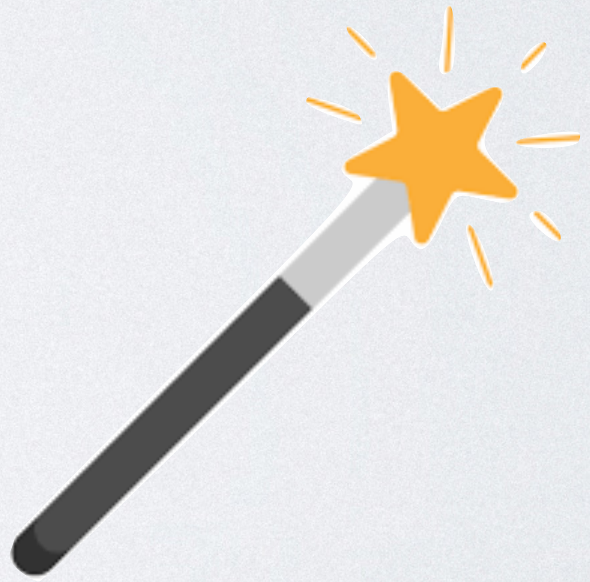
- **A flexible framework to reason about sharing and separation**
- **Program logic**
  - **Judgement:** $\{P\}C\{Q\}$
- **Assertion logic (logic of Bunched Implications, BI)**
  - $P, Q ::= \boxed{p \in \mathcal{AP} \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q} \; P * Q \mid P -\!\!* \, Q$

Separation Logic

- **A flexible framework to reason about sharing and separation**
- **Program logic**
  - **Judgement:** $\{P\}C\{Q\}$
- **Assertion logic (logic of Bunched Implications, BI)**
  - $P, Q ::= \boxed{p \in \mathscr{A}\mathscr{P} \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q} \; P * Q \mid P -\!\!* Q$

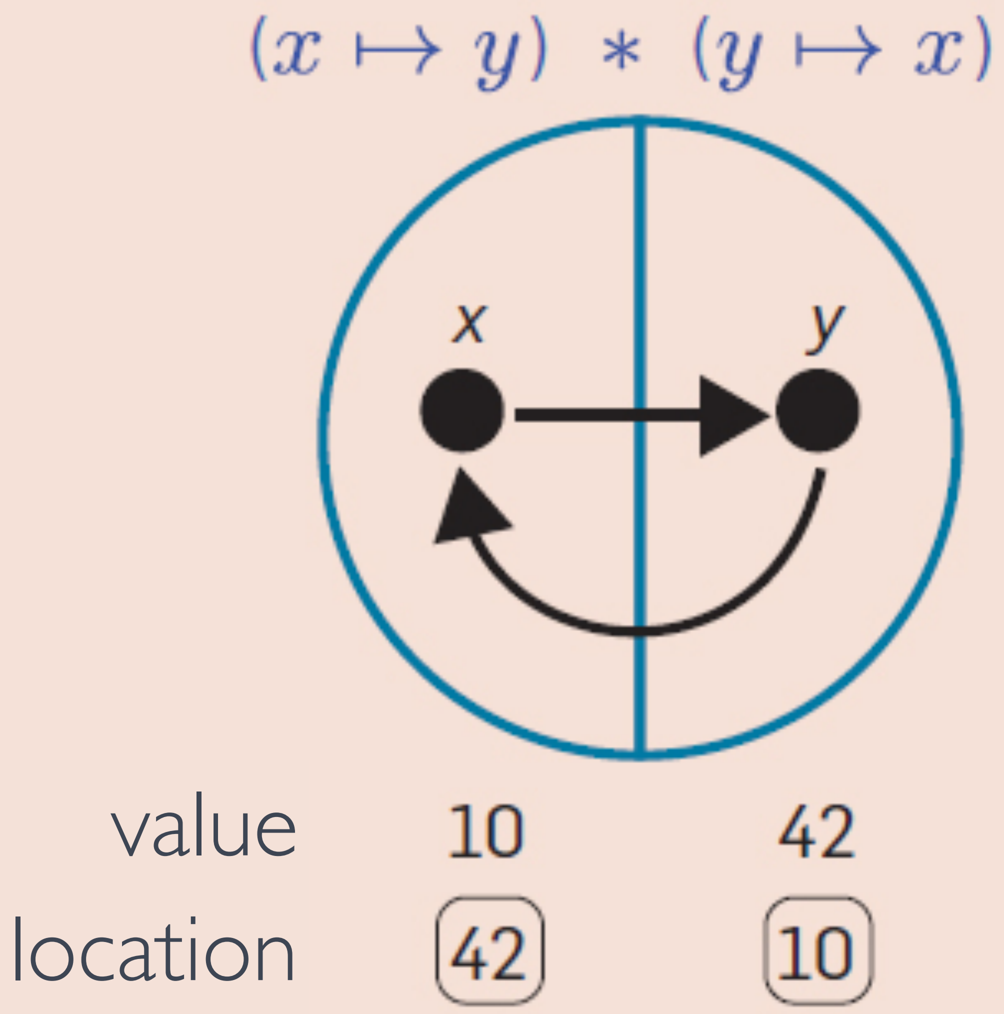
Separation Logic

- **A flexible framework to reason about sharing and separation**
- **Program logic**
  - **Judgement:** $\{P\}C\{Q\}$
- **Assertion logic (logic of Bunched Implications, BI)**
  - $P, Q ::= \boxed{p \in \mathscr{AP} \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q} \; P * Q \mid P \mathrel{-\!\!*} Q$

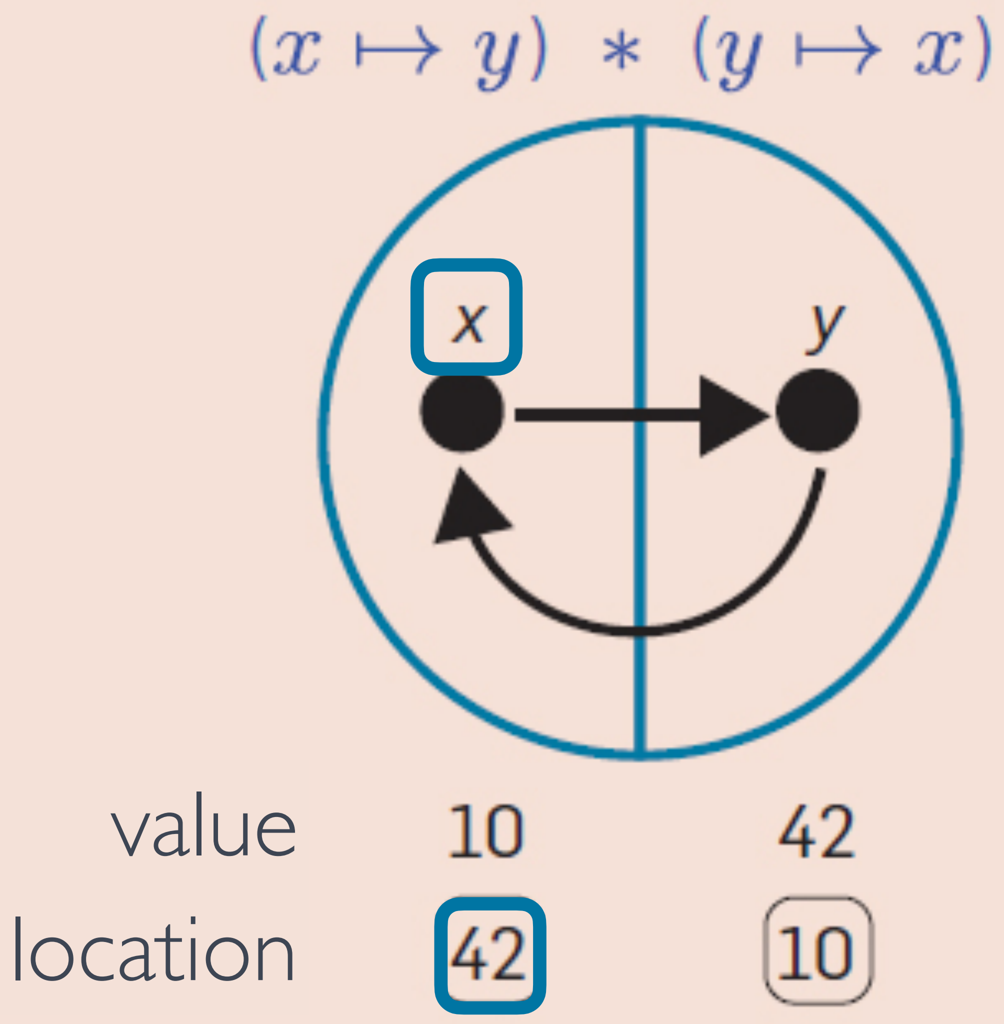


Separation Logic

- A flexible framework to reason about sharing and separation
- Program logic
  - Judgement: $\{P\}C\{Q\}$
- Assertion logic (logic of Bunched Implications, BI)
  - $P, Q ::= \boxed{p \in \mathscr{AP} \mid \top \mid \bot \mid P \wedge Q \mid P \vee Q \mid P \Rightarrow Q} \; P * Q \mid P \mathrel{-\!\!*} Q$
- Outline:
  - Intuition of $P * Q$
  - Semantics of BI
  - Programs and atomic propositions
  - Proof rules of program logic

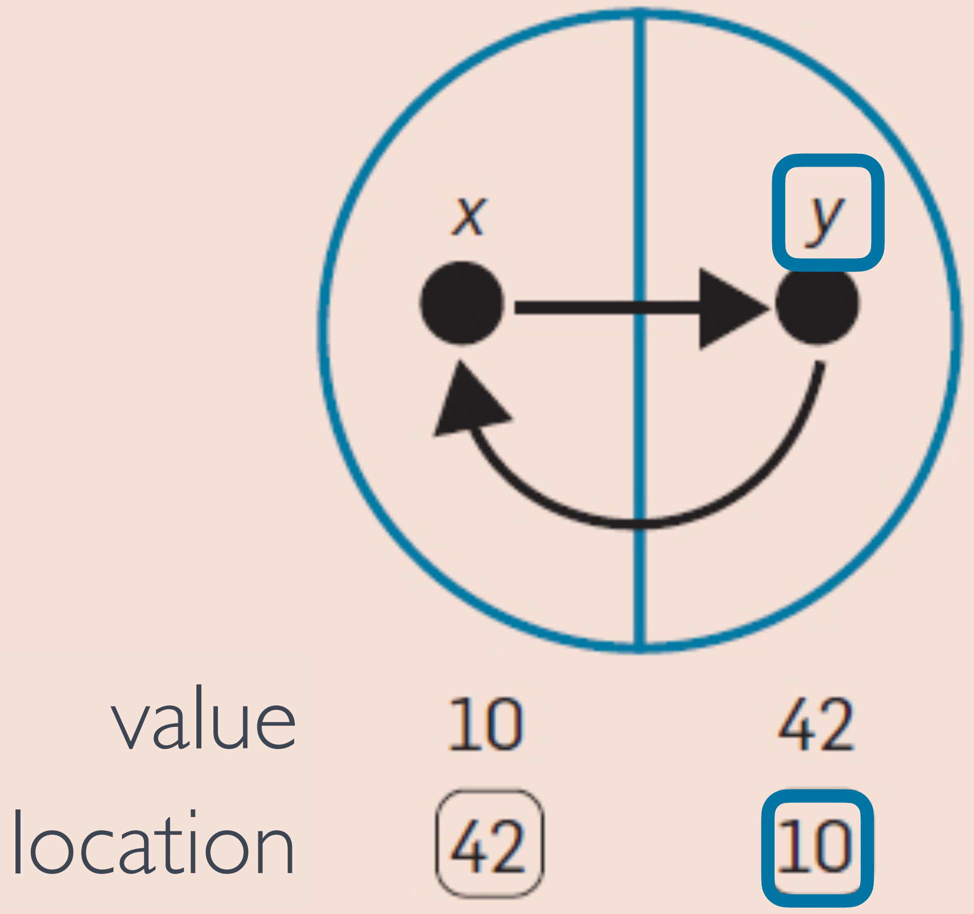


Adapted from an image in "Separation Logic" in CACM

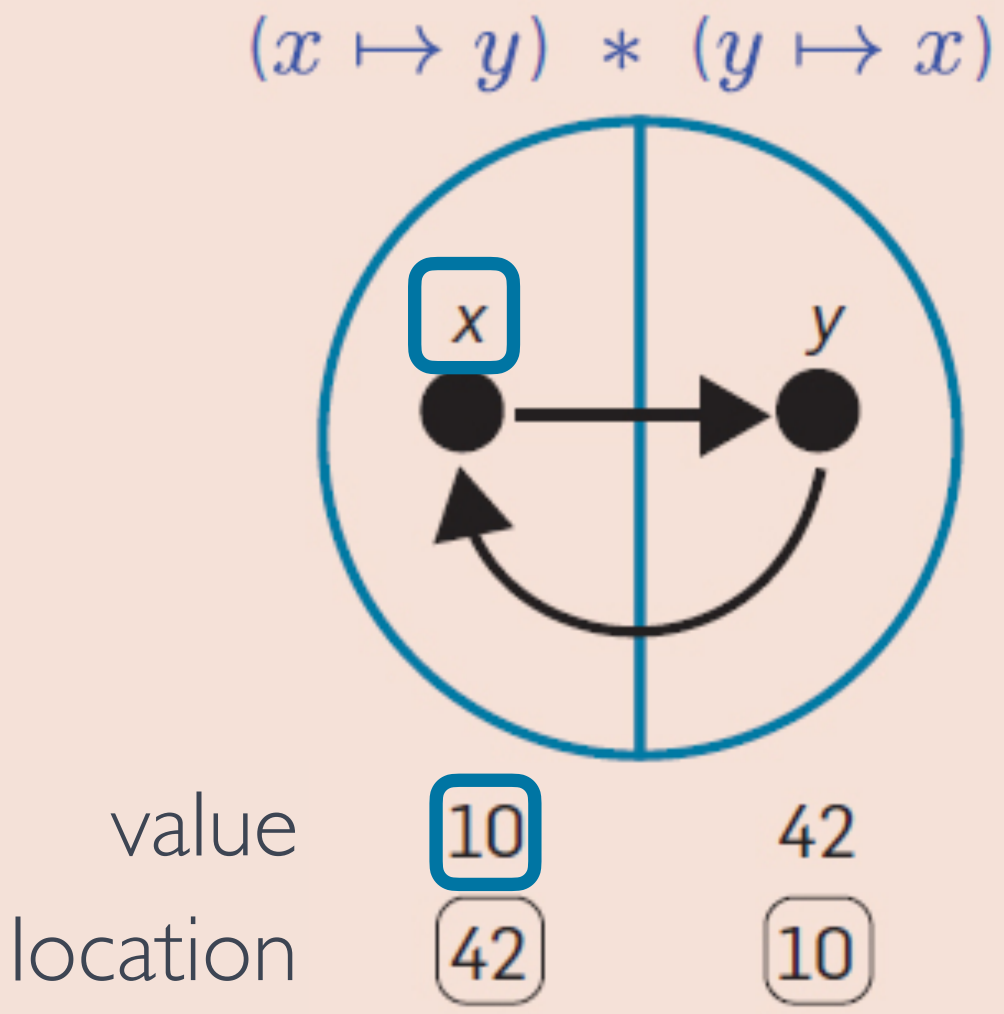


Adapted from an image in "Separation Logic" in CACM

$(x \mapsto y) * (y \mapsto x)$

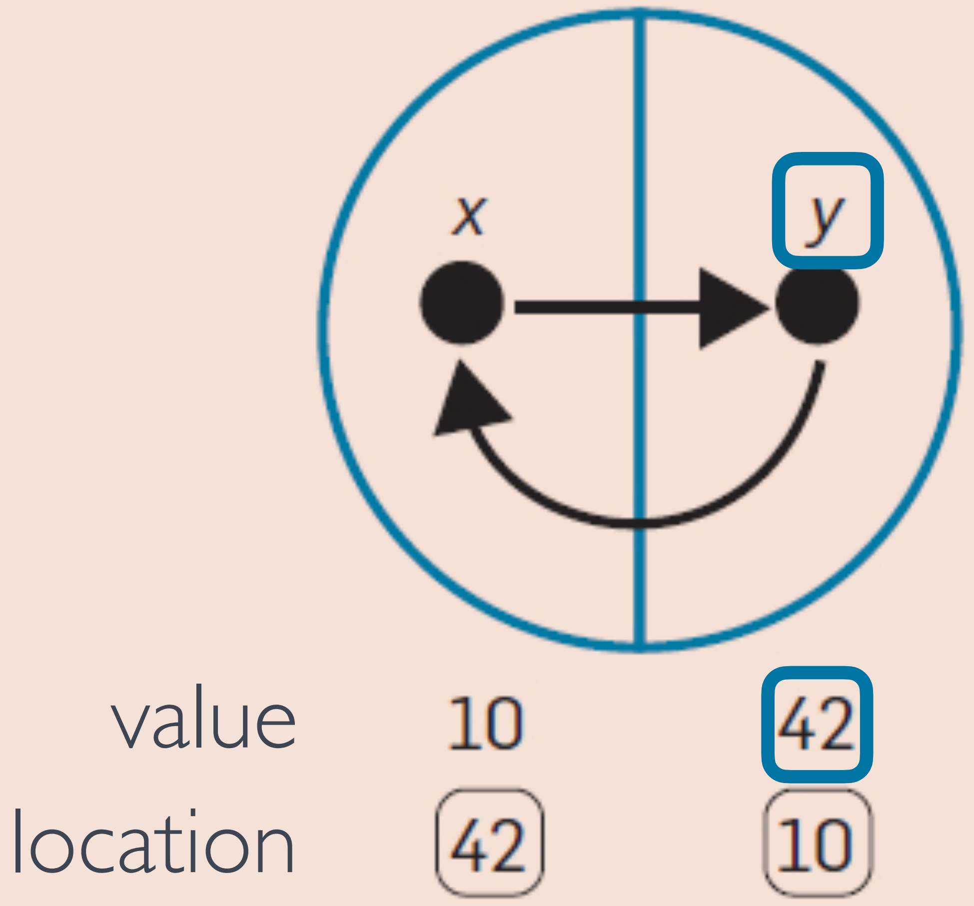


Adapted from an image in "Separation Logic" in CACM

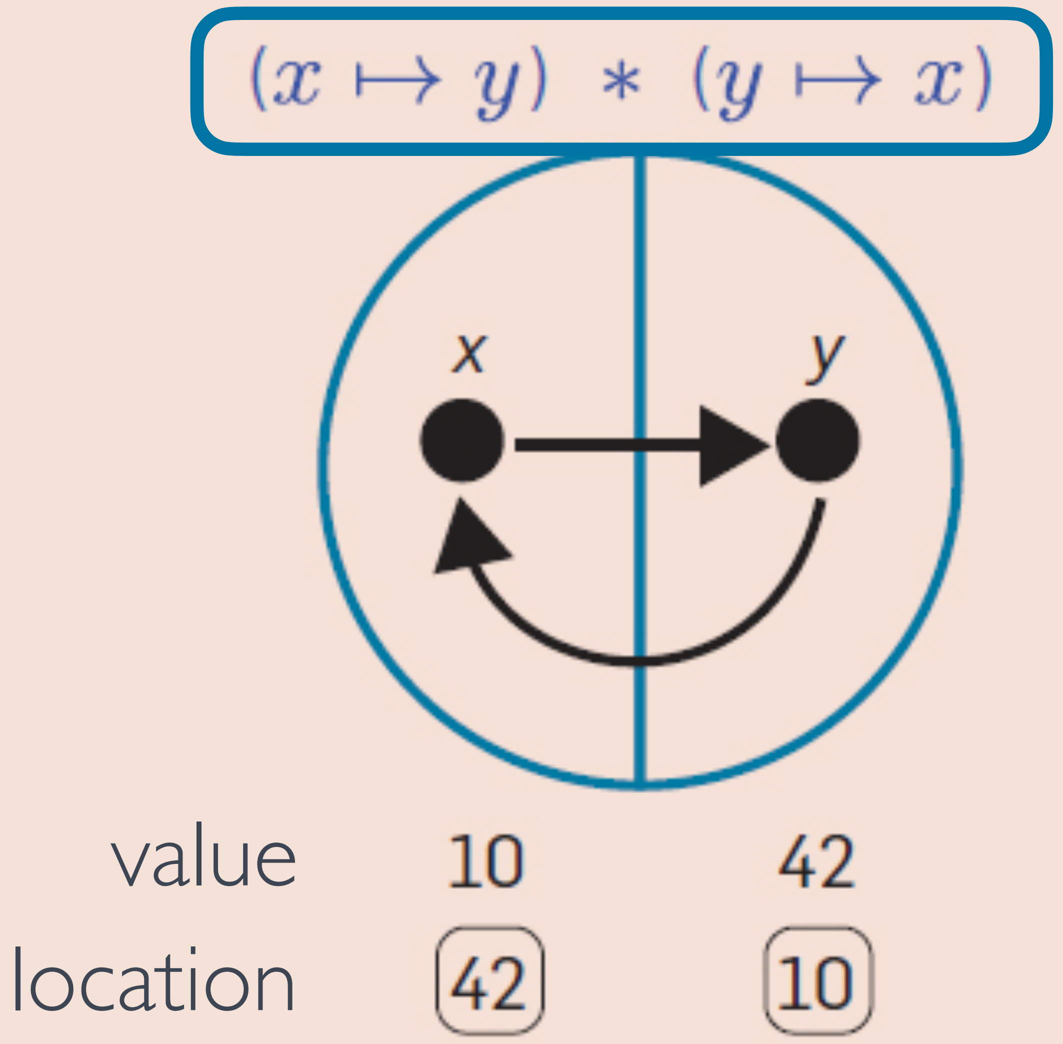


Adapted from an image in "Separation Logic" in CACM

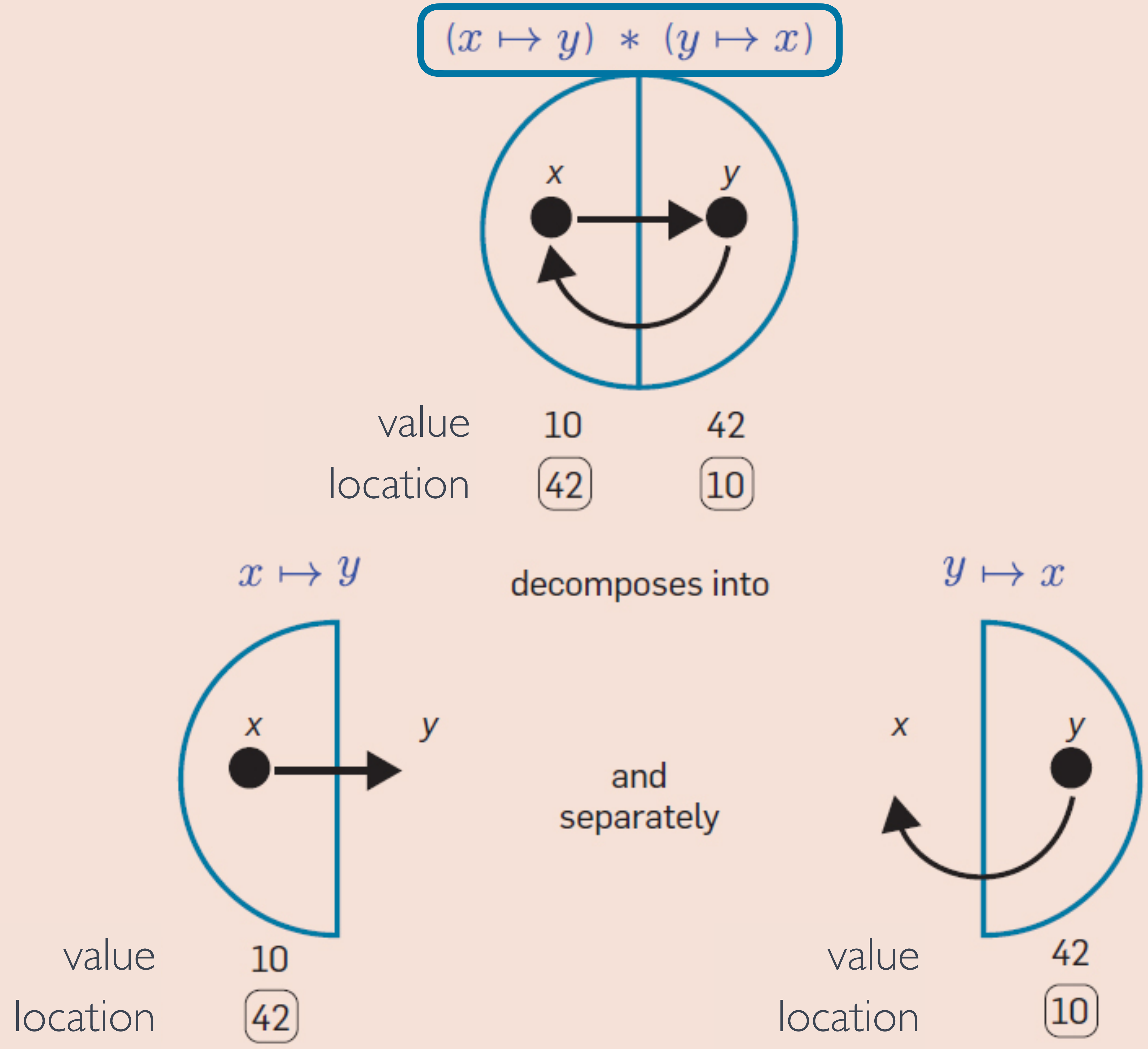
$(x \mapsto y) * (y \mapsto x)$



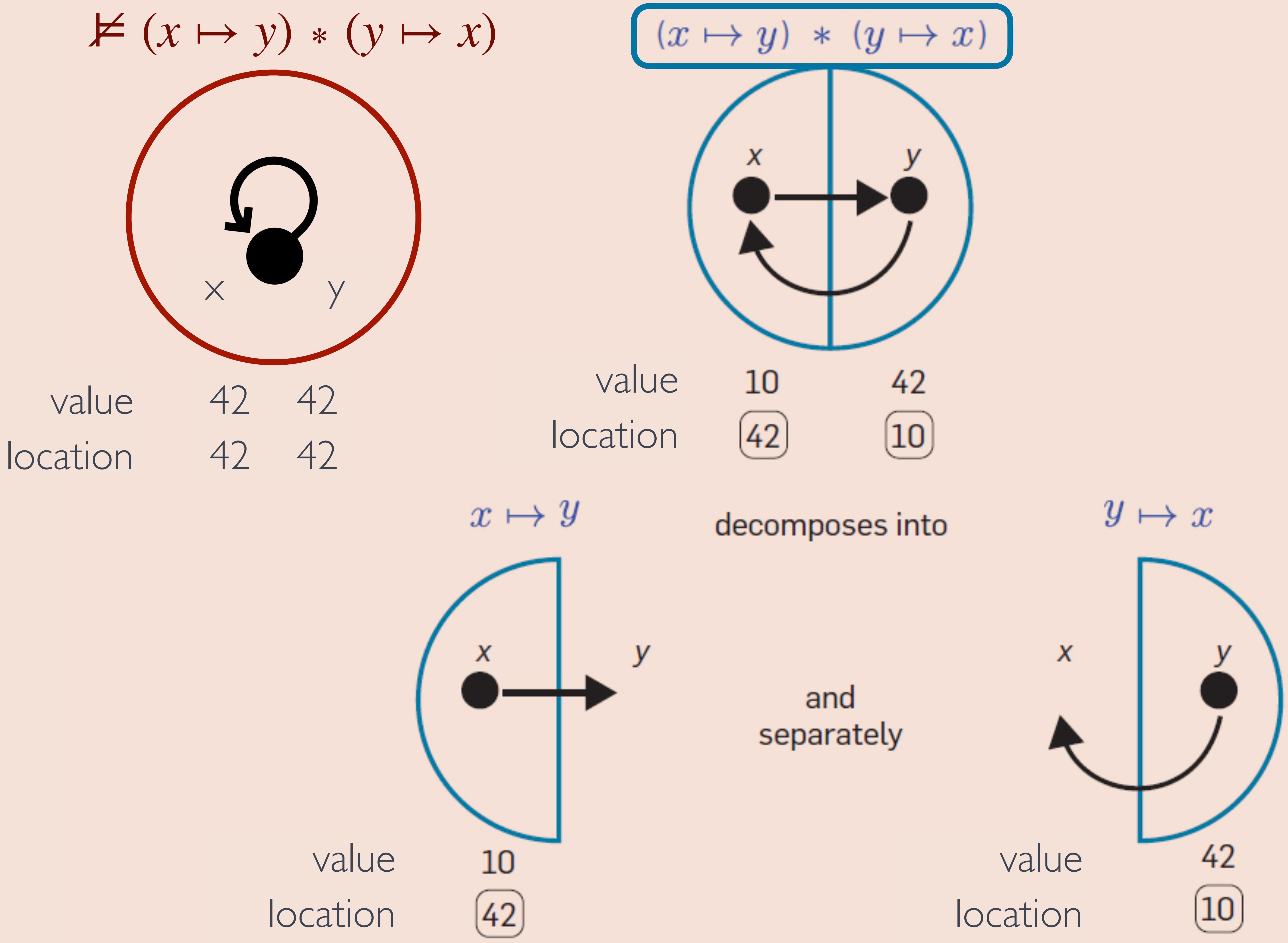
Adapted from an image in "Separation Logic" in CACM



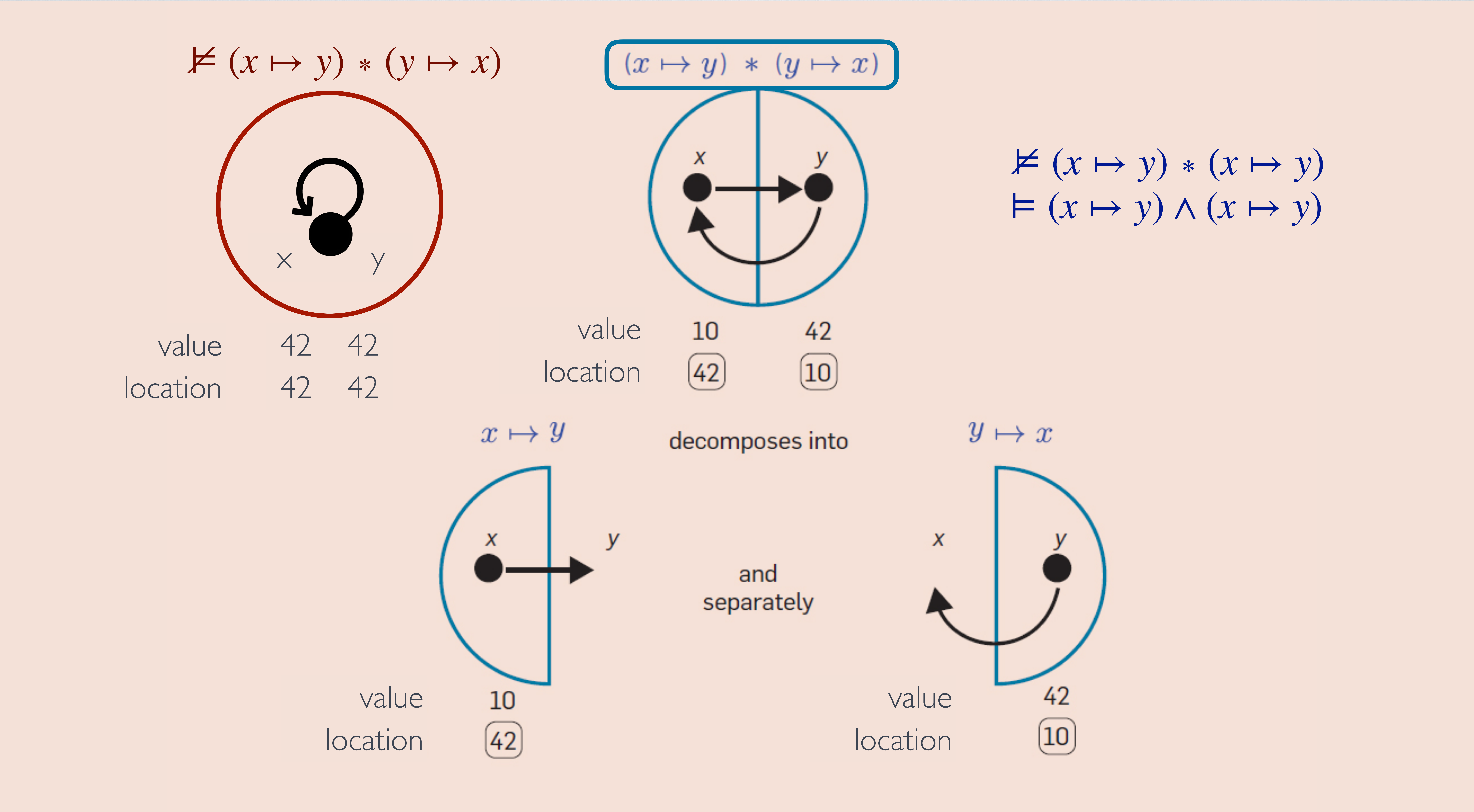
Adapted from an image in "Separation Logic" in CACM



Adapted from an image in "Separation Logic" in CACM



Adapted from an image in "Separation Logic" in CACM



Adapted from an image in "Separation Logic" in CACM

Structures for Interpreting BI

- **A Kripke resource monoid is a set M with**
  - **a partial binary operation $\circ : M \times M \rightharpoonup M$ that is**
    - **associative**
    - **commutative**
  - **an identity element $e \in M$**
  - **a pre-order $\sqsubseteq$ on M**

Structures for Interpreting BI

- A Kripke resource monoid is a set M with
  - a partial binary operation $\circ : M \times M \rightharpoonup M$ that is
    - associative: $x \circ (y \circ z) = (x \circ y) \circ z$,
    - commutative
  - an identity element $e \in M$
  - a pre-order $\sqsubseteq$ on M

Structures for Interpreting BI

- A Kripke resource monoid is a set M with
  - a partial binary operation $\circ : M \times M \rightharpoonup M$ that is
    - associative: $x \circ (y \circ z) = (x \circ y) \circ z$,
    - commutative: $y \circ x = x \circ y$,
  - an identity element $e \in M$
  - a pre-order $\sqsubseteq$ on M

Structures for Interpreting BI

- **A Kripke resource monoid is a set M with**
  - **a partial binary operation $\circ : M \times M \rightharpoonup M$ that is**
    - **associative: $x \circ (y \circ z) = (x \circ y) \circ z$,**
    - **commutative: $y \circ x = x \circ y$,**
  - **an identity element $e \in M$: $e \circ x = x \circ e = x$,**
  - **a pre-order $\sqsubseteq$ on M**

Structures for Interpreting BI

- **A Kripke resource monoid is a set M with**
  - **a partial binary operation $\circ : M \times M \rightharpoonup M$ that is**
    - **associative: $x \circ (y \circ z) = (x \circ y) \circ z$,**
    - **commutative: $y \circ x = x \circ y$,**
  - **an identity element $e \in M$: $e \circ x = x \circ e = x$,**
  - **a pre-order $\sqsubseteq$ on M:**
    - **transitive: if $x \sqsubseteq y$ and $y \sqsubseteq z$, then $x \sqsubseteq z$;**
    - **reflexive: $x \sqsubseteq x$ for any x**

Examples of Kripke Resource Monoid

Examples of Kripke Resource Monoid

Distribution model

Examples of Kripke Resource Monoid

Distribution model

- **Let M be the set of distributions over memories,**

Examples of Kripke Resource Monoid

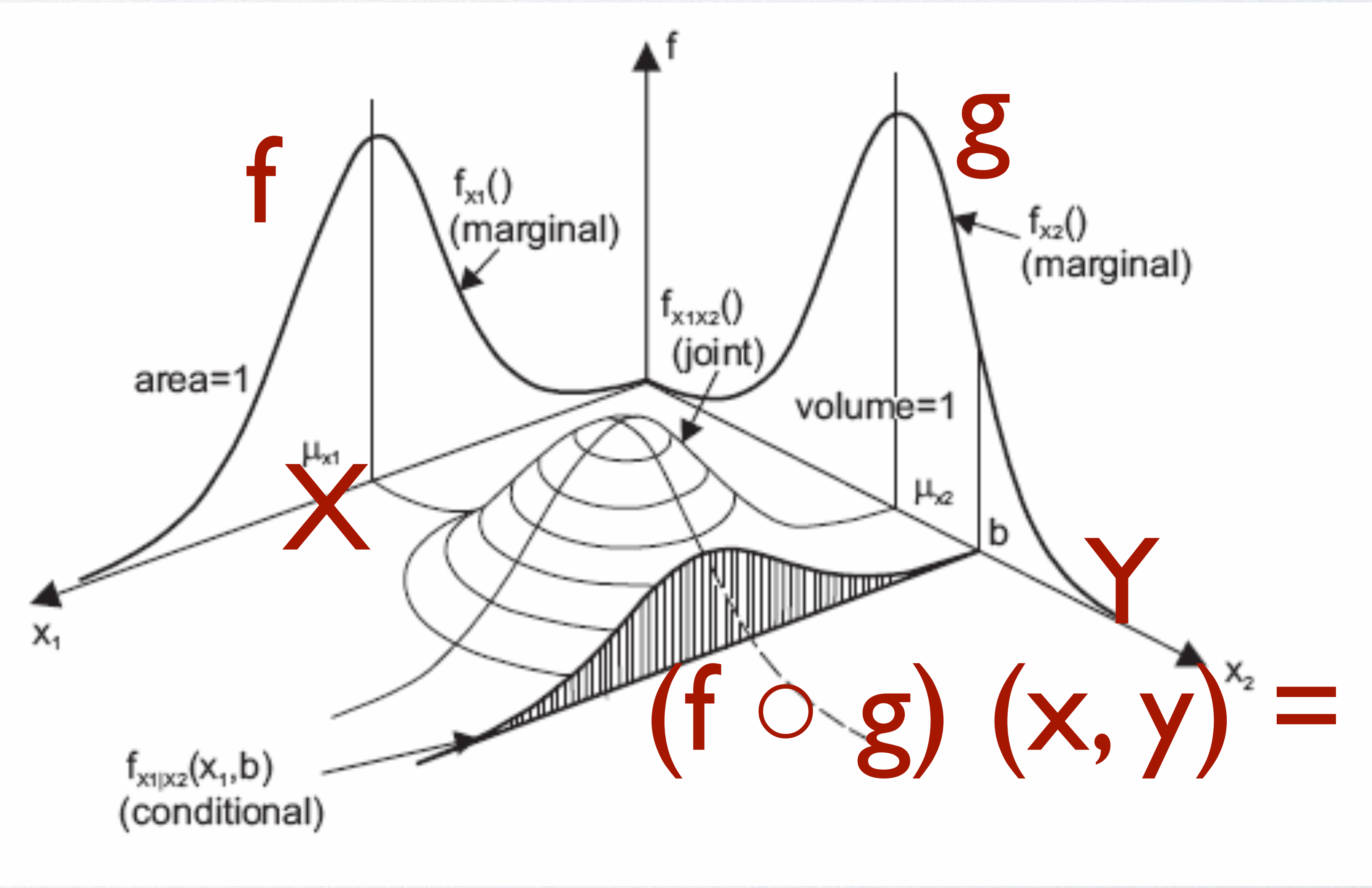
Distribution model

- **Let M be the set of distributions over memories,**
  - **for distributions** $f : X \to [0,1]$ **and** $g : Y \to [0,1]$ **,** $f \circ g$ **defined to be their independent product if** X, Y **are disjoint, otherwise undefined**

Examples of Kripke Resource Monoid

Distribution model

- Let M be the set of distributions over memories,
  - for distributions $f : X \to [0,1]$ and $g : Y \to [0,1]$, $f \circ g$ defined to be their independent product if X, Y are disjoint, otherwise undefined

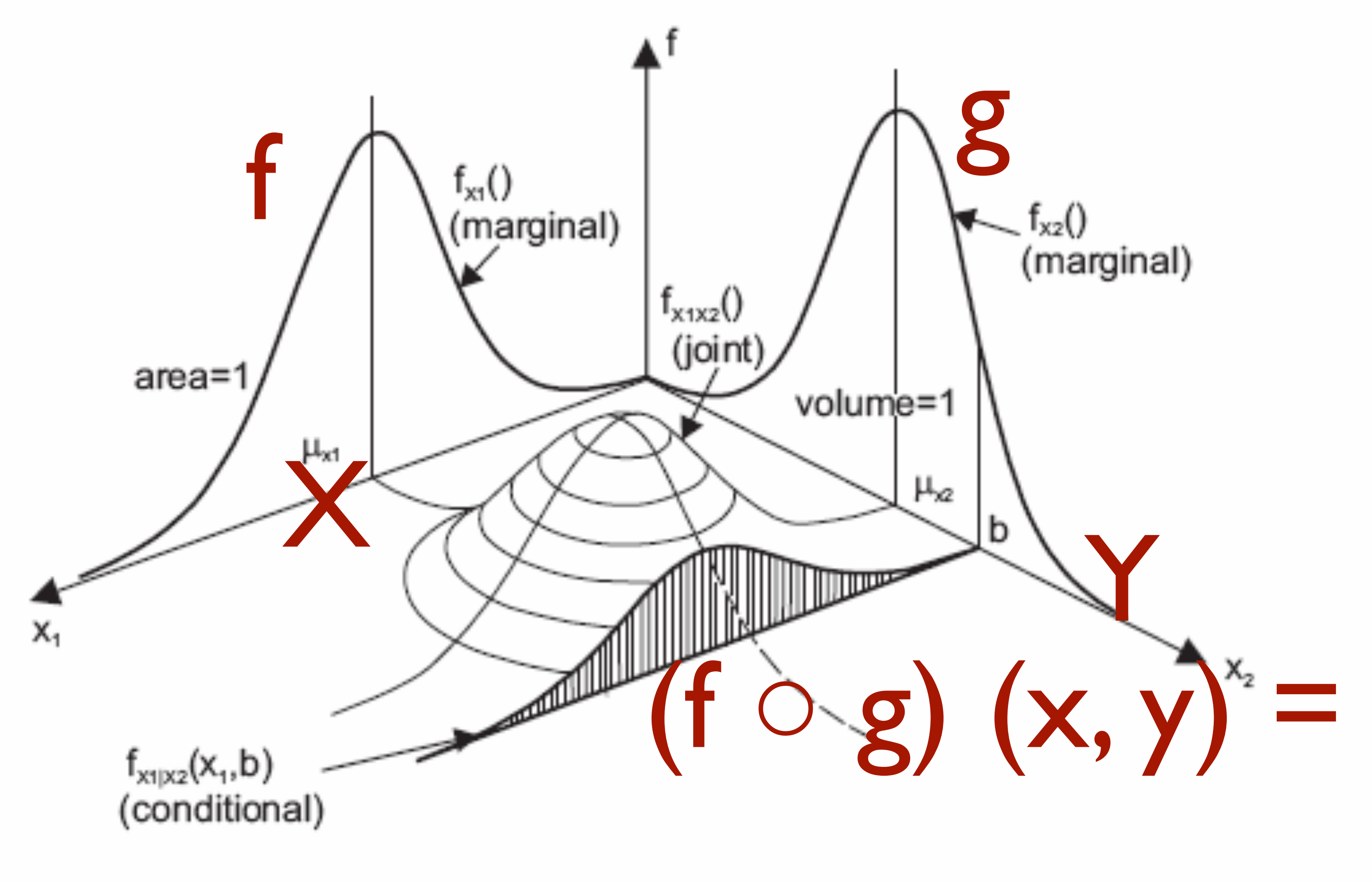


$(f \circ g)\ (x, y) = f(x) \cdot g(y)$

Examples of Kripke Resource Monoid

Distribution model

- **Let M be the set of distributions over memories,**
  - **for distributions** $f : X \to [0,1]$ **and** $g : Y \to [0,1]$ **,** $f \circ g$ **defined to be their independent product if** X, Y **are disjoint, otherwise undefined**
  - **an identity element e: deterministic distribution on empty memory**



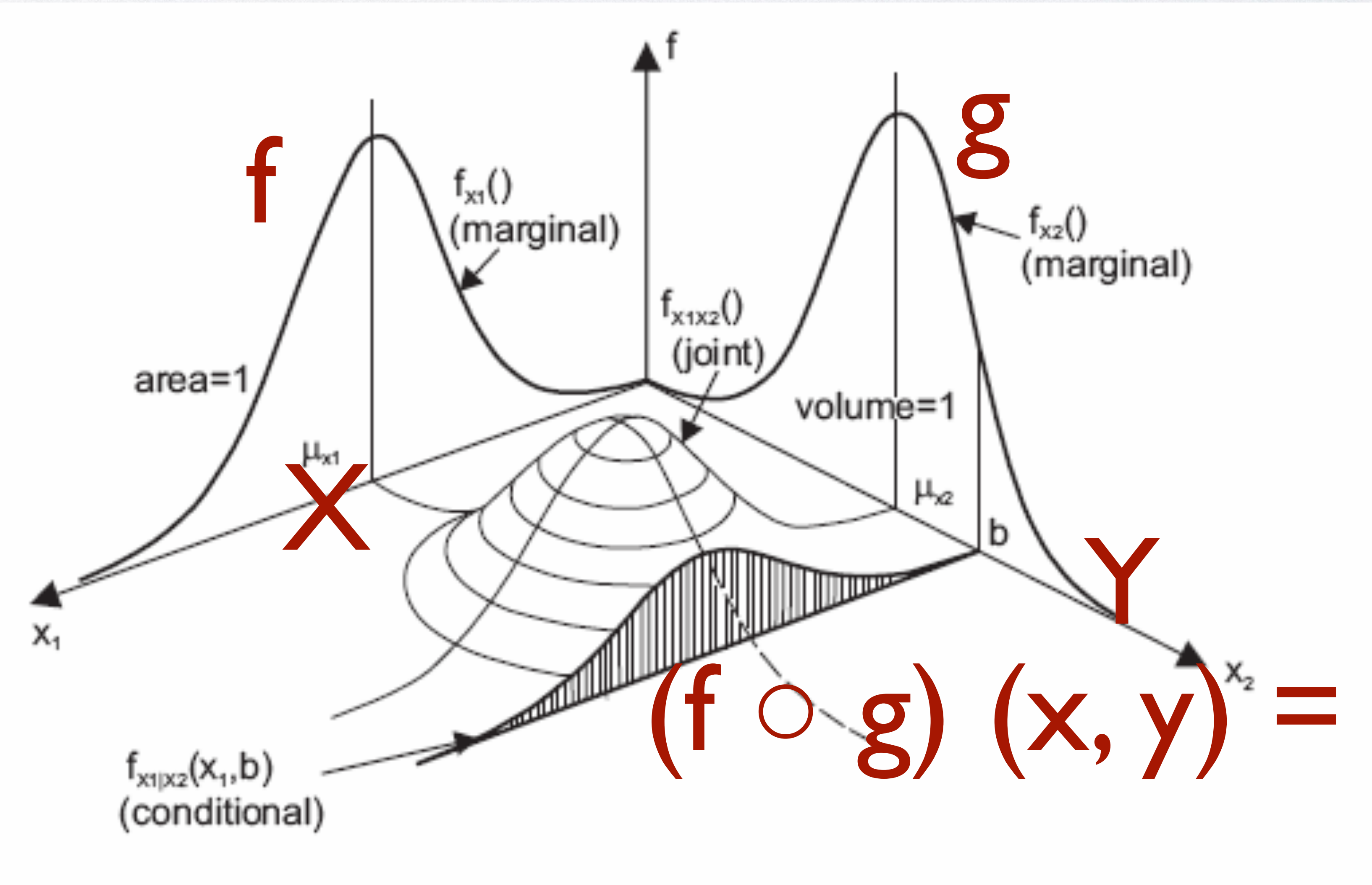
$(f \circ g)\,(x, y) = f(x) \cdot g(y)$

Examples of Kripke Resource Monoid

Distribution model

- Let M be the set of distributions over memories,
  - for distributions $f : X \to [0,1]$ and $g : Y \to [0,1]$, $f \circ g$ defined to be their independent product if X, Y are disjoint, otherwise undefined
  - an identity element e: deterministic distribution on empty memory

$$e(\langle\rangle) = 1$$



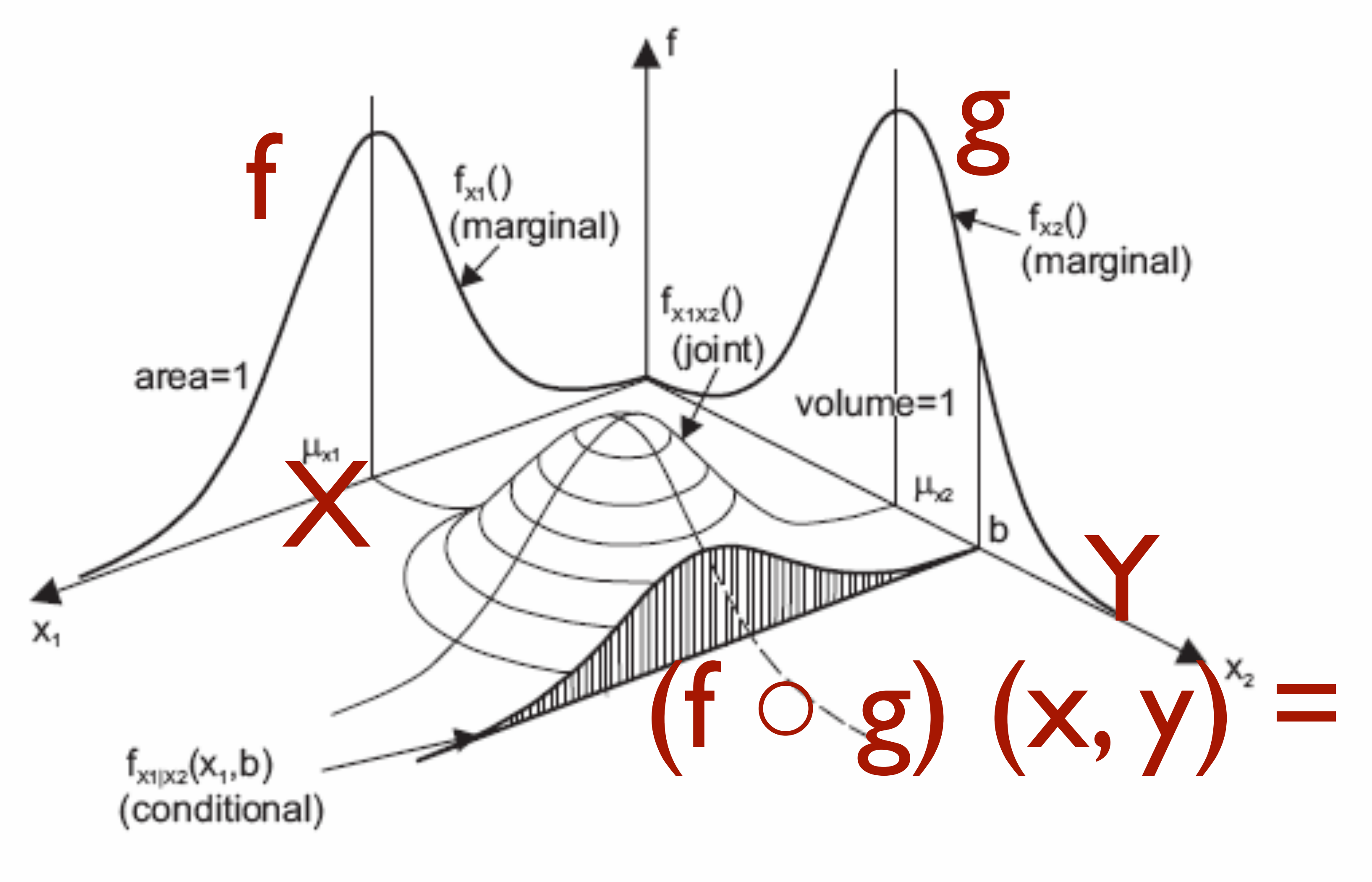
$$(f \circ g)\,(x, y) = f(x) \cdot g(y)$$

Examples of Kripke Resource Monoid

Distribution model

- Let M be the set of distributions over memories,
  - for distributions $f : X \rightarrow [0,1]$ and $g : Y \rightarrow [0,1]$, $f \circ g$ defined to be their independent product if X, Y are disjoint, otherwise undefined
  - an identity element e: deterministic distribution on empty memory
  - f ⊑ h if h marginalizes into f

$$e(\langle\rangle) = 1$$



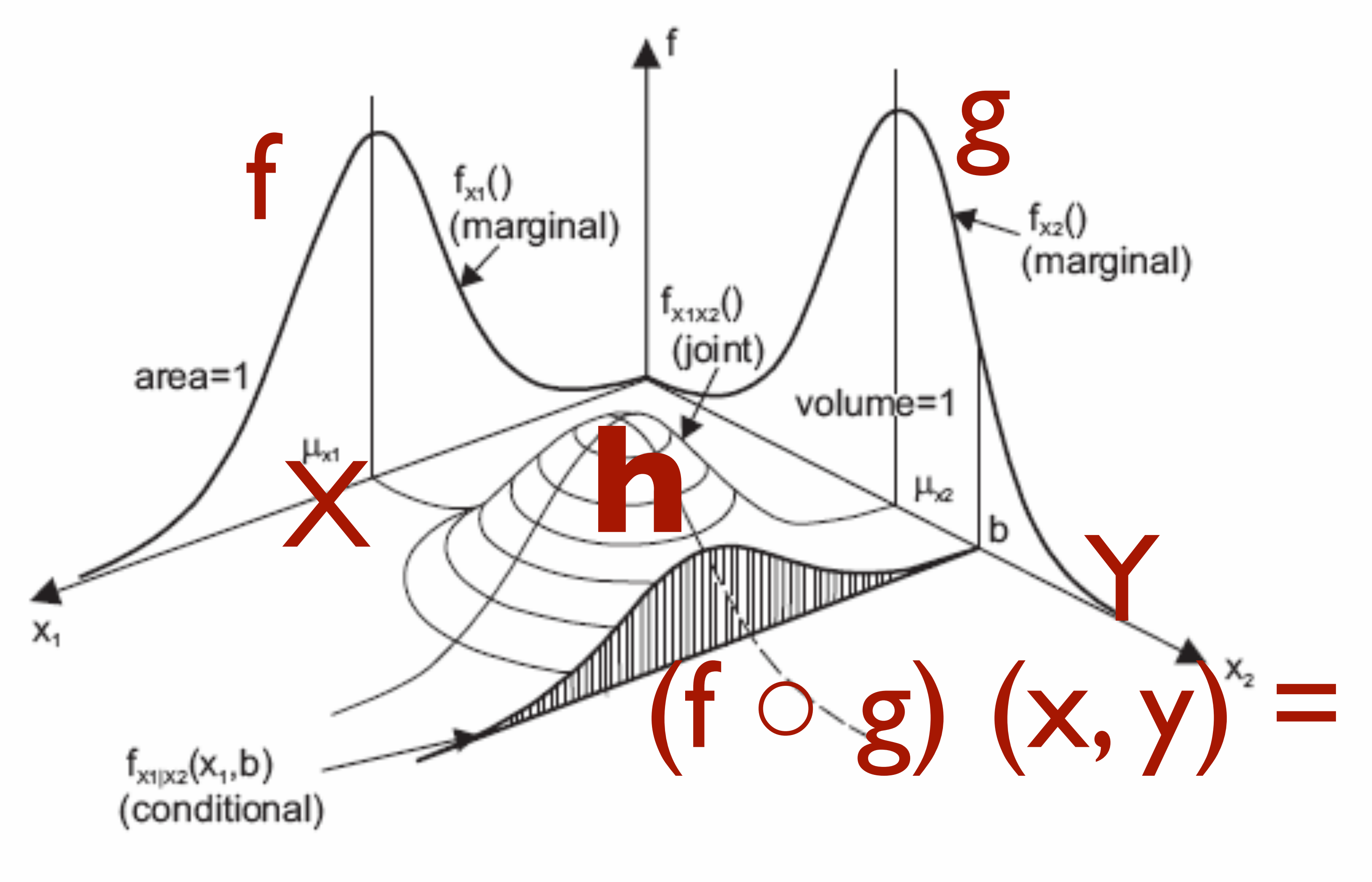
$(f \circ g)\,(x, y) = f(x) \cdot g(y)$

Examples of Kripke Resource Monoid

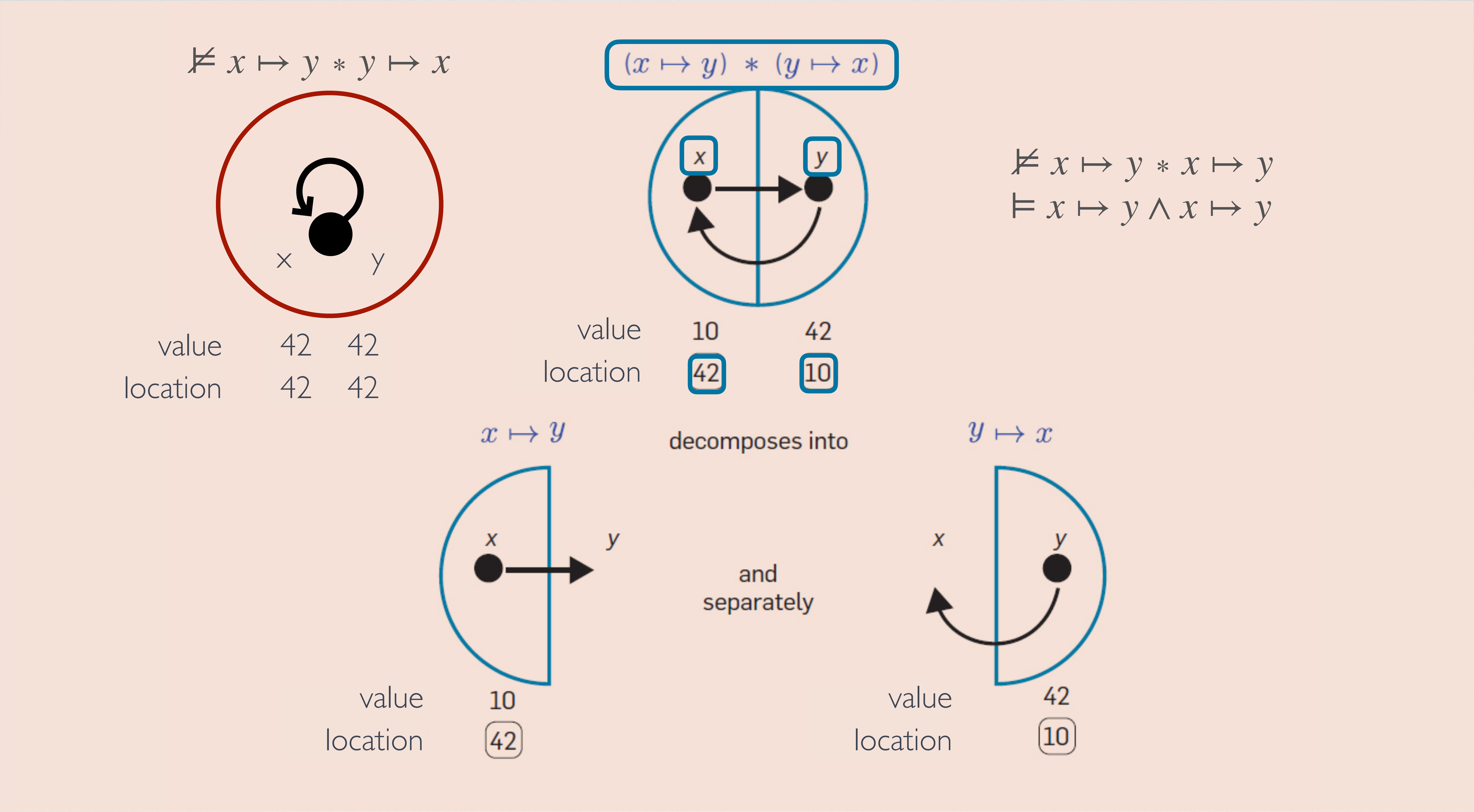
Distribution model

- Let M be the set of distributions over memories,
  - for distributions $f : X \to [0,1]$ and $g : Y \to [0,1]$, $f \circ g$ defined to be their independent product if X, Y are disjoint, otherwise undefined
  - an identity element e: deterministic distribution on empty memory
  - f ⊑ h if h marginalizes into f

$$e(\langle\rangle) = 1$$

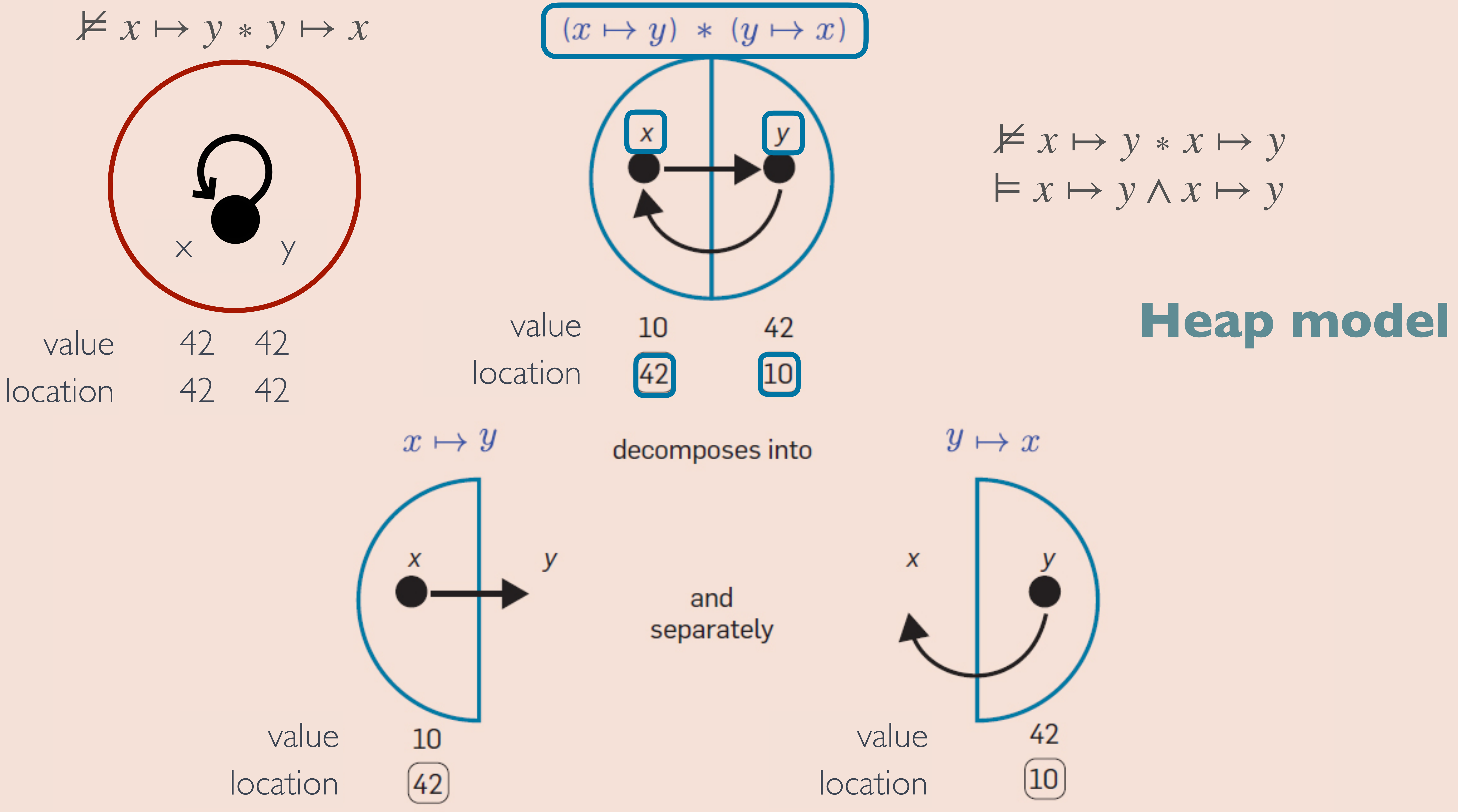


$(f \circ g)\ (x, y) = f(x) \cdot g(y)$



Adapted from an image in "Separation Logic" in CACM

Heap model



Adapted from an image in "Separation Logic" in CACM


Pumpkin Spice Latte
Probabilistic Separation Logic
credit to Joe Cutler

Satisfactions on Kripke Monoid

Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**

Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$

Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$
  - …

Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$
  - …
  - $m \vDash P \wedge Q$ **iff** $m \vDash P$ **and** $m \vDash Q$

Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$
  - …
  - $m \vDash P \wedge Q$ **iff** $m \vDash P$ **and** $m \vDash Q$
  - $m \vDash P * Q$ **iff exist** m_1, m_2 **with** $m_1 \circ m_2$ **defined and** $m_1 \circ m_2 \sqsubseteq m$ **such that** $m_1 \vDash P$ **and** $m_2 \vDash Q$

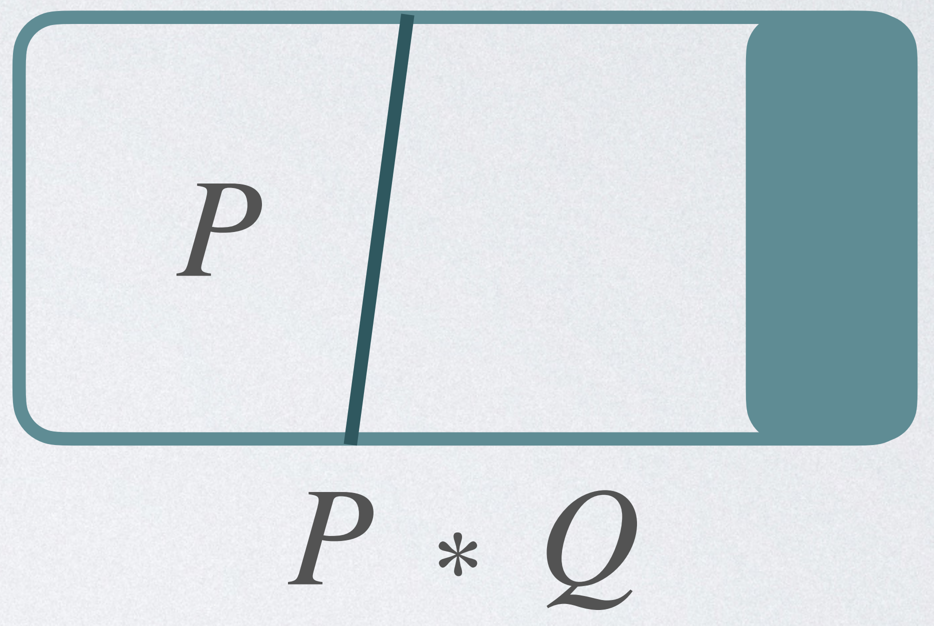
Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$
  - …
  - $m \vDash P \wedge Q$ **iff** $m \vDash P$ **and** $m \vDash Q$
  - $m \vDash P * Q$ **iff exist** m_1, m_2 **with** $m_1 \circ m_2$ **defined and** $m_1 \circ m_2 \sqsubseteq m$ **such that** $m_1 \vDash P$ **and** $m_2 \vDash Q$

$P * Q$

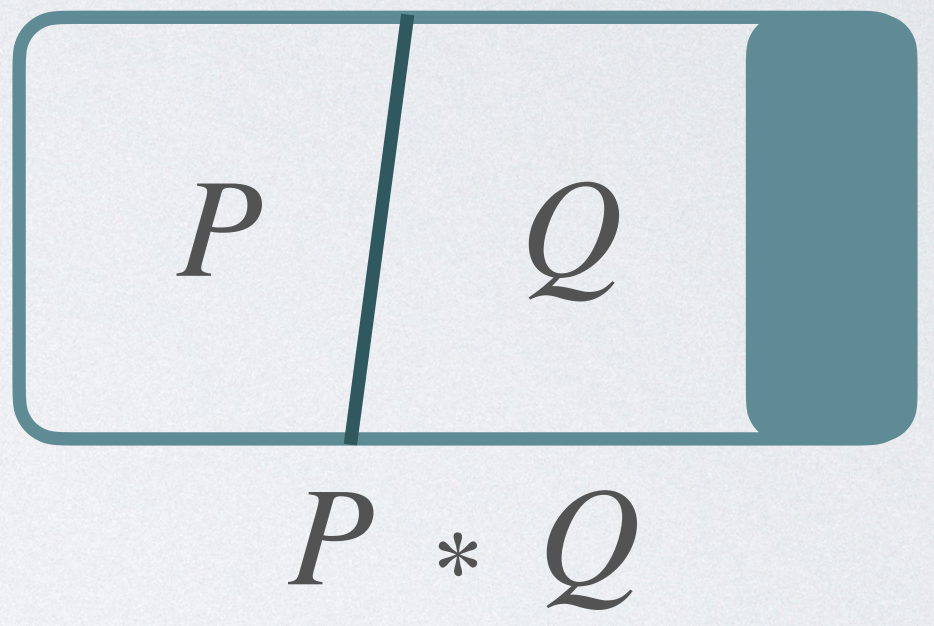
Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$
  - …
  - $m \vDash P \wedge Q$ **iff** $m \vDash P$ **and** $m \vDash Q$
  - $m \vDash P * Q$ **iff exist** m_1, m_2 **with** $m_1 \circ m_2$ **defined and** $m_1 \circ m_2 \sqsubseteq m$ **such that** $m_1 \vDash P$ **and** $m_2 \vDash Q$



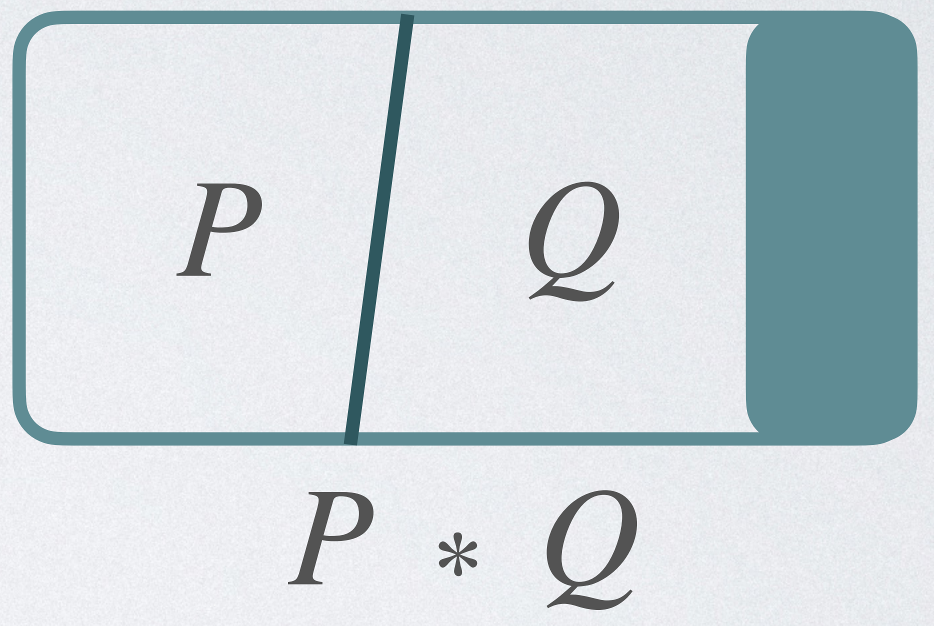
Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$
  - …
  - $m \vDash P \wedge Q$ **iff** $m \vDash P$ **and** $m \vDash Q$
  - $m \vDash P * Q$ **iff exist** m_1, m_2 **with** $m_1 \circ m_2$ **defined and** $m_1 \circ m_2 \sqsubseteq m$ **such that** $m_1 \vDash P$ **and** $m_2 \vDash Q$



Satisfactions on Kripke Monoid

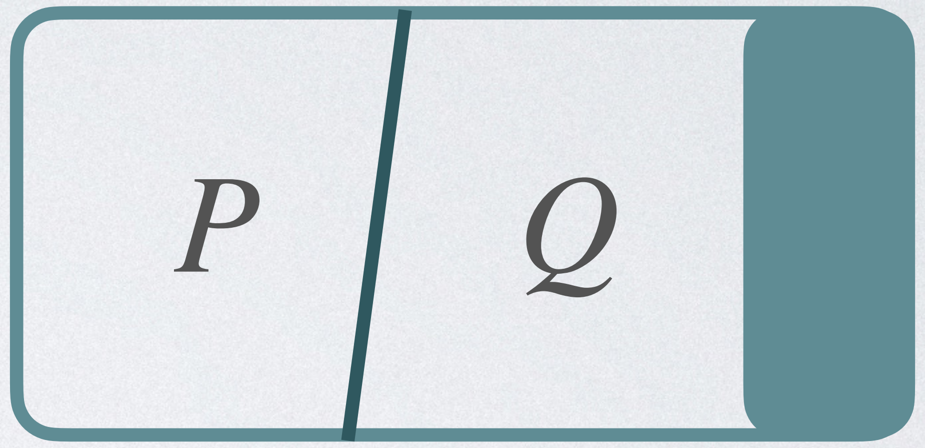
- **We inductively define the satisfaction relations on** $m \in M$ **and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$
  - …
  - $m \vDash P \wedge Q$ **iff** $m \vDash P$ **and** $m \vDash Q$
  - $m \vDash P * Q$ **iff exist** m_1, m_2 **with** $m_1 \circ m_2$ **defined and** $m_1 \circ m_2 \sqsubseteq m$ **such that** $m_1 \vDash P$ **and** $m_2 \vDash Q$



- **In the independence model:**

Satisfactions on Kripke Monoid

- **We inductively define the satisfaction relations on $m \in M$ and assertions:**
  - $m \vDash p$ **iff** $m \in \mathscr{V}(p)$
  - …
  - $m \vDash P \wedge Q$ **iff** $m \vDash P$ **and** $m \vDash Q$
  - $m \vDash P * Q$ **iff exist** m_1, m_2 **with** $m_1 \circ m_2$ **defined and** $m_1 \circ m_2 \sqsubseteq m$ **such that** $m_1 \vDash P$ **and** $m_2 \vDash Q$



$P * Q$

- **In the independence model:**
  - $m \vDash \langle X \rangle * \langle Y \rangle$ **iff variables** X, Y **are independent in** m

Program logic

Program logic

- **Judgement:** $\{P\}C\{Q\}$

Program logic

- **Judgement:** $\{P\}C\{Q\}$
- **Programs:**

Program logic

- **Judgement:** $\{P\}C\{Q\}$
- **Programs:**
  - **Standard imperative language + sampling from uniform distribution**

Program logic

- **Judgement:** $\{P\}C\{Q\}$
- **Programs:**
  - **Standard imperative language + sampling from uniform distribution**
- **Atomic propositions in the distribution model**

Program logic

- **Judgement:** $\{P\}C\{Q\}$
- **Programs:**
  - **Standard imperative language + sampling from uniform distribution**
- **Atomic propositions in the distribution model**
  - $\mu \vDash \mathbf{U}_T\langle e \rangle$

Program logic

- **Judgement:** $\{P\}C\{Q\}$
- **Programs:**
  - **Standard imperative language + sampling from uniform distribution**
- **Atomic propositions in the distribution model**
  - $\mu \models \mathbf{U}_T\langle e\rangle$
  - $\mu \models \mathbf{Detm}\langle e\rangle$

Program logic

- **Judgement:** $\{P\}C\{Q\}$
- **Programs:**
  - **Standard imperative language + sampling from uniform distribution**
- **Atomic propositions in the distribution model**
  - $\mu \models \mathbf{U}_T\langle e \rangle$
  - $\mu \models \mathbf{Detm}\langle e \rangle$
  - $\mu \models e \sim e'$

Program logic

- **Judgement:** $\{P\}C\{Q\}$
- **Programs:**
  - **Standard imperative language + sampling from uniform distribution**
- **Atomic propositions in the distribution model**
  - $\mu \models \mathbf{U}_T\langle e \rangle$
  - $\mu \models \mathbf{Detm}\langle e \rangle$
  - $\mu \models e \sim e'$
  - $\mu \models \langle e \rangle$ **iff** $\mu \models e \sim e$

Proof Rules

Proof Rules

$$\textsc{RSamp}^{*}\ \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

Proof Rules

$$\textsc{RSamp}^{*}\ \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\textsc{Frame}\ \frac{\vdash \{\phi\}c\{\psi\} \qquad c \text{ does not modifies } FV(\eta) \qquad \text{side conditions}}{\vdash \{\phi * \eta\}c\{\psi * \eta\}}$$

A SEPARATION LOGIC FOR NEGATIVE DEPENDENCE

Independence → Negative Association

on assertion logic

Independence → Negative Association

on assertion logic

$\langle X_1 \rangle * \langle X_2 \rangle * \ldots * \langle X_n \rangle$ **asserts** $X_1, \ldots, X_n$ **independent in distribution model**

Independence → Negative Association

on assertion logic

$\langle X_1 \rangle * \langle X_2 \rangle * \ldots * \langle X_n \rangle$ **asserts** $X_1, \ldots, X_n$ **independent in distribution model**

Can we add another conjunction $\circledast$ such that $\langle X_1 \rangle \circledast \langle X_2 \rangle \circledast \ldots \circledast \langle X_n \rangle$ asserts $X_1, \ldots, X_n$ NA?

Challenge in the simplest case

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Define some $\oplus$: M × M ⇀ M, and let $\mu \models \langle X_1 \rangle \circledast \langle X_2 \rangle$ iff exist μ_1, μ_2 with $\mu_1 \oplus \mu_2$ defined and $\mu_1 \oplus \mu_2 \sqsubseteq \mu$ such that $\mu_1 \models \langle X_1 \rangle$ and $\mu_2 \models \langle X_2 \rangle$

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Define some $\oplus$: M × M ⇀ M, and X_1, X_2 NA in μ **iff exist μ_1, μ_2 with $\mu_1 \oplus \mu_2$ defined and $\mu_1 \oplus \mu_2 \sqsubseteq \mu$ such that $\mu_1 \models \langle X_1 \rangle$ and $\mu_2 \models \langle X_2 \rangle$**

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Define some $\oplus$: M × M ⇀ M, and X_1, X_2 NA in μ iff exist μ_1, μ_2 with $\mu_1 \oplus \mu_2$ defined and $\mu_1 \oplus \mu_2 \sqsubseteq \mu$ such that $\mu_1 \models \langle X_1 \rangle$ and $\mu_2 \models \langle X_2 \rangle$

$$\mu_1(X_1 = 1) = \mu_2(X_2 = 1) = \frac{1}{3}$$

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Define some $\oplus$: M × M ⇀ M, and X_1, X_2 NA in μ **iff exist μ_1, μ_2 with $\mu_1 \oplus \mu_2$ defined and $\mu_1 \oplus \mu_2 \sqsubseteq \mu$ such that $\mu_1 \models \langle X_1 \rangle$ and $\mu_2 \models \langle X_2 \rangle$**

$$\mu_1(X_1 = 1) = \mu_2(X_2 = 1) = \frac{1}{3}$$

$$\mu_1(X_1 = 0) = \mu_2(X_2 = 0) = \frac{2}{3}$$

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Define some $\oplus$: M × M ⇀ M, and X_1, X_2 NA in μ **iff exist μ_1, μ_2 with $\mu_1 \oplus \mu_2$ defined and $\mu_1 \oplus \mu_2 \sqsubseteq \mu$ such that $\mu_1 \vDash \langle X_1 \rangle$ and $\mu_2 \vDash \langle X_2 \rangle$**

$$\mu_1(X_1 = 1) = \mu_2(X_2 = 1) = \frac{1}{3}$$

$$\mu_1(X_1 = 0) = \mu_2(X_2 = 0) = \frac{2}{3}$$

	X_1	X_2
$\frac{2}{9}$	1	0
$\frac{2}{9}$	0	1
$\frac{4}{9}$	0	0
$\frac{1}{9}$	1	1

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Define some $\oplus$: M × M ⇀ M, and **X_1, X_2 NA in μ** **iff exist μ_1, μ_2 with $\mu_1 \oplus \mu_2$ defined and $\mu_1 \oplus \mu_2 \sqsubseteq \mu$ such that $\mu_1 \vDash \langle X_1 \rangle$ and $\mu_2 \vDash \langle X_2 \rangle$**

$$\mu_1(X_1 = 1) = \mu_2(X_2 = 1) = \frac{1}{3}$$

$$\mu_1 \oplus \mu_2 =$$

	X_1	X_2
$\frac{2}{9}$	1	0
$\frac{2}{9}$	0	1
$\frac{4}{9}$	0	0
$\frac{1}{9}$	1	1

?

$$\mu_1(X_1 = 0) = \mu_2(X_2 = 0) = \frac{2}{3}$$

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Define some $\oplus$: M × M ⇀ M, and X_1, X_2 NA in μ **iff exist μ_1, μ_2 with $\mu_1 \oplus \mu_2$ defined and $\mu_1 \oplus \mu_2 \sqsubseteq \mu$ such that $\mu_1 \models \langle X_1 \rangle$ and $\mu_2 \models \langle X_2 \rangle$**

$$\mu_1(X_1 = 1) = \mu_2(X_2 = 1) = \frac{1}{3}$$

$$\mu_1 \oplus \mu_2 = \text{?}$$

	X_1	X_2
$\frac{2}{9}$	1	0
$\frac{2}{9}$	0	1
$\frac{4}{9}$	0	0
$\frac{1}{9}$	1	1

$$\mu_1(X_1 = 0) = \mu_2(X_2 = 0) = \frac{2}{3}$$

	X_1	X_2
$\frac{1}{3}$	1	0
$\frac{1}{3}$	0	1
$\frac{1}{3}$	0	0

Challenge in the simplest case

Say we want $\langle X_1 \rangle \circledast \langle X_2 \rangle$ asserts X_1, X_2 NA in distribution model

Define some $\oplus$: M × M ⇀ M, and X_1, X_2 NA in μ **iff exist μ_1, μ_2 with $\mu_1 \oplus \mu_2$ defined and $\mu_1 \oplus \mu_2 \sqsubseteq \mu$ such that $\mu_1 \models \langle X_1 \rangle$ and $\mu_2 \models \langle X_2 \rangle$**

$$\mu_1(X_1 = 1) = \mu_2(X_2 = 1) = \frac{1}{3}$$

$$\mu_1(X_1 = 0) = \mu_2(X_2 = 0) = \frac{2}{3}$$

	X_1	X_2
$\frac{2}{9}$	1	0
$\frac{2}{9}$	0	1
$\frac{4}{9}$	0	0
$\frac{1}{9}$	1	1

$$\mu_1 \oplus \mu_2 =$$

	X_1	X_2
$\frac{1}{3}$	1	0
$\frac{1}{3}$	0	1
$\frac{1}{3}$	0	0

?

Solution for the Challenge

Solution for the Challenge

- A Kripke resource monoid is a set M with
  - a partial binary operation $\circ : M \times M \rightharpoonup M$ that is
    - associative
    - commutative
  - an identity element $e \in M$
  - a pre-order $\sqsubseteq$ on M

Solution for the Challenge

- A Kripke resource monoid is a set M with
  - a binary operation $\circ : M \times M \to \mathscr{P}(M)$ that is
    - associative
    - commutative
  - an identity element e $\in$ M
  - a pre-order $\sqsubseteq$ on M

Solution for the Challenge

- A BI frame [Docherty 2019] is a set M with
  - a binary operation $\circ : M \times M \to \mathscr{P}(M)$ that is
    - associative
    - commutative
  - an identity element e ∈ M
  - a pre-order ⊑ on M

Solution for the Challenge

- A BI frame [Docherty 2019] is a set M with
  - a binary operation $\circ : M \times M \to \mathscr{P}(M)$ that is
    - associative
    - commutative
  - an identity element $E \subseteq M$ compatible with $\circ$ and $\sqsubseteq$
  - a pre-order $\sqsubseteq$ on M

Solution for the Challenge

- A BI frame [Docherty 2019] is a set M with
  - a binary operation $\circ : M \times M \to \mathscr{P}(M)$ that is
    - associative
    - commutative
  - an identity element $E \subseteq M$ compatible with $\circ$ and $\sqsubseteq$
  - a pre-order $\sqsubseteq$ on M

$\mu_1 \oplus \mu_2 = \{\mu \mid$ variables in μ_1, μ_2 satisfy some sort of NA in $\mu\}$

Skipping other challenges, we have

$\langle X_1 \rangle \circledast \langle X_2 \rangle \circledast \ldots \circledast \langle X_n \rangle$ **asserts** $X_1, X_2, \ldots, X_n$ **NA**

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1
- Uniformly random permutations

Closure of Negative Association:

- Subsets of NA variables are NA
- Union of independent NA sets is also NA
- Monotonically increasing map preserves NA

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1
- Uniformly random permutations

All valid axioms!

Closure of Negative Association:

- Subsets of NA variables are NA
- Union of independent NA sets is also NA
- Monotonically increasing map preserves NA

$$P * Q \vdash P \circledast Q$$

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1
- Uniformly random permutations

All valid axioms!

Closure of Negative Association:

- Subsets of NA variables are NA
- Union of independent NA sets is also NA
- Monotonically increasing map preserves NA

$$P * Q \vdash P \circledast Q$$

$$\models \mathbf{OH}_N\langle [x_0, \ldots, x_{N-1}] \rangle \to \mathop{\circledast}_{y=0}^{N} \langle x_y \rangle$$

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1
- Uniformly random permutations

All valid axioms!

Closure of Negative Association:

- Subsets of NA variables are NA
- Union of independent NA sets is also NA
- Monotonically increasing map preserves NA

Examples of NA random variables:

- Deterministic variables
- Independent random variables
- Bernoulli random variables that sum to 1
- Uniformly random permutations

All valid axioms!

$$P * Q \vdash P \circledast Q$$

$$\models \mathbf{OH}_N\langle[x_0, \ldots, x_{N-1}]\rangle \to \bigcircledast_{\gamma=0}^{N} \langle x_\gamma\rangle$$

Closure of Negative Association:

- Subsets of NA variables are NA
- Union of independent NA sets is also NA
- Monotonically increasing map preserves NA

Mono-map Axiom

$$\models \bigcircledast_{\gamma=0}^{N}\left(\bigwedge_{\alpha=0}^{K_\gamma+1}\langle x_{\gamma,\alpha}\rangle\right) \wedge \bigwedge_{\gamma=0}^{N} y_\gamma = f_\gamma\left(x_{\gamma,0}, \ldots, x_{\gamma,K_\gamma}\right) \to \bigcircledast_{\gamma=0}^{N}\langle y_\gamma\rangle$$

when $f_1, \ldots, f_N$ all monotone or all antitone

Independence → Negative Association

on program logic

Independence → Negative Association

on program logic

A RSamp rule for NA?

Independence → Negative Association

on program logic

A RSamp rule for NA?

$$\textsc{RSamp}^* \; \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r \rangle\}}$$

Independence → Negative Association

on program logic

A RSamp rule for NA?

$$\text{RS\small{AMP}}^{*}\ \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r \rangle\}}$$

A frame rule for NA?

Independence → Negative Association

on program logic

A RSamp rule for NA?

$$\textsc{RSamp}^* \; \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r \rangle\}}$$

A frame rule for NA?

$$\frac{\vdash \{\phi\}c\{\psi\} \qquad c \text{ does not modifies } FV(\eta) \qquad \text{side conditions}}{\vdash \{\phi \;\; * \;\; \eta\}c\{\psi \;\; * \;\; \eta\}}$$

Independence → Negative Association

on program logic

A RSamp rule for NA?

$$\textsc{RSamp}^{*}\ \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

A frame rule for NA?

$$\frac{\vdash \{\phi\}c\{\psi\} \qquad c \text{ does not modifies } FV(\eta) \qquad \text{side conditions}}{\vdash \{\phi \circledast \eta\}c\{\psi \circledast \eta\}}$$

Independence → Negative Association

on program logic

A RSamp rule for NA?

$$\textsc{RSamp}^* \; \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\; x_r \xleftarrow{\$} \mathbf{U}_S \; \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

A frame rule for NA?

c is a monotonically increasing map from $dom(\phi)$ to $dom(\psi)$

$$\frac{\vdash \{\phi\}c\{\psi\} \qquad c \text{ does not modifies } FV(\eta) \qquad \text{side conditions}}{\vdash \{\phi \circledast \eta\}c\{\psi \circledast \eta\}}$$

Independence → Negative Association

on program logic

A RSamp rule for NA?

$$\textsc{RSamp}^* \; \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\; x_r \xleftarrow{\$} \mathbf{U}_S \; \{\phi * \mathbf{U}_S\langle x_r \rangle\}}$$

A frame rule for NA? **NA preserved under monotone maps**

c is a monotonically increasing map from $dom(\phi)$ to $dom(\psi)$

$$\frac{\vdash \{\phi\} c \{\psi\} \qquad c \text{ does not modifies } FV(\eta) \qquad \text{side conditions}}{\vdash \{\phi \circledast \eta\} c \{\psi \circledast \eta\}}$$

Independence → Negative Association

on program logic

A frame rule for NA

$$\frac{\vdash \{\phi\}c\{\psi\} \qquad c \text{ does not modifies } FV(\eta) \qquad \text{side conditions}}{\vdash \{\phi \circledast \eta\}c\{\psi \circledast \eta\}}$$

Independence → Negative Association

on program logic

A frame rule for NA

$$\frac{\vdash \{\phi\}c\{\langle y\rangle\} \qquad c \text{ does not modifies } FV(\eta)}{\vdash \{\phi \circledast \eta\}c\{\langle y\rangle \circledast \eta\}}$$

$\langle y\rangle$ **obtained from a monotonically increasing map on** $dom(\phi)$

side conditions

Independence → Negative Association

on program logic

A frame rule for NA

$$\textsc{NA-Frame}\ \frac{\vdash \{\phi\}c\{\langle y\rangle\} \qquad c \text{ does not modifies } FV(\eta) \qquad \langle y\rangle \textbf{ obtained from a monotonically increasing map on } dom(\phi) \qquad \text{side conditions}}{\vdash \{\phi \circledast \eta\}c\{\langle y\rangle \circledast \eta\}}$$

APPLICATIONS

to the motivating example

```
tasks = [A, ..., Z]
loads = [0, 0, 0]

for task in tasks:
    new_load = one-hot(3)
    loads = loads + new_load

overflow = [n >= 10 for n in loads]
```

```
tasks = [A, ..., Z]
loads = [0, 0, 0]
i = 0

while i < |tasks|:
    i = i + 1
    new_load = one-hot(3)
    loads = loads + new_load

overflow = [n >= 10 for n in loads]
```

In our informal proof

```
tasks = [A, ..., Z]
loads = [0, 0, 0]
i = 0
```

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i]\}$

```
while i < |tasks|:
    i = i + 1
    new_load = one-hot(3)
    loads = loads + new_load
```

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i]\}$

```
overflow = [n >= 10 for n in loads]
```

```
tasks = [A, ..., Z]
loads = [0, 0, 0]
i = 0

while i < |tasks|:
    i = i + 1
    new_load = one-hot(3)
    loads = loads + new_load

overflow = [n >= 10 for n in loads]
```

$$\textsc{Loop}\ \frac{\vdash \{\phi \wedge b \sim tt\}\ c\ \{\phi\} \qquad \models \phi \to \mathbf{Detm}\langle b\rangle}{\vdash \{\phi\}\ \mathbf{while}\ b\ \mathbf{do}\ c\ \{\phi \wedge b \sim f\!f\}}$$

```
tasks = [A, ..., Z]
loads = [0, 0, 0]
i = 0
```
$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$
```
while i < |tasks|:
    i = i + 1
    new_load = one-hot(3)
    loads = loads + new_load
```
$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$
```
overflow = [n >= 10 for n in loads]
```

$$\textsc{Loop}\ \frac{\vdash \{\phi \wedge b \sim tt\}\ c\ \{\phi\} \qquad \models \phi \to \mathbf{Detm}\langle b\rangle}{\vdash \{\phi\}\ \mathbf{while}\ b\ \mathbf{do}\ c\ \{\phi \wedge b \sim f\!f\}}$$

```
tasks = [A, ..., Z]
loads = [0, 0, 0]
i = 0
```

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

```
while i < |tasks|:
    i = i + 1
    new_load = one-hot(3)
    loads = loads + new_load
```

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$

```
overflow = [n >= 10 for n in loads]
```

$$\textsc{Loop}\ \frac{\vdash \{\phi \wedge b \sim tt\}\ c\ \{\phi\} \qquad \models \phi \to \mathbf{Detm}\langle b\rangle}{\vdash \{\phi\}\ \mathbf{while}\ b\ \mathbf{do}\ c\ \{\phi \wedge b \sim f\!f\}}$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

while i < |tasks|:

i = i + 1

new_load = one-hot(3)

loads = loads + new_load

$$\textsc{Loop}\ \frac{\vdash \{\phi \wedge b \sim tt\}\ c\ \{\phi\} \qquad \models \phi \to \mathbf{Detm}\langle b\rangle}{\vdash \{\phi\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{\phi \wedge b \sim f\!f\}}$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \textbf{Detm}[\mathsf{i}] \wedge \textbf{Detm}[\mathsf{task}]\}$

```
while i < |tasks|:
```

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \textbf{Detm}[\mathsf{i}] \wedge \textbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}|)\}$

```
    i = i + 1

    new_load = one-hot(3)

    loads = loads + new_load
```

$$\textsc{Loop}\ \frac{\vdash \{\phi \wedge b \sim tt\}\ c\ \{\phi\} \qquad \models \phi \to \textbf{Detm}\langle b\rangle}{\vdash \{\phi\}\ \textbf{while}\ b\ \textbf{do}\ c\ \{\phi \wedge b \sim ff\}}$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \textbf{Detm}[\mathsf{i}] \wedge \textbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \textbf{Detm}[\mathsf{i}] \wedge \textbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

```
while i < |tasks|:
```

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}|)\}$

```
    i = i + 1

    new_load = one-hot(3)

    loads = loads + new_load
```

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$

$$\textsc{DAssn}\ \frac{}{\vdash \{\psi[e_d/x_d]\}\ x_d \leftarrow e_d\ \{\psi\}}$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

while i < |tasks|:

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}|)\}$

i = i + 1

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$$\text{DAssn}\ \frac{}{\vdash \{\psi[e_d/x_d]\}\ x_d \leftarrow e_d\ \{\psi\}}$$

loads = loads + new_load

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \land \mathbf{Detm}[\mathsf{i}] \land \mathbf{Detm}[\mathsf{task}]\}$

while i < |tasks|:

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \land \mathbf{Detm}[\mathsf{i}] \land \mathbf{Detm}[\mathsf{task}] \land (\mathsf{i} < |\mathsf{task}|)\}$

i = i + 1

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \land \mathbf{Detm}[\mathsf{i}] \land \mathbf{Detm}[\mathsf{task}] \land (\mathsf{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$$\textsc{DAssn}\ \frac{}{\vdash \{\psi[e_d/x_d]\}\ x_d \leftarrow e_d\ \{\psi\}}$$

loads = loads + new_load

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \land \mathbf{Detm}[\mathsf{i}] \land \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \land \mathbf{Detm}[\mathsf{i}] \land \mathbf{Detm}[\mathsf{task}] \land (\mathsf{i} \geq |\mathsf{task}|)\}$

$\{\circledast_{i \in \{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

loads = loads + new_load

$\{\circledast_{i \in \{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$$\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$$

new_load = one-hot(3)

$$\textsc{RSamp}^* \quad \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

loads = loads + new_load

$$\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$$\textsc{RSamp}^{*}\ \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

loads = loads + new_load

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

loads = loads + new_load

$\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$$\text{RSAMP}^* \quad \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle[x_0, \ldots, x_{N-1}]\rangle \to \bigcircledast_{\gamma=0}^{N} \langle x_\gamma\rangle$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \circledast_{i\in\{0,1,2\}} \mathsf{new\_load}[i]) \wedge \ldots\}$

$$\textsc{RSamp}^* \quad \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle[x_0, \ldots, x_{N-1}]\rangle \rightarrow \bigcircledast_{\gamma=0}^{N}\langle x_\gamma\rangle$$

loads = loads + new_load

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathrm{task}] \wedge (\mathrm{i} < |\mathrm{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \circledast_{i\in\{0,1,2\}}\mathsf{new\_load}[i]) \wedge \ldots\}$

loads = loads + new_load

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathrm{task}]\}$

$$\textsc{RSamp}^* \; \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\; x_r \xleftarrow{\$} \mathbf{U}_S\; \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle[x_0,\ldots,x_{N-1}]\rangle \to \bigcircledast_{\gamma=0}^{N}\langle x_\gamma\rangle$$

$$P * Q \vdash P \circledast Q$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathrm{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \circledast_{i\in\{0,1,2\}} \mathsf{new\_load}[i]) \wedge \ldots\}$

$\{((\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i]) \circledast (\circledast_{i\in\{0,1,2\}}\mathsf{new\_load}[i])) \wedge \ldots\}$

loads = loads + new_load

$$\textsc{RSamp}^* \; \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\; x_r \xleftarrow{\$} \mathbf{U}_S \; \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle[x_0, \ldots, x_{N-1}]\rangle \to \bigcircledast_{\gamma=0}^{N}\langle x_\gamma\rangle$$

$$P * Q \vdash P \circledast Q$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathrm{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \circledast_{i\in\{0,1,2\}}\,\mathsf{new\_load}[i]) \wedge \ldots\}$

$\left\{\left((\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i]) \circledast (\circledast_{i\in\{0,1,2\}}\mathsf{new\_load}[i])\right) \wedge \ldots\right\}$

updates = loads + new_loads

loads = updates

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$$\text{RS}\textsc{amp}^{*}\ \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle[x_0,\ldots,x_{N-1}]\rangle \to \bigcircledast_{\gamma=0}^{N}\langle x_\gamma\rangle$$

$$P * Q \vdash P \circledast Q$$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathrm{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \circledast_{i\in\{0,1,2\}} \mathsf{new\_load}[i]) \wedge \ldots\}$

$\{((\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i]) \circledast (\circledast_{i\in\{0,1,2\}}\mathsf{new\_load}[i])) \wedge \ldots\}$

updates = loads + new_loads

loads = updates

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$$\textsc{RSamp}^* \ \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle[x_0, \ldots, x_{N-1}]\rangle \to \bigcircledast_{\gamma=0}^{N}\langle x_\gamma\rangle$$

$$P * Q \vdash P \circledast Q$$

Mono-map Axiom

$\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

$\{(\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] * \circledast_{i\in\{0,1,2\}} \mathsf{new\_load}[i]) \wedge \ldots\}$

$\left\{\left((\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i]) \circledast (\circledast_{i\in\{0,1,2\}} \mathsf{new\_load}[i])\right) \wedge \ldots\right\}$

updates = loads + new_loads

$\{\circledast_{i\in\{0,1,2\}} \mathsf{updates}[i] \wedge \ldots\}$

loads = updates

$\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$$\textsc{RSamp}^* \; \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\; x_r \xleftarrow{\$} \mathbf{U}_S \;\{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle [x_0, \ldots, x_{N-1}]\rangle \rightarrow \bigcircledast_{\gamma=0}^{N} \langle x_\gamma\rangle$$

$$P * Q \vdash P \circledast Q$$

Mono-map Axiom

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathrm{i} < |\mathsf{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \mathbf{OH}_3[\mathsf{new\_loads}]) \wedge \ldots\}$

$\{(\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] * \circledast_{i\in\{0,1,2\}} \mathsf{new\_load}[i]) \wedge \ldots\}$

$\{((\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i]) \circledast (\circledast_{i\in\{0,1,2\}}\mathsf{new\_load}[i])) \wedge \ldots\}$

updates = loads + new_loads

$\{\circledast_{i\in\{0,1,2\}}\mathsf{updates}[i] \wedge \ldots\}$

loads = updates

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathrm{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$$\textsc{RSamp}^* \quad \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle[x_0,\ldots,x_{N-1}]\rangle \to \bigcircledast_{\gamma=0}^{N}\langle x_\gamma\rangle$$

$$P * Q \vdash P \circledast Q$$

Mono-map Axiom

$\{\circledast_{i\in\{0,1,2\}}\text{loads}[i] \wedge \mathbf{Detm}[\text{i}] \wedge \mathbf{Detm}[\text{task}] \wedge (\text{i} < |\text{task}| + 1)\}$

new_load = one-hot(3)

$\{(\circledast_{i\in\{0,1,2\}}\text{loads}[i] * \mathbf{OH}_3[\text{new\_loads}]) \wedge \ldots\}$

$\{(\circledast_{i\in\{0,1,2\}}\text{loads}[i] * \circledast_{i\in\{0,1,2\}} \text{new\_load}[i]) \wedge \ldots\}$

$\{((\circledast_{i\in\{0,1,2\}}\text{loads}[i]) \circledast (\circledast_{i\in\{0,1,2\}}\text{new\_load}[i])) \wedge \ldots\}$

updates = loads + new_loads

$\{\circledast_{i\in\{0,1,2\}}\text{updates}[i] \wedge \ldots\}$

loads = updates

$\{\circledast_{i\in\{0,1,2\}}\text{loads}[i] \wedge \mathbf{Detm}[\text{i}] \wedge \mathbf{Detm}[\text{task}]\}$

$$\textsc{RSamp}^* \ \frac{x_r \notin \mathrm{FV}(\phi)}{\vdash \{\phi\}\ x_r \xleftarrow{\$} \mathbf{U}_S\ \{\phi * \mathbf{U}_S\langle x_r\rangle\}}$$

$$\models \mathbf{OH}_N\langle[x_0,\ldots,x_{N-1}]\rangle \to \bigcircledast_{\gamma=0}^{N}\langle x_\gamma\rangle$$

$$P * Q \vdash P \circledast Q$$

Mono-map Axiom

$$\textsc{DAssn}\ \frac{}{\vdash \{\psi[e_d/x_d]\}\ x_d \leftarrow e_d\ \{\psi\}}$$

Scoping back …

tasks = [A, …, Z]

loads = [0, 0, 0]

i = 0

$\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

while i < |tasks|:

    i = i + 1

    $\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

    new_load = one-hot(3)

    loads = loads + new_load

    $\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}} \mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$

overflow = [n >= 10 for n in loads]

Scoping back …

tasks = [A, …, Z]

loads = [0, 0, 0]

i = 0

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

while i < |tasks|:

    i = i + 1

    $\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

    new_load = one-hot(3)

    loads = loads + new_load

    $\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$

overflow = [n >= 10 for n in loads]

Mono-map Axiom

Scoping back …

tasks = [A, …, Z]

loads = [0, 0, 0]

i = 0

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

while i < |tasks|:

    i = i + 1

    $\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} < |\mathsf{task}| + 1)\}$

    new_load = one-hot(3)

    loads = loads + new_load

    $\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}]\}$

$\{\circledast_{i\in\{0,1,2\}}\mathsf{loads}[i] \wedge \mathbf{Detm}[\mathsf{i}] \wedge \mathbf{Detm}[\mathsf{task}] \wedge (\mathsf{i} \geq |\mathsf{task}|)\}$

overflow = [n >= 10 for n in loads]

$\{\circledast_{i\in\{0,1,2\}}\mathsf{overflow}[i]\}$

Mono-map Axiom

More contents in our paper

More contents in our paper

- **M-BI logic: a sound and complete extension of BI that supports ordered separating conjunctions**

More contents in our paper

- **M-BI logic: a sound and complete extension of BI that supports ordered separating conjunctions**
- **Details of the (M)-BI model for negative association**

More contents in our paper

- **M-BI logic: a sound and complete extension of BI that supports ordered separating conjunctions**
- **Details of the (M)-BI model for negative association**
- **Details of the NA-Frame rule**

More contents in our paper

- **M-BI logic: a sound and complete extension of BI that supports ordered separating conjunctions**
- **Details of the (M)-BI model for negative association**
- **Details of the NA-Frame rule**
- **Applications to various probabilistic data structure**
  - Bloom filter
  - Permutation Hashing [Ding and König 2011]
  - Fully-dynamic dictionary [Bercea and Even 2019]
  - Repeated balls-into-bins [Becchetti et al. 2019]

A SEPARATION LOGIC FOR NEGATIVE DEPENDENCE

Jialu Bao at PLDG, Oct. 6, 2021
Joint work with Marco Gaboardi, Justin Hsu, Joseph Tassarotti