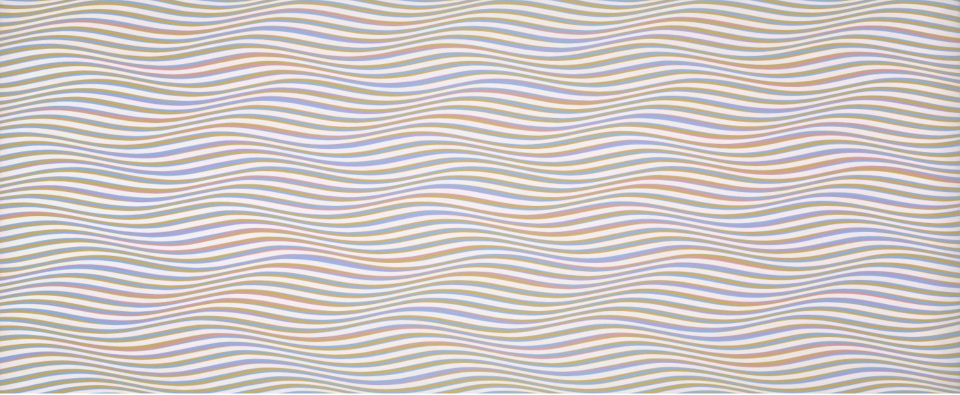
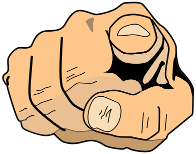


Exact Bayesian inference by symbolic disintegration

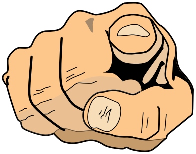
Chung-chieh Shan
Indiana University

Norman Ramsey
Tufts University

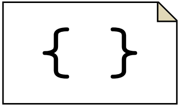
POPL, 18 January 2017

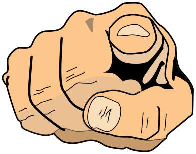


1. Probabilistic programs denote distributions
2. Exact inference by transforming terms

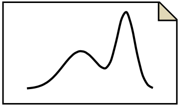


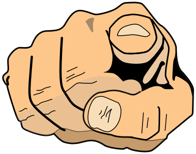
1. **Probabilistic programs** denote distributions
2. Exact inference by transforming terms



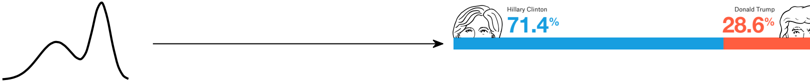


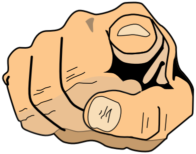
1. **Probabilistic programs** denote distributions
2. Exact inference by transforming terms



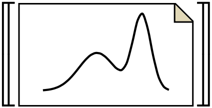


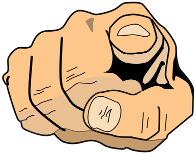
1. **Probabilistic programs** denote distributions
2. Exact inference by transforming terms



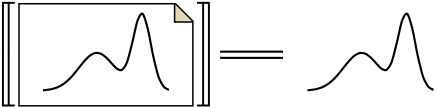


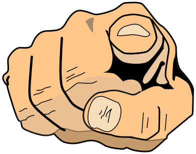
1. Probabilistic programs **denote distributions**
2. Exact inference by transforming terms



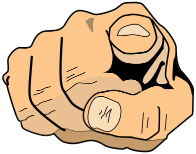


1. Probabilistic programs **denote distributions**
2. Exact inference by transforming terms

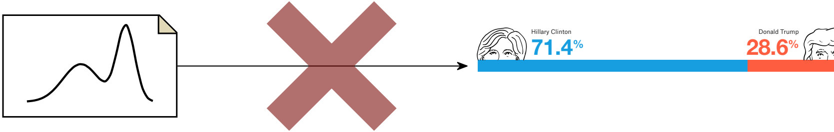


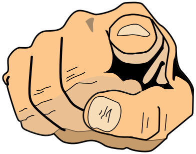


1. Probabilistic programs denote distributions
2. **Exact inference** by transforming terms

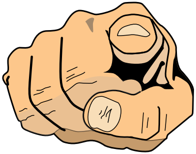


1. Probabilistic programs denote distributions
2. **Exact inference** by transforming terms

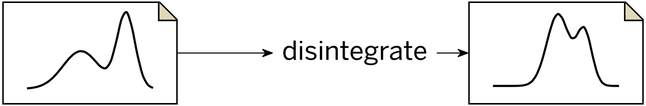


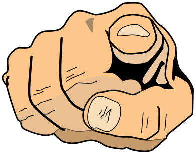


1. Probabilistic programs denote distributions
2. Exact inference **by transforming terms**

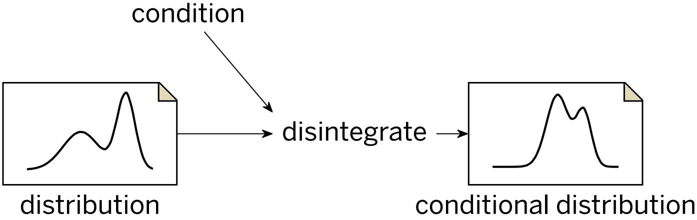


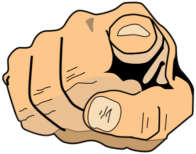
1. Probabilistic programs denote distributions
2. Exact inference **by transforming terms**



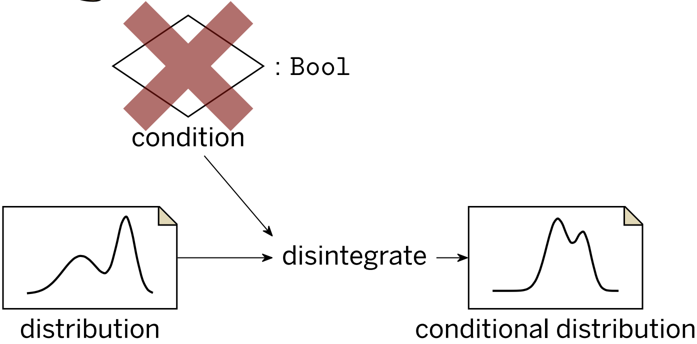


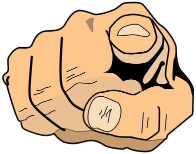
1. Probabilistic programs denote distributions
2. Exact inference **by transforming terms**





1. Probabilistic programs denote distributions
2. Exact inference **by transforming terms**

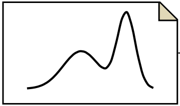




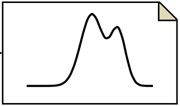
1. Probabilistic programs denote distributions
2. Exact inference **by transforming terms**

condition : α

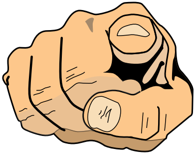
dependent variable	of regression
noisy measurement	of location
total momentum	of point masses
detected amplitude	of seismic event
...	...



distribution → disintegrate → conditional distribution (condition → disintegrate)



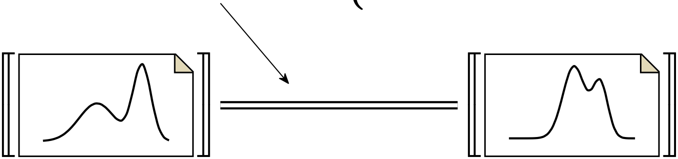
conditional distribution



1. Probabilistic programs denote distributions
2. Exact inference **by transforming terms**

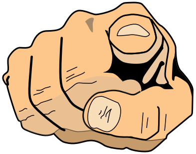
condition $: \mathbb{R}$

dependent variable	of regression
noisy measurement	of location
total momentum	of point masses
detected amplitude	of seismic event
...	...



distribution

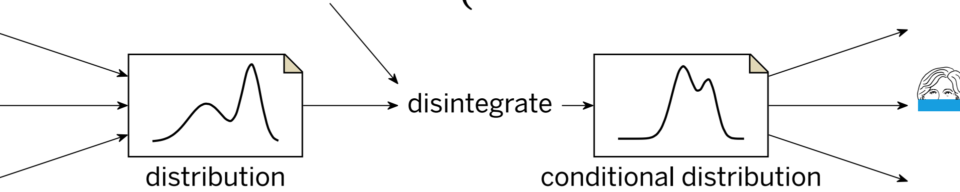
conditional distribution

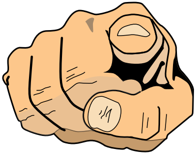


1. Probabilistic programs denote distributions
2. Exact inference **by transforming terms**

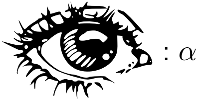
condition $: \alpha$

dependent variable	of regression
noisy measurement	of location
total momentum	of point masses
detected amplitude	of seismic event
...	...

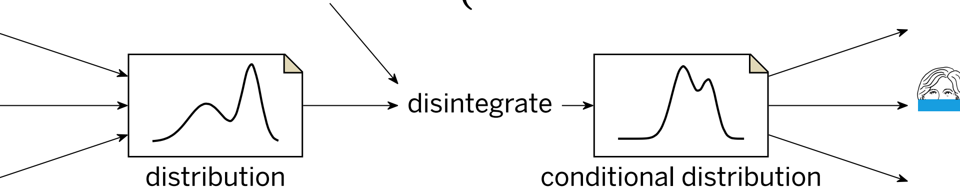




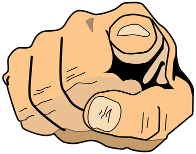
1. Probabilistic programs denote distributions
2. Exact inference by transforming terms



dependent variable	of regression
noisy measurement	of location
total momentum	of point masses
detected amplitude	of seismic event
…	…



1. **Motivate** by puzzle
2. **Specify** by semantics
3. **Implement** by derivation

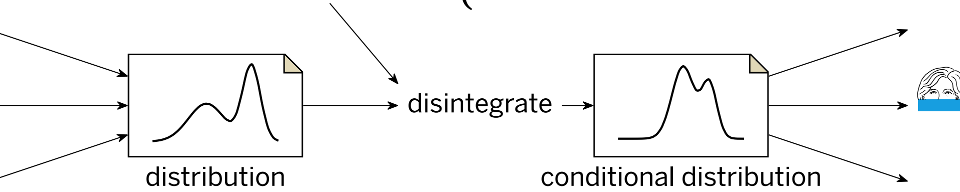


1. Probabilistic programs denote distributions
2. Exact inference by transforming terms

: α

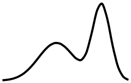
condition

dependent variable	of regression
noisy measurement	of location
total momentum	of point masses
detected amplitude	of seismic event
…	…



1. **Motivate** by puzzle
2. **Specify** by semantics
3. **Implement** by derivation

Bayesian probabilistic inference



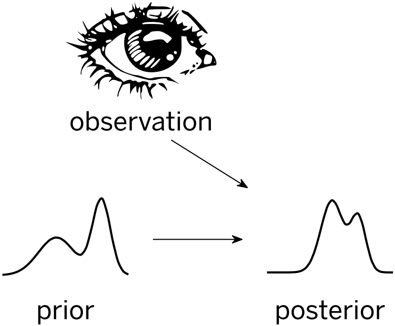
prior

Bayesian probabilistic inference

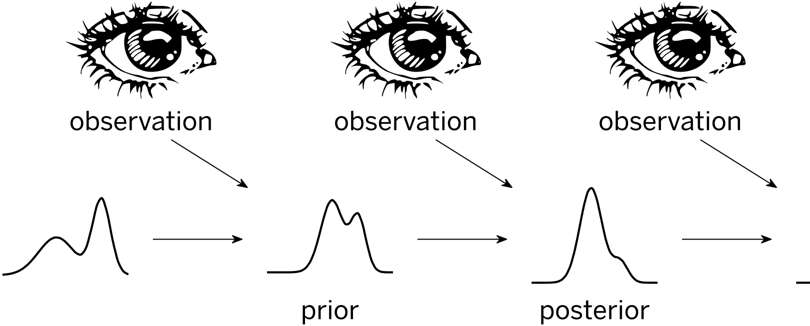
observation

prior

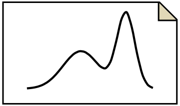
Bayesian probabilistic inference



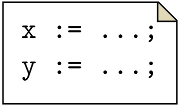
Bayesian probabilistic inference



Bayesian probabilistic inference

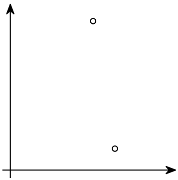


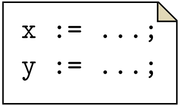
Bayesian probabilistic inference



generative model

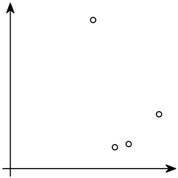
Bayesian probabilistic inference





generative model

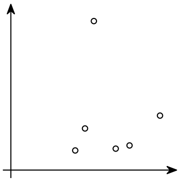
Bayesian probabilistic inference

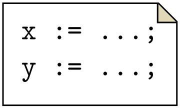


```
x := ...;
y := ...;
```

generative model

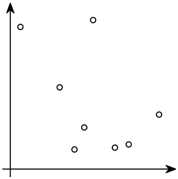
Bayesian probabilistic inference





generative model

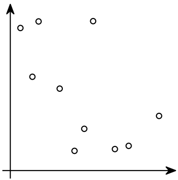
Bayesian probabilistic inference



```
x := ...;
y := ...;
```

generative model

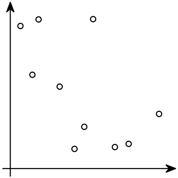
Bayesian probabilistic inference



```
x := ...;
y := ...;
```

generative model

Bayesian probabilistic inference



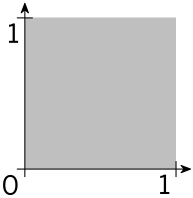
```
x := ...;
y := ...;
```

generative model

$E(x)$

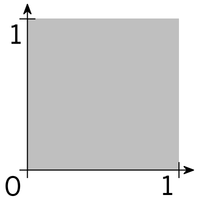
$P(A)$

Observation, inference, and query in core Hakaru



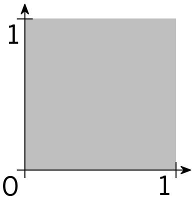
$$
\begin{aligned}
m0 = \textbf{do}\ \{&x \leftsquigarrow \textbf{uniform}\ 0\ 1;\\
&y \leftsquigarrow \textbf{uniform}\ 0\ 1;\\
&\textbf{return}\ (x, y)\}
\end{aligned}
$$

Observation, inference, and query in core Hakaru



$$m0 = \textbf{do}\ \{x \leftarrow\!\!\sim \textbf{uniform}\ 0\ 1;\ y \leftarrow\!\!\sim \textbf{uniform}\ 0\ 1;\ \textbf{return}\ (x, y)\}$$

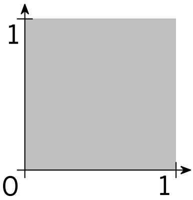
Observation, inference, and query in core Hakaru



$$m0 = \textbf{do}\ \{x \leftarrow\!\!\sim \textbf{uniform}\ 0\ 1;\ y \leftarrow\!\!\sim \textbf{uniform}\ 0\ 1;\ \textbf{return}\ (x, y)\}$$

$E(x)$

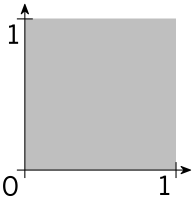
Observation, inference, and query in core Hakaru



$$m0 = \textbf{do}\ \{x \leftarrow \textbf{uniform}\ 0\ 1;\ y \leftarrow \textbf{uniform}\ 0\ 1;\ \textbf{return}\ (x, y)\}$$

$$E_{m0}(\lambda(x,y).\,x)$$

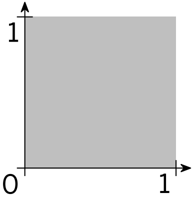
Observation, inference, and query in core Hakaru



$$m0 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;\ y \leftsquigarrow \textbf{uniform}\ 0\ 1;\ \textbf{return}\ (x, y)\}$$

$$E_{m0}(\lambda(x,y).\,x) = \frac{\int_{m0} x \, d(x,y)}{\int_{m0} 1 \, d(x,y)} = \frac{1/2}{1} = 1/2$$

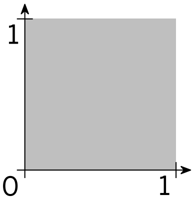
Observation, inference, and query in core Hakaru



$$m0 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;\ y \leftsquigarrow \textbf{uniform}\ 0\ 1;\ \textbf{return}\ (x,y)\}$$

$$E_{m0}(\lambda(x,y).\,x) = \frac{\int_{m0} x\, d(x,y)}{\int_{m0} 1\, d(x,y)} = \frac{1/2}{1} = 1/2$$

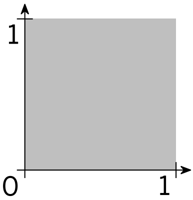
Observation, inference, and query in core Hakaru



$$m0 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;\ y \leftsquigarrow \textbf{uniform}\ 0\ 1;\ \textbf{return}\ (x, y)\}$$

$$E_{m0}(\lambda(x,y).\,x) = \frac{\int_{m0} x\, d(x,y)}{\int_{m0} 1\, d(x,y)} = \frac{1/2}{1} = 1/2$$

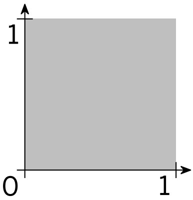
Observation, inference, and query in core Hakaru



$$m0 = \textbf{do}\ \{x \leftarrow\!\!\sim \textbf{uniform}\ 0\ 1;\ y \leftarrow\!\!\sim \textbf{uniform}\ 0\ 1;\ \textbf{return}\ (x, y)\}$$

$$E_{m0}(\lambda(x,y).\,x) = \frac{\int_{m0} x\, d(x,y)}{\int_{m0} 1\, d(x,y)} = \frac{1/2}{1} = 1/2$$

Observation, inference, and query in core Hakaru

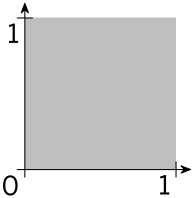


$$
\begin{aligned}
m0 = \textbf{do}\ \{ & x \leftsquigarrow \textbf{uniform}\ 0\ 1; \\
& y \leftsquigarrow \textbf{uniform}\ 0\ 1; \\
& \textbf{return}\ (x, y) \}
\end{aligned}
$$

$$E_{m0}(\lambda(x,y).\,x) = \frac{\int_{m0} x \, d(x,y)}{\int_{m0} 1 \, d(x,y)} = \frac{1/2}{1} = 1/2$$

$$P(A) = E(\langle A \rangle)$$

Observation, inference, and query in core Hakaru



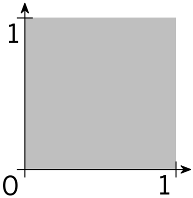
$y < 2 \cdot x$

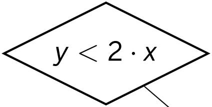
observation

$$
\begin{aligned}
m0 = \textbf{do}\ \{ & x \leftsquigarrow \textbf{uniform}\ 0\ 1; \\
& y \leftsquigarrow \textbf{uniform}\ 0\ 1; \\
& \textbf{return}\ (x, y) \}
\end{aligned}
$$

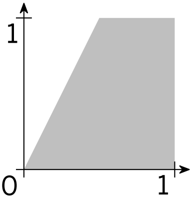
prior

Observation, inference, and query in core Hakaru





observation

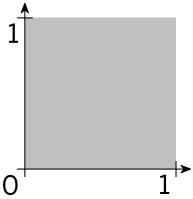


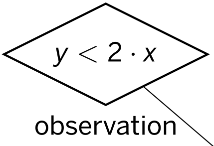
$m0 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad y \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad \textbf{return}\ (x, y)\}$

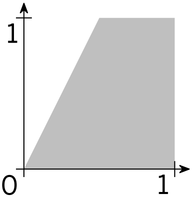
prior

$m1 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad y \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad \textbf{observe}\ y < 2 \cdot x;$
$\quad \textbf{return}\ (x, y)\}$

posterior







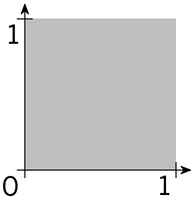
$m0 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad y \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad \textbf{return}\ (x, y)\}$

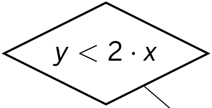
prior

$m1 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad y \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad \textbf{observe}\ y < 2 \cdot x;$
$\quad \textbf{return}\ (x, y)\}$

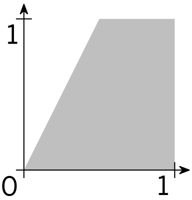
posterior

Observation, inference, and query in core Hakaru





observation

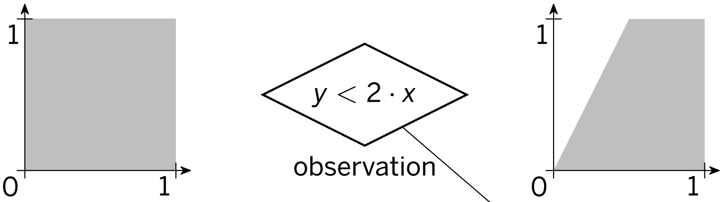


$m0 =$ **do** $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$y \leftsquigarrow$ **uniform** $0\ 1;$
return $(x, y)\}$

$m1 =$ **do** $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$y \leftsquigarrow$ **uniform** $0\ 1;$
observe $y < 2 \cdot x;$
return $(x, y)\}$

$$E_{m1}(\lambda(x,y).\, x) = \frac{\int_{m1} x \, d(x,y)}{\int_{m1} 1 \, d(x,y)} = \frac{\int_{m0} \langle y < 2 \cdot x \rangle \cdot x \, d(x,y)}{\int_{m0} \langle y < 2 \cdot x \rangle \cdot 1 \, d(x,y)} = \frac{11/24}{3/4} = 11/18$$

Observation, inference, and query in core Hakaru

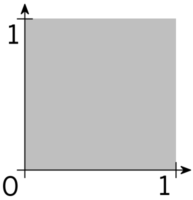


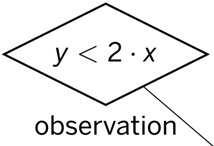
$$m0 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;\ y \leftsquigarrow \textbf{uniform}\ 0\ 1;\ \textbf{return}\ (x, y)\}$$

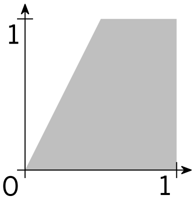
$$m1 = \textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;\ y \leftsquigarrow \textbf{uniform}\ 0\ 1;\ \textbf{observe}\ y < 2 \cdot x;\ \textbf{return}\ (x, y)\}$$

$$E_{m1}(\lambda(x,y).\,x) = \frac{\int_{m1} x\, d(x,y)}{\int_{m1} 1\, d(x,y)} = \frac{\int_{m0} \langle y < 2 \cdot x \rangle \cdot x\, d(x,y)}{\int_{m0} \langle y < 2 \cdot x \rangle \cdot 1\, d(x,y)} = \frac{11/24}{3/4} = 11/18$$

Observation, inference, and query in core Hakaru





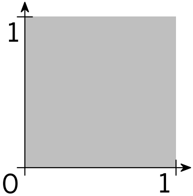


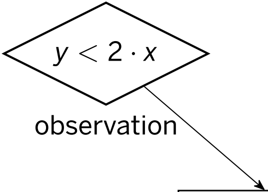
$$\begin{aligned} m0 = \textbf{do}\ \{ & x \leftsquigarrow \textbf{uniform}\ 0\ 1;\\ & y \leftsquigarrow \textbf{uniform}\ 0\ 1;\\ & \textbf{return}\ (x,y)\} \end{aligned}$$

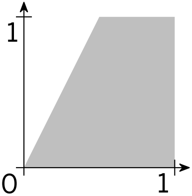
$$\begin{aligned} m1 = \textbf{do}\ \{ & x \leftsquigarrow \textbf{uniform}\ 0\ 1;\\ & y \leftsquigarrow \textbf{uniform}\ 0\ 1;\\ & \textbf{observe}\ y < 2\cdot x;\\ & \textbf{return}\ (x,y)\} \end{aligned}$$

$$E_{m1}(\lambda(x,y).\,x) = \frac{\int_{m1} x\, d(x,y)}{\int_{m1} 1\, d(x,y)} = \frac{\int_{m0} \langle y < 2\cdot x\rangle \cdot x\, d(x,y)}{\int_{m0} \langle y < 2\cdot x\rangle \cdot 1\, d(x,y)} = \frac{11/24}{3/4} = 11/18$$

Observation, inference, and query in core Hakaru





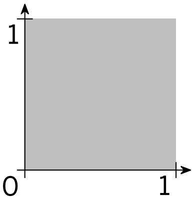


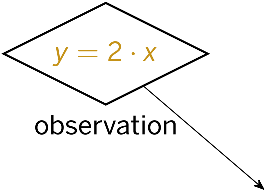
$$
\begin{aligned}
m0 = \textbf{do}\ \{&x \leftsquigarrow \textbf{uniform}\ 0\ 1;\\
&y \leftsquigarrow \textbf{uniform}\ 0\ 1;\\
&\textbf{return}\ (x, y)\}
\end{aligned}
$$

$$
\begin{aligned}
m1 = \textbf{do}\ \{&x \leftsquigarrow \textbf{uniform}\ 0\ 1;\\
&y \leftsquigarrow \textbf{uniform}\ 0\ 1;\\
&\textbf{observe}\ y < 2 \cdot x;\\
&\textbf{return}\ (x, y)\}
\end{aligned}
$$

$$E_{m1}(\lambda(x,y).\,x) = \frac{\int_{m1} x\, d(x,y)}{\int_{m1} 1\, d(x,y)} = \frac{\int_{m0} \langle y < 2 \cdot x \rangle \cdot x\, d(x,y)}{\int_{m0} \langle y < 2 \cdot x \rangle \cdot 1\, d(x,y)} = \frac{11/24}{3/4} = 11/18$$

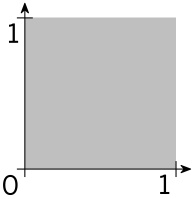
Observation, inference, and query in core Hakaru

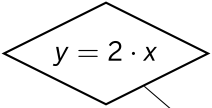




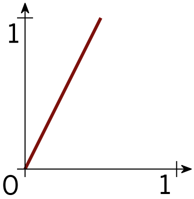
$$
\begin{aligned}
m0 = \textbf{do}\ \{ & x \leftsquigarrow \textbf{uniform}\ 0\ 1; \\
& y \leftsquigarrow \textbf{uniform}\ 0\ 1; \\
& \textbf{return}\ (x, y) \}
\end{aligned}
$$

Observation, inference, and query in core Hakaru





observation

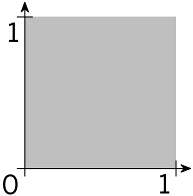


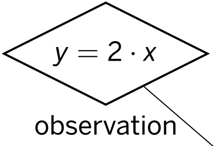

```
m0 = do {x ↜ uniform 0 1;
         y ↜ uniform 0 1;
         return (x, y)}
```

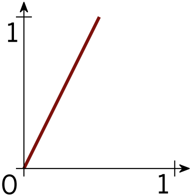
```
m2 = do {x ↜ uniform 0 1;
         y ↜ uniform 0 1;
         observe y = 2 · x;
         return (x, y)}
```

$$E_{m2}(\lambda(x,y).\,x) = \frac{\int_{m2} x\, d(x,y)}{\int_{m2} 1\, d(x,y)} = \frac{\int_{m0} \langle y = 2\cdot x\rangle \cdot x\, d(x,y)}{\int_{m0} \langle y = 2\cdot x\rangle \cdot 1\, d(x,y)} = \frac{0}{0}$$

Observation, inference, and query in core Hakaru





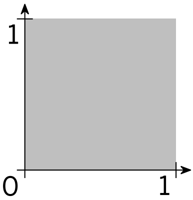



```
m0 = do {x ↜ uniform 0 1;
         y ↜ uniform 0 1;
         return (x, y)}
```

```
m2 = do {x ↜ uniform 0 1;
         y ↜ uniform 0 1;
         observe y = 2 · x;
         return (x, y)}
```

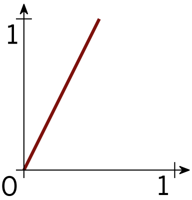
$$E_{m2}(\lambda(x,y).\,x) = \frac{\int_{m2} x \, d(x,y)}{\int_{m2} 1 \, d(x,y)} = \frac{\int_{m0} \langle y = 2 \cdot x \rangle \cdot x \, d(x,y)}{\int_{m0} \langle y = 2 \cdot x \rangle \cdot 1 \, d(x,y)} = \frac{0}{0}$$

Observation, inference, and query in core Hakaru



$y = 2 \cdot x$

observation

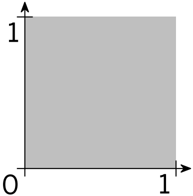


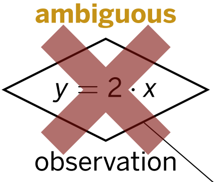
$$\begin{aligned} m0 = \textbf{do}\ \{ & x \leftsquigarrow \textbf{uniform}\ 0\ 1; \\ & y \leftsquigarrow \textbf{uniform}\ 0\ 1; \\ & \textbf{return}\ (x, y)\} \end{aligned}$$

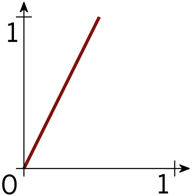
$$\begin{aligned} m2 = \textbf{do}\ \{ & x \leftsquigarrow \textbf{uniform}\ 0\ 1; \\ & y \leftsquigarrow \textbf{uniform}\ 0\ 1; \\ & \textbf{observe}\ y = 2 \cdot x; \\ & \textbf{return}\ (x, y)\} \end{aligned}$$

$$E_{m2}(\lambda(x,y).\,x) = \frac{\int_{m2} x \, d(x,y)}{\int_{m2} 1 \, d(x,y)} = \frac{\int_{m0} \langle y = 2 \cdot x \rangle \cdot x \, d(x,y)}{\int_{m0} \langle y = 2 \cdot x \rangle \cdot 1 \, d(x,y)} = \frac{0}{0}$$

Observation, inference, and query in core Hakaru





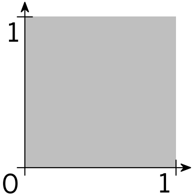


$m0 = \textbf{do}\ \{x \leftharpoondown \textbf{uniform}\ 0\ 1;$
$\quad y \leftharpoondown \textbf{uniform}\ 0\ 1;$
$\quad \textbf{return}\ (x, y)\}$

$m2 = \textbf{do}\ \{x \leftharpoondown \textbf{uniform}\ 0\ 1;$
$\quad y \leftharpoondown \textbf{uniform}\ 0\ 1;$
$\quad \textbf{observe}\ y = 2 \cdot x;$
$\quad \textbf{return}\ (x, y)\}$

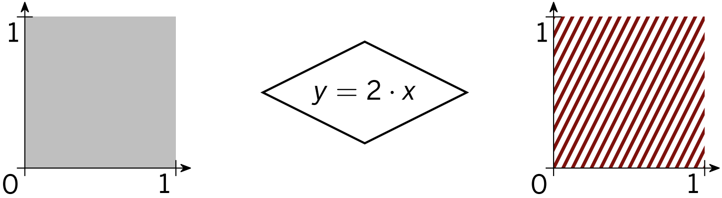
$$E_{m2}(\lambda(x,y).\,x) = \frac{\int_{m2} x \, d(x,y)}{\int_{m2} 1 \, d(x,y)} = \frac{\int_{m0} \langle y = 2 \cdot x \rangle \cdot x \, d(x,y)}{\int_{m0} \langle y = 2 \cdot x \rangle \cdot 1 \, d(x,y)} = \frac{0}{0}$$

Observation of measure-zero sets is paradoxical

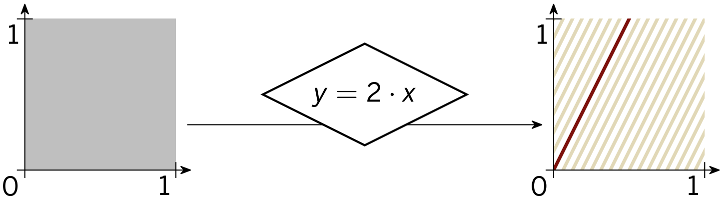


$$y = 2 \cdot x$$

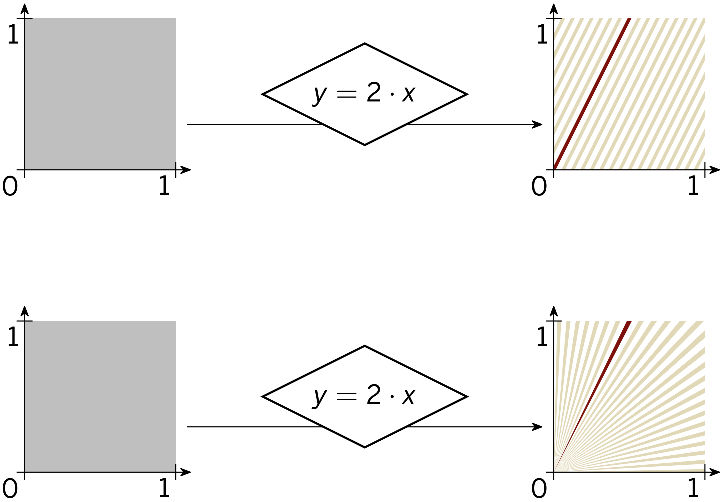
Observation of measure-zero sets is paradoxical



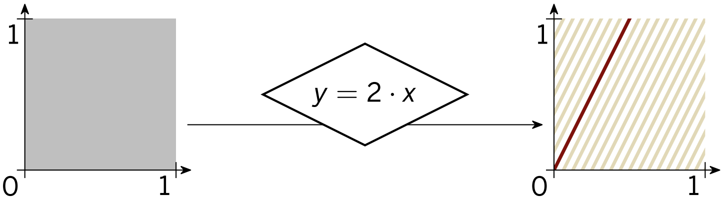
Observation of measure-zero sets is paradoxical



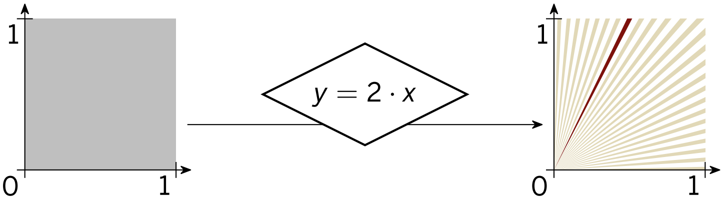
Observation of measure-zero sets is paradoxical



Observation of measure-zero sets is paradoxical

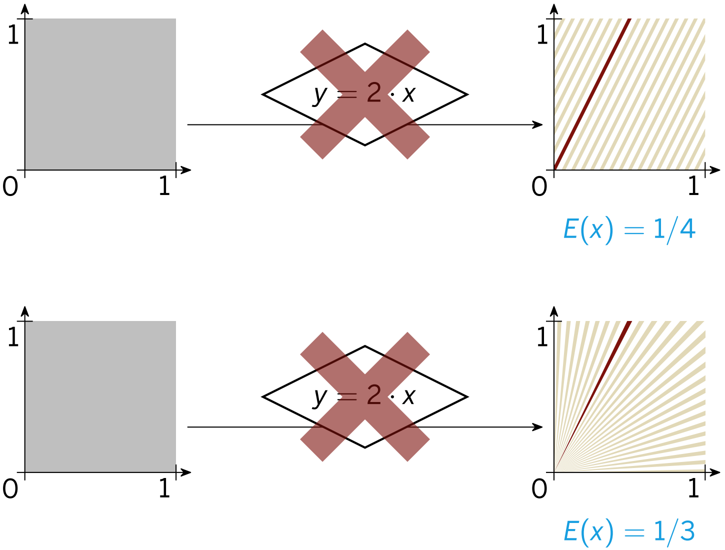


$E(x) = 1/4$

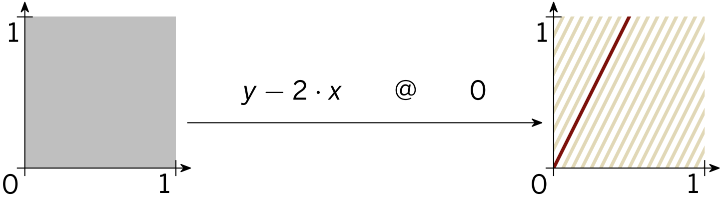


$E(x) = 1/3$

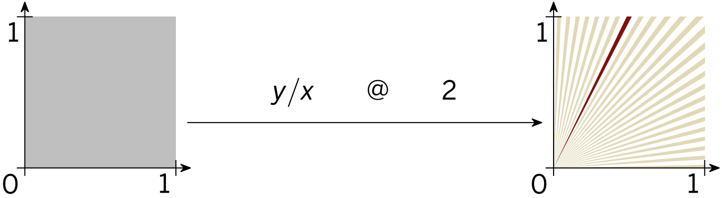
Observation of measure-zero sets is paradoxical



Resolving the paradox via disintegration

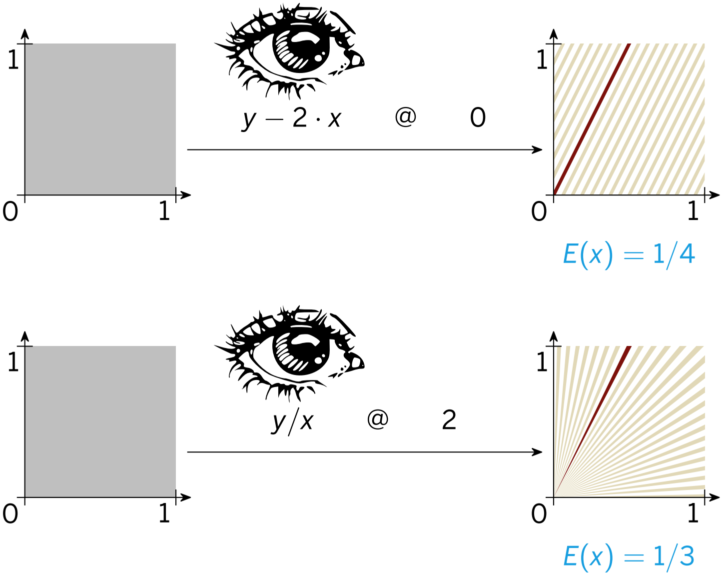


$E(x) = 1/4$

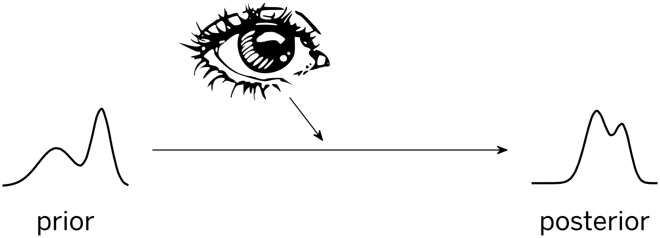


$E(x) = 1/3$

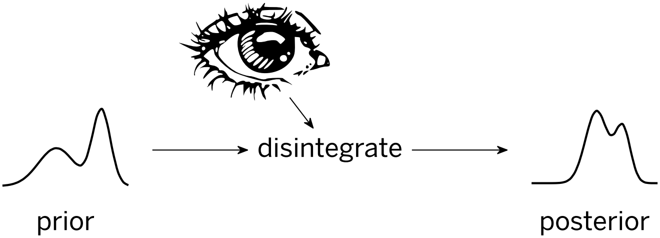
Resolving the paradox via disintegration



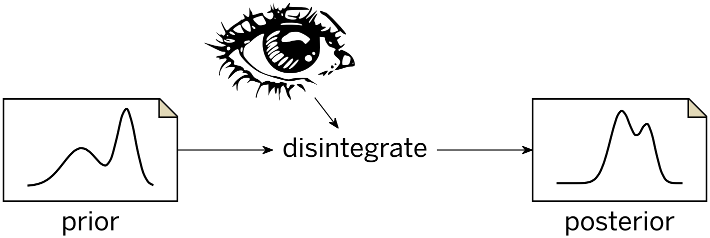
Resolving the paradox via disintegration



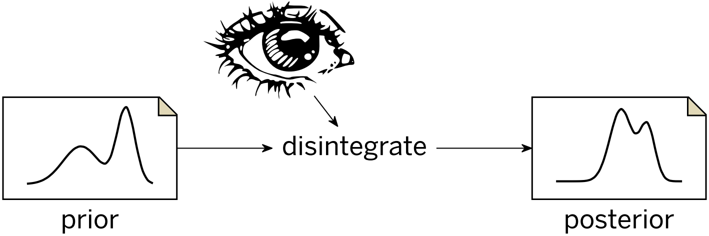
Resolving the paradox via disintegration



Resolving the paradox via disintegration



Resolving the paradox via disintegration



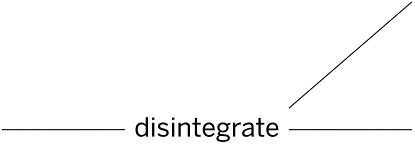
Soundness: If the disintegrator succeeds then the result is correct.

1. **Motivate** by puzzle
2. **Specify** by semantics
3. **Implement** by derivation

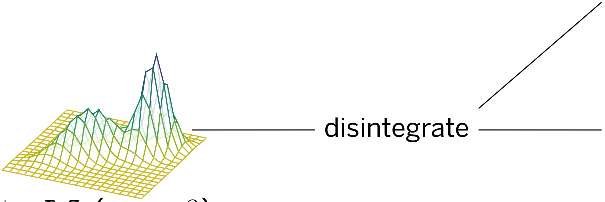
Specifying disintegration by semantics

disintegrate

Specifying disintegration by semantics

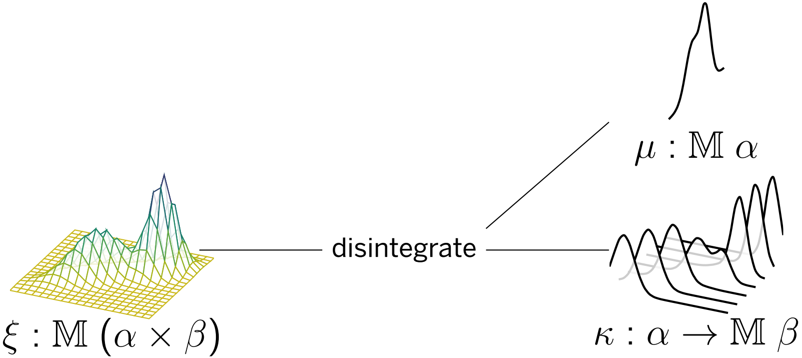


Specifying disintegration by semantics

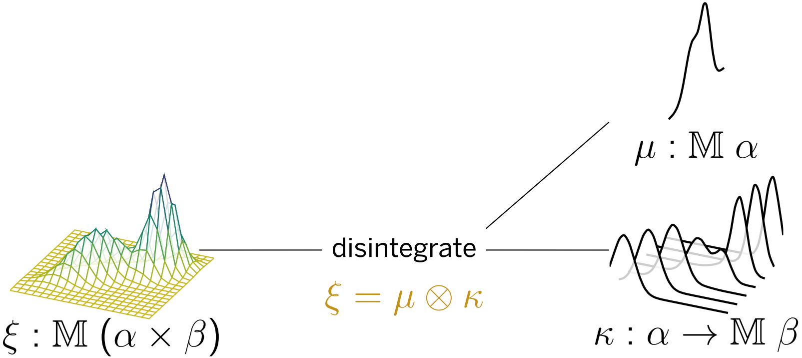


$$\xi : \mathbb{M}\,(\alpha \times \beta)$$

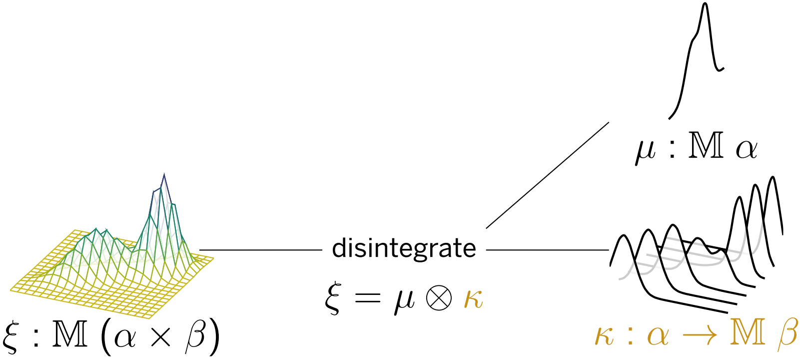
Specifying disintegration by semantics



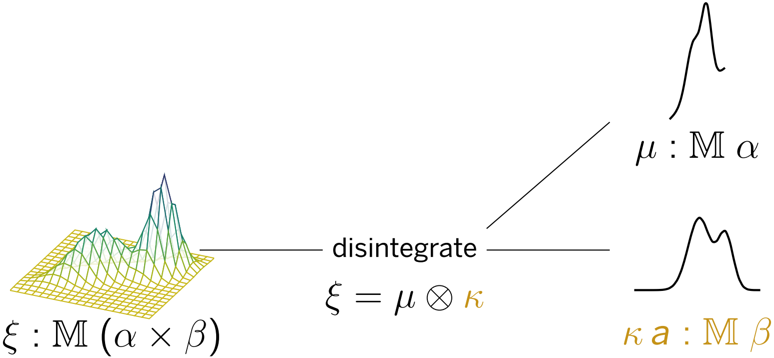
Specifying disintegration by semantics



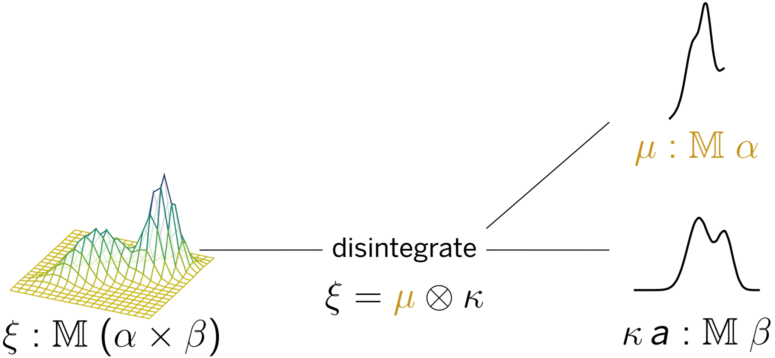
Specifying disintegration by semantics



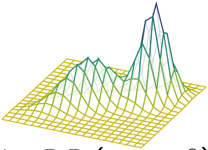
Specifying disintegration by semantics



Specifying disintegration by semantics



Specifying disintegration by semantics



$\xi : \mathbb{M}\,(\alpha \times \beta)$

$\xi = \mu \otimes \kappa$

do $\{a \leftsquigarrow$  ;

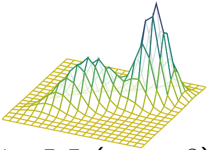
$\mu : \mathbb{M}\ \alpha$

$b \leftsquigarrow$  ;

$\kappa\ a : \mathbb{M}\ \beta$

return $(a, b)\}$

Specifying disintegration by semantics



$\xi : \mathbb{M}\,(\alpha \times \beta)$

$$\Longrightarrow$$

$\xi = \mu \otimes \kappa$

do $\{a \leftsquigarrow$

;

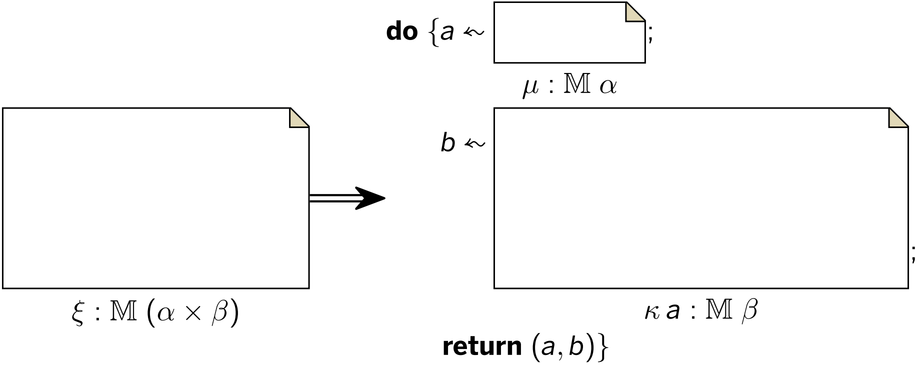
$\mu : \mathbb{M}\ \alpha$

$b \leftsquigarrow$

;

$\kappa\, a : \mathbb{M}\ \beta$

return $(a, b)\}$


do $\{a \leftsquigarrow$;
$\mu : \mathbb{M}\ \alpha$
$b \leftsquigarrow$;
$\xi : \mathbb{M}\ (\alpha \times \beta)$
$\kappa\ a : \mathbb{M}\ \beta$
return $(a, b)\}$

$\alpha = \mathbb{R}$

$\beta = \mathbb{R} \times \mathbb{R}$

do $\{a \leftsquigarrow$;

$\mu : \mathbb{M}\ \alpha$

$b \leftsquigarrow$;

$\kappa\, a : \mathbb{M}\ \beta$

return $(a, b)\}$

$\xi : \mathbb{M}\ (\alpha \times \beta)$ $\Rightarrow$

do $\{x \leftsquigarrow$ **uniform** 0 1;
$y \leftsquigarrow$ **uniform** 0 1;
return $(x, y)\}$

prior : $\mathbb{M}\ \beta$

$y - 2 \cdot x$

$: \alpha$

observation

$\alpha = \mathbb{R}$

$\beta = \mathbb{R} \times \mathbb{R}$

do $\{a \leftsquigarrow$;

$\mu : \mathbb{M}\ \alpha$

$\xi : \mathbb{M}\ (\alpha \times \beta)$

$b \leftsquigarrow$;

$\kappa\ a : \mathbb{M}\ \beta$

return $(a, b)\}$

do $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$y \leftsquigarrow$ **uniform** $0\ 1;$
return $(x, y)\}$

prior $: \mathbb{M}\ \beta$

$\alpha = \mathbb{R}$

$\beta = \mathbb{R} \times \mathbb{R}$

$y - 2 \cdot x$

$: \alpha$

observation

do $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$y \leftsquigarrow$ **uniform** $0\ 1;$
let $a = y - 2 \cdot x;$
return $(a, (x, y))\}$

$\xi : \mathbb{M}\ (\alpha \times \beta)$

do $\{a \leftsquigarrow$ [ ] ;

$\mu : \mathbb{M}\ \alpha$

$b \leftsquigarrow$ [ ] ;

$\kappa\ a : \mathbb{M}\ \beta$

return $(a, b)\}$

do $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$y \leftsquigarrow$ **uniform** $0\ 1;$
return $(x, y)\}$

prior $: \mathbb{M}\ \beta$

$\alpha = \mathbb{R}$

$\beta = \mathbb{R} \times \mathbb{R}$

$y - 2 \cdot x$

$: \alpha$

observation

do $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$\quad y \leftsquigarrow$ **uniform** $0\ 1;$
$\quad$ **let** $a = y - 2 \cdot x;$
$\quad$ **return** $(a, (x, y))\}$

$\xi : \mathbb{M}\ (\alpha \times \beta)$

do $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$\quad y \leftsquigarrow$ **uniform** $0\ 1;$
$\quad$ **return** $(x, y)\}$

prior $: \mathbb{M}\ \beta$

do $\{a \leftsquigarrow$ **lebesgue**$;$

$\mu : \mathbb{M}\ \alpha$

$b \leftsquigarrow$ **do** $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$\quad$ **observe** $0 < a + 2 \cdot x < 1;$
$\quad$ **return** $(x, a + 2 \cdot x)\}$ $;$

$\kappa\, a : \mathbb{M}\ \beta$

return $(a, b)\}$

$\alpha = \mathbb{R}$

$\beta = \mathbb{R} \times \mathbb{R}$

$y - 2 \cdot x$

$: \alpha$

observation

do $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$\quad y \leftsquigarrow$ **uniform** $0\ 1;$
$\quad$ **let** $a = y - 2 \cdot x;$
$\quad$ **return** $(a, (x, y))\}$

$\xi : \mathbb{M}\ (\alpha \times \beta)$

do $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$\quad y \leftsquigarrow$ **uniform** $0\ 1;$
$\quad$ **return** $(x, y)\}$

prior $: \mathbb{M}\ \beta$

$\Longrightarrow$

do $\{a \leftsquigarrow$ **lebesgue**;

$\mu : \mathbb{M}\ \alpha$

$b \leftsquigarrow$ **do** $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$\quad$ **observe** $0 < a + 2 \cdot x < 1;$
$\quad$ **return** $(x, a + 2 \cdot x)\}$;

$\kappa\ a : \mathbb{M}\ \beta$

return $(a, b)\}$

do $\{x \leftsquigarrow$ **uniform** $0\ 1;$
$\quad$ **observe** $0 < 0 + 2 \cdot x < 1;$
$\quad$ **return** $(x, 0 + 2 \cdot x)\}$

$\kappa\ 0 : \mathbb{M}\ \beta$

$\alpha = \mathbb{R}$

$\beta = \mathbb{R} \times \mathbb{R}$

y/*x*

: α

observation

```
do {x ⇜ uniform 0 1;
    y ⇜ uniform 0 1;
    let a = y/x;
    return (a, (x, y))}
```

$\xi : \mathbb{M}\ (\alpha \times \beta)$

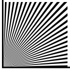
```
do {x ⇜ uniform 0 1;
    y ⇜ uniform 0 1;
    return (x, y)}
```

prior : $\mathbb{M}\ \beta$

```
do {a ⇜ lebesgue;
```

$\mu : \mathbb{M}\ \alpha$

```
    b ⇜ do {x ⇜ uniform 0 1;
            observe 0 < a · x < 1;
            factor x;
            return (x, a · x)};
```



$\kappa\ a : \mathbb{M}\ \beta$

```
    return (a, b)}
```

```
do {x ⇜ uniform 0 1;
    observe 0 < 2 · x < 1;
    factor x;
    return (x, 2 · x)}
```

$\kappa\ 2 : \mathbb{M}\ \beta$

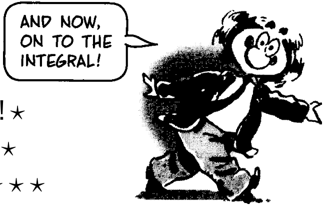
$$[\![\; \square \;]\!] = [\![\; \mathbf{do}\ \{a \leftsquigarrow \square;\quad b \leftsquigarrow \square;\quad \mathbf{return}\ (a, b)\} \;]\!]$$

Measure semantics

⋆ Compositional denotation! ⋆

⋆⋆ Equational reasoning! ⋆⋆

⋆⋆⋆ Integrator formulation! ⋆⋆⋆



Integrator semantics

$$[\![\mathbb{M}\ \alpha]\!] = \overbrace{([\![\alpha]\!] \to \mathbb{R})}^{\text{integrand}} \to \mathbb{R}$$

$$[\![\textbf{uniform}\ 0\ 1]\!] = \lambda f.\int_0^1 f(x)\,dx$$

$$[\![\textbf{lebesgue}]\!] = \lambda f.\int_{-\infty}^{\infty} f(x)\,dx$$

$$[\![\textbf{return}\ (x,y)]\!] = \lambda f.\,f(x,y)$$

$$[\![\textbf{do}\ \{x \leftsquigarrow m;\ M\}]\!] = \lambda f.\,[\![m]\!](\lambda x.\,[\![M]\!]f)$$

$$\left[\!\!\left[\begin{array}{r}\textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;\\ y \leftsquigarrow \textbf{uniform}\ 0\ 1;\\ \textbf{return}\ (x,y)\}\end{array}\right]\!\!\right] = \lambda f.\int_0^1\int_0^1 f(x,y)\,dy\,dx$$

Integrator semantics

$$
\begin{aligned}
[\![\mathbb{M}\ \alpha]\!] &= \overbrace{([\![\alpha]\!] \to \mathbb{R})}^{\text{integrand}} \to \mathbb{R} \\
[\![\textbf{uniform}\ 0\ 1]\!] &= \lambda f.\ \int_0^1 f(x)\,dx \\
[\![\textbf{lebesgue}]\!] &= \lambda f.\ \int_{-\infty}^{\infty} f(x)\,dx \\
[\![\textbf{return}\ (x,y)]\!] &= \lambda f.\ f(x,y) \\
[\![\textbf{do}\ \{x \leftsquigarrow m;\ M\}]\!] &= \lambda f.\ [\![m]\!](\lambda x.\ [\![M]\!]f) \\
\left[\!\!\left[\begin{array}{l}\textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1; \\ \quad y \leftsquigarrow \textbf{uniform}\ 0\ 1; \\ \quad \textbf{return}\ (x,y)\}\end{array}\right]\!\!\right] &= \lambda f.\ \int_0^1 \int_0^1 f(x,y)\,dy\,dx
\end{aligned}
$$

Integrator semantics

$$[\![\mathbb{M}\ \alpha]\!] = \overbrace{([\![\alpha]\!] \to \mathbb{R})}^{\text{integrand}} \to \mathbb{R}$$

$$[\![\textbf{uniform}\ 0\ 1]\!] = \lambda f.\ \int_0^1 f(x)\,dx$$

$$[\![\textbf{lebesgue}]\!] = \lambda f.\ \int_{-\infty}^{\infty} f(x)\,dx$$

$$[\![\textbf{return}\ (x,y)]\!] = \lambda f.\ f(x,y)$$

$$[\![\textbf{do}\ \{x \leftsquigarrow m;\ M\}]\!] = \lambda f.\ [\![m]\!](\lambda x.\ [\![M]\!]f)$$

$$\left[\!\!\left[\begin{array}{l}\textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;\\ \quad y \leftsquigarrow \textbf{uniform}\ 0\ 1;\\ \quad \textbf{return}\ (x,y)\}\end{array}\right]\!\!\right] = \lambda f.\ \int_0^1 \int_0^1 f(x,y)\,dy\,dx$$

"fantastic introduction!"

★ "a pleasure to read!" ★

"very polished!"

★ "*best written of the last 30 papers I have read!*" ★

★ "loved reading!" ★

★ "deft!"

★ "self contained!" ★

"gentle!" ★

★ "easy to follow!" ★

"beautifully explained!"

"fantastic introduction!"

★ "a pleasure to read!" ★

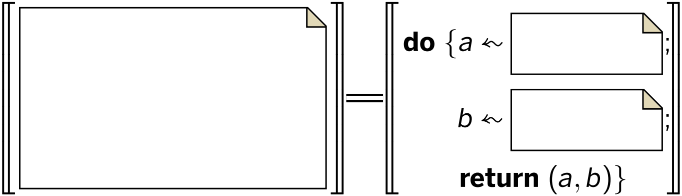
"very polished!"

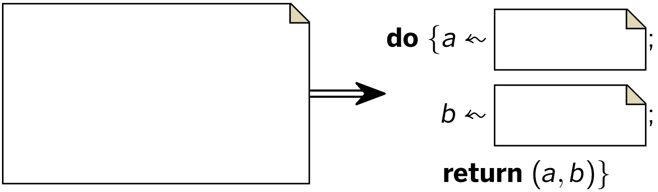
★ "best written of the last 30 papers I have read!" ★ "loved reading!" ★

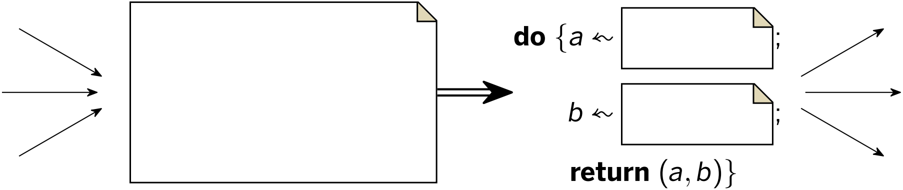
"PLDI readers without lots of background in probability theory should be able to follow; this is impressive"

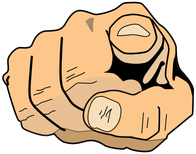
★ "easy to follow!" ★

"beautifully explained!"

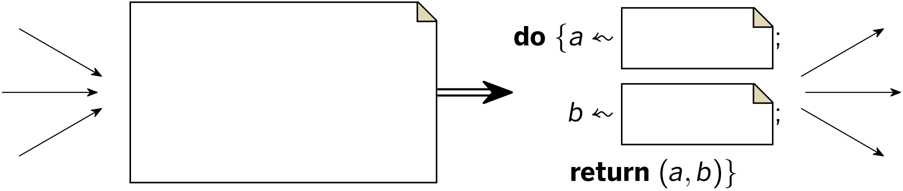

do {a ⇜
;
b ⇜
;
return (a, b)}

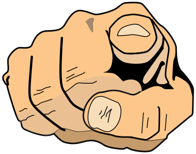

do {a ⇜ ;
b ⇜ ;
return (a, b)}


do {a ↜ ;
b ↜ ;
return (a, b)}

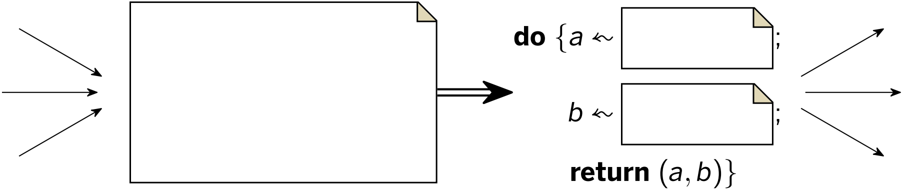


1. Probabilistic programs denote distributions
2. Exact inference by transforming terms





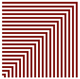
1. Probabilistic programs denote distributions
2. Exact inference by transforming terms



1. **Motivate** by puzzle
2. **Specify** by semantics
3. **Implement** by derivation

When it works

- $y - 2 \cdot x$ $\quad y/x$ $\quad \max(x, y)$ $\quad \ldots$

- multivariate Gaussian distributions
  (for regression and dynamics)
- mixtures of distributions
  (for classifying points and documents)
- seismic event detection (Arora et al.)
- point masses' total momentum (Afshar et al.)

When it works

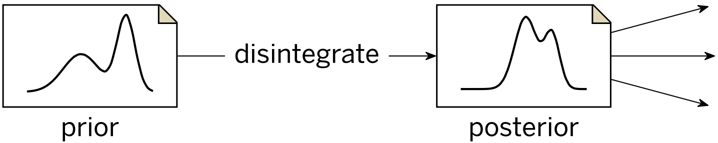
- $y - 2 \cdot x \qquad y/x \qquad \max(x, y) \qquad \ldots$

- multivariate Gaussian distributions
  (for regression and dynamics)
- mixtures of distributions
  (for classifying points and documents)
- seismic event detection (Arora et al.)
- point masses' total momentum (Afshar et al.)

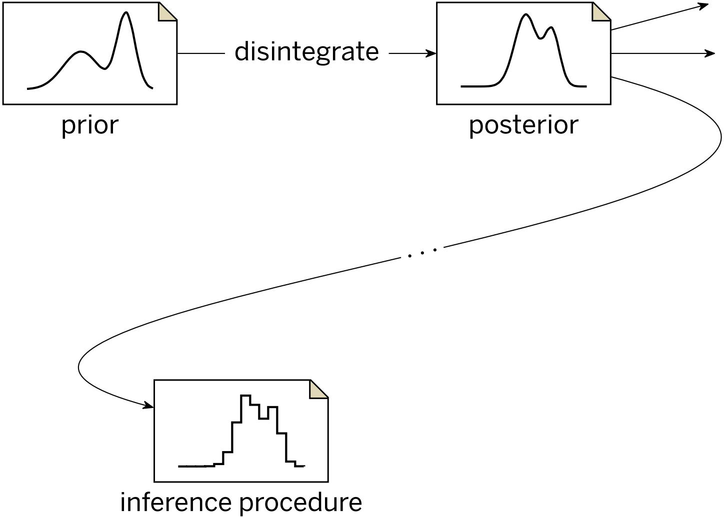
$$
\begin{aligned}
\textbf{do}\ \{ & x \leftsquigarrow \cdots; \\
& y \leftsquigarrow \cdots; \\
& z \leftsquigarrow \cdots; \\
& \textbf{return}\ (f(x, y, z), \ldots)\}
\end{aligned}
$$

invertible

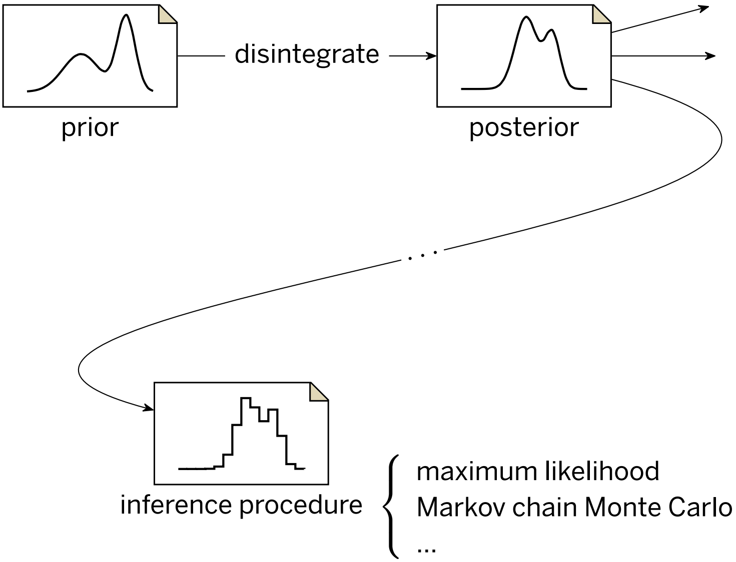
Where it helps



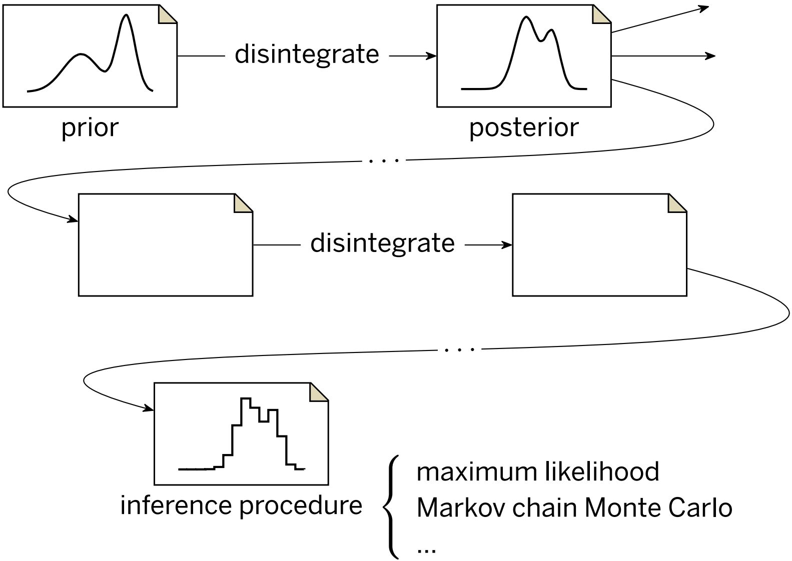
Where it helps



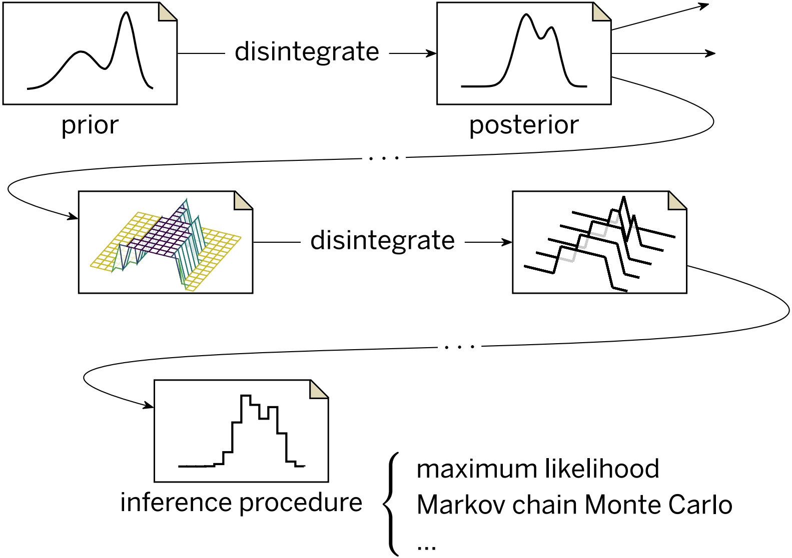
Where it helps



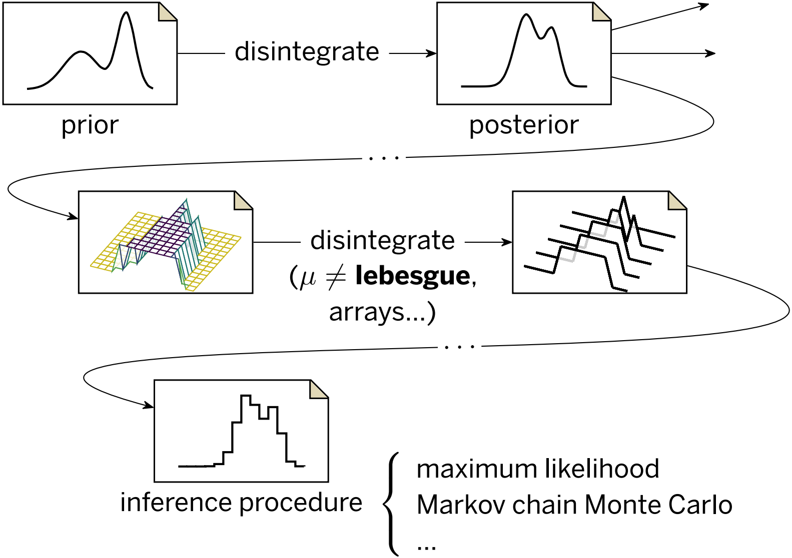
Where it helps

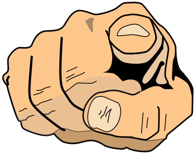


Where it helps



Where it helps

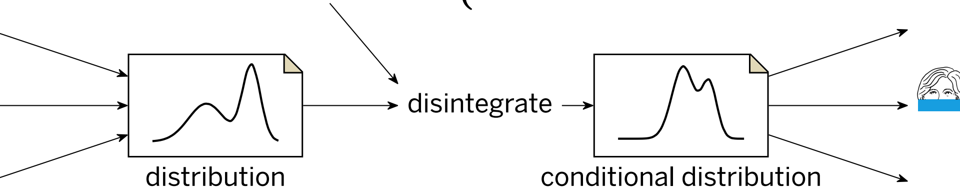




1. Probabilistic programs denote distributions
2. Exact inference by transforming terms

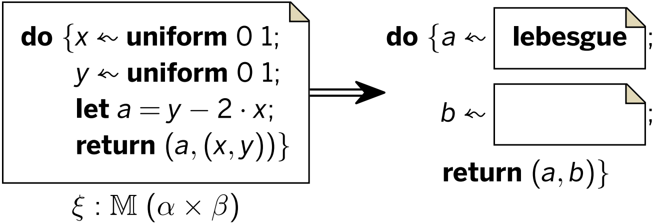
condition $: \alpha$

- dependent variable of regression
- noisy measurement of location
- total momentum of point masses
- detected amplitude of seismic event
- … …



1. **Motivate** by puzzle
2. **Specify** by semantics
3. **Implement** by derivation

Induction hypothesis for automatic disintegrator



Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\,\mathbf{do}\ \{h';\ M\}$.

$$\begin{aligned}&\mathbf{do}\ \{x \leftsquigarrow \mathbf{uniform}\ 0\ 1;\\ &\quad y \leftsquigarrow \mathbf{uniform}\ 0\ 1;\\ &\quad \mathbf{let}\ a = y - 2 \cdot x;\\ &\quad \mathbf{return}\ (a, (x, y))\}\end{aligned} \quad\Longrightarrow\quad \begin{aligned}&\mathbf{do}\ \{a \leftsquigarrow \boxed{\mathbf{lebesgue}};\\ &\quad b \leftsquigarrow \boxed{};\\ &\quad \mathbf{return}\ (a, b)\}\end{aligned}$$

$\xi : \mathbb{M}\ (\alpha \times \beta)$

Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\, \textbf{do}\ \{h';\, M\}$.

$$\textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;\ y \leftsquigarrow \textbf{uniform}\ 0\ 1;\ \textbf{let}\ a = y - 2 \cdot x;\ \textbf{return}\ (a, (x, y))\}$$

$\xi : \mathbb{M}\ (\mathbb{R} \times \beta)$

$\Longrightarrow$

$$\textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ b \leftsquigarrow \square;\ \textbf{return}\ (a, b)\}$$

Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\,\textbf{do}\ \{h';\, M\}$.

$\textbf{do}\ \{x \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad y \leftsquigarrow \textbf{uniform}\ 0\ 1;$
$\quad \textbf{let}\ a = y - 2 \cdot x;$
$\quad \textbf{return}\ (a, (x, y))\}$

$\xi : \mathbb{M}\ (\mathbb{R} \times \beta)$

$\Longrightarrow$

$\textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};$
$\quad b \leftsquigarrow \square;$
$\quad \textbf{return}\ (a, b)\}$

Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\,\mathbf{do}\ \{h';\ M\}$.

$$\begin{aligned}&\mathbf{do}\ \{x \leftsquigarrow \mathbf{uniform}\ 0\ 1;\\ &\quad y \leftsquigarrow \mathbf{uniform}\ 0\ 1;\\ &\quad a \leftsquigarrow m;\\ &\quad \mathbf{return}\ (a, (x, y))\}\end{aligned}$$

$\xi : \mathbb{M}\ (\mathbb{R} \times \beta)$

$\Longrightarrow$

$$\begin{aligned}&\mathbf{do}\ \{a \leftsquigarrow \mathbf{lebesgue};\\ &\quad b \leftsquigarrow \square;\\ &\quad \mathbf{return}\ (a, b)\}\end{aligned}$$

Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\,\mathbf{do}\ \{h'; M\}$.

$$\mathbf{do}\ \{x \leftsquigarrow \mathbf{uniform}\ 0\ 1;\ y \leftsquigarrow \mathbf{uniform}\ 0\ 1;\ a \leftsquigarrow m;\ \mathbf{return}\ (a, (x, y))\}$$

$$\xi : \mathbb{M}\ (\mathbb{R} \times \beta)$$

$$\Longrightarrow$$

$$\mathbf{do}\ \{a \leftsquigarrow \mathbf{lebesgue};\ b \leftsquigarrow \square;\ \mathbf{return}\ (a, b)\}$$

Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\,\mathbf{do}\ \{h';\, M\}$.

do {
    h;
    $a \leftsquigarrow m$;
    return $(a, (x, y))$}

$\xi : \mathbb{M}\,(\mathbb{R} \times \beta)$

$\Longrightarrow$

do $\{a \leftsquigarrow$ **lebesgue**;
    $b \leftsquigarrow$ [ ];
    return $(a, b)\}$

Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\,$**do** $\{h';\, M\}$.

do $\{$
h;
$a \leftsquigarrow m$;
return $(a, (x, y))\}$

$\xi : \mathbb{M}\,(\mathbb{R} \times \beta)$

$\Longrightarrow$

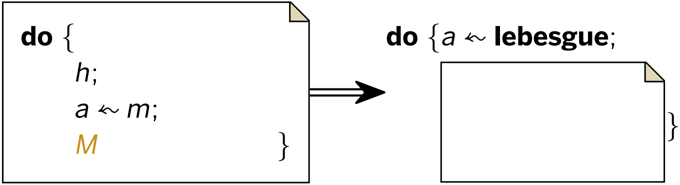
do $\{a \leftsquigarrow$ **lebesgue**;
$b \leftsquigarrow$ [ ] ;
return $(a, b)\}$

Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\,\textbf{do}\ \{h';\ M\}$.



Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\, \textbf{do}\ \{h';\ M\}$.

$$\textbf{do}\ \{\ h;\ a \leftsquigarrow m;\ M\ \} \quad \Longrightarrow \quad \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll m\ a\ \overline{M}\ h\ \}$$

Induction hypothesis for automatic disintegrator

Specialize: $\alpha = \mathbb{R}$, $\mu =$ **lebesgue**.

Generalize: observable action m, heap h, final action M.

Define continuation $\overline{M} = \lambda h'.\,\textbf{do}\ \{h';\, M\}$.

$$\textbf{do}\ \{\ h;\ a \leftsquigarrow m;\ M\ \} \implies \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll m\ a\ \overline{M}\ h\ \}$$

Implement $\lll$ by equational reasoning from this specification.

Case analysis on m:

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll m\ a\ \overline{M}\ h\}$

Case $m = \textbf{uniform}\ 0\ 1$:

$$
\begin{aligned}
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{uniform}\ 0\ 1;\ M\} \\
= \quad & \{\ \text{probability density of } m \text{ (Bhat et al.)}\ \} \\
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1 \rangle;\ M\} \\
= \quad & \{\ \text{exchange integrals using Tonelli's theorem}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1 \rangle;\ h;\ M\} \\
= \quad & \{\ \text{beta; recall } \overline{M} = \lambda h'.\,\textbf{do}\ \{h';\ M\}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1 \rangle;\ \overline{M}\ h\}
\end{aligned}
$$

So define

$$\lll (\textbf{uniform}\ 0\ 1)\ a\ c\ h \quad = \quad \textbf{do}\ \{\textbf{factor}\ \langle 0 < a < 1 \rangle;\ c\ h\}$$

Similarly for other primitive continuous distributions.
The disintegration fused into most inference methods ends here.

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll m\ a\ \overline{M}\ h\}$

Case $m = \textbf{uniform}\ 0\ 1$:

$$
\begin{aligned}
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{uniform}\ 0\ 1;\ M\} \\
= \quad & \{\ \text{probability density of } m \text{ (Bhat et al.)}\ \} \\
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\ M\} \\
= \quad & \{\ \text{exchange integrals using Tonelli's theorem}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\ h;\ M\} \\
= \quad & \{\ \text{beta; recall } \overline{M} = \lambda h'.\,\textbf{do}\ \{h';\ M\}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\ \overline{M}\ h\}
\end{aligned}
$$

So define

$$\lll (\textbf{uniform}\ 0\ 1)\ a\ c\ h \quad = \quad \textbf{do}\ \{\textbf{factor}\ \langle 0 < a < 1\rangle;\ c\ h\}$$

Similarly for other primitive continuous distributions.
The disintegration fused into most inference methods ends here.

Goal: $\textbf{do}\ \{h;\, a \leftsquigarrow m;\, M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \overline{M}\ h\}$

Case $m = \textbf{uniform}\ 0\ 1$:

$$
\begin{aligned}
& \textbf{do}\ \{h;\, a \leftsquigarrow \textbf{uniform}\ 0\ 1;\, M\} \\
= \quad & \{\text{ probability density of } m \text{ (Bhat et al.) }\} \\
& \textbf{do}\ \{h;\, a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\, M\} \\
= \quad & \{\text{ exchange integrals using Tonelli's theorem }\} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\, h;\, M\} \\
= \quad & \{\text{ beta; recall } \overline{M} = \lambda h'.\, \textbf{do}\ \{h';\, M\}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\, \overline{M}\ h\}
\end{aligned}
$$

So define

$$\lll\ (\textbf{uniform}\ 0\ 1)\ a\ c\ h \quad = \quad \textbf{do}\ \{\textbf{factor}\ \langle 0 < a < 1\rangle;\, c\ h\}$$

Similarly for other primitive continuous distributions.
The disintegration fused into most inference methods ends here.

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll m\ a\ \overline{M}\ h\}$

Case $m = \textbf{uniform}\ 0\ 1$:

$$
\begin{aligned}
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{uniform}\ 0\ 1;\ M\} \\
= \quad & \{\text{ probability density of } m \text{ (Bhat et al.) }\} \\
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\ M\} \\
= \quad & \{\text{ exchange integrals using Tonelli's theorem }\} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\ h;\ M\} \\
= \quad & \{\text{ beta; recall } \overline{M} = \lambda h'.\ \textbf{do}\ \{h';\ M\}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{factor}\ \langle 0 < a < 1\rangle;\ \overline{M}\ h\}
\end{aligned}
$$

So define

$$\lll (\textbf{uniform}\ 0\ 1)\ a\ c\ h \quad = \quad \textbf{do}\ \{\textbf{factor}\ \langle 0 < a < 1\rangle;\ c\ h\}$$

Similarly for other primitive continuous distributions.
The disintegration fused into most inference methods ends here.

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \overline{M}\ h\}$

Case $m = \textbf{return}\ x$: Look up x in $h = (h_1;\ x \leftsquigarrow m;\ h_2)$

$$
\begin{aligned}
& \textbf{do}\ \{h_1;\ x \leftsquigarrow m;\ h_2;\ a \leftsquigarrow \textbf{return}\ x;\ M\} \\
= \quad & \{\ \text{monad laws, beta, alpha}\ \} \\
& \textbf{do}\ \{h_1;\ a \leftsquigarrow m;\ \textbf{let}\ x = a;\ h_2;\ M\} \\
= \quad & \{\ \text{induction hypothesis}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \left(\overline{\textbf{do}\ \{\textbf{let}\ x = a;\ h_2;\ M\}}\right)\ h_1\} \\
= \quad & \{\ \text{beta; recall}\ \overline{M} = \lambda h'.\ \textbf{do}\ \{h';\ M\}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \left(\lambda h'.\ \overline{M}\ (h';\ \textbf{let}\ x = a;\ h_2)\right)\ h_1\}
\end{aligned}
$$

So define

$$\lll\ x\ a\ c\ (h_1;\ x \leftsquigarrow m;\ h_2) \quad = \quad \lll\ m\ a\ \left(\lambda h'.\ \overline{M}\ (h';\ \textbf{let}\ x = a;\ h_2)\right)\ h_1$$

The continuation memoizes.

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \overline{M}\ h\}$

Case $m = \textbf{return}\ x$: Look up x in $h = (h_1;\ x \leftsquigarrow m;\ h_2)$

$$
\begin{aligned}
& \textbf{do}\ \{h_1;\ x \leftsquigarrow m;\ h_2;\ a \leftsquigarrow \textbf{return}\ x;\ M\} \\
= & \quad \{\text{ monad laws, beta, alpha }\} \\
& \textbf{do}\ \{h_1;\ a \leftsquigarrow m;\ \textbf{let}\ x = a;\ h_2;\ M\} \\
= & \quad \{\text{ induction hypothesis }\} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \big(\overline{\textbf{do}\ \{\textbf{let}\ x = a;\ h_2;\ M\}}\big)\ h_1\} \\
= & \quad \{\text{ beta; recall } \overline{M} = \lambda h'.\,\textbf{do}\ \{h';\ M\}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \big(\lambda h'.\,\overline{M}\ (h';\ \textbf{let}\ x = a;\ h_2)\big)\ h_1\}
\end{aligned}
$$

So define

$$\lll\ x\ a\ c\ (h_1;\ x \leftsquigarrow m;\ h_2) \quad = \quad \lll\ m\ a\ \big(\lambda h'.\,\overline{M}\ (h';\ \textbf{let}\ x = a;\ h_2)\big)\ h_1$$

The continuation memoizes.

Goal: $\mathbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \mathbf{do}\ \{a \leftsquigarrow \mathbf{lebesgue};\ \lll\ m\ a\ \overline{M}\ h\}$

Case $m = \mathbf{return}\ x$: Look up x in $h = (h_1;\ x \leftsquigarrow m;\ h_2)$

$$
\begin{aligned}
& \mathbf{do}\ \{h_1;\ x \leftsquigarrow m;\ h_2;\ a \leftsquigarrow \mathbf{return}\ x;\ M\} \\
= & \quad \{\ \text{monad laws, beta, alpha}\ \} \\
& \mathbf{do}\ \{h_1;\ a \leftsquigarrow m;\ \mathbf{let}\ x = a;\ h_2;\ M\} \\
= & \quad \{\ \text{induction hypothesis}\ \} \\
& \mathbf{do}\ \{a \leftsquigarrow \mathbf{lebesgue};\ \lll\ m\ a\ \left(\overline{\mathbf{do}\ \{\mathbf{let}\ x = a;\ h_2;\ M\}}\right)\ h_1\} \\
= & \quad \{\ \text{beta; recall}\ \overline{M} = \lambda h'.\,\mathbf{do}\ \{h';\ M\}\ \} \\
& \mathbf{do}\ \{a \leftsquigarrow \mathbf{lebesgue};\ \lll\ m\ a\ \left(\lambda h'.\,\overline{M}\ (h';\ \mathbf{let}\ x = a;\ h_2)\right)\ h_1\}
\end{aligned}
$$

So define

$$\lll\ x\ a\ c\ (h_1;\ x \leftsquigarrow m;\ h_2) \ = \ \lll\ m\ a\ \left(\lambda h'.\,\overline{M}\ (h';\ \mathbf{let}\ x = a;\ h_2)\right)\ h_1$$

The continuation memoizes.

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \overline{M}\ h\}$

Case $m = \textbf{return}\ (-e)$:

$$\begin{aligned}
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{return}\ (-e);\ M\} \\
= & \quad \{\ \text{monad laws, beta}\ \} \\
& \textbf{do}\ \{h;\ b \leftsquigarrow \textbf{return}\ e;\ \textbf{let}\ a = -b;\ M\} \\
= & \quad \{\ \text{induction hypothesis ...}\ \} \\
& \textbf{do}\ \{b \leftsquigarrow \textbf{lebesgue};\ \textbf{let}\ a = -b;\ \lll\ (\textbf{return}\ e)\ b\ \overline{M}\ h\} \\
= & \quad \{\ \text{change integration variable from } b \text{ to } a\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{let}\ b = -a;\ \lll\ (\textbf{return}\ e)\ b\ \overline{M}\ h\} \\
= & \quad \{\ \text{“parametricity” of } \lll \text{ ...}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ (\textbf{return}\ e)\ (-a)\ \overline{M}\ h\}
\end{aligned}$$

So define

$$\lll\ (\textbf{return}\ (-e))\ a\ c\ h \quad = \quad \lll\ (\textbf{return}\ e)\ (-a)\ c\ h$$

Similarly for other invertible functions: $\log x,\quad y - 2 \cdot x.$

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \overline{M}\ h\}$

Case $m = \textbf{return}\ (-e)$:

$$
\begin{aligned}
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{return}\ (-e);\ M\} \\
= & \quad \{\text{ monad laws, beta }\} \\
& \textbf{do}\ \{h;\ b \leftsquigarrow \textbf{return}\ e;\ \textbf{let}\ a = -b;\ M\} \\
= & \quad \{\text{ induction hypothesis ... }\} \\
& \textbf{do}\ \{b \leftsquigarrow \textbf{lebesgue};\ \textbf{let}\ a = -b;\ \lll\ (\textbf{return}\ e)\ b\ \overline{M}\ h\} \\
= & \quad \{\text{ change integration variable from } b \text{ to } a\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{let}\ b = -a;\ \lll\ (\textbf{return}\ e)\ b\ \overline{M}\ h\} \\
= & \quad \{\text{ "parametricity" of } \lll \text{ ... }\} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ (\textbf{return}\ e)\ (-a)\ \overline{M}\ h\}
\end{aligned}
$$

So define

$$\lll\ (\textbf{return}\ (-e))\ a\ c\ h \quad = \quad \lll\ (\textbf{return}\ e)\ (-a)\ c\ h$$

Similarly for other invertible functions: $\log x, \quad y - 2 \cdot x.$

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \overline{M}\ h\}$

Case $m = \textbf{return}\ (-e)$:

$$
\begin{aligned}
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{return}\ (-e);\ M\} \\
= \quad & \{\ \text{monad laws, beta}\ \} \\
& \textbf{do}\ \{h;\ b \leftsquigarrow \textbf{return}\ e;\ \textbf{let}\ a = -b;\ M\} \\
= \quad & \{\ \text{induction hypothesis ...}\ \} \\
& \textbf{do}\ \{b \leftsquigarrow \textbf{lebesgue};\ \textbf{let}\ a = -b;\ \lll\ (\textbf{return}\ e)\ b\ \overline{M}\ h\} \\
= \quad & \{\ \text{change integration variable from } b \text{ to } a\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{let}\ b = -a;\ \lll\ (\textbf{return}\ e)\ b\ \overline{M}\ h\} \\
= \quad & \{\ \text{“parametricity” of } \lll \text{ ...}\ \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ (\textbf{return}\ e)\ (-a)\ \overline{M}\ h\}
\end{aligned}
$$

So define

$$\lll\ (\textbf{return}\ (-e))\ a\ c\ h \quad = \quad \lll\ (\textbf{return}\ e)\ (-a)\ c\ h$$

Similarly for other invertible functions: $\log x,\quad y - 2 \cdot x.$

Goal: $\textbf{do}\ \{h;\ a \leftsquigarrow m;\ M\} = \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ m\ a\ \overline{M}\ h\}$

Case $m = \textbf{return}\ (-e)$:

$$
\begin{aligned}
& \textbf{do}\ \{h;\ a \leftsquigarrow \textbf{return}\ (-e);\ M\} \\
= \quad & \{ \text{monad laws, beta} \} \\
& \textbf{do}\ \{h;\ b \leftsquigarrow \textbf{return}\ e;\ \textbf{let}\ a = -b;\ M\} \\
= \quad & \{ \text{induction hypothesis ...} \} \\
& \textbf{do}\ \{b \leftsquigarrow \textbf{lebesgue};\ \textbf{let}\ a = -b;\ \lll\ (\textbf{return}\ e)\ b\ \overline{M}\ h\} \\
= \quad & \{ \text{change integration variable from } b \text{ to } a \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \textbf{let}\ b = -a;\ \lll\ (\textbf{return}\ e)\ b\ \overline{M}\ h\} \\
= \quad & \{ \text{"parametricity" of } \lll \text{ ...} \} \\
& \textbf{do}\ \{a \leftsquigarrow \textbf{lebesgue};\ \lll\ (\textbf{return}\ e)\ (-a)\ \overline{M}\ h\}
\end{aligned}
$$

So define

$$\lll\ (\textbf{return}\ (-e))\ a\ c\ h \quad = \quad \lll\ (\textbf{return}\ e)\ (-a)\ c\ h$$

Similarly for other invertible functions: $\log x, \quad y - 2 \cdot x.$