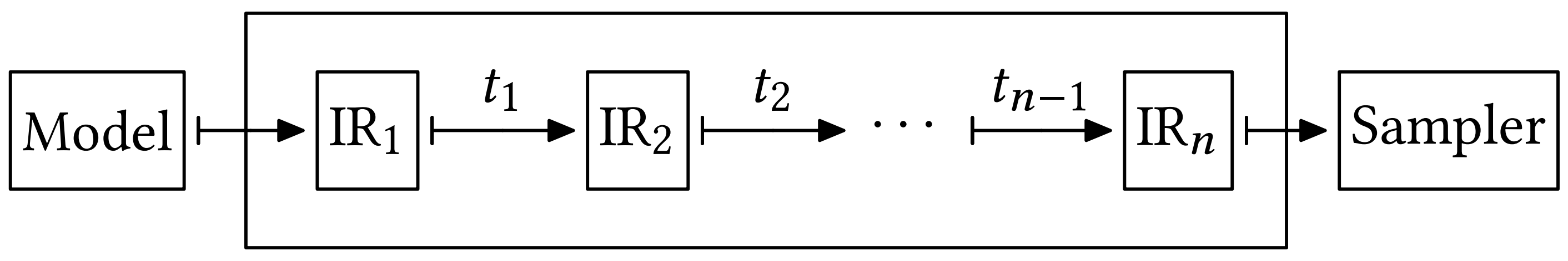
FOUNDATIONS OF PROBABILISTIC PROGRAMMING

Compositionality

composing program fragments

let v = **sample**$(\Gamma^{-1}(\alpha, 2))$ **in**
let x = **sample**$(\mathcal{N}(1.5 + m, \sqrt{v}))$ **in**
observe x **from** $\mathcal{N}(0, 0.1)$;
x

composing compiler transformations



⇒

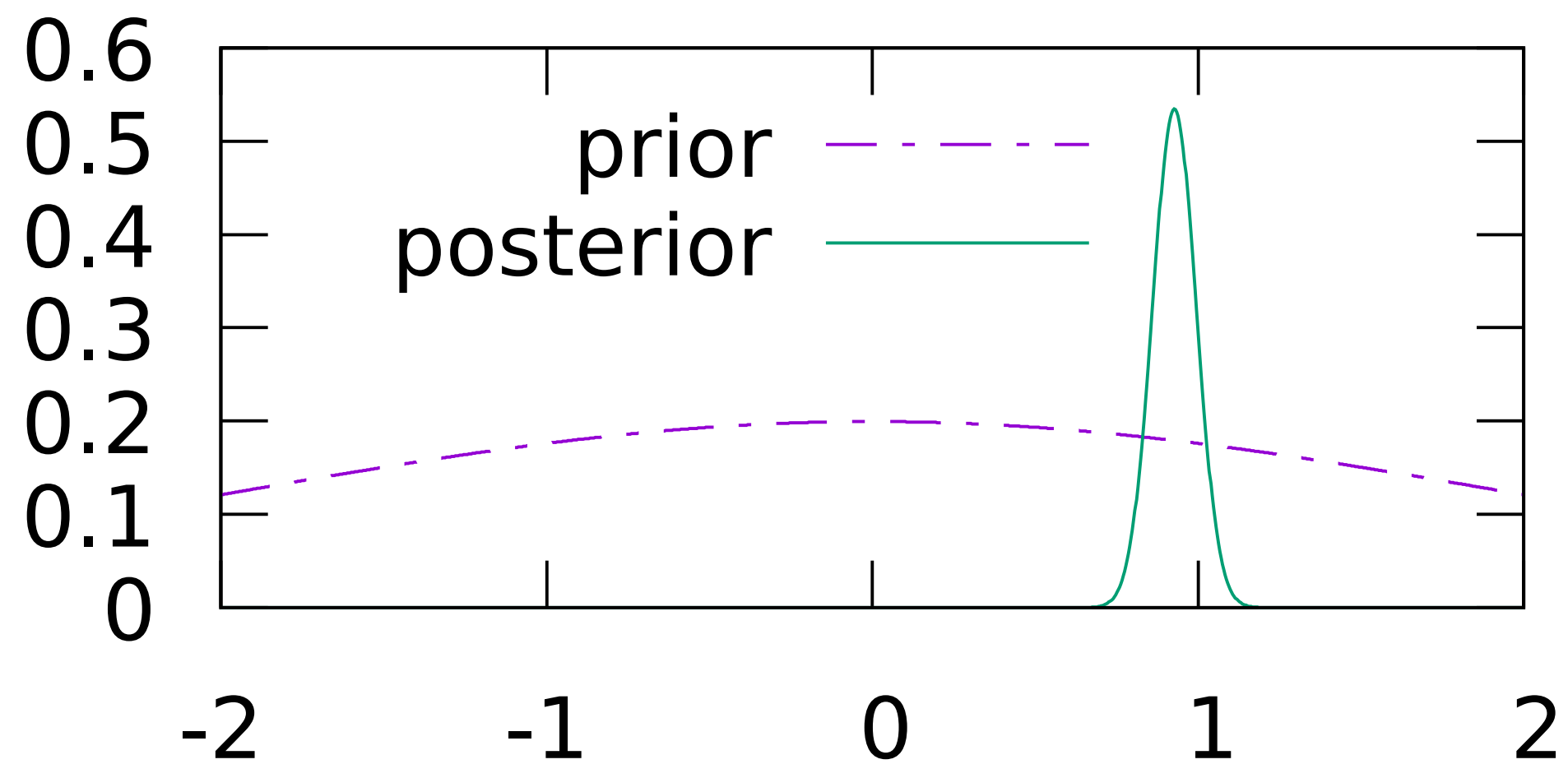
ProbProg challenges for ProgLang foundations

complexities of continuous mathematics

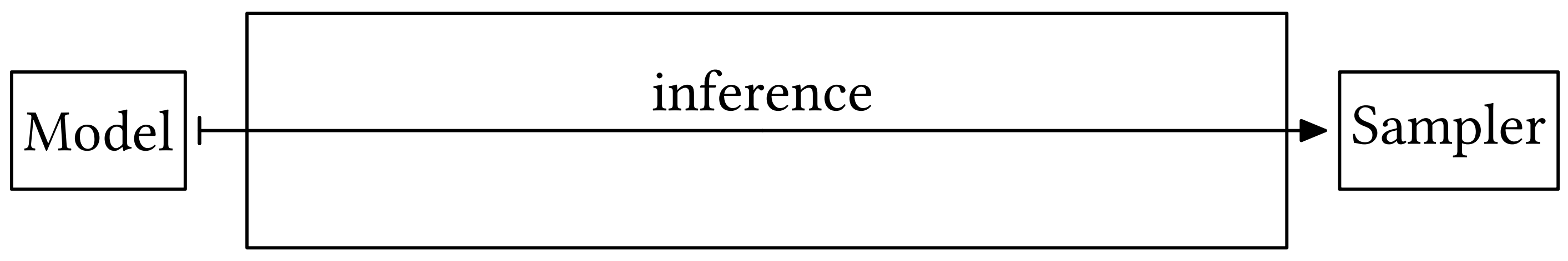
let $a = \mathit{normal\_rng}(0, 2)$ **in**

score $(\mathit{normal\_pdf}(1.1 \mid a * 1, 0.25))$;
score $(\mathit{normal\_pdf}(1.9 \mid a * 2, 0.25))$;
score $(\mathit{normal\_pdf}(2.7 \mid a * 3, 0.25))$;
a

Prior: $a \sim \mathcal{N}(0, 2)$
Observations:
$1.1 \sim \mathcal{N}(1a, \frac{1}{4})$
$1.9 \sim \mathcal{N}(2a, \frac{1}{4})$
$2.7 \sim \mathcal{N}(3a, \frac{1}{4})$



inference: global transformation



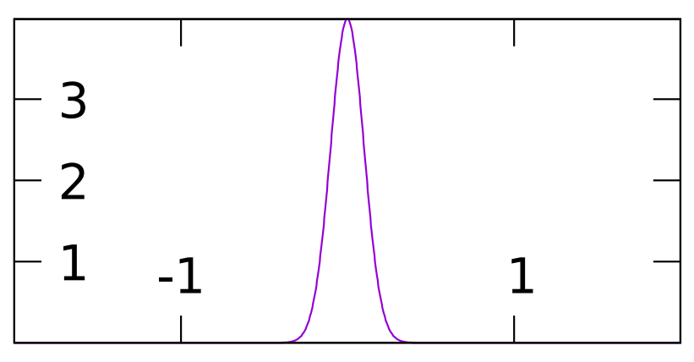
⇓

Solution: measures as invariants

code: measure-kernels/measure-valued functions

observe x **from** $\mathcal{N}(0, 0.1)$;
x

$\mathbb{R} \rightsquigarrow \mathbb{R}$



$\mathit{normal\_pdf}(x \mid 0, 0.1)$

s-finite kernels: guaranteed exchangeability[4]

let $x = M$ **in**
let $y = N$ **in**
P

=

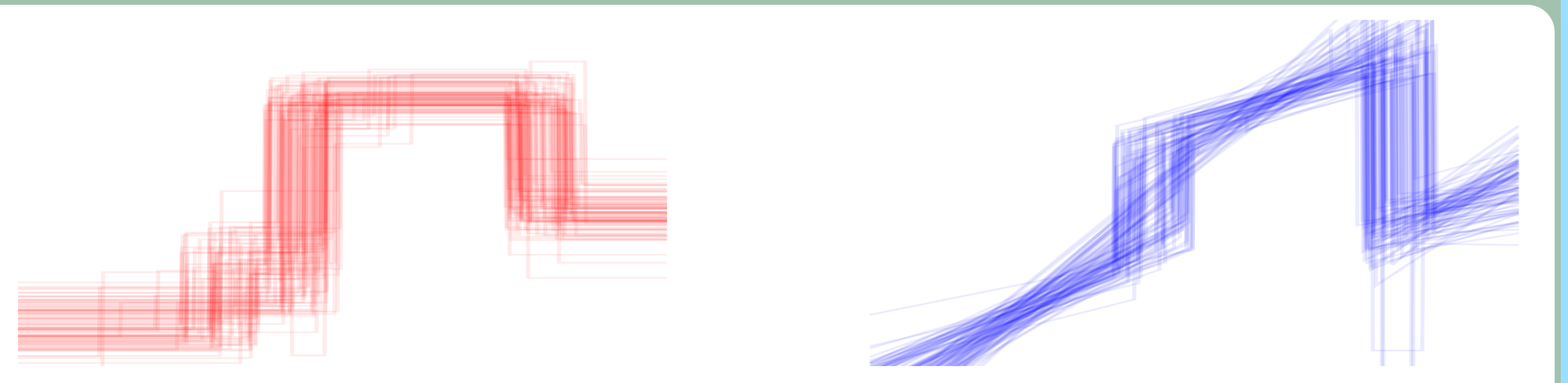
let $y = N$ **in**
let $x = M$ **in**
P

$$\int M(\mathrm{d}x) \int N(\mathrm{d}y) P(x, y) = \int N(\mathrm{d}y) \int M(\mathrm{d}x) P(x, y)$$

⇐

Expressive ProbProg

higher-order functions

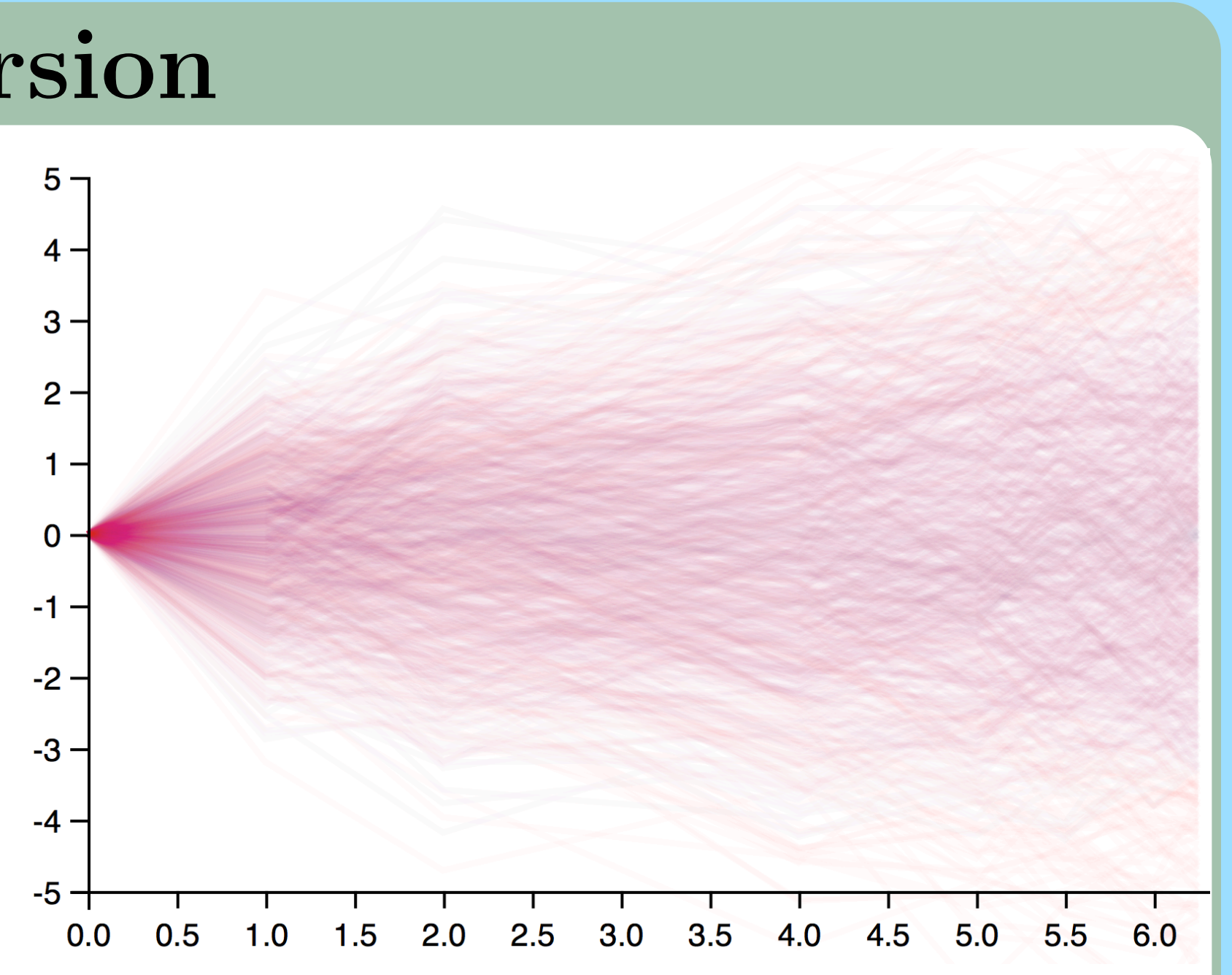


piecewise(random-constant) *piecewise(random-linear)*

generative random function models

recursion

$rw(x, \sigma) = \lambda().$ // thunk
let y = **sample**$(\mathcal{N}(x, \sigma))$
in $(x, rw\,(y, \sigma))$



Gaussian random walk

dynamic types

Church *WebPPL* Venture

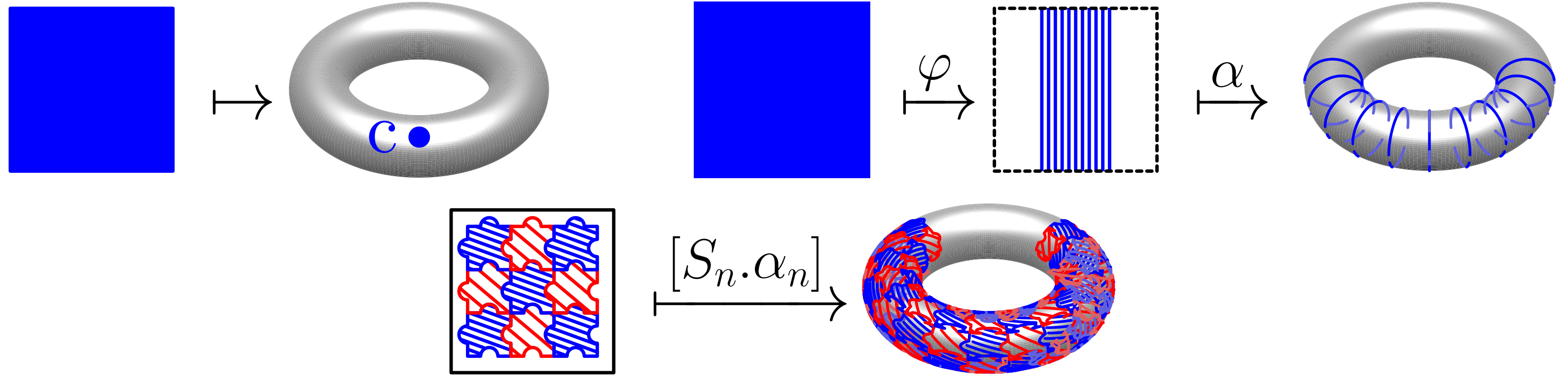
non-compositional in measure theory!

⇒

Quasi-Borel spaces: a compositional alternative[1]

set (S, M) random elements $\alpha : \mathbb{R} \to S$

qbs axioms:

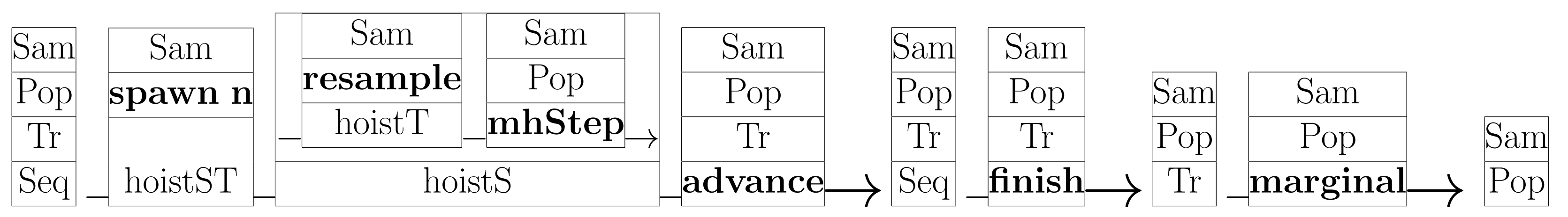


⇐

Modular inference[3, 2]

particles moves resamples

```
rmsmc k n t =
  marginal . finish . compose k (
    advance . hoistS (
      compose t mhStep . hoistT resample
    )
  ) . hoistST (spawn n >>)
```



Poster by Ohad Kammar, Sam Staton, Matthijs Vákár

Poster partly based on the following papers:

[1] C. Heunen, O. Kammar, S. Staton, and H. Yang. A convenient category for higher-order probability theory. In *LICS*, 2017.

[2] A. Ścibior, O. Kammar, and Z. Ghahramani. Functional programming for modular bayesian inference. *PACMPL*, 2(ICFP):83:1–83:29, 2018.

[3] A. Ścibior, O. Kammar, M. Vákár, S. Staton, H. Yang, Y. Cai, K. Ostermann, S.K. Moss, C. Heunen, and Z. Ghahramani. Denotational validation of higher-order bayesian inference. *PACMPL*, 2(POPL):60:1–60:29, 2017.

[4] S. Staton. Commutative semantics for probabilistic programming. In *ESOP 2017*, 2017.

LaTeX Ti*k*Z**poster**