面向概率编程的确定性流采样：语义与验证

Fredrik Dahlqvist 伦敦玛丽女王大学和伦敦大学学院 英国伦敦
f.dahlqvist@qmul.ac.uk

Alexandra Silva 计算机科学系 康奈尔大学 美国伊萨卡
alexandra.silva@cornell.edu

William Smith 计算机科学系 伦敦大学学院 英国伦敦
willi
am.smith.19@ucl.ac.uk

arXiv:2304.13504v1 [cs.PL] 26 Apr 2023

摘要——概率编程语言从根本上依赖某种采样概念；对于使用蒙特卡罗技术进行贝叶斯推断的概率编程语言而言，更是如此。验证采样器——证明其能够从正确的分布中生成样本——对于将概率编程语言用于统计建模和推断至关重要。然而，概率程序的典型指称语义与确定性采样概念不兼容。考虑到大多数统计推断都是使用伪随机数生成器完成的，这一问题尤为突出。

我们提出了一种以采样器和采样器操作为核心的高阶概率编程语言。我们以拓扑空间之间的连续映射为基础，为该语言赋予操作语义和指称语义。我们的语言还支持不连续操作，例如实数之间的比较，其实现方式是利用类型系统跟踪不连续性。此特性可能具有独立的研究价值，例如在可微编程的上下文中。

使用这种语言，我们开发了用于采样器正确性形式化验证的工具。我们提出了一种等式演算，用于推理采样器等价性；还提出了一种可靠的演算，用于证明采样器的语义正确性，即采样器通过构造能够正确地以给定测度为目标。

索引词——概率编程、操作语义与指称语义、验证

I. 引言

不具备条件化功能的概率编程语言——也就是说，能够抽取随机样本的编程语言——以及蒙特卡罗方法和随机化算法的概念，与真正的计算机一样历史悠久[1]；然而，具备条件化、高阶特性、连续变量和递归功能的语言，以及将其应用于统计建模和机器学习，则是二十一世纪的产物[36]，[26]，[15]，[27]，[40]，[8]，[4]。由于具备条件化功能的概率编程语言必须配备一系列推断算法和采样方法，而近年来这些算法和方法的引入速度不断加快，因此必须开发新的形式化方法来验证这些算法。

本研究得到 Leverhulme 项目资助“机器学习算法的验证”的支持。[1] 参见 [5]，了解概率编程语言概述；参见 [7]，了解蒙特卡洛方法的历史概述；参见 [23]，了解伪随机数生成的历史。

近年来，新的形式化方法必须被开发出来，用于验证这些算法。

我们的目标是通过引入一种具有采样器类型的语言，使推断算法的验证变得简单直接。该语言包含这些推断算法中使用的许多采样器操作，并提供一种演算，用于推理这些采样器相对于预期“目标”分布的正确性。为了能够简单且无悖论地使用确定性（伪随机）采样器，我们语言的语义完全是确定性的。

在为概率程序赋予操作语义和指称语义时，会出现一种有趣的不对称性：具有连续数据类型的一阶语言最简单且合理的指称语义，是以概率测度的形式给出的 [21]，；而最简单且合理的操作语义，则是以抽样值的形式给出的——后者更贴近那些能够抽取样本的语言程序员所采用的直觉。

概率编程语言的指称语义，以广义理解的测度为基础，已经得到了充分研究 [21]，[16]，[12]，[38]，[9]。正如 [12], 所强调的，这类语言操作语义最常见的方法是轨迹语义和马尔可夫链语义。后者被许多概率 λ-演算 [11], [22], [6], [13], [12]，[14] 以及概率编程语言 [34], [38], 选作操作语义，但它并不讨论抽样值，而只讨论执行路径上的分布。从上述不对称性的角度来看，它因此更接近指称语义。

轨迹语义最初在 [21] 中发展起来，后来应用于 [32], [6], [10], [2], ，它假设语言中的每个分布都预先生成了一个无限的样本集合。当请求一个样本时，会从该序列的头部弹出一个样本并用于计算，而序列的尾部则保留下来以供进一步采样。这一观点将采样器（例如 rand()）建模为函数，并使其对机器状态产生隐藏的副作用，这与程序员对 rand() 顺序调用本质的直觉一致。关于指称语义，自然的充分性概念是证明：在假设这些序列中的每个元素都从其对应分布中独立采样的前提下，经程序产生的推前

与程序的指称语义完全相同。我们认为这种方法存在两个问题。

首先，假设所有样本都预先计算好，与“随机”值的伪随机生成不相容，因为计算生成的样本事实上可以与真正随机样本区分开来。例如，如果（x_0、x_1、.....）是一个以分布 P 为目标、通过迭代可计算映射 $x_{n+1} = T(x_n)$ 生成的样本序列，那么如果 x 是来自 P 的“真正随机”样本，程序 $T(x) - x$ 的行为将不同于 x 由上述迭代过程生成时的行为，因此操作语义与指称语义将不再协调；对于任何伪随机数生成器，都存在类似的反例。如果（x_0、x_1、.....）是一个确定性序列，那么只有在假设该序列是Martin-Löf随机时，才能确保操作序列与指称序列之间具有有意义的一致性（该概念最早在 [29], 中定义，后来在 [17] 中推广到可计算度量空间）；遗憾的是，所有Martin-Löf随机序列都是不可计算的。事实上，仿真中使用伪随机数的频率远高于使用“真正的”物理随机性，因为伪随机数通常获取速度更快，在给定特定种子的情况下还具有可复现的优势，并且在某些假设成立时，甚至可以拥有更好的收敛性质[2]。我们认为，轨迹语义无法描述伪随机数生成，是一个重大缺陷。

其次，从轨迹语义的角度来看，采样器类型这一概念与有关副作用的问题密不可分，这使得验证颇具挑战。为了恰当地为不带括号的语法 rand 赋予作为函数的有意义语义，我们必须像 [32]，中那样对其给出单子解释，以说明它对轨迹所产生的隐含影响。输入和输出采样器的代码（即采样器操作）的正确性，取决于计算开始时轨迹的状态，而这一状态既不包含在相关采样器操作的代码中，也不包含在其所输入的采样器代码中。使用采样器操作创建复合采样器，并使其利用其他“原始”采样器，这种模式是计算统计学中的一个核心主题。尤其是，贝叶斯统计学中的推断算法——概率编程语言通过条件化编译到的对象——大量使用了这一技术。常用的采样方法，例如重要性采样、拒绝采样和粒子马尔可夫链蒙特卡罗方法，都可以自然地理解为此类复合采样器 [1], [33]。我们更倾向于采用一种无副作用的视角：在这种视角下，可以证明采样器操作的正确性，但须以关于输入这些程序的采样器的假设为前提；相较之下，另一种视角认为，其正确性还取决于运行这些操作的机器状态。

贡献。我们围绕采样器类型和采样器操作的理念开发了一种语言，使人们能够对确定性采样器和实值采样器进行推理，并旨在使对这些采样器的验证变得自然。

[2]此处的评论涉及拟蒙特卡罗技术这一相关领域，相关内容概述见 [24]。

我们在第 III 节介绍了该语言的语法、操作语义和指称语义，并给出了联系它们的适切性结果。第 IV 节提出了采样器等价性的概念，并将其应用于简化程序。最后，在第 V 节中，我们讨论了证明采样器以期望的概率测度为目标的方法（即验证采样器），并介绍了一种用于验证复合采样器正确性的可靠演算；该演算能够证明重要性采样和拒绝采样等常见蒙特卡罗技术的可靠性。

II. 示例

本节有两个目的。首先，我们通过示例展示如何变换采样器以创建新的采样器。我们借此机会非正式地介绍一种带有采样器类型构造器 Σ 以及用于构造和操作采样器的运算的语言。其次，我们介绍用于推理采样算法正确性的技术。这些技术分为两类：我们通过等式推理采样器之间的等价性（见 § IV），并从语义层面推理采样器是否完成了其设计目的——即生成服从目标分布的随机变量（见 § V）。

A. 冯·诺伊曼提取器

我们从一类简单的离散采样器开始，这类采样器称为冯·诺伊曼提取器。此例将说明采样器的自乘积这一概念，它是支持采样器类型的语言中的核心概念。冯·诺伊曼提取器 [37] 是一种简单的过程：给定从一个有偏硬币获得的结果，该硬币的结果为 {True、False}，且出现 True 的概率为 $p \in (0,1)$，它会生成来自无偏硬币的结果，出现 True 的概率为 $1/2$。我们将其视为一个布尔类型的采样器 v，记作 v : ΣB；它以另一个布尔值采样器 flip : ΣB 为输入，该采样器表示我们的有偏硬币，并据此构造一个无偏的布尔值采样器。冯·诺伊曼提取器的一种简单实现见清单 1。

```
令 choice = λb : B × B。若（第一投影(b) 且
第二投影(b)）或（非 第一投影(b) 且非第二投影
(b)）则 0 否则 1，令proj = λb : B × B。第
一投影(b)，并 映射(proj，重新加权(choice，flip²)）
```

列表 1：冯·诺伊曼提取器

该算法背后的思想是：如果从参数为 p 的伯努利分布中独立采样 (b_1, b_2)，那么结果为 (b_1, b_2) =(False, True) 和 (b_1, b_2) =(True, False) 的概率都为 $p(1-p)$，因此每一对中的第一个元素 b_1 都是无偏翻转。在清单 1 中，样本 (b_1, b_2) 中满足 $b_1 = b_2$ 的样本会被重新加权操作移除，该操作将此类样本对的权重设为零；随后，命令 map 对每一对 (b_1, b_2) 应用函数 proj，返回输出仅包含第一个元素 b_1的采样器。

$$\dfrac{\dfrac{\vdash \texttt{flip}^2 :\ \Sigma(\mathtt{B}\times\mathtt{B})\ \text{伯努利}(p)^2 \quad \vdash \texttt{choice} :\ \mathtt{B}\times\mathtt{B}\to\mathtt{R}^+}{\vdash \texttt{reweight}(\texttt{choice},\texttt{flip}^2) :\ \Sigma(\mathtt{B}\times\mathtt{B})\ [\![\texttt{choice}]\!]\cdot\text{伯努利}(p)^2} \quad \vdash \texttt{proj} :\ \mathtt{B}\times\mathtt{B}\to\mathtt{B}}{\vdash \texttt{map}(\texttt{proj},\texttt{reweight}(\texttt{choice},\texttt{flip}^2)) :\ \Sigma\mathtt{B} \rightsquigarrow [\![\texttt{proj}]\!]_*\,([\![\texttt{choice}]\!]\cdot\text{伯努利}(p)^2) = \text{伯努利}(1/2)}$$

图 1：冯·诺伊曼提取器的有效性

$$\dfrac{\dfrac{\vdash \texttt{rand}^2 :\ \Sigma(\mathtt{R}\times\mathtt{R})\ U^2 \quad \vdash \texttt{plus} :\ \mathtt{R}\times\mathtt{R}\to\mathtt{R}}{\vdash \texttt{map}(\texttt{plus},\texttt{rand}^2) :\ \Sigma\mathtt{R} \rightsquigarrow [\![\texttt{plus}]\!]_*\,U^2} \quad \vdash \texttt{phi} :\ \mathtt{R}\to\mathtt{R}^+}{\vdash \texttt{reweight}(\texttt{phi},\texttt{map}(\texttt{plus},\texttt{rand}^2)) :\ \Sigma\mathtt{R} \rightsquigarrow [\![\texttt{phi}]\!]\cdot([\![\texttt{plus}]\!]_*\,U^2) = P}$$

图 2：清单 2 中重要性采样的有效性

请注意 flip² 在冯·诺伊曼提取器中的出现；这是采样器 flip 的“自乘积”。其中，flip : ΣB 生成取布尔值的样本，而 flip² : Σ(B × B) 生成由两个布尔值组成的样本对。给定一个类型为 T 的采样器 t : ΣT，自乘积 t^2 : Σ(T×T) 是一个采样器，其元素是来自 T 的相邻样本；同样的构造（详见第 III 节）可以轻松扩展到任意自幂 t^K : Σ(T^K)。

我们语言的主要应用是为其采样器的形式化验证提供一个环境。设 v : ΣB 为清单 1 中定义的冯·诺伊曼提取器；验证 v 的任务就是证明 v “以.....为目标”，即目标分布为均匀分布 Ber (1/2)——非正式地说，这意味着随着样本量趋于极限，v 生成无偏翻转。在第 V-B 节中，我们定义了采样器与分布之间的一种关系，将这一概念具体化：我们将 ⊢ v : ΣB Ber (1/2) 读作“采样器 v 以测度 Ber (1/2) 为目标”。前述自乘积操作在采样器验证中发挥着关键作用，因为为了得出 v 以 Ber (1/2) 为目标这一结论，仅仅假设 flip 以 Ber(p) 为目标（其中 $p \in (0,1)$）是不够的。相反，我们必须作出更强的假设，即 flip² 以 Ber(p)²为目标：本质上，这意味着从 flip 得到的相邻样本表现得如同彼此独立。沿用伪随机数生成领域的术语，我们将这一性质称为 K-均匀分布（在本例中为 $K=2$）：我们将称采样器 s 相对于分布 P 是 K-均匀分布的，若 s^K 以 P^K 为目标。例如，断言一个取值于 $\{0,\dots,N-1\}$ 的伪随机数生成器相对于均匀分布是 K-均匀分布的，就是断言所有长度为 K的词 $w \in \{0,\dots,N-1\}^K$ 均以相同比例生成。若干常用的离散伪随机数生成器，例如 xorshift 和梅森旋转算法，都具有众所周知的K-均匀分布保证；参见 [39], [30]。

在我们的渐近目标计算中，冯·诺伊曼提取器的有效性如图 1 所示。在开始这一推导之前，必须证明，在假设可以访问一个布尔值类型的采样器 flip 的上下文中，冯·诺伊曼提取器 v 等价于简化采样器映射(proj, 重新加权(choice, flip²))，其中 proj 和 choice 的定义如清单 1 所示。我们将这种等价性写作 flip : ΣB ⊢ v ≈ 映射(proj, 重新加权(choice, flip²)) : ΣB；这一等价关系将在第 IV 节中讨论，并将在这一具体案例中使用表 IV 的 let 绑定规则予以证明。

以这种方式重写 v，并利用 2-等分布假设 ⊢ flip² : ΣB Ber(p)², ，我们通过应用 § V-B 中与重新加权和映射这两种采样器操作相对应的规则，得出图 1 中的结论。这些规则表明，v 的目标测度为 $[\![\texttt{proj}]\!]_*\,([\![\texttt{choice}]\!]\cdot\text{Ber}(p)^2)$。（如我们将在 § V-B 中讨论的，$[\![f]\!]_*\,\mu$ 表示测度 μ 经函数 f推前所得的测度，符号 $f\cdot\mu$ 表示经密度 f重新加权后的测度 μ 。）为完成证明，我们证明该测度与 Ber(1/2) 相同，也就是 B 上的均匀测度；这很容易做到。首先很容易看出，$[\![\texttt{choice}]\!]\cdot\text{Ber}(p)^2$给样本 (True, False) 和 (False, True) 赋予概率 1/2 ，并给所有其他样本赋予零概率；接着只需注意到函数 proj 会丢弃第二个样本，即可得到所需结果。

B. 重要性采样

重新加权操作的一个核心应用是其在重要性采样中的作用。这是一种在贝叶斯学习和统计推断中常用的技术 [33], [7] ，它将潜空间 X 上的“提议”分布 Q 中的样本转换为来自“目标”分布 P的近似样本，其中 P 相对于 Q 绝对连续，其拉东-尼科迪姆导数为 $\frac{dP}{dQ}(x)$。其操作非常直接：对于每个样本 $x_n \sim Q$，计算该样本的权重 $w_n = \frac{dP}{dQ}(x)$，然后将加权样本(x_n, w_n)非正式地理解为来自目标分布 P的近似样本。形式上，归一化经验测度$\sum_{n=1}^{N} \frac{w_n}{\sum_{i=1}^{N} w_i}\delta_{x_n}$（其中 δ_x是位于 $x \in X$处的狄拉克测度）在 $N\to\infty$时弱收敛到目标测度 P。

例如，考虑如下贝叶斯推断问题：先验分布 P_0 是定义在 $[0,2]$ 上的三角分布，而给定潜在值 x 时，观测值 $y=3$ 的似然是标准高斯分布 $L(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(3-x)^2}{2}\right)$。设 P表示相应的后验分布，其密度正比于三角分布密度与高斯密度的逐点乘积；在我们的语言中，从 P 采样的一个简单重要性采样过程如代码清单 2 所示。这里，我们假设可以使用一个采样器 rand，它以 $[0,1]$上的均匀分布为目标；回想一下，三角分布随机变量是两个独立均匀分布随机变量之和，并假设 rand 具有 2-等分布所需的独立性性质，我们

$$\frac{\vdash \texttt{tri} \otimes \texttt{rand} : \ \Sigma \texttt{T} \rightsquigarrow \text{三元组} \ \otimes U \qquad \vdash \texttt{accept} : \ \texttt{T} \to \texttt{R}^+}{\vdash \texttt{reweight(accept, tri} \otimes \texttt{rand)} : \ \Sigma \texttt{T} \rightsquigarrow [\![\texttt{accept}]\!] \cdot (\text{三元组} \ \otimes U) \quad \vdash \texttt{proj} : \ \texttt{T} \to \texttt{R}}$$
$$\vdash \texttt{map(proj, reweight(accept, tri} \otimes \texttt{rand))} : \ \Sigma \texttt{R} \rightsquigarrow [\![\texttt{proj}]\!]_* ([\![\texttt{accept}]\!] \cdot (\text{三元组} \ \otimes U)) = P$$

图 3：列表 3 中拒绝采样的有效性

将 rand 的两次抽样相加，生成一个三角分布随机变量。最后，我们根据似然 $L(x)$ 对结果重新加权，从而得到一个以观测数据 $y = 3$对应的后验分布 P 为目标的采样器。

```
令 phi = λx : R . 1/sqrt(2*pi)
*exp(-1/2*(3-x)*(3-x))其中令 加 = λu : R × R。 第
一投影(u) + 第二投影(u)其中 重新加权(phi, 映射(加, 随机²)
```

列表 2：重要性采样

该采样器的有效性——即它指向正确后验分布这一事实——可以在我们的目标演算中轻易得到。假设 rand 生成关于均匀分布 U 的 2-等分布样本，图 2 中的推导证明，列表 2 所定义的采样器指向测度 $[\![\texttt{phi}]\!] \cdot ([\![\texttt{plus}]\!]_* U^2)$。接下来需要证明该测度就是所需的 P；一旦证明两个独立均匀随机变量之和服从三角分布，该结论便可根据重新加权操作 $\cdot$的定义得到。对于任意观测值 y，同样的论证也足够。

C. 拒绝采样

我们关于采样器验证的最后一个例子，是一种称为拒绝采样的技术实例。列表 3 从先验分布应用拒绝采样于 § II-B 中讨论的同一贝叶斯推断问题，从而得到一个指向相同后验分布 P 的采样器。尤其重要的是接受函数-拒绝函数步骤的不连续性；在存在伪随机数生成的情况下，这显著增加了采样器验证论证的复杂性。

```
let phi = λx : R . 1/sqrt(2*pi)
*exp(-1/2*(3-x)*(3-x))in let 接受函数 = λ(u,v) :
T .if v ≤ phi(u)*sqrt(2*pi) then1 else 0in
let proj = λz: T. 第一投影(cast⟨R×R⟩(z)) in映射
(proj, 重新加权(接受函数, tri ⊗ rand))
```

列表 3：拒绝采样

为了证明从先验分布进行拒绝采样的有效性，我们必须假设能够访问先验分布上的采样器（此处为 tri）、独立的标准均匀随机采样器（此处为 rand），以及似然的上界 $\sup_{x \in \mathbb{R}} f($ $\frac{1}{2\pi} \quad x \in \mathbb{R} \quad -\frac{(3-x)^2}{2} = \frac{1}{2\pi}$) sup exp( )，该上界用于接受条件。图 3 证明，在对采样器 tri 和 rand 作出自然的独立性假设后，由列表 3 定义的拒绝采样器以测度 $[\![\texttt{proj}]\!]_* ([\![\texttt{accept}]\!] \cdot (\text{Tri} \otimes U))$ 为目标。随后，我们可以使用标准方法证明，该测度与 P 相同；列表 2 的目标同样是这个后验分布。

目前我们省略了证明中的一个关键部分。请注意，在列表 3 和图 3 中，accept 函数的类型看起来应为 $\mathrm{R} \times \mathrm{R} \to \mathrm{R}^+$，但实际上却是 $\mathrm{T} \to \mathrm{R}^+$。相应地，乘积 tri $\otimes$ rand 必须假定生成类型为 T 的样本，而非 $\mathrm{R} \times \mathrm{R}$；proj 必须接受类型为 T 的输入，而非 $\mathrm{R} \times \mathrm{R}$ 对。类型 T 是 $\mathrm{R} \times \mathrm{R}$ 的一个子类型，其性质将在第 III 节中说明；它编码了这样一个事实：当把 accept 视为标准拓扑上的函数时，它是不连续的，同时也编码了这些不连续性允许出现的位置。列表 3 的类型推断（详细说明了 T 的结构）见附录中的图 6 和图 7。

III. 语言

A. 语法

我们使用一种带有子类型概念和采样器类型构造器的 λ-演算。

1) 类型：类型由图 4a 中基本遵循标准的文法生成，其中基类型集合 Ground 为

$$\{\mathtt{N}, \mathtt{R}, \mathtt{R}^+\} \cup \{f^{-1}(i) \mid f \in \{\leq, <, \geq, >, =, \neq\}, i = 0, 1\}.$$

我们的基类型包括自然数、实数和非负实数，以及一些重要的实数对集合：例如， $<^{-1}(1)$ 将按照该记号所暗示的方式，表示第一分量严格小于第二分量的实数对集合。布尔类型 B $1 + 1$ 将被视为一种基类型。

唯一不寻常的类型构造器是拉回类型 $\mathrm{T}_{s \ t}$ ，顾名思义，它们将被解释为拉回（实际上是逆像）；以及采样器类型 ΣT，它们将被定义为由（语法上的）函子 $\mathrm{T} \times \mathrm{R}$ $^+ \times$ –定义的余归纳（流）类型。换句话说，我们假设采样器可以带权；这涵盖了无权采样器这一特殊情况，其中每个权重都设为 1。由于这些是我们所需的唯一余归纳类型，并且为了凸显采样器所发挥的核心作用，我们选择不将通用余归纳类型添加到语言中。

类型上的子类型关系 $\lhd$ 是由图 4b 中的规则生成的关系的自反传递闭包。

2）项：图 4c 展示了生成我们语言中项的集合 Expr 的文法。我们假设存在一个内置函数集合 Func，其中的函数都带有类型信息 $f : \mathrm{T} \to \mathrm{G}$，G 为基类型。一些内置函数相对于通常拓扑是连续的，例如加法运算 $+ : \mathrm{R} \times \mathrm{R} \to \mathrm{R}$，但另一些则是不连续的，例如比较运算符 $\{\leq, <, \geq, >, =, \neq\} : \mathrm{R} \times \mathrm{R} \to \mathrm{B}$。正如我们将在第 III-C 节中讨论的那样，处理这类函数是将余积加入文法的主要原因。我们还采用语法糖

$$\texttt{if}\ b\ \texttt{then}\ s_{\mathtt{True}}\ \texttt{else}\ s_{\mathtt{False}} \triangleq \texttt{case}\ (b, _)\ \texttt{of}\ \{(i, _) \Rightarrow s_i\}_{i \in \mathbb{B}}.$$

$$\mathtt{S},\mathtt{T} ::= \mathtt{G} \in \text{Ground} \mid 1 \mid \mathtt{S} \times \mathtt{T} \mid \mathtt{S} + \mathtt{T} \mid {}_{s}\mathtt{T}_{t} \mid \mathtt{S} \to \mathtt{T} \mid \Sigma\,\mathtt{T} \qquad s,t : \mathtt{T}$$

(a) Type grammar

$$\frac{}{f^{-1}(0) + f^{-1}(1) \lhd \mathtt{R} \times \mathtt{R}}\ f \in \{<,\leq,>,\geq,=,\neq\} \qquad \frac{\mathtt{S}_1 \lhd \mathtt{S}_2 \qquad \mathtt{T}_1 \lhd \mathtt{T}_2}{\mathtt{S}_1 \times \mathtt{T}_1 \lhd \mathtt{S}_2 \times \mathtt{T}_2} \qquad \frac{\mathtt{S}_1 \lhd \mathtt{S}_2 \qquad \mathtt{T}_1 \lhd \mathtt{T}_2}{\mathtt{S}_1 + \mathtt{T}_1 \lhd \mathtt{S} + \mathtt{T}_2}$$

$$\frac{\mathtt{S} \lhd \mathtt{T}}{\Sigma\,\mathtt{S} \lhd \Sigma\,\mathtt{T}} \qquad \frac{\sum_{i\in n} \mathtt{S}_i \lhd \mathtt{S} \quad \sum_{j\in m} \mathtt{S}'_i \lhd \mathtt{S}}{\sum_{i\in n, j\in m} \mathtt{S}_i \cap \mathtt{S}'_j \lhd \sum_{i\in n} \mathtt{S}_i} \qquad \frac{\sum_{i\in n} \mathtt{S}_i \lhd \mathtt{S} \quad \sum_{j\in m} \mathtt{S}'_j \lhd \mathtt{S}}{\sum_{i\in n, j\in m} \mathtt{S}_i \cap \mathtt{S}'_j \lhd \sum_{j\in m} \mathtt{S}_j}$$

(b) Subtyping rules

$$\begin{array}{lll}
t ::= & x \in \text{Var} \mid b \in \{\mathtt{True},\mathtt{False}\} \mid n \in \mathbb{N} \mid r \in \mathbb{R} \mid & \textit{Variables and constants} \\
& f(t,\ldots,t), f \in \text{Func} \mid \mathtt{cast}\langle \mathtt{T}\rangle t \mid & \textit{Built-in functions} \\
& \mathtt{case}\ t\ \mathtt{of}\ \{(i,x_i) \Rightarrow s_i\}_{i\in n} \mid \mathtt{in}_i\,(t) \mid \lambda x\colon \mathtt{T}.t \mid t(t) \mid \mathtt{let}\ x = t\ \mathtt{in}\ t \mid & \textit{Programming constructs} \\
& (t,t) \mid \mathtt{fst}(t) \mid \mathtt{snd}(t) \mid & \textit{Products} \\
& \mathtt{prng}(t,t) \mid t \otimes t \mid \mathtt{map}(t,t) \mid \mathtt{reweight}(t,t) \mid \mathtt{hd}(t) \mid \mathtt{wt}(t) \mid \mathtt{tl}(t) \mid \mathtt{thin}(t,t) & \textit{Sampler operations}
\end{array}$$

(c) Term grammar

图 4：文法和子类型规则

我们的大多数语言构造对于一种不含递归的类型化函数式语言而言都是标准的，但我们为语言赋予了若干非标准的（采样器）操作：

- 操作 prng(f,t) 用于将伪随机数生成器构造为采样器，使用初始值 t 和确定性自映射 f。
- $s \otimes t$ 表示采样器 s和 t的乘积。
- 语法 map(f,t) 将函数 f 映射到采样器 t 生成的元素上，从而生成新的采样器，这类似于测度的推前。
- 操作 reweight(f,t) 将重加权方案 f 应用于采样器 t，以形成新的采样器。
- 给定采样器 t，操作 hd(t) 返回 t生成的第一个样本，wt(t) 返回 t生成的第一个样本的权重，而 tl(t) 返回采样器 t，但会丢弃其第一个样本-权重对。
- 操作 thin(n,t) 接受一个自然数 n 和一个采样器 t，返回一个仅包含 t 中索引为 n 倍数的元素的采样器。

这些语言结构中的大多数，其直观含义和用途已在 § II 中说明；引入其语义后，我们将阐明它们的确切含义。

3) 良构项：我们的类型系统基本上是标准的，见表 I。唯一非标准的规则是表 I 第二行的上下文限制规则，以及采样器操作的类型规则；根据上文对这些操作的描述，这些规则应该很容易理解。简而言之，上下文限制规则的作用是：将一个类型为 T 的计算结果传递给另一个计算。前者关于类型 T 的指称上的某个拓扑 τ 是连续的；后者使用一个类型为 T 的变量，但关于类型 T 的指称上的一个更细拓扑 $\tau' \supset \tau$ 是连续的。应用此规则后，便无法再对上下文中的各个变量进行 λ-抽象。此特性有充分的语义原因，我们将在第 III-C 节讨论。为便于阅读和直观理解，该规则采用以下语法糖表示

$$t^{-1}(\mathtt{T}_i) \triangleq {}_{\mathtt{cast}\langle \mathtt{T}\rangle \mathtt{in}_i(x)}\mathtt{T}_t \qquad \text{其中 } x : \mathtt{T}_i \tag{1}$$

对于图 4b 中的子类型规则，我们使用以下语法糖

$$\mathtt{S}_i \cap \mathtt{S}'_j \triangleq {}_{\mathtt{cast}\langle \mathtt{S}\rangle \mathtt{in}_i(x_i)}\mathtt{S}_{\mathtt{cast}\langle \mathtt{S}\rangle \mathtt{in}_j(x'_j)} \quad \text{where } x_i : \mathtt{S}_i, x'_j : \mathtt{S}'_j$$

我们的类型化 lambda 演算不包含递归，原因有二。首先，递归并非必需：任何可计算概率测度都可以表示为单位区间 [18], [17]，上均匀测度的可计算推前，因此，任何具有采样器操作 map 的采样器语言，只要给定一个均匀采样器，就能以任意可计算概率测度为目标。特别是，许多通常以递归方式实现的拒绝采样器，也可以改用重新加权操作实现，如列表 3 所示。其次，带递归的类型化概率高阶 λ 演算的范畴语义是一个非常新的研究领域 [38]；我们将递归采样器的纳入留待后续工作。

B. 操作语义

在实践中，为了计算包含采样器的程序，必须指定要生成的有限数量的样本 $N \in \mathbb{N}$。因此，我们的（大步）操作语义采取归约关系 $(t,N) \to v$的形式，其中左侧由一个类型正确的闭项 $t \in$ Expr 和一定数量的样本 $N \in \mathbb{N}$组成，右侧则是一个值 $v \in$ Value，即由以下文法生成的项

$$v ::= x \in \text{Var} \mid g \in \mathtt{G} \mid (v,v) \mid \mathtt{in}_i\,(v) \mid \lambda x\colon \mathtt{T}.\ v \tag{2}$$

该大步操作语义的规则完整列于附录表 VI 中，包括标准语言构造的通常规则，以及针对我们所实现的采样器操作的附加规则；后者列于表 II 中。为简化记号，这些操作使用列表（a、b、c、d），但在我们的语言中实际上将其解释为嵌套对（a，（b，（c、d））），为了使

$$\frac{}{\Gamma \vdash g : \mathtt{G}}\ g \in [\![\mathtt{G}]\!] \qquad \frac{}{\Gamma, x : \mathtt{T}, \Delta \vdash x : \mathtt{T}} \qquad \frac{\Gamma \vdash t : \mathtt{T}}{\Gamma \vdash f(t) : \mathtt{G}}\ \mathrm{Func} \ni f : \mathtt{T} \to \mathtt{G} \qquad \frac{\Delta \vdash t : \mathtt{S}}{\Gamma \vdash \mathtt{cast}\langle \mathtt{T} \rangle t : \mathtt{T}}\ \mathtt{S} \lhd \mathtt{T}, \Gamma \lhd \Delta$$

$$\frac{\Gamma \vdash t : \mathtt{T},}{(x_1, \ldots, x_n) : \sum_{i \in m} t^{-1}(\mathtt{T}_i) \vdash t : \sum_{i \in m} \mathtt{T}_i}\ \sum_{i \in m} \mathtt{T}_i \lhd \mathtt{T}, \Gamma = x_1 : \mathtt{S}_1, \ldots, x_n : \mathtt{S}_n$$

$$\frac{\Gamma \vdash s : \mathtt{S} \quad \Gamma \vdash t : \mathtt{T}}{\Gamma \vdash (s,t) : \mathtt{S} \times \mathtt{T}} \qquad \frac{\Gamma \vdash t : \mathtt{S} \times \mathtt{T}}{\Gamma \vdash \mathtt{fst}(t) : \mathtt{S}} \qquad \frac{\Gamma \vdash t : \mathtt{S} \times \mathtt{T}}{\Gamma \vdash \mathtt{snd}(t) : \mathtt{T}} \qquad \frac{\Gamma, \mathtt{x} : \mathtt{S} \vdash t : \mathtt{T} \quad \Gamma \vdash s : \mathtt{S}}{\Gamma \vdash \mathtt{let}\ x = s\ \mathtt{in}\ t : \mathtt{T}}$$

$$\frac{\Gamma, x : \mathtt{S} \vdash t : \mathtt{T}}{\Gamma \vdash \lambda x : \mathtt{S}\ .\ t : \mathtt{S} \to \mathtt{T}} \qquad \frac{\Gamma \vdash s : \mathtt{S} \quad \Gamma \vdash t : \mathtt{S} \to \mathtt{T}}{\Gamma \vdash t(s) : \mathtt{T}} \qquad \frac{\Gamma \vdash t : \mathtt{T}_j}{\Gamma \vdash \mathtt{in}_j\,(t) : \sum_{i \in n} \mathtt{T}_i}\ j \in n \qquad \frac{\Gamma \vdash t : \sum_{i \in I} \mathtt{T}_i \quad \Gamma, x_i : \mathtt{T}_i \vdash s_i : \mathtt{T}}{\Gamma \vdash \mathtt{case}\ t\ \mathtt{of}\ \{(i, x_i) \Rightarrow s_i\}_{i \in I} : \mathtt{T}}$$

$$\frac{\Gamma \vdash t : \Sigma \mathtt{T}}{\Gamma \vdash \mathtt{hd}(t) : \mathtt{T}} \qquad \frac{\Gamma \vdash t : \Sigma \mathtt{T}}{\Gamma \vdash \mathtt{wt}(t) : \mathtt{R}^+} \qquad \frac{\Gamma \vdash t : \Sigma \mathtt{T}}{\Gamma \vdash \mathtt{tl}(t) : \Sigma \mathtt{T}} \qquad \frac{\Gamma \vdash s : \Sigma \mathtt{S} \quad \Gamma \vdash t : \Sigma \mathtt{T}}{\Gamma \vdash s \otimes t : \Sigma(\mathtt{S} \times \mathtt{T})}$$

$$\frac{\Gamma \vdash s : \mathtt{T} \to \mathtt{T} \quad \Gamma \vdash t : \mathtt{T}}{\Gamma \vdash \mathtt{prng}(s,t) : \Sigma \mathtt{T}} \qquad \frac{\Gamma \vdash t : \Sigma \mathtt{T} \quad \Gamma \vdash n : \mathtt{N}}{\Gamma \vdash \mathtt{thin}(t,n) : \Sigma \mathtt{T}} \qquad \frac{\Gamma \vdash s : \Sigma \mathtt{S} \quad \Gamma \vdash t : \mathtt{S} \to \mathtt{T}}{\Gamma \vdash \mathtt{map}(t,s) : \Sigma \mathtt{T}} \qquad \frac{\Gamma \vdash s : \mathtt{T} \to \mathtt{R}^+ \quad \Gamma \vdash t : \Sigma \mathtt{T}}{\Gamma \vdash \mathtt{reweight}(s,t) : \Sigma \mathtt{T}}$$

表 I：类型规则

规则易于阅读，我们还引入简写（t、 N） $\to$（v_1、 w_1）...
...、（v_N、 w_N））来表示 N归约

$$(\mathtt{hd}(t), \mathtt{wt}(t)) \to (v_1, w_1),$$
$$(\mathtt{hd}(\mathtt{tl}(t)),\ \mathtt{wt}(\mathtt{tl}(t))) \to (v_2, w_2), \ldots,$$
$$(\mathtt{hd}(\mathtt{tl}^{N-1}(t)),\ \mathtt{wt}(\mathtt{tl}^{N-1}(t))) \to (v_N, w_N).$$

请注意，两个加权采样器的乘积，其权重是各因子权重的乘积。乘积和重新加权操作是仅有的会修改采样器权重的操作。

下面的命题表明，操作语义是良构的：对于任意 $N \in \mathbb{N}$，采样器只能规约为长度为 N 的加权列表。

命题 III.1. *[附录 A]* 若 $\vdash s : \Sigma \mathtt{S}$ 是一个闭采样器，则对于任意 $N \in \mathbb{N}$，若 $(s, N) \to v$，那么 v具有如下形式：$((v_1, w_1) \ldots, (v_N, w_N))$，其中 v_n 是值，$w_n \in \mathbb{R}_{\geq 0}$ 是权重。若 S 不是采样器类型，则 $v_n : \mathtt{S}$；更一般地，每个 v_n 本身也可能是一个加权列表。

自乘积操作：阐明乘积和 thin 操作的含义后，我们现在可以正式说明这一操作的合理性；在 § II 中，我们将其称为采样器的“自乘积”。为了说明其动机，考虑一个采样器 $t : \Sigma \mathtt{T}$，它的求值结果为 $(\mathtt{t}\ 2N) \to (x_1, \ldots, x_{2N})$，为简洁起见，我们省略了权重。根据上述操作语义，滞后采样器 $\mathtt{thin}(2, \mathtt{t} \otimes \mathtt{tl}(\mathtt{t})) : \Sigma(\mathtt{T} \times \mathtt{T})$ 的求值结果为

$$(\mathtt{thin}(2, \mathtt{t} \otimes \mathtt{tl}(\mathtt{t})), N) \to ((x_1, x_2), (x_3, x_4), \ldots, (x_{2N-1}, x_{2N})).$$

这就是 § II 中记作 $\mathtt{t}^2$ 的“自乘积”。这个概念很重要，因为它是一种构造，能够让我们从给定的采样器生成一对独立样本。请注意，简单地取 $t \otimes t$ 会生成完全相关的样本对：操作语义给出 $(t \otimes t, N) \to ((x_1, x_1) \ldots (x_N, x_N))$。更一般地，对于任意$K \in \mathbb{N}$，我们将采样器的 K 重自乘积定义为

$$t^K \triangleq \mathtt{thin}(K, t \otimes \mathtt{tl}(t) \otimes \ldots \otimes \mathtt{tl}^{K-1}(t)). \tag{3}$$

从 t^K 中采样旨在允许对由采样器 K 生成的独立 K-元组进行采样。

归根结底，包含采样器操作 thin 的唯一目的就是定义这一自乘积操作，因为它在某种程度上是一个不自然的构造。

C. 指称语义

1) 指称宇宙：我们将在 § V 中看到，连续映射在采样器性质的验证中发挥着特殊作用。因此，我们需要一个能够清晰定义连续性概念的指称域。我们还需要一个笛卡尔闭模型，因为我们希望对演算中的 λ 抽象操作进行解释。一种标准做法是考虑紧生成拓扑空间范畴 [35]–[31], [25]（以下简称 CG 空间）。若拓扑空间 X 满足以下条件，则称其为紧生成空间：它是豪斯多夫的，并且对于每个属于 X [35, §1] 的紧 K，$C \subseteq X$ 闭当且仅当 $C \cap K$ 在 K 中闭。我们无需关心这些空间的理论，但以下事实对于后文至关重要。

命题 III.2（[35], [25]）。1）CG 范畴由 CG 空间和连续函数构成，是笛卡尔闭的。2）CG 范畴是完备且余完备的。3）每个可度量化拓扑空间都是 CG 空间。4）CG 空间的局部闭子集（即一个开子集与一个闭子集的交）是紧生成的。

下面简要介绍 CG 的笛卡尔闭结构。乘积通常不同于 Top（拓扑空间范畴）中的乘积：如果通常的积拓扑还不是紧生成的，就需要对其进行修改，以确保紧生成性 [35, §4]。不过，在大多数实际情况下，通常的积拓扑已经是紧生成的——例如，任意可数个可度量化空间的乘积都是可度量化的，因此根据命题 III.2，它是紧生成的。CG 空间 XY 之间的内部同态 $[X, Y]$ 由连续映射 $X \to Y$的集合以及紧集上一致收敛的拓扑构成，这一拓扑也称为紧开拓扑 [35, §5]。

2) 类型语义：在建立这一范畴模型后，我们定义类型的语义。基类型的语义符合预期：$[\![\mathtt{N}]\!] = \mathbb{N}$，赋予离散拓扑，以及 $[\![\mathtt{R}]\!] = \mathbb{R}, [\![\mathtt{R}^+]\!] = [0, \infty)$，赋予通常

$$\frac{((s(\mathtt{hd}(t)),\mathtt{wt}(t)),N)\to(v_1,w_1)\quad\dots\quad((s(\mathtt{hd}(\mathtt{tl}^{N-1}(t)),\mathtt{wt}(\mathtt{tl}^{N-1}(t))),N)\to(v_N,w_N)}{(\mathtt{map}(s,t),N)\to((v_1,w_1),\dots,(v_N,w_N))}$$

$$\frac{((\mathtt{hd}(t),s(\mathtt{hd}(t))\cdot\mathtt{wt}(t)),N)\to(v_1,w_1)\quad\dots\quad((\mathtt{hd}(\mathtt{tl}^{N-1}(t)),s(\mathtt{hd}(\mathtt{tl}^{N-1}(t)))\cdot\mathtt{wt}(\mathtt{tl}^{N-1}(t))),N)\to(v_N,w_N)}{(\mathtt{reweight}(s,t),N)\to((v_1,w_1),\dots,(v_N,w_N))}$$

$$\frac{(s,N)\to((v_1,w_1),\dots,(v_N,w_N))\quad(t,N)\to((v'_1,w'_1),\dots,(v'_N,w'_N))}{(s\otimes t,N)\to(((v_1,v'_1),w_1\cdot w'_1),\dots,((v_N,v'_N),w_N\cdot w'_N))}$$

$$\frac{(t,N)\to((v_1,w_1),\dots,(v_N,w_N))}{(\mathtt{hd}(t),N)\to v_1}\qquad\frac{(t,N)\to((v_1,w_1),\dots,(v_N,w_N))}{(\mathtt{tl}(t),N-1)\to((v_2,w_2),\dots,(v_N,w_N))}\qquad\frac{(t,N)\to((v_1,w_1),\dots,(v_N,w_N))}{(\mathtt{wt}(t),N)\to w_1}$$

$$\frac{(s,N)\to i\quad(t,Ni)\to((v_1,w_1),\dots,(v_{Ni},w_{Ni}))}{(\mathtt{thin}(s,t),N)\to((v_1,w_1),(v_{i+1},w_{i+1}),(v_{2i+1},w_{2i+1}),\dots,(v_{(N-1)i+1},w_{(N-1)i+1}))}$$

$$\frac{(t,N)\to v_1\quad(s(t),N)\to v_2\quad\dots\quad(s^{N-1}(t),N)\to v_N}{(\mathtt{prng}(s,t),N)\to((v_1,1),\dots,(v_N,1))}$$

TABLE II: Big-step operational semantics of sampler operations

拓扑。空间 $f^{-1}(i)$ $f\in\{\le,<,\ge,>,=,\neq\},i\in 2$ 的解释正如该记法所示，例如

$$[\![<^{-1}\,(0)]\!]=\{(x,y)\mid x,y\in\mathbb{R}\wedge x\ge y\},$$
$$[\![=^{-1}\,(1)]\!]=\{(x,x)\mid x\in\mathbb{R}\}$$

以及从 $\mathbb{R}\times\mathbb{R}$ 继承的子空间拓扑。由于所有这些空间都是可度量化的，根据命题 III.2，我们的基类型在 CG 中得到解释。

乘积（包括单位类型）和函数类型以显然的方式，利用 CG 的笛卡尔闭结构进行解释。余积类型通过 CG 中的余积进行解释；给定两个项 s、$t:\mathsf{T}$，它们被解释为 CG-态射 $[\![s]\!]\colon A\to[\![\mathsf{T}]\!]$、$[\![t]\!]\colon B\to[\![\mathsf{T}]\!]$，则拉回类型 $\mathsf{T}_{s\ t}$ 解释为沿 $[\![t]\!]$ 的 $A\times_{[\![\mathsf{T}]\!]}B$ 对 $[\![s]\!]$ 的拉回。根据命题 III.2，所有这些空间都属于 CG。

由于采样器类型是余归纳类型，其语义将取决于终余代数的存在性。

定理 III.1（Adámek）。设 $\mathscr{C}$ 是一个带有终对象 1 的范畴，且 $F\colon\mathscr{C}\to\mathscr{C}$ 是一个函子。如果 $\mathscr{C}$ 具有并且 F 保持 ω^{op}-指标化极限，那么 νF 的极限 $1\xleftarrow{!}F1\xleftarrow{F}FF1\xleftarrow{FF}\cdots$ 就是 F 的终余代数。

由于 CG 是完备的，它具有由 ω^{op} 索引的极限。回顾一下，我们希望将 $\Sigma\mathsf{T}$ 解释为由“函子” $\mathsf{T}\times\mathsf{R}^+\times-$ 定义的余归纳类型。形式上，给定一个类型 T，我们希望

$$[\![\Sigma\,\mathsf{T}]\!]\triangleq\nu([\![\mathsf{T}]\!]\times\mathbb{R}^+\times\mathrm{Id}).\tag{4}$$

由于积是极限，而极限与极限可交换，因此函子 $[\![\mathsf{T}]\!]\times\mathbb{R}^+\times$ 恒等函子保持极限，尤其保持由 ω^{op} 索引的极限。由此，Adámek 定理保证了满足（4）的对象存在。更具体地说，由于终对象 1 显然是可度量化的，且由于$\mathbb{R}^+$ 是可度量化的，只要 $[\![\mathsf{T}]\!]$ 是可度量化的，终序列中的每个对象也将是可度量化的。因此，只要 $[\![\mathsf{T}]\!]$ 是可度量化的，$\prod_n([\![\mathsf{T}]\!]\times\mathbb{R}^+)^n$ 就是可度量化的，并因而赋有通常的积拓扑。定义（4）的极限是该积的一个闭子空间，这意味着当 $[\![\mathsf{T}]\!]$ 是可度量化的时，在 CG 中定义 $[\![\Sigma\mathsf{T}]\!]$ 的极限与 Top 中的极限相同（例如，当 T 是一个基类型或基类型的积时）。然而，通过以余归纳的方式定义 $[\![\Sigma\mathsf{T}]\!]$，而不是简单地将其定义为 $([\![\mathsf{T}]\!]\times\mathbb{R}^+)^\omega$，我们在 $[\![\Sigma\mathsf{T}]\!]$ 上获得了一个终余代数结构，从而能够以余归纳的方式定义采样器操作。

3）子类型关系的语义：我们的语言包含谓词 $f\in\{\le,<,\ge,>,=,\neq\}$（对于拒绝采样 § II-C 至关重要），但其目标解释空间却是由拓扑空间和连续映射构成的宇宙。对于 $\mathbb{R}\times\mathbb{R}\to 2$ 上的通常拓扑而言，这些谓词并不是连续映射 $\mathbb{R}\times\mathbb{R}$。然而，对于每个这样的谓词 f，集合 $[\![f^{-1}(0)]\!]$ 和 $[\![f^{-1}(1)]\!]$ 都是局部闭集，也就是说，它们是开集与闭集的交集（关于 $\mathbb{R}\times\mathbb{R}$ 上的通常拓扑），因此根据命题 III.2，它们是 CG 空间。例如，$[\![<^{-1}(0)]\!]$ 是闭集，而 $[\![<^{-1}(1)]\!]$ 是开集。

我们处理不连续性的核心思路是：由于 CG 是余完备的，空间 $[\![f^{-1}(0)]\!]+[\![f^{-1}(1)]\!]$ 是一个 CG 空间。该空间具有一个良好性质，即 f 作为映射 $f\colon[\![f^{-1}(0)+f^{-1}(1)]\!]\to 2$是连续的。由于每个 $f^{-1}(i)$ 都是一个类型，我们只需将 Func 中的这些内置函数类型标注为 $f\colon f^{-1}(0)+f^{-1}(1)\to\mathsf{B}$，即可强制实现这一语义。

$[\![f^{-1}(0)+f^{-1}(1)]\!]$ 上的拓扑比 $\mathbb{R}\times\mathbb{R}$上的通常拓扑更精细，这意味着恒等映射 $\mathrm{Id}:[\![f^{-1}(0)+f^{-1}(1)]\!]\to\mathbb{R}\times\mathbb{R}$ 是连续的。这是图 4b 中公理的语义基础。根据其他规则，通过归纳可以很容易地看出，子类型关系始终存在于共享同一载体集合的空间之间，并且在语义上由拓扑粗化给出。换言之，如果 $\mathsf{S}\triangleleft\mathsf{T}$，那么 $[\![\mathsf{S}]\!]$ 和 $[\![\mathsf{T}]\!]$ 共享同一载体，并且相应的恒等映射 $\mathrm{Id}:[\![\mathsf{S}]\!]\to[\![\mathsf{T}]\!]$ 是连续的。

例 III.1. 令 $\mathsf{p}\triangleq\text{if }x=0\text{ then 1 else }-1$；我们首先展示上下文限制规则如何帮助我们对该程序进行类型检查。为便于阅读，令 $\mathsf{Eq}=^{-1}(1)$ 且 $\mathsf{Neq}=^{-1}(0)$。现在，使用 $=$，我们推导出：$\mathsf{Neq}+\mathsf{Eq}\to$实数集，

$$\dfrac{\dfrac{\dfrac{x:\mathtt{R}\vdash x:\mathtt{R}\qquad\vdash 0:\mathtt{R}}{x:\mathtt{R}\vdash (x,0):\mathtt{R}\times\mathtt{R}}}{x:(x,0)^{-1}\mathtt{Neq}+(x,0)^{-1}\mathtt{Eq}\vdash (x,0):\mathtt{Neq}+\mathtt{Eq}}\ \mathtt{Neq}+\mathtt{Eq}\triangleleft\mathtt{R}\times\mathtt{R}}{\dfrac{x:(x,0)^{-1}\mathtt{Neq}+(x,0)^{-1}\mathtt{Eq}\vdash x=0:\mathtt{B}\qquad\vdash 1:\mathtt{R}\qquad\vdash -1:\mathtt{R}}{x:(x,0)^{-1}\mathtt{Neq}+(x,0)^{-1}\mathtt{Eq}\vdash \mathtt{if}\ x=0\ \mathtt{then}\ 1\ \mathtt{else}\ -1:\mathtt{R}}}$$

预期将在下文讨论项的语义，可以很容易地证明

$$[\![(x,0)^{-1}\mathtt{Neq}+(x,0)^{-1}\mathtt{Eq}]\!]=((-\infty,0)\cup(0,\infty))+\{0\}$$

因此 $[\![\mathtt{p}]\!]$ 是连续映射

$$[\![\mathtt{p}]\!]:((-\infty,0)\cup(0,\infty))+\{0\}\to\mathbb{R},x\mapsto\begin{cases}1 & \textit{if } x=0\\ -1 & \textit{else}\end{cases}$$

4）良构项的语义：公理、弱化、子类型、乘积、投影、let 绑定、 λ-抽象、函数应用、注入和模式匹配均按预期方式进行解释（鉴于 CG 是带余积的笛卡尔闭范畴）。

连续内置函数，例如 $+:\ \mathtt{R}\times\mathtt{R}\to\mathtt{R}$ 或 $\exp:\ \mathtt{R}\to\mathtt{R}$，按显然的方式进行解释。如上文所述，不连续内置函数 $\{\leq,<,\geq,>,=,\neq\}$ 的类型定义方式使得其自然解释按定义必然是连续的。

现在可以描述上下文限制规则的语义。根据前提、我们在第 III-C3 节中的观察结果以及附加条件，我们有态射

$$[\![t]\!]:\prod_{j\in n}[\![\mathtt{S}_j]\!]\to[\![\mathtt{T}]\!],\quad\text{and}\quad\mathrm{Id}:\coprod_{i\in m}[\![\mathtt{T}_i]\!]\to[\![\mathtt{T}]\!].$$

根据公式（1），我们将每个“逆像类型” $t^{-1}(\mathtt{T}_i)$解释为 $[\![t]\!]$ 沿包含映射 $[\![\mathtt{T}]\!]_i\hookrightarrow\coprod_{i\in m}[\![\mathtt{T}_i]\!]$ 的拉回（逆像），正如该记号所暗示的，它简单地表示为$[\![t]\!]^{-1}([\![\mathtt{T}_i]\!])$。由于 $\coprod_{i\in m}[\![\mathtt{T}_i]\!]$ 和 $[\![\mathtt{T}]\!]$ 具有相同的载体，显然这定义了 $[\![\Gamma]\!]$的一个划分，因此我们可以将 t 重新类型化为连续映射 $\coprod_{i\in m}[\![t^{-1}(\mathtt{T}_i)]\!]\to\coprod_{i\in m}[\![\mathtt{T}_i]\!]$，从而解释该规则。

如本节前文所述，上下文限制阻止了 λ-抽象；下面的示例说明了为什么必须如此。

示例 III.2. 考虑由以下方式导出的程序 $x<y$：

$$\dfrac{x:\mathtt{R},y:\mathtt{R}\vdash(x,y):\mathtt{R}\times\mathtt{R}}{(x,y):(x,y)^{-1}(<^{-1}(0))+(x,y)^{-1}(<^{-1}(1))\vdash(x,y):<^{-1}(0)+<^{-1}(1)}$$
$$(x,y):(x,y)^{-1}(<^{-1}(0))+(x,y)^{-1}(<^{-1}(1))\vdash x<y:\mathtt{B}$$

$x<y$ 的解释由连续函数给出

$$[\![<]\!]:\{(x,y)\mid x<y\}+\{(x,y)\mid x\geq y\}\to 2.$$

尽管它具有相同的载体 $\mathbb{R}\times\mathbb{R}$，但该映射的定义域不再是拓扑空间的乘积；现在它是拓扑空间的余积。这意味着，利用 CG 的笛卡尔闭结构，已经无法再对该函数的某个变量进行 λ-抽象。

为了能够对映射 $<$进行 λ-抽象，我们需要在 $\mathbb{R}\times\mathbb{R}$ 上定义一种拓扑，使得对于任意给定的 $x_0\in\mathbb{R}$，函数 $x_0<-:\ \mathbb{R}\to 2$ 都是连续的。这将把开集 $[x_0,\infty)$ 引入 $\mathbb{R}$ 的拓扑中（对每个 $x_0\in\mathbb{R}$而言），这意味着我们必须为 $\mathbb{R}$ 赋予众所周知的问题重重的下限拓扑（又称 Sorgenfrey 直线）。它是否为 CG 空间似乎是一个棘手的问题，可能独立于 ZF *[20]*。

最后，我们利用采样器类型的余归纳性质来定义采样器操作的指称语义。回忆一下，对于类型 T，有 $[\![\Sigma\mathtt{T}]\!]$， ν($[\![\mathtt{T}]\!]\times\mathbb{R}^+\times$恒等函子)。特别地，$[\![\Sigma\mathtt{T}]\!]$ 自带一个余代数结构映射

$$\mathrm{unfold}_{\mathtt{T}}:[\![\Sigma\,\mathtt{T}]\!]\to[\![\mathtt{T}]\!]\times\mathbb{R}^+\times[\![\Sigma\,\mathtt{T}]\!].$$

此外，对于任意其他的（连续）余代数结构映射γ： $X\to[\![\mathtt{T}]\!]\times\mathbb{R}^+\times X$， $[\![\Sigma\,\mathtt{T}]\!]$的终对象性质提供了一个唯一的 $[\![\mathtt{T}]\!]\times\mathbb{R}^+\times$恒等函子-余代数态射

$$\mathrm{beh}(\gamma):X\to[\![\Sigma\,\mathtt{T}]\!].$$

由于 $[\![\Sigma\mathtt{T}]\!]$ 在 CG 中进行解释，因此可以自动推出 $\mathrm{unfold}_{\mathtt{T}}$ 和 beh(γ) 都是连续的。然而，并不立即清楚的是，beh 实际上关于 γ 是连续的。

命题 III.3。 *[附录 A]* 设 $F:\mathrm{CG}\to\mathrm{CG}$ 满足定理 III.1 的条件以及 $\mathrm{int}(\nu F)\neq\emptyset$in $\prod_i F^i1$ 的条件，并设 beh_X: $[X,FX]\to[X,\nu F]$为将 F-余代数结构关联到终余代数中唯一余代数态射的（行为）映射。映射 beh_X 是连续的，即为一个 CG-态射。

利用 unfold 和 beh，我们定义表 III 中所有采样器操作的指称语义。这些定义恰好是表 II 操作语义中所定义的有限变换的无限（余归纳的）版本。这些定义中涉及的所有映射都是连续的；这是命题 III.3 以及求值和函数复合是 CG 的内部同态集上的连续操作这一事实的直接推论（[35, 5.2,5.9]）。

D. 充分性

这门语言有一个有趣的不对称之处：其指称语义是用余归纳的采样器类型 $[\![\Sigma\mathtt{T}]\!]$来表述的，而其操作语义则是用有限样本序列上的有限操作来表述的。此外，操作语义通过归约到值来给出，即归约到类型由不含类型构造器 Σ 的类型构成的项；而指称语义并不作此区分。为了建立二者之间的联系，我们首先按照操作语义背后的思路，定义一种将任意类型的项转换为值的通用方法。给定类型 T 和整数 N，我们归纳地定义其对应的值类型 $\mathrm{val}^N(\mathtt{T})\in\mathrm{Value}$，如下：

$$\mathrm{val}^N(\mathtt{G})=\mathtt{G}\qquad\mathrm{val}^N(\Sigma\,\mathtt{T})=\left(\mathrm{val}^N\mathtt{T}\right)^N$$
$$\mathrm{val}^N(\mathtt{S}*\mathtt{T})=\mathrm{val}^N(\mathtt{S})*\mathrm{val}^N(\mathtt{T}),\quad *\in\{\times,+,\to\}$$

$$\frac{[\![\Gamma \vdash t:\ \Sigma \mathtt{T}]\!] = f}{[\![\Gamma \vdash \mathtt{hd}(t):\ \mathtt{T}]\!] = \pi_1 \circ \text{展开}_{\mathtt{T}} \circ f} \qquad \frac{[\![\Gamma \vdash t:\ \Sigma \mathtt{T}]\!] = f}{[\![\Gamma \vdash \mathtt{wt}(t):\ \mathtt{R}^+]\!] = \pi_2 \circ \text{展开}_{\mathtt{T}} \circ f} \qquad \frac{[\![\Gamma \vdash t:\ \Sigma \mathtt{T}]\!] = f}{[\![\Gamma \vdash \mathtt{tl}(t):\ \Sigma \mathtt{T}]\!] = \pi_3 \circ \text{展开}_{\mathtt{T}} \circ f}$$

$$\frac{[\![\Gamma \vdash s:\ \mathtt{N}]\!] = f \quad [\![\Gamma \vdash t:\ \Sigma \mathtt{T}]\!] = g}{[\![\Gamma \vdash \mathtt{thin}(s,t):\ \Sigma(\mathtt{T})]\!] = \mathrm{ev}_{\Sigma \mathtt{T},\ \Sigma \mathtt{T}} \circ (\text{恒等函子}_{\Sigma \mathtt{T}} \times \mathrm{beh}_{\Sigma \mathtt{T}}) \circ (\text{恒等函子}_{\Sigma \mathtt{T}} \times (\text{展开}_{\mathtt{T}} \circ (\pi_3 \circ \text{展开}_{\mathtt{T}})(\cdot\ -1))) \circ \langle f,\ g \rangle}$$

$$\frac{[\![\Gamma \vdash s:\ \Sigma \mathtt{S}]\!] = f \quad [\![\Gamma \vdash t:\ \Sigma \mathtt{T}]\!] = g}{[\![\Gamma \vdash s \otimes t:\ \Sigma(\mathtt{S} \times \mathtt{T})]\!] = \mathrm{beh}_{\Sigma \mathtt{S},\ \Sigma \mathtt{T}}(\pi_1 \times \pi_4 \times (\pi_2 \cdot \pi_5) \times \pi_3 \times \pi_6 \circ \text{展开}_{\mathtt{S}} \times \text{展开}_{\mathtt{T}}) \circ \langle f,\ g \rangle}$$

$$\frac{[\![\Gamma \vdash s:\ \Sigma \mathtt{S}]\!] = f \quad [\![\Gamma \vdash t:\ \mathtt{S} \to \mathtt{T}]\!] = g}{[\![\Gamma \vdash \mathtt{map}(t,s):\ \Sigma \mathtt{T}]\!] = \mathrm{ev}_{\Sigma \mathtt{S},\ \Sigma \mathtt{T}} \circ (\mathrm{id}_{\Sigma \mathtt{S}} \times \mathrm{beh}_{\Sigma \mathtt{S}}) \circ (\mathrm{id}_{\Sigma \mathtt{S}} \times ((- \times \mathrm{id}_{\mathtt{R}^+} \times \mathrm{id}_{\Sigma \mathtt{S}}) \circ \text{展开}_{\mathtt{S}})) \circ \langle f,\ g \rangle}$$

$$\frac{[\![\Gamma \vdash s:\ \Sigma \mathtt{T}]\!] = f \quad [\![\Gamma \vdash t:\ \mathtt{T} \to \mathtt{R}^+]\!] = g}{[\![\Gamma \vdash \mathtt{reweight}(t,s)]\!] = \mathrm{ev}_{\Sigma \mathtt{T},\ \Sigma \mathtt{T}} \circ (\mathrm{id}_{\Sigma \mathtt{T}} \times \mathrm{beh}_{\Sigma \mathtt{T}}) \circ (\mathrm{id}_{\Sigma \mathtt{T}} \times ((\mathrm{id}_{\mathtt{T}} \times - \times \mathrm{id}_{\Sigma \mathtt{T}}) \circ \text{展开}_{\mathtt{T}})) \circ \langle f,\ g \rangle}$$

$$\frac{[\![\Gamma \vdash t:\ \mathtt{T}]\!] = f \quad [\![\Gamma \vdash s:\ \mathtt{T} \to \mathtt{T}]\!] = g}{[\![\Gamma \vdash \mathtt{prng}(s,t):\ \Sigma \mathtt{T}]\!] = \text{求值}_{\mathtt{T},\ \Sigma \mathtt{T}} \circ (\text{恒等函子}_{\mathtt{T}} \times \mathrm{beh}_{\mathtt{T}}) \circ (\text{恒等函子}_{\mathtt{T}} \times (\text{恒等函子}_{\mathtt{T}} \times 1 \times -)) \circ \langle f,\ g \rangle}$$

表 III：采样器操作的指称语义

其中 $\mathtt{G} \in \mathrm{Ground}$.[3] 现在，我们递归地定义广义投影映射 $p_{\mathtt{T}}^N \colon [\![\mathtt{T}]\!] \to \left[\!\left[\mathrm{val}^N(\mathtt{T})\right]\!\right]$：

$$p_{\mathtt{G}}^N = \mathrm{id}_{[\![\mathtt{G}]\!]}, \qquad p_{\mathtt{S}*\mathtt{T}}^N = p_{\mathtt{S}}^N * p_{\mathtt{T}}^N, * \in \{\times, +\}$$
$$p_{\mathtt{S}\to\mathtt{T}}^N = \mathrm{id}_{[\![\mathtt{S}\to\mathtt{T}]\!]} \qquad p_{\Sigma\mathtt{T}}^N = \pi_{1:N} \circ (p_{\mathtt{T}\times\mathtt{R}^+}^N)^\omega$$

读者可能已经注意到，我们对 $p_{\mathtt{S}\to\mathtt{T}}^N$的定义是平凡的。原因在于，正如快速查看表 II 中的规则便可发现，类型为 $(t, N) \to v$ 且其中 t 为函数类型的情形，既不会出现在结论中，也不会出现在前提中。函数类型项仅出现在求值中，或者是值，即平凡地归约为自身的项。

定理 III.2。 *[附录 A]* 对于任意程序 $\vdash t : \mathtt{T}$，我们有

$$(t, N) \to v \Leftrightarrow p_{\mathtt{T}}^N([\![t]\!]) = [\![v]\!] .$$

IV. 采样器等价性

为了实现一个系统，以推理确定性采样器是否以某个特定概率分布为目标，必须先定义采样器之间的等价性概念。这样的系统为验证采样器提供了一条自然路径：首先将给定的采样器 s改写为一个等价但更简单的形式，然后证明这个简化形式以正确的分布为目标。§ II 中的推导采用了这种方法，并隐式地使用了若干等价性结果——特别是 let 归约，以及对任意采样器 s，嵌套自乘积 $(s^m)^n$ 与自乘积 s^{m*n} 等价这一结果。本节将引入程序上的关系 $\approx$，以证明这类推理是合理的。

定义 IV.1. 我们称两个程序 $\Gamma \vdash s : \mathtt{T}$ 和 $\Gamma \vdash t : \mathtt{T}$ 等价，记作 $\Gamma \vdash s \approx t : \mathtt{T}$，若它们由包含表 IV 规则的、定义在类型正确项上的最小同余关系联系起来。[4]

[3]由于我们在此只关注封闭采样器，并且拉回类型只会出现在上下文中，因此无需在拉回类型上定义 val^N。

[4]同余关系是指：$\approx$ 它是一种等价关系，并且在语言中的所有操作下保持不变。例如，如果 $\Gamma \vdash s \approx t : \Sigma\mathtt{T}$成立，那么 $\Gamma \vdash \mathtt{tl}(s) \approx \mathtt{tl}(t) : \Sigma\mathtt{T}$ 也必须成立；对于语言中的所有操作，情况同样如此。

表 IV 中的规则采用了一些简写约定，以便更简洁地呈现。我们引入恒等函数 $\mathrm{id}_{\mathtt{S}}$, $\lambda x : \mathtt{S}.\ x : \mathtt{S} \to \mathtt{S}$，常值函数 $1_{\mathtt{S}}$, $\lambda x : \mathtt{S}. \mathbf{1} : \mathtt{S} \to \mathtt{R}^+$，函数复合 $t \circ s$, $\lambda x : \mathtt{S}.\ t(s(x)) : \mathtt{S} \to \mathtt{U}$，其中 $s : \mathtt{S} \to \mathtt{T}$ $t : \mathtt{T} \to \mathtt{U}$，复合 $\mathtt{f}^0 \triangleq \mathrm{id}_{\mathtt{S}} : \mathtt{S} \to \mathtt{S}$，$\mathtt{f}^n$，$\mathtt{f} \circ \mathtt{f}^{n-1}$ 对任意 $n \in \mathbb{N}$，实值函数 $s : \mathtt{S} \to \mathtt{R}^+$，$t : \mathtt{T} \to \mathtt{R}^+$ 的逐点乘积 $s \cdot t$, $\lambda x : \mathtt{S}\ y : \mathtt{T}.\ s(x) * t(y) : \mathtt{S} \times \mathtt{T} \to \mathtt{R}^+$，以及最后的函数 $s : \mathtt{S} \to \mathtt{S}'$，$t : \mathtt{T} \to \mathtt{T}'$ 的笛卡尔积 $s \times t$, $\lambda x : \mathtt{S}\ y : \mathtt{T}.\ (s(x)\ t(y)) : \mathtt{S} \times \mathtt{T} \to \mathtt{S}' \times \mathtt{T}'$。

定理 IV.1. *[附录 A]* 表 IV 中的规则是可靠的：若 $\Gamma \vdash s \approx t : \mathtt{T}$，则 $[\![\Gamma \vdash s : \mathtt{T}]\!] = [\![\Gamma \vdash t : \mathtt{T}]\!]$。

该证明是余归纳推理中的一道简单练习，可在附录中找到，那里还列出了完整的等价规则。根据抽象，这些规则相对于操作等价的可靠性也就随之成立，不过直接证明这一点同样很简单。

回顾一下，我们的采样器操作的一个重要应用，是为式 (3) 中给出的采样器的自乘积提供形式化定义。我们的等价规则能够表明该自乘积是良定义的，这一点至关重要。

命题 IV.1. *[附录 A]* 对于任意 $\Gamma \vdash s : \Sigma\mathtt{S}$，$m\,n \in \mathbb{N}$，该自乘积满足 $\Gamma \vdash (s^m)^n \approx s^{mn} : \Sigma(\mathtt{S}^{mn})$。

表 IV 中的等价规则给出了一种简化采样器的过程。考虑不含操作 prng 的采样器。在其余的采样器操作中，我们确定了两组：{tl、hd、wt、thin $\otimes$} 和{map、reweight}。应用表 IV 中的规则可以看到，对于第一组和第二组操作的每一种组合，都存在一条规则，使我们能够将第一种操作移入第二种操作的主体中。因此，任何不含 prng 实例的采样器都可以改写为：采样器操作 tl、hd、wt、thin $\otimes$ 全部向内移动。

命题 IV.2。设 $\Gamma \vdash t : \Sigma\mathtt{T}$ 是一个不包含任何伪随机数生成器实例的采样器。应用表 IV 中的规则可知，我们可以等价地将此类采样器重写为以下任一种形式

$\Gamma,\ s:\mathsf{S}\vdash(\lambda x:\mathsf{S}.t)(s)\approx t[x\leftarrow s]:\mathsf{T}\ \Gamma\vdash\lambda x:\mathsf{S}.t(x)\approx t:\mathsf{S}\to\mathsf{T}\Gamma,\ s:\mathsf{S}\vdash$令 $x=s$，则 $t\approx(\lambda x:\mathsf{S}.t)(s):\mathsf{T}$ $\Gamma\vdash$若 True 则 s 否则 $t\approx s:\mathsf{T}\ \Gamma\vdash$若 False 则 s 否则 $t\approx t:\mathsf{T}\Gamma\vdash$第一投影$((s,t))\approx s:\mathsf{S}\ \Gamma\vdash$第二投影$((s,t))\approx t:\mathsf{T}$ $\Gamma\vdash\mathtt{hd}(\mathtt{map}(s,t))\approx s(\mathtt{hd}(t)):\ \mathtt{T}$ $\Gamma\vdash\mathtt{wt}(\mathtt{map}(s,t))\approx\mathtt{wt}(t):\ \mathtt{R}^+$ $\Gamma\vdash\mathtt{tl}(\mathtt{map}(s,t))\approx\mathtt{map}(s,\mathtt{tl}(t)):\ \Sigma\mathsf{T}\Gamma\vdash(\mathtt{hd}(s)\ \mathtt{hd}(t))\approx\mathtt{hd}(s\otimes t):\ \mathtt{S}\times\mathtt{T}$ $\Gamma\vdash\mathtt{wt}(s)*\mathtt{wt}(t)\approx\mathtt{wt}(s\otimes t):\ \mathtt{R}^+$ $\Gamma\vdash\mathtt{tl}(s)\otimes\mathtt{tl}(t)\approx\mathtt{tl}(s\otimes t):\ \Sigma(\mathtt{S}\times\mathtt{T})\Gamma\vdash\mathtt{hd}(\mathtt{thin}(n,t))\approx\mathtt{hd}(t):\ T$ $\Gamma\vdash\mathtt{wt}(\mathtt{thin}(n,t))\approx\mathtt{wt}(t):\ \mathtt{R}^+$ $\{\Gamma\vdash\mathtt{tl}(\mathtt{thin}(n,t))\approx\mathtt{thin}(n,\mathtt{tl}^n(t)):\ \Sigma\mathsf{T}\mid n\in\mathbb{N}\}$ $\Gamma\vdash\mathtt{thin}(1,t)\approx t:\ \Sigma\mathsf{T}\Gamma\vdash\mathtt{hd}(\mathtt{prng}(s,t))\approx t:\ \mathtt{T}$ $\Gamma\vdash\mathtt{wt}(\mathtt{prng}(s,t))\approx 1:\ \mathtt{R}^+$ $\Gamma\vdash\mathtt{tl}(\mathtt{prng}(s,t))\approx\mathtt{prng}(s,s(t)):\ \Sigma\mathsf{T}\Gamma\vdash\mathtt{hd}(\mathtt{reweight}(s,t))\approx\mathtt{hd}(t):\ \mathtt{T}$ $\Gamma\vdash\mathtt{wt}(\mathtt{reweight}(s,t))\approx s(\mathtt{hd}(t))*\mathtt{wt}(t):\ \mathtt{R}^+$ $\Gamma\vdash\mathtt{tl}(\mathtt{reweight}(s,t))\approx\mathtt{reweight}(s,\mathtt{tl}(t)):\ \Sigma\mathsf{T}\Gamma\vdash\mathsf{thin}(n,\mathsf{thin}(m,t))\approx\mathsf{thin}(n*m,t)\approx\Sigma\mathsf{T}\ \Gamma\vdash$映射$(g,$映射$(f,t))\approx$映射$(g\circ f,t):\ \Sigma\mathsf{T}\Gamma\vdash$重新加权$(g,\mathsf{reweight}(f,t))\approx$重新加权$(f\cdot g,t):\ \Sigma\mathsf{T}\{\Gamma\vdash\mathsf{thin}(n,$伪随机数生成器$(s,t))\approx$伪随机数生成器$(s^n,t):\ \Sigma\mathsf{T}\mid n\in\mathbb{N}\}$ $\Gamma\vdash\mathsf{thin}(n,$映射$(s,t))\approx$映射$(s,\mathsf{thin}(n,t)):\ \Sigma\mathsf{T}\Gamma\vdash s\otimes$映射$(f,t)\approx$映射$(\mathsf{id}_\mathsf{S}\times f,s\otimes t):\ \Sigma(\mathsf{S}\times\mathsf{T})\ \Gamma\vdash$映射$(f,s)\otimes t\approx$映射$(f\times\mathsf{id}_\mathsf{T},s\otimes t):\ \Sigma(\mathsf{S}\times\mathsf{T})\Gamma\vdash s\otimes$重新加权$(g,t)\approx$重新加权$(1_\mathsf{S}\cdot g,s\otimes t):\ \Sigma(\mathsf{S}\times\mathsf{T})\ \Gamma\vdash$重新加权$(f,s)\otimes t\approx$重新加权$(f\cdot 1_\mathsf{T},s\otimes t):\ \Sigma(\mathsf{S}\times\mathsf{T})\Gamma\vdash$伪随机数生成器$(f,a)\otimes$伪随机数生成器$(g,b)\approx$伪随机数生成器$(f\times g,(a,b)):\ \Sigma(\mathsf{S}\times\mathsf{T})$ $\Gamma\vdash\mathsf{thin}(n,s)\otimes\mathsf{thin}(n,t)\approx\mathsf{thin}(n,s\otimes t):\ \Sigma(\mathsf{S}\times\mathsf{T})$

表 IV：采样器等价规则

形式为 $\Gamma\vdash\mathsf{map}(f,\mathsf{reweight}(g,\mathsf{map}(f',\ldots,s))):\ \Sigma\mathsf{T}$ 或 $\Gamma\vdash\mathsf{reweight}(g,\mathsf{map}(f,\mathsf{reweight}(g',\ldots,s))):\ \Sigma\mathsf{T}$，即由 map 和 reweight 的调用组成（包括二者出现次数均为零的平凡情况），其中关键在于，采样器 s 不包含采样器操作 map 或 reweight。

我们还可以证明，自乘积可分配于 map 和 reweight 操作之上；此类操作有助于以更简单的形式表示复合采样器的自乘积，随后即可验证该形式的正确性。

命题 IV.3。*[附录 A]* 对于任意映射采样器 $\Gamma\vdash\mathsf{map}(f,s):\ \Sigma\mathsf{T}$ 以及任意 $n\in\mathbb{N}$，可推出 $\Gamma\vdash\mathsf{map}(f,s)^n\approx\mathsf{map}(f\times\ldots\times f,s^n):\ \Sigma(\mathsf{T}^n)$；对于重新加权的采样器 $\Gamma\vdash\mathsf{reweight}(f,s):\ \Sigma\mathsf{S}$，可推出 $\Gamma\vdash\mathsf{reweight}(f,s)^n\approx\mathsf{reweight}(f\cdot\ldots\cdot f,s^n):\ \Sigma(\mathsf{S}^n)$。

V. 采样器的语义正确性

采样器的基本正确性标准是：它应生成服从所需目标分布的样本。本节旨在概述一种简单的“目标演算”，以组合地验证这一性质。我们通过测度的弱收敛来表述这种正确性；虽然也可以采用其他收敛概念，但弱收敛是标准做法，足以满足我们的需求。

A. 经验变换

首先，我们需要形式化说明：当我们说采样器 $s:\Sigma\mathsf{T}$ 的目标是在 $[\![\mathsf{T}]\!]$ 上的概率分布时，具体指什么。给定一个拓扑空间 X，记 $\mathcal{P}X$ 为在（由 X生成的 Borel σ-代数上定义的）概率测度空间，并赋予弱收敛拓扑；也就是说，若对于任意有界连续映射 $f:X\to\mathbb{R}$，$\lim_{n\to\infty}\int f\,d\mu_n=\int f\,d\mu$，则 $\lim_{n\to\infty}\mu_n=\mu$ 在 $\mathcal{P}X$ 中。事实上，$\mathcal{P}$ 定义了一个函子 Top $\to$ Top：若 $f:X\to Y$ 是连续映射，则 $\mathcal{P}(f)\ f_*:\mathcal{P}X\to\mathcal{P}Y$ 是推前映射，它很容易证明是连续的。我们不知道当 X 是 CG 空间时，$\mathcal{P}X$ 是否也是 CG 空间；尤其是，我们不知道能否在 CG 上为 $\mathcal{P}$赋予单子结构。不过，这些问题与本文的工作无关，因为 $\mathcal{P}$ 在第 III-C 节的语义中不起作用。

对于任意流 $\sigma:\mathbb{N}\to X\times\mathbb{R}^+$，我们将 $\hat{}\sigma_n\in\mathcal{P}X$ 定义为

$$\hat{\sigma}_n\triangleq\frac{1}{n}\sum_{i=1}^{n}\frac{\pi_2(\sigma(i))}{\sum_{j=1}^{n}\pi_2(\sigma(j))}\delta_{\pi_1(\sigma(i))},$$

基于 σ 的前 n 个（加权）样本的经验分布。我们还定义 $\mathcal{P}_\perp X$、$\mathcal{P}X+1$，其中 $1=\{\perp\}$ 是终对象，$+$ 是 Top 中的余积。

定义 V.1。经验测度变换是 $\mathsf{Top}^{\mathrm{obj}}$-族映射

$$\varepsilon_X:(X\times\mathbb{R}^+)^{\mathbb{N}}\to\mathcal{P}_\perp X,\sigma\mapsto\begin{cases}\lim_{n\to\infty}\hat{\sigma}_n & \textit{if it exists}\\ \perp\in 1 & \textit{else}\end{cases}$$

如下例所示，经验测度变换不可能是自然的。

例 V.1. 令 $\sigma:\mathbb{N}\to X$ 为 X上的一个发散无权采样器，即 $\varepsilon_X(\sigma)=\perp$，并考虑到终对象的映射 $!:X\to 1$。那么 $\varepsilon_1(!\circ\sigma)=\varepsilon_1(\perp,\perp,\ldots)=\delta_\perp$，但 $\mathcal{P}_\perp(!)(\varepsilon_X(\sigma))=\mathcal{P}_\perp(!)(\perp)=\perp$。

然而，如果某个采样器确实通过ε定义了一个概率测度，那么连续映射会保留这一性质。

命题 V.1. *[附录 A]* 令 $\sigma:\mathbb{N}\to X\times\mathbb{R}^+$ 和$f:X\times\mathbb{R}^+\to Y\times\mathbb{R}^+$ 为连续映射。若 $\varepsilon_X(\sigma)=\mu$，则$\varepsilon_Y(f\circ\sigma)=f_*(\mu)$。

我们很容易想要将连续映射这一优良性质推广到更一般的映射，例如可测映射。下面的例子表明，这是不可能的。

例 V.2. 设 $X=[0,1]$ 和 $\sigma:\mathbb{N}\to[0,1]$ 是任意无权采样器，使得 $\varepsilon(\sigma)$ 为勒贝格测度

于 $[0,1]$ 上。现在考虑映射 $f\colon [0,1]\to\{0,1\}$，其定义为：若对某个 i， $x=\sigma(i)$，则 $f(x)=1$；否则为 0。该函数是一个可数集（因此也是闭集）的示性函数，所以是 Borel 可测的。一方面，我们有 $\varepsilon(f\circ\sigma)=\delta_1$，因为 $f\circ\sigma$ 是取值恒为 1 的常值流；但另一方面，我们有 $f_*(\varepsilon(\sigma))(1)=\varepsilon(\sigma)(f^{-1}(1))=0$，因为只有一个可数集会被 f 映射到 1。

即使对于只有有限个不连续性的函数，也不可能扩展满足命题 V.1 的函数类。然而，第 III-C 节所采用的语义框架使我们能够完全绕过这一问题。我们通过重新考察例 III.1 来说明这两点。

例 V.3。考虑采样器 $\mathsf{s}\triangleq$ 伪随机数生成器$(\lambda x:$实数集$.\ x/2, 1)$ 以及例 III.1 中的项 $\mathsf{p}\triangleq$ if $x=0$ then 1 else -1。首先假设 $\mathbb{R}$ 赋备标准拓扑，即 $[\![\mathsf{p}]\!]$ 在 0 处不连续。由于 $\mathbb{R}$ 是一个度量空间，我们可以使用 Portmanteau 引理，将弱收敛重新表述为仅针对有界 Lipschitz 函数。于是很容易证明 $\varepsilon([\![\mathsf{s}]\!])=\delta_0$：令 $f\colon\mathbb{R}\to\mathbb{R}$ 为有界 Lipschitz 函数，则有

$$\lim_{n\to\infty}\left|\int f\ d\widehat{[\![\mathsf{s}]\!]}_n-\int f\ d\delta_0\right|=\lim_{n\to\infty}\left|\frac{1}{n}\sum_{i=1}^{n}f\left(\frac{1}{2^i}\right)-f(0)\right|$$
$$\le\lim_{n\to\infty}\left|\frac{1}{n}\sum_{i=1}^{n}\frac{1}{2^i}\right|\le\lim_{n\to\infty}\frac{2}{n}=0$$

此时，命题 V.1 在 $[\![\mathsf{p}]\!]$ 上失效，因为 $\varepsilon([\![\mathsf{p}]\!]\circ[\![\mathsf{s}]\!])=\varepsilon(-1,-1,\ldots)=\delta_{-1}\neq\mathcal{P}([\![\mathsf{p}]\!])(\varepsilon([\![\mathsf{s}]\!]))=[\![\mathsf{p}]\!]_*(\delta_0)=\delta_1$。

现在，让我们为 $\mathbb{R}$ 赋予通过如例 III.1 中所述对 p 进行类型检查所给出的拓扑。这使得 $[\![\mathsf{p}]\!]$ 有界且连续，因此我们不再有 $\varepsilon([\![\mathsf{s}]\!])=\delta_0$；事实上，$\lim_n\int[\![\mathsf{p}]\!]\ d[\![\mathsf{s}]\!]_n=-1\neq[\![\mathsf{p}]\!](0)=1$。实际上，我们现在有 $\varepsilon([\![\mathsf{s}]\!])=\bot$，也就是说，对于该拓扑，s 不再是指向任何对象的采样器，这阻止了命题 V.1 在 $[\![\mathsf{p}]\!]$ 上失效。

这个例子还表明，与只考虑相对于类型指称上的通常拓扑连续的程序相比，我们的语义为我们提供了更多满足命题 V.1 的态射。我们的语义允许我们通过任意分段连续函数对采样器 s 进行推前，唯一例外是该函数存在一个不连续点，并且该不连续点被 s渐近地赋予正质量的狭窄情形。我们将在下一个例子中进一步说明这一点。

示例 V.4。考虑示例 V.3 中的采样器，但现在令 $\mathsf{p}\triangleq$ 若 $x=2$ 则为 1，否则为 -1。为了使这个函数连续，我们的语义在 $\mathbb{R}$的通常拓扑中加入开集 $\{2\}$。这不会影响推导 $\varepsilon([\![\mathsf{s}]\!])=\delta_0$，因为我们可以写作

$$\widehat{[\![\mathsf{s}]\!]}_n=\int_{\{2\}^c}f\ d\widehat{[\![\mathsf{s}]\!]}_n+\int_{\{2\}}f\ d\widehat{[\![\mathsf{s}]\!]}_n=\int_{\{2\}^c}f\ d\widehat{[\![\mathsf{s}]\!]}_n,$$

对于 δ_0也同理。由于 $[\![\mathsf{p}]\!]$ 的不连续性在 δ_0下的质量为零，使 $[\![\mathsf{p}]\!]$连续的 $\mathbb{R}$ 上的拓扑不再妨碍 $\varepsilon([\![\mathsf{s}]\!])$ 收敛，因此我们可以安全地使用映射将 s 经 p 向前推送。

B. 渐近目标计算

定义 V.2。我们称采样器 $\Gamma\vdash s:\Sigma\mathsf{S}$ 渐近目标于连续映射 $\mu:[\![\Gamma]\!]\to\mathcal{P}[\![\mathsf{S}]\!]$，或简称目标于该映射，如果对于每个 $\gamma\in[\![\Gamma]\!]$，

$$\varepsilon_{[\![\mathsf{S}]\!]}\circ[\![s]\!](\gamma)=\mu(\gamma).$$

特别是，$[\![s]\!](\gamma)_n$ 在 $n\to\infty$时总是收敛；发散采样器不指向任何对象。

我们称 $\Gamma\vdash s:\Sigma\mathsf{S}$ 是关于态射 $\mu:[\![\Gamma]\!]\to\mathcal{P}[\![\mathsf{S}]\!]$ 的 K-等分布，如果对于每个 $\gamma\in[\![\Gamma]\!]$，都有 $\varepsilon_{[\![\mathsf{S}]\!]}\circ[\![s^K]\!](\gamma)=\mu^K(\gamma)$；其中，自乘积 s^K定义于（3），而 $\mu^K(\gamma)\in\mathcal{P}([\![\mathsf{S}]\!]^K)$ 是测度 $\mu(\gamma)$ 与自身的 K 重乘积。

我们在表 V 中引入一个关于渐近目标而言可靠的关系；这正是 § II 的证明中所用的关系。也就是说，若 $\Gamma\vdash s:\Sigma\mathsf{S}$ μ，那么 s 是 S 上的一个参数化采样器，它渐近目标于 $[\![\mathsf{S}]\!]$ 上的参数化分布 μ。这里，我们用希腊小写字母 μ、ν 表示（参数化）分布，以强调它们作为元变量的作用：它们仅用于目标演算的上下文中，而不出现在语言本身中。在重新加权规则中，我们将对 X 上的测度 μ 按 $X\to\mathbb{R}_{\ge0}$：f 重新加权的操作简写为乘积 $(f\cdot\mu)(A)=\int_A f(x)\,d\mu(x)\int_X f(x)\,d\mu(x)$，假设相关积分有限且非零。

表 V 纳入了一条从头构建采样器的规则：将其定义为由确定性自映射 $t:\mathsf{T}\to\mathsf{T}$ 和初始值 $x:\mathsf{T}$ 通过 $\mathsf{prng}(t,x):\Sigma\mathsf{T}$.[5] 构成的伪随机数生成器。不过，表 V 还纳入了一组针对内置采样器 rand_i 的“公理”，每个采样器都指向分布 $\mu_i\in\mathcal{P}[\![\mathsf{T}_i]\!]$；在某些情况下，假定可以使用通过物理过程生成样本的“真正随机”采样器或许是合理的。

读者可能会疑惑，为什么表 V 没有关于 thin 操作的规则：毕竟，如果 σ 是一个指向分布 μ 的采样器，那么只保留每 n 个样本中的一个，也应该能生成一个良好的采样器。尽管对于“真正的”独立同分布采样器（例如物理采样器）生成的序列而言，这一说法以概率 1 成立，但一般来说，这条规则并不健全，下面的简单例子说明了这一点。

例 V.5。考虑定义在 $\{0,1\}$ 上的采样器，其程序为 $\mathsf{prng}(\lambda x:\mathsf{R}\ .\ 1-x, 0)$。该采样器生成未加权样本 (0, 1, 0, 1, . . .)，指向均匀伯努利分布；然而，对其应用 $\mathsf{thin}(2,-)$ 后，会得到一个指向狄拉克测度 δ_0 的采样器。

这个例子凸显了这样一个事实：采样器显然可以是非随机的，但从推断的角度来看——也就是说，从弱收敛拓扑的角度来看——它们指向的是正当的概率分布。

[5]应用此规则需要证明采样器的初始点是典型的；如果点 $x\in[\![\mathsf{T}]\!]$ 属于 μ-质量为 1 的子集 $X\subseteq[\![\mathsf{T}]\!]$，且遍历定理在其中成立，则称该点为典型点 [19, 定理 9.6]。

$$\frac{}{\Gamma_i \vdash \mathtt{rand}_i : \Sigma\,\mathtt{T}_i \rightsquigarrow \mu_i}\ i \in I \qquad \frac{\Gamma \vdash s \approx t : \Sigma\,\mathtt{T} \quad \Gamma \vdash s : \Sigma\,\mathtt{T} \rightsquigarrow \mu}{\Gamma \vdash t : \Sigma\,\mathtt{T} \rightsquigarrow \mu} \qquad \frac{\Gamma \vdash s : \Sigma\,\mathtt{S} \rightsquigarrow \mu}{\Gamma \vdash \mathtt{tl}(s) : \Sigma\,\mathtt{S} \rightsquigarrow \mu}$$

$$\frac{\Gamma \vdash s : \Sigma\,\mathtt{S} \rightsquigarrow \mu \quad \Gamma \vdash f : \mathtt{S} \to \mathtt{T}}{\Gamma \vdash \mathtt{map}(f, s) : \Sigma\,\mathtt{T} \rightsquigarrow \gamma \mapsto (\llbracket f \rrbracket(\gamma))_* \mu(\gamma)} \qquad \frac{\Gamma \vdash s : \Sigma\,\mathtt{S} \rightsquigarrow \mu \quad \Gamma \vdash f : \mathtt{S} \to \mathtt{R}^+}{\Gamma \vdash \mathtt{reweight}(f, s) : \Sigma\,\mathtt{S} \rightsquigarrow \gamma \mapsto \llbracket f \rrbracket(\gamma) \cdot \mu(\gamma)}\ \int_{\llbracket S \rrbracket} \llbracket f \rrbracket(\gamma)\, d\mu(\gamma) \in (0, \infty)$$

$$\frac{}{\Gamma \vdash \mathtt{prng}(f, x) \rightsquigarrow \mu}\ \llbracket f \rrbracket : \llbracket \mathtt{T} \rrbracket \to \llbracket \mathtt{T} \rrbracket \text{ ergodic w.r.t. } \mu, x \text{ typical}$$

TABLE V: Rules for asymptotic targeting

$$\frac{\dfrac{\vdash \mathtt{rand}^3 : \Sigma\,\mathtt{S} \rightsquigarrow U^3 \quad \vdash \mathtt{id_R} \times \mathtt{box} : \mathtt{S} \to \mathtt{T}}{\vdash \mathtt{map}(\mathtt{id_R} \times \mathtt{box}, \mathtt{rand}^3) : \Sigma\,\mathtt{T} \rightsquigarrow \llbracket \mathtt{id_R} \times \mathtt{box} \rrbracket_* (U^3) = U \otimes \mathrm{N}(0,1)} \quad \vdash \mathtt{accept} : \mathtt{T} \to \mathtt{R}^+}{\vdash \mathtt{reweight}(\mathtt{accept}, \mathtt{joint}) : \Sigma\,\mathtt{T} \rightsquigarrow \llbracket \mathtt{accept} \rrbracket \cdot (U \otimes \mathrm{N}(0,1)) \qquad \vdash \mathtt{produce} : \mathtt{T} \to \mathtt{R}}$$
$$\vdash \mathtt{map}(\mathtt{produce}, \mathtt{reweight}(\mathtt{accept}, \mathtt{joint})) : \Sigma\,\mathtt{R}^+ \rightsquigarrow \llbracket \mathtt{produce} \rrbracket_* (\llbracket \mathtt{accept} \rrbracket \cdot (U \otimes \mathrm{N}(0,1))) = \Gamma(\alpha, 1)$$

图 5：代码清单 4 中 Marsaglia 采样器的有效性

定理 V.1. *[附录 A]* 指向是可靠的：若 $\Gamma \vdash s : \Sigma\mathtt{S}\ \mu$，则 $\varepsilon_{\llbracket \mathtt{S} \rrbracket} \circ \llbracket s \rrbracket = \mu$。

容易看出，相应的自然完备性概念，即 $\varepsilon_{\llbracket \mathtt{S} \rrbracket} \circ \llbracket \mathtt{S} \rrbracket = \mu \to \Gamma \vdash s : \Sigma\mathtt{S}\ \mu$，并不成立；参见以下示例。

示例 V.6。令 s，prng(λn：实数集.n + 1, 0)；显然，s 不会指向任何（概率）测度。将变换映射应用于 s map(λn：实数集 . 2^(−1 ∗ n) s)。虽然 s 不指向任何对象，因此我们的目标演算无法证明变换后的采样器指向任何对象，但显然该采样器确实指向狄拉克测度 δ_0。

我们在 § V-A 中看到，我们的（子）类型系统可用于将采样器安全地推前通过仅分段连续的映射。我们的类型系统还允许我们向采样器添加额外约束。具体而言，我们可以确保采样器无穷多次访问某些子集。

命题 V.2. *[附录 A]* 假设 $\Gamma \vdash s : \Sigma\mathtt{S}\mu$，$\mathtt{S} \triangleleft \mathtt{T}$ 且 $\llbracket \mathtt{T} \rrbracket$ 第二可数；则 $\Gamma \vdash$ 映射($\lambda x : \mathtt{S}.\mathtt{cast}\langle \mathtt{T} \rangle x,\ s$)：$\Sigma\,\mathtt{T}$ 指向 T 上的同一测度 μ。此外，若 $\llbracket \mathtt{T} \rrbracket$ 可度量化，若 U 属于 $\llbracket \mathtt{S} \rrbracket$的拓扑但不属于 $\llbracket \mathtt{T} \rrbracket$，且 $\mu(\partial_{\mathtt{T}} U) > 0$（其中 $\partial_{\mathtt{T}}$ 表示 $\llbracket \mathtt{T} \rrbracket$中的边界），则 s 必须无穷多次访问 $\partial_{\mathtt{T}} U$。

示例 V.7. 假设我们希望 $s : \Sigma\mathtt{R}\ \mathrm{Bern}(1/2)$。一个在采样器 z、Ex. V.3 中的 prng($\lambda x : \mathrm{R} . x/2,\ 1$) 及其平移版本 map($\lambda x : \mathrm{R} . 1 + x,\ z$) 之间交替运行的采样器将满足该条件，但永远不会访问 0 或 1！我们可以利用前面的结果，通过构造 s 使其类型为 $\Sigma\big((x,\ 0)^{-1}\mathrm{Neq} + (x,\ 0)^{-1}\mathrm{Eq}\big)$（见 Ex. III.1），来强制一个 s 指向 Bern(1/2) 的采样器无穷多次访问 0。以同样的方式，我们可以强制一个s' 指向 Bern(1/2) 的采样器无穷多次访问 1。最后，利用图 4b 中构造两个拓扑的最粗公共细化的最后两条规则，我们可以组合 s 和 s'，创建一个指向 Bern(1/2) 且保证无穷多次访问 0、1 的采样器。

示例 V.8。最后，我们通过串联两种广为人知的采样算法，给出一个突出采样器组合性的示例。考虑以下程序：

```
令盒函数 = λu: 实数集+×实数集+ . sqrt(-2*log(第一投影(u))) *
    cos(2*pi*第二投影(u)) 属于
令联合分布 = 映射(恒等函子R×盒函数, 随机) 属于
令d = α - 1/3 属于
令c = 1/ (3*sqrt(d)) 属于
令接受函数 = λ(u, x) : T 。
  令 v = (1+c*x)^3 于
    如果 v > 0 且log(u) < x^2 + d - d*v + d*
    log(v) 则 1 否则 0于
令生成函数 = λz : T 。
  映射(proj, 重新加权(接受函数, 联合分布))
```

代码清单 4：用于伽马随机变量的 Marsaglia 采样器

首先，Box-Muller方法是一种利用两个独立均匀样本生成标准正态随机变量的知名方法；使用表 V 的映射规则对其进行验证十分直接。随后，我们构造一个由独立均匀样本和高斯样本组成的联合采样器，并消耗这两类样本，使用一种广为人知的拒绝采样方法生成形状参数为 $\alpha \geq 1$ 的伽马分布随机变量 $z \sim \Gamma(\alpha,1)$；证明见 *[28]*。图 5 概述了该采样器的有效性；为简洁起见，我们省略了一些类型。

VI. 讨论

我们提出了一种旨在以组合方式构造采样器的“概率”语言。我们为该语言给出了直观的操作语义，以及在 CG 空间范畴中的指称语义，并证明对于封闭采样器二者是等价的。这个指称宇宙足够丰富，能够余归纳地解释采样器类型，并解释在类型系统（第 III-C 节）给出的标准拓扑上仅分片连续的函数。

借助这一语言，我们展示了如何组合式地推理采样器构造的有效性，这是概率编程实践中的一个关键方面。我们的方法依托一个可靠的等式系统，用于推理构造同一采样器的等价方式（§ IV），以及一个用于推理语义正确性的可靠系统（§ V）。

我们的方法的独特之处在于，实际上为概率程序提供了纯确定性的语义。这种方法更贴近概率编程的实践，在该实践中，样本和采样器是最重要的具体实体；采样器与其所针对的测度之间的这种区分，对于支持伪随机数生成是必要的。测度论实体通常是概率语言的指称语义的一部分，例如 [21], [16], [12], [38], [9], ，在这里则承担验证标准的元理论角色。

我们的演算中缺少两种常用的采样器生成方案：马尔可夫链蒙特卡罗方法和重采样技术，例如粒子滤波器中所采用的技术。我们将把这些构造及其在目标演算中的相应正确性证明加入语言，视为未来的工作。

参考文献

[1] ANDRIEU, C., DOUCET, A., AND HOLENSTEIN, R. 粒子马尔可夫链蒙特卡罗方法。Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72, 3 (2010), 269–342. [2] BAGNALL, A., STEWART, G., AND BANERJEE, A. 对含循环和条件化的概率程序进行形式化验证的采样器，2023。[3] BILLINGSLEY, P. 概率测度的收敛性。John Wiley & Sons, 2013. [4] BINGHAM, E., CHEN, J. P., JANKOWIAK, M., OBERMEYER, F., PRAD- HAN, N., KARALETSOS, T., SINGH, R., SZERLIP, P. A., HORSFALL, P., AND GOODMAN, N. D. Pyro：深度通用概率编程。J. Mach. Learn. Res. 20 (2019), 28:1–28:6. [5] BLACKWELL, A., KOHN, T., ERWIG, M., BAYDIN, A. G., CHURCH, L., GEDDES, J., GORDON, A., GORINOVA, M., GRAM-HANSEN, B., LAWRENCE, N., MANSINGHKA, V., PAIGE, B., PETRICEK, T., ROBIN- SON, D., SARKAR, A., AND STRICKSON, O. 概率编程语言的易用性。载于 Psychology of Programming Interest Group Annual Workshop (PPIG 2019), Newcastle, UK, 28–30 August 2019 (2019). [6] BORGSTRÖM, J., DAL LAGO, U., GORDON, A. D., AND SZYMCZAK, M. 面向通用概率编程的 λ 演算基础。ACM SIGPLAN Notices 51, 9 (2016), 33–46. [7] BROOKS, S., GELMAN, A., JONES, G., AND MENG, X.-L. 马尔可夫链蒙特卡罗手册。CRC press, 2011. [8] CARPENTER, B., GELMAN, A., HOFFMAN, M. D., LEE, D., GOODRICH, B., BETANCOURT, M., BRUBAKER, M., GUO, J., LI, P., AND RIDDELL, A. Stan：一种概率编程语言。Journal of statistical software 76, 1 (2017). [9] DAHLQVIST, F., AND KOZEN, D. 带条件化的高阶概率程序语义。Proceedings of the ACM on Programming Languages 4, POPL (2019), 1–29. [10] DAHLQVIST, F., KOZEN, D., AND SILVA, A. 概率编程语义：简明导论。Cambridge University Press, 2020, pp. 1–42. [11] EHRHARD, T., PAGANI, M., AND TASSON, C. 概率相干空间的计算含义。载于 2011 IEEE 第 26 届年度计算机科学逻辑研讨会 (2011), IEEE, pp. 87–96. [12] EHRHARD, T., PAGANI, M., AND TASSON, C. 可测锥与稳定可测函数：概率高阶编程模型。Proceedings of the AC M on Programming Languages 2, POPL (2017), 1–28. [13] EHRHARD, T., PAGANI, M., AND TASSON, C. 概率 PCF 的完全抽象。Journal of the ACM (JACM) 65, 4 (2018), 1–44. [14] FAGGIAN, C., AND DELLA ROCCA, S. R. λ 演算与概率计算。载于 2019 年第 34 届 ACM/IEEE 计算机科学逻辑研讨会 (LICS) (2019), IEEE, pp. 1–13. [15] GOODMAN, N., MANSINGHKA, V., ROY, D. M., BONAWITZ, K., AND TENENBAUM, J. B. Church：一种生成模型语言。arXiv 预印本 arXiv:1206.3255 (2012). [16] HEUNEN, C., KAMMAR, O., STATON, S., AND YANG, H. 适用于高阶概率论的范畴。载于 2017 年第 32 届 ACM/IEEE 计算机科学逻辑研讨会 (LICS) (2017), IEEE, pp. 1–12. [17] HOYRUP, M., AND ROJAS, C. 度量空间上概率测度的可计算性与 Martin-Löf 随机性。Information and Computation 207, 7 (2009), 830–847. [18] HUANG, D., MORRISETT, G., AND SPITTERS, B. 可计算分布在概率程序语义中的应用。Cambridge University Press, 2020, p. 75–120. [19] KALLENBERG, O. 现代概率论基础。Springer, 1997. [20] KEREMEDIS, K., ÖZEL, C., PI .EKOSZ, A., SHUMRANI, M. A., AND WAJCH, E. 紧补拓扑与 k-空间。arXiv 预印本 arXiv:1806.10177 (2018). [21] KOZEN, D. 概率程序的语义。J. Comput. Syst. Sci. 22, 3 (1981 年 6 月), 328–350. [22] LAGO, U. D., AND ZORZI, M. λ 演算的概率操作语义。RAIRO-Theoretical Informatics and Applications- Informatique Théorique et Applications 46, 3 (2012), 413–450. [23] L’ECUYER, P. 均匀随机数生成的历史。载于 2017 Winter Simulation Conference (WSC) (2017), pp. 202–230. [24] LEOBACHER, G., AND PILLICHSHAMMER, F. 准蒙特卡罗积分及其应用导论。Springer, 2014.

[25] LEWIS, L. G. 稳定范畴与广义 Thom 谱。附录 A。博士论文，芝加哥大学数学系，1978。[26] LUNN, D., THOMAS, A., BEST, N., AND SPIEGELHALTER, D. Winbugs——贝叶斯建模框架：概念、结构与可扩展性。Statistics and Computing 10 (10 2000), 325–337. [27] MANSINGHKA, V., SELSAM, D., AND PEROV, Y. Venture：具有可编程推断功能的高阶概率编程平台。arXiv 电子预印本 (2014 年 3 月), arXiv:1404.0099. [28] MARSAGLIA, G., AND TSANG, W. W. 一种生成伽马变量的简单方法。ACM Trans. Math. Softw. 26, 3 (2000 年 9 月), 363–372. [29] MARTIN-LÖF, P. 随机序列的定义。Information and control 9, 6 (1966), 602–619. [30] MATSUMOTO, M., AND NISHIMURA, T. 梅森旋转算法：623 维均匀分布的均匀伪随机数生成器。ACM Trans. Model. Comput. Simul. 8, 1 (1998), 3–30. [31] MCCORD, M. C. 分类空间与无限对称积。Transactions of the American Mathematical Society 146 (1969), 273–298.[32] PARK, S., PFENNING, F., AND THRUN, S. 一种基于采样函数的概率语言。ACM SIGPLAN Notices 40, 1 (2005), 171–182.[33] ROBERT, C., AND CASELLA, G. 蒙特卡罗统计方法。Springer Science & Business Media, 2013. [34] STATON, S., WOOD, F., YANG, H., HEUNEN, C., AND KAMMAR, O. 概率编程语义：高阶函数、连续分布与软约束。载于第 31 届 ACM/IEEE 计算机科学逻辑研讨会论文集 (LICS) (2016), IEEE, pp. 1–10. [35] STEENROD, N. E. 一个方便的拓扑空间范畴。Michigan Mathematical Journal 14, 2 (1967), 133–152. [36] THOMAS, A. Bugs：一个统计建模软件包。RTA/BCS Modular Languages Newsletter 2 (1994), 36–38. [37] TREVISAN, L., AND VADHAN, S. 从可采样分布中提取随机性。载于第 41 届计算机科学基础研讨会论文集（美国，2000），FOCS '00，IEEE Computer Society, p. 32。[38] VÁKÁR, M., KAMMAR, O., AND STATON, S. 统计概率编程的域理论。Proceedings of the ACM on Programming Languages 3, POPL (2019), 1–29. [39] VIGNA, S. 对 Marsaglia 的 xorshift 生成器进行进一步扰动。Journal of Computational and Applied Mathematics 315 (2017), 175–181.[40] WOOD, F., VAN DE MEENT, J. W., AND MANSINGHKA, V. 一种概率编程推断的新方法。载于第 17 届人工智能与统计学国际会议论文集 (2014), pp. 1024–1032.

$$\frac{}{(v,N)\to v}\ v \text{ a value} \qquad \frac{(t,N)\to v}{(f(t),N)\to f(v)}\ \text{Func} \ni f:\mathtt{T}\to\mathtt{G}$$

$$\frac{((\lambda x:T.t)(s),N)\to v}{(\mathtt{let}\ x=s\ \mathtt{in}\ t,N)\to v} \qquad \frac{(t[x\leftarrow s],N)\to v}{((\lambda x:\mathtt{T}.t)(s),N)\to v}$$

$$\frac{(t,N)\to v}{(\mathtt{cast}\langle\mathtt{T}\rangle t,N)\to v} \qquad \frac{(c,N)\to j\in I \quad (s_j[x_j\leftarrow t],N)\to v}{(\mathtt{case}\ (c,t)\ \mathtt{of}\ \{(i,x_i)\Rightarrow s_i\}_{i\in I}),N)\to v} \qquad \frac{(t,N)\to v}{(\mathtt{in}_j\ (t)\ ,N)\to v}$$

$$\frac{(s,N)\to v_1 \quad (t,N)\to v_2}{((s,t),N)\to(v_1,v_2)} \qquad \frac{(t,N)\to(v_1,v_2)}{(\mathtt{fst}(t),N)\to v_1} \qquad \frac{(t,N)\to(v_1,v_2)}{(\mathtt{snd}(t),N)\to v_2}$$

$$\frac{((s(\mathtt{hd}(t)),\mathtt{wt}(t)),N)\to(v_1,w_1) \quad \ldots \quad ((s(\mathtt{hd}(\mathtt{tl}^{N-1}(t)),\mathtt{wt}(\mathtt{tl}^{N-1}(t))),N)\to(v_N,w_N)}{(\mathtt{map}(s,t),N)\to((v_1,w_1),\ldots,(v_N,w_N))}$$

$$\frac{((\mathtt{hd}(t),s(\mathtt{hd}(t))\cdot\mathtt{wt}(t)),N)\to(v_1,w_1) \quad \ldots \quad ((\mathtt{hd}(\mathtt{tl}^{N-1}(t)),s(\mathtt{hd}(\mathtt{tl}^{N-1}(t)))\cdot\mathtt{wt}(\mathtt{tl}^{N-1}(t))),N)\to(v_N,w_N)}{(\mathtt{reweight}(s,t),N)\to((v_1,w_1),\ldots,(v_N,w_N)}$$

$$\frac{(s,N)\to((v_1,w_1),\ldots,(v_N,w_N)) \quad (t,N)\to((v'_1,w'_1),\ldots,(v'_N,w'_N))}{(s\otimes t,N)\to(((v_1,v'_1),w_1\cdot w'_1),\ldots,((v_N,v'_N),w_N\cdot w'_N))}$$

$$\frac{(t,N)\to((v_1,w_1),\ldots,(v_N,w_N))}{(\mathtt{hd}(t),N)\to v_1} \qquad \frac{(t,N)\to((v_1,w_1),\ldots,(v_N,w_N))}{(\mathtt{tl}(t),N-1)\to((v_2,w_2),\ldots,(v_N,w_N))} \qquad \frac{(t,N)\to((v_1,w_1),\ldots,(v_N,w_N))}{(\mathtt{wt}(t),N)\to w_1}$$

$$\frac{(s,N)\to i \quad (t,Ni)\to((v_1,w_1),\ldots,(v_{Ni},w_{Ni}))}{(\mathtt{thin}(s,t),N)\to((v_1,w_1),(v_{i+1},w_{i+1}),(v_{2i+1},w_{2i+1}),\ldots,(v_{(Ni)+1},w_{(Ni)+1}))}$$

$$\frac{(t,N)\to v_1 \quad (s(t),N)\to v_2 \quad \ldots \quad (s^{N-1}(t),N)\to v_N}{(\mathtt{prng}(s,t),N)\to((v_1,1),\ldots,(v_N,1))}$$

TABLE VI: Full big-step operational semantics

附录

拒绝采样的类型检查（§ II-C）

我们首先对 § II-C 中的接受函数进行类型检查。这确定了此前未指定的类型 T。为使推导易于阅读，我们定义 t、(y、phi(x) $*$ sqrt(2 $*$ pi)。

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{x:\text{实数集}\ y:\text{实数集}\vdash x:\text{实数集}} \quad \vdash \text{phi}:\text{实数集}\to\text{实数集} \quad \vdash 2:\text{实数集}}{x:\text{实数集}\ y:\text{实数集}\vdash \text{phi}(x)*2*\text{pi}:\text{实数集}} \quad \vdash \text{pi}:\text{实数集}}{x:\text{实数集}\ y:\text{实数集}\vdash(y,\text{phi}(x)*\text{sqrt}(2*\text{pi})):\text{实数集}\times\text{实数集}}}{(x,y):\ t^{-1}(<^{-1}(0))+t^{-1}(<^{-1}(1))\vdash(y,\text{phi}(x)*\text{sqrt}(2*\text{pi})):<^{-1}(0)+<^{-1}(1)} \ <^{-1}(0)+<^{-1}(1)\triangleleft\text{实数集}\times\text{实数集}}{(x,y):\ t^{-1}(<^{-1}(0))+t^{-1}(<^{-1}(1))\vdash y<\text{phi}(x)*\text{sqrt}(2*\text{pi}):\ \mathrm{B}} \quad \vdash 0:\ \text{实数集}^+ \quad \vdash 1:\ \text{实数集}^+}{(x,y):\ t^{-1}(<^{-1}(0))+t^{-1}(<^{-1}(1))\vdash \text{若}\ y<\text{phi}(x)*\text{sqrt}(2*\text{pi})\ \text{则}\ 1\ \text{否则}\ 0:\text{实数集}^+}$$

$$\vdash\lambda(x,y):\ t^{-1}(<^{-1}(0))+t^{-1}(<^{-1}(1))\,.\ \text{若}\ y<\text{phi}(x)*\text{sqrt}(2*\text{pi})\ \text{则}\ 1\ \text{否则}\ 0:\ t^{-1}(<^{-1}(0))+t^{-1}(<^{-1}(1))\to\text{实数集}^+$$

图 6：§ II-C 中 accept 的推导

我们现在定义 T $= t^{-1}(<^{-1}(0))+t^{-1}(<^{-1}(1))$ 以及 proj $\lambda u:\mathrm{T}$。fst(cast$\langle\mathrm{R}\times\mathrm{R}\rangle u$)。

$$\dfrac{\dfrac{\dfrac{\dfrac{u:\mathrm{T}\vdash u:\mathrm{T}}{u:\mathrm{T}\vdash\text{cast}\langle\text{实数集}\times\text{实数集}\rangle u:\text{实数集}\times\text{实数集}}\ \mathrm{T}\triangleleft\text{实数集}\times\text{实数集}}{u:\mathrm{T}\vdash\text{第一投影}(\text{cast}\langle\text{实数集}\times\text{实数集}\rangle u):\text{实数集}\times\text{实数集}}}{\vdash\text{proj}:\mathrm{T}\to\text{实数集}} \quad \vdash\text{接受函数}:\mathrm{T}\to\text{实数集}^+}{\vdash\text{重新加权}(\text{接受函数},\text{tri}\otimes\text{rand})):\ \Sigma\,\mathrm{T}} \qquad \dfrac{\vdash\text{tri}:\ \Sigma\text{实数集} \quad \vdash\text{rand}:\ \Sigma\text{实数集}}{\vdash\text{tri}\otimes\text{rand}:\ \Sigma(\text{实数集}\times\text{实数集})}\ \Sigma\,\mathrm{T}\triangleleft\Sigma(\text{实数集}\times\text{实数集})\ \Gamma=\emptyset}{\vdash\text{映射}(\text{重新加权}(\text{接受函数},\text{tri}\otimes\text{rand}),\text{proj}):\ \Sigma\,\text{实数集}}$$

图 7：拒绝采样算法的推导 § II-C

命题 III.1 的证明。

为了通过对 $(t, N) \to v$ 的推导树进行归纳来证明这一结果，我们必须首先将其推广以包含高阶采样器。

命题 A.1。如果 $\vdash s : \Sigma^k\mathsf{S}$ 是一个闭合的 k 阶采样器，其中 S 不是采样器类型，并且 $k \in \{0, 1, 2, \ldots\}$，那么对于任意 $N \in \mathbb{N}$，如果 $(s, N) \to v$，则 v 具有一个由类型为 S 的值构成的 k 层嵌套加权列表的形式。例如，对于 $k = 0$，$v :$ S 就是一个类型为 S 的值；对于 $k = 1$，$v =$，$((v_1, w_1) \ldots, (v_N, w_N))$ 是一个值为 $v_n : \mathsf{S}$ 且具有 $w_n \geq 0$ 的加权列表；对于 $k = 2$，$v =$，$(((v_1^1, w_1^1) \ldots, (v_N^1, w_N^1))\ w_1) \ldots, (((v_1^N, w_1^N) \ldots, (v_N^N, w_N^N))\ w_N))$ 是一个由类型为 $v_n^n : \mathsf{S}$ 的值构成的加权列表的加权列表，依此类推。

基例。由于值 v 不可能具有采样器类型，因此对于某个类型 T，推导 $(v, N) \to v$ 其中 $v : \Sigma^k\mathsf{T}$ 的唯一可能性是 $k = 0$，这使得我们的结论立即成立。

归纳情形。我们根据推导 $(t, N) \to v$ 的最后一条规则，分别说明每种情形下的归纳论证，其中 $\vdash t : \Sigma^k\mathsf{T}$ 是某个 $k \in \{0, 1, 2, \ldots\}$ 的 k 阶采样器，并且 T 不是采样类型（即不包含 Σ 的出现）。

1) 内置函数。不存在输入或输出采样器类型的内置函数，因此 $k = 0$。根据归纳假设，每个输入 $s_i : \mathsf{G}_i$ 都会归约为一个值 $v_i : \mathsf{G}_i$，其中 G_n 是基类型，我们立即得到 $(f(s_1, \ldots, s_n)\ N) \to v$ 会求值为一个基类型 G 的值 v，从而得出我们的结论。

2) Case 分支。假设 $t = \mathsf{case}\ (c, t')\ \mathsf{of}\ \{(i, x_i) \Rightarrow s_i\}_{i \in n}$，则必有 $\vdash s_i : \Sigma^k\mathsf{T}$。根据归纳假设，$(s_i[x_i \leftarrow t'], N) \to v$ 求值为一个 k-嵌套加权列表；若 $(c, N) \to i \in n$，则立即可得 $(t, N) \to v$ 也同样如此。

3) 函数应用。令 $t = (\lambda x : \Sigma^{k'}\mathsf{S}.t')(s)$ 为函数应用的一个实例，其中所涉及的函数输入一个 k'-采样器并输出一个 k-采样器，而 S 本身不包含任何采样器类型；为了使该表达式类型正确，必须有 $\vdash s : \Sigma^{k'}\mathsf{S}$。为了使 $(t, N) \to v$ 求值为一个值，我们的操作语义要求 $(t'[x \leftarrow s], N) \to v$；根据归纳假设，$t'[x \leftarrow s]$ 求值为一个由 S 类型的值构成的 k-嵌套列表，因此即可得到所需结论。

4) let 绑定。由函数应用可直接得出，因为 $(\mathsf{let}\ x = s\ \mathsf{in}\ t', N) \to v$ 当且仅当 $((\lambda x : S.t')(s)\ N) \to v$。5) 乘积。若 $t = (s, s')$，则结果显然成立，因为 $\vdash t : \Sigma^k\mathsf{T}$ 蕴含 $k = 0$，因此 $\mathsf{T} = \mathsf{S} \times \mathsf{S}'$，其中 $\vdash s : \mathsf{S} \vdash s' : \mathsf{S}'$ 均不是采样器类型；所以，(s, s') 显然是类型 T 的一个值（即一个 0-嵌套加权列表）。

6) 投影。若 $t = \mathsf{fst}((s, s'))$，则 $\vdash s : \Sigma^k\mathsf{T}$，因此归纳假设 $(s, N) \to v$ 立即推出我们的结果；同样的论证也适用于 snd 和 t。

7) 头部。若 $t = \mathsf{hd}(s)$，则 $\vdash s : \Sigma^{k+1}\mathsf{T}$。根据归纳假设 $(s, N) \to v$ 蕴含 v 是一个由类型 T 的值构成的 $(k+1)$-嵌套加权列表，我们只需注意到该列表的第一个元素本身就是一个由类型 T 的值构成的 k-嵌套加权列表。

8) **Weig**ht. If $t = \mathsf{wt}\ (s)$, our result is triviall y true, as the out put of $\mathsf{wt}\ (s)$ can onl y be a nonne g实数 $\vdash t : R^+$。

9）尾部。如果 $t = \mathsf{tl}(s)$，那么根据我们的归纳假设，$(s, N + 1) \to ((v_1, w_1)\ldots\ldots, (v_{N+1}, w_{N+1}))$，其中每个 v_n 都是一个由类型 T 的元素组成的（$k-1$）层嵌套加权列表。由此立即可知，$(\mathsf{tl}(s)\ N)$ 求值为一个包含 N 个元素的加权列表，其中每个元素都是类型 T 的（$k-1$）层嵌套列表。

10）thin。如果 $t = \mathsf{thin}(i, s)$，那么 $\vdash n : \mathbb{N}$ 且 $\vdash s : \Sigma^k\mathsf{T}$；根据我们的归纳假设，$(s, Ni) \to ((v_1, w_1)\ldots\ldots, (v_{Ni+1}, w_{Ni+1}))$，其中每个 v_n 都是一个由类型 T 的元素组成的（$k-1$）层嵌套加权列表。由此立即可知，$(\mathsf{thin}(i, s)\ N)$ 求值为一个包含 N 个元素的加权列表，其中每个元素都是类型 T 的（$k-1$）层嵌套列表。

11) 映射。如果 $t = \mathsf{map}(s, t')$ 和 $(t, N) \to v$，map 的操作语义要求

$$\left((s(\mathtt{hd}(\mathtt{tl}^{n-1}(t'))), \mathtt{wt}(\mathtt{tl}^{n-1}(t'))), N\right) \to (v_n, w_n)$$

对每个 $n \in \{1, \ldots, N\}$ 都成立。由于 t' 是 t 的子项，根据归纳假设，如果对某个 $k' \in \{1, 2, \ldots\}$，$\vdash t' : \Sigma^{k'}\mathsf{S}$，那么对于任意 $N \in \mathbb{N}$，t' 会求值为由类型 S 的值构成的 k' 层嵌套加权列表。注意，要使 t 类型正确，必须有 $\vdash s : \Sigma^{k'-1}\mathsf{S} \to \Sigma^{k-1}\mathsf{T}$。我们已经证明，在这种情况下，$\mathsf{tl}^{n-1}(t')$ 会求值为由类型 S 的值构成的 k' 层嵌套加权列表；接着，$\mathsf{hd}(\mathsf{tl}^{n-1}(t'))$ 会求值为由类型 S 的值构成、长度为 N 的 $(k'-1)$ 层嵌套加权列表；然后，$s(\mathsf{hd}(\mathsf{tl}^{n-1}(t')))$ 会求值为由类型 T 的值构成、长度为 N 的 $(k-1)$ 层嵌套加权列表。注意到，对于每个 $n \in \{1, \ldots, N\}$，如果其中每个 v_n 都是由类型 T 的值构成的 $k-1$ 层嵌套加权列表，且 $((s(\mathsf{hd}(\mathsf{tl}^{n-1}(t'))), \mathsf{wt}(\mathsf{tl}^{n-1}(t')))\ N) \to (v_n, w_n)$，那么表达式 $((v_1, w_1) \ldots, (v_N, w_N))$ 就是由类型 T 的值构成的 k 层嵌套加权列表，证明完毕。

12）重新加权。该证明与映射的证明完全相同。

13）采样器乘积。如果 $t = s \otimes s'$ 和 $\vdash t : \Sigma^k\mathsf{T}$，其中 T 不包含 Σ 的任何实例，那么必然有 $k \geq 1$，有 $\vdash s : \Sigma^k\mathsf{S}$，并且有 $\vdash s' : \Sigma^k\mathsf{S}'$。我们的归纳假设表明，$(s, N) \to ((v_1, w_1) \ldots, (v_N, w_N))$，其中

每个 v_1 都是一个由 S 类型的值构成的 $(k-1)$-嵌套加权列表，并且 $(s', N) \to ((v'_1, w'_1) \ldots, (v'_N, w'_N))$，其中每个 v'_1 都是一个由 S′ 类型的值构成的 $(k-1)$-嵌套加权列表。我们的结果随即由以下事实得出：乘积 $(((v_1, v'_1)\ w_1\ w'_1) \ldots, ((v_N, v'_N)\ w'_N))$ 是一个由 S 类型的值构成的 k-嵌套加权列表。

14）伪随机数生成器。最后，假设 $t = \mathsf{prng}(s, t')$；为了使该表达式类型正确，我们必须有 $k \geq 1, \vdash t' : \Sigma^{k-1}\mathsf{T}$，以及 $\vdash s : \Sigma^{k-1}\mathsf{T} \to \Sigma^{k-1}\mathsf{T}$。为了使 (t, N) 能够求值，我们必须对每个 $n \in \{2, \ldots, N\}$ 都有 $(s^{n-1}(t)\ N) \to v_n$；由于我们已经证明了函数抽象的情形，因此知道每个 v_n 都是一个由 $k-1$ 个值为类型 T 的加权列表嵌套而成的列表。于是我们只需注意到，$((v_1, 1) \ldots, (v_N, 1))$ 显然是一个由 k 个值为类型 T 的加权列表嵌套而成的列表。

□

命题 III.3 的证明。

设 $f_n \to f$ 是 $[X, FX]$ 中的一个余代数映射收敛序列；我们需要证明 $\mathsf{beh}_X(f_n) \to \mathsf{beh}_X(f)$ 在 $[X, \nu F]$ 中成立。$[X, \nu F]$ 上的拓扑是紧开拓扑，这意味着它由形如以下形式的开集子基生成

$$(K, V) \triangleq \{h : X \to \nu F \mid h[K] \subset U\}$$

对于某个固定的紧集 $K \subseteq X$ 和开集 $U \subseteq \nu F$。此外，根据 νF 的构造（见定理 III.1），我们知道，$\prod_i F^i 1$上的拓扑由积拓扑诱导。该拓扑的一个基由柱集与 νF的交集构成。由于我们还假设 $\mathsf{int}(\nu F) \neq \emptyset$于 $\prod_i F^i 1$中，它包含这样一个开集，因此我们可以直接从 $\mathsf{beh}_X f$的一个形如 $(K, \prod_i V_i)$ 的开邻域开始，其中除了有限多个指标 $V_i = F^i 1$之外，对于其他指标，V_i是 $F^i 1$的一个开子集（而且我们不必担心与 νF取交集）。给定这样一个开集，我们需要找到 $N \in \mathbb{N}$，使得对于所有 $n > N$ $\mathsf{beh}_X(f_n) \in (K, \prod_i V_i)$。

根据定理 III.1 的构造，我们有

$$\mathrm{beh}_X(f)(x) = (!_X(x), F!_X(f(x)), F^2!_X(Ff(f(x))), \ldots)$$

其中 $!_X :\ X \to 1$ 是到终对象的唯一态射。对于有限多个非平凡开子集中的每一个$V_{i_k} \subset F^{i_k}1 1 \leq k \leq M$，由于 $f_n \to f$，且在 CG ([35, 5.9]) 中，与连续函数复合是内部同态集上的连续操作，因此可推出存在 N_k，使得对于每个 $n > N_k$

$$F^{i_k}!_X \circ F^{i_k - 1} f_n \circ \ldots \circ f_n \in (K, V_{i_k})$$

取 $N =$ 最大值$_{1 \leq k \leq M} N_k$，可得对于所有 $i \in \mathbb{N}$ 以及所有 $n > N$

$$F^i!_X \circ F^{i-1} f_n \circ \ldots \circ f_n \in (K, V_i)$$

换言之，对于任意 $n > N$，$\mathsf{beh}_X(f_n) \in (K, \prod_i V_i)$，从而证明完毕。

□

定理 III.2 的证明。

⇐) 对 $(t, N) \to v$的推导树进行归纳。

基例。基例显然成立：表 II 允许的唯一长度为 0 的推导假设 $t = v$ 是一个值。容易验证，如果 $t : \mathsf{T}$ 是一个值，那么 $p^N_{\mathsf{T}} = \mathsf{id}_{[\![\mathsf{T}]\!]}$，因此 $p^N_{\mathsf{T}}([\![t]\!]) = [\![t]\!] = [\![v]\!]$ 显然成立。

归纳情形。假设 $(t, N) \to v$ 的推导的最后一条规则是

(i) 内置函数。$t = f(s_1, \ldots, s_n)$，其中某个 $s_i : \mathsf{G}_i, 1 \leq i \leq n$。由于对于基类型 G，有 $p^N_{\mathsf{G}} = \mathsf{id}_{[\![\mathsf{G}]\!]}$，我们立即得到

$$\begin{aligned} p^N_{\mathsf{G}}([\![f(s_1, \ldots, s_n)]\!]) &\triangleq [\![f]\!]([\![s_1]\!], \ldots [\![s_n]\!]) \\ &= [\![f]\!]([\![v_1]\!], \ldots, [\![v_n]\!]) \qquad \text{归纳假设} \end{aligned}$$

（ii）Case 分支。$t =$对（c，t'）进行 {Case 分支（i，x_i）$\Rightarrow s_i\}_{i \in I}$。如果 c选择分支 $j \in n$，那么

$$\begin{aligned} p^N_{\mathsf{T}}\left([\![\mathtt{case}\ (c, t')\ \mathtt{of}\ \{(i, x_i) \Rightarrow s_i\}_{i \in I}]\!]\right) &\triangleq [\![s_j]\!]([\![t']\!]) \\ &= [\![v]\!] \qquad \text{归纳假设} \end{aligned}$$

（iii） λ-抽象。 $t = (\lambda x : \mathtt{S}.\ t')\ (s) : \mathtt{T}$，其中 $s : \mathtt{S}$，且 $t' : \mathtt{T}$。

$$
\begin{aligned}
p^N_{\mathtt{T}}\left([\![(\lambda x : \mathtt{T}.\ t')(s)]\!]\right) &\triangleq p^N_{\mathtt{T}} \circ \mathrm{ev}_{[\![\mathtt{S}]\!],[\![\mathtt{T}]\!]}\left([\![\lambda x : \mathtt{T}.\ t']\!] \times [\![s]\!]\right) \\
&\triangleq p^N_{\mathtt{T}} \circ \mathrm{ev}_{[\![\mathtt{S}]\!],[\![\mathtt{T}]\!]}\left(\widehat{[\![t']\!]} \times [\![s]\!]\right) && \text{Currying } [\![t']\!] \\
&= p^N_{\mathtt{T}} \circ \mathrm{ev}_{[\![\mathtt{S}]\!],[\![\mathtt{T}]\!]}\left([\![t']\!] \times [\![s]\!]\right) && [\![t']\!] \text{ 只有一个变量} \\
&= p^N_{\mathtt{T}}\left([\![t']\!]\left([\![s]\!]\right)\right) \\
&= [\![v]\!] && \text{归纳假设}
\end{aligned}
$$

(iv) let 绑定。 $t = \mathtt{let}\ x = s\ \mathtt{in}\ t'$：对于某个 $s : \mathtt{S}$ 和 $t' : \mathtt{T}$。

$$
\begin{aligned}
p^N_{\mathtt{T}}\left([\![\mathtt{let}\ x = s\ \mathtt{in}\ t]\!]\right) &\triangleq [\![t']\!]\left([\![s]\!]\right) \\
&= p^N_{\mathtt{T}}\left([\![\lambda x.\ t']\!]\left([\![s]\!]\right)\right) \\
&= [\![v]\!] && \text{归纳假设}
\end{aligned}
$$

(v) 乘积。 $t = (s, s')$，其中某个 $s : \mathtt{S}\ \ s' : \mathtt{S}'$

$$
\begin{aligned}
p^N_{\mathtt{S}\times\mathtt{S}'}([\![(s, s')]\!]) &\triangleq p^N_{\mathtt{S}\times\mathtt{S}'}\left(\langle [\![s]\!], [\![s']\!]\rangle\right) \\
&= \langle p^N_{\mathtt{S}}([\![s]\!]), p^N_{\mathtt{S}'}([\![s']\!])\rangle && p^N_{\mathtt{T}} \text{的归纳定义} \\
&= \langle [\![v_1]\!], [\![v_2]\!]\rangle && \text{归纳假设} \\
&= [\![(v_1, v_2)]\!]
\end{aligned}
$$

（vi）对于某个 $s : \mathtt{S}\ \ s' : \mathtt{S}'$，投影 $t = \mathrm{fst}(s, s')$

$$
\begin{aligned}
p^N_{\mathtt{S}}\left([\![\mathtt{fst}(s, s')]\!]\right) &\triangleq p^N_{\mathtt{S}}\left(\pi_1 \langle [\![s]\!], [\![s']\!]\rangle\right) \\
&= p^N_{\mathtt{S}}([\![s]\!]) \\
&= [\![v_1]\!] && \text{归纳假设}
\end{aligned}
$$

snd 也同理。

（vii）推前。对于某个 $s : \mathtt{S} \to \mathtt{T}$ 以及 $t : \Sigma\,\mathtt{S}$，有 $t = \mathrm{map}(s, t)$。为使推导清晰，我们将使用 s 代替 $[\![s]\!]$，使用 t 代替 $[\![t]\!]$，并引入以下记号。令 F 表示函子 $[\![\mathrm{T}]\!] \times \mathbb{R}^+ \times \mathrm{Id}$，令 $\gamma : \nu F \to F\nu F$ 表示终余代数结构映射 $\mathrm{unfold}_{\mathtt{T}}$，令 δ、$[\![s]\!] \times \mathrm{id}_{\mathbb{R}^+} \times \mathrm{id}_{\Sigma\,\mathtt{S}} \circ \mathrm{unfold}_{\mathtt{S}}$ 表示定义映射运算的余代数结构映射，令 $b = \mathrm{beh}(\delta)$，并令

$$
\begin{aligned}
h &\triangleq \pi_1 \circ \mathrm{unfold}_{\mathtt{S}} \\
w &\triangleq \pi_2 \circ \mathrm{unfold}_{\mathtt{S}} \\
f &\triangleq \pi_3 \circ \mathrm{unfold}_{\mathtt{S}}
\end{aligned}
$$

即，$h(t)$ 是 t 的第一个样本；即，$w(t)$ 是 t 的第一个样本的权重；即，$f(t)$ 是 t 的尾部

由此，我们现在可以推导出

$$
\begin{aligned}
&p^N_{\Sigma\,\mathtt{T}}([\![\mathtt{map}(s, t)]\!]) \\
&\triangleq \pi_{1:N} \circ \left(p^N_{\mathtt{T}\times\mathrm{R}^+}\right)^{\omega}\left([\![\mathtt{map}(s, t)]\!]\right) \\
&\triangleq \pi_{1:N} \circ \left(p^N_{\mathtt{T}\times\mathrm{R}^+}\right)^{\omega}(b(t)) \\
&\overset{(1)}{=} \left(p^N_{\mathtt{T}\times\mathrm{R}^+}\right)^{N} \circ \pi_{1:N}(b(t)) \\
&\overset{(2)}{=} \left(p^N_{\mathtt{T}\times\mathrm{R}^+}\right)^{N} \circ F\pi_{1:N-1} \circ \gamma \circ (b(t)) \\
&\overset{(3)}{=} \left(p^N_{\mathtt{T}\times\mathrm{R}^+}\right)^{N} \circ F^{N-1}\pi_1 \circ F^{N-2}\gamma \circ \ldots \circ F^0\gamma(b(t)) \\
&\overset{(4)}{=} \left(p^N_{\mathtt{T}\times\mathrm{R}^+}\right)^{N} \circ F^{N-1}\pi_1 \circ F^{N-1}b \circ F^{N-2}\delta \circ \ldots \circ F^0\delta(t) \\
&\overset{(5)}{=} \left(p^N_{\mathtt{T}\times\mathrm{R}^+}\right)^{N} \circ F^{N-1}\pi_1 \circ F^{N-1}b\left((s(h(t)), w(t)), \ldots, (s(h(f^{N-1}(t))), w(f^{N-1}(t))), f^{N-1}(t)\right) \\
&\overset{(6)}{=} \left(p^N_{\mathtt{T}\times\mathrm{R}^+}\right)^{N} \circ \left((s(h(t))), w(t)), \ldots, (s(h(f^{N-1}(t))), w(f^{N-1}(t)))\right) \\
&= \left(p^N_{\mathtt{T}\times\mathrm{R}^+}(s(h(t))), w(t)), \ldots, p^N_{\mathtt{T}\times\mathrm{R}^+}(s(h(f^{N-1}(t))), w(f^{N-1}(t)))\right) \\
&= \left((p^N_{\mathtt{T}}(s(h(t))), w(t)), \ldots, (p^N_{\mathtt{T}}(s(h(f^{N-1}(t)))), w(f^{N-1}(t)))\right) \\
&\overset{(7)}{=} [\![((v_1, w_1), \ldots, (v_N, w_N))]\!]
\end{aligned}
$$

其中，(1) 是对 $\pi_{1:N} \circ (p_{\mathtt{T}}^N)^\omega = (p_{\mathtt{T}}^N)^N \circ \pi_{1:N}$ 的简单观察，(2) 根据 γ 的定义得出，(3) 是通过迭代 (2) 得出，(4) 源于 b 是一个余代数态射这一事实，(5) 根据 δ 的定义得出，(6) 根据 F、 b 和 $p_{\mathtt{T}}^1$ 的定义得出，而 (7) 则依据该规则的 N 前提上的归纳假设得出。

（viii）重新加权。证明与映射的 Case 分支非常相似。同样地，记

$$\delta \triangleq \mathrm{id}_{[\![\mathtt{T}]\!]} \times (- \cdot -) \times \mathrm{id}_{[\![\Sigma\,\mathtt{T}]\!]} \circ \langle \mathrm{id}_{[\![\mathtt{T}]\!]}, [\![s]\!] \rangle \times \mathrm{id}_{\mathbb{R}^+} \times \mathrm{id}_{[\![\Sigma\,\mathtt{T}]\!]} \circ \gamma$$

定义重新加权和 $b =$beh(δ) 的余代数结构，可得

$$\begin{aligned}
&p_{\Sigma\,\mathtt{T}}^N(\mathtt{reweight}(s,t)) \\
&\triangleq \pi_{1:N} \circ \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^\omega (\mathtt{reweight}(s,t)) \\
&\triangleq \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^N \circ \pi_{1:N} \circ b(t) \\
&\overset{(1)}{=} \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^N \circ F^{N-1}\pi_1 \circ F^{N-1} b\left((h(t), s(h(t))w(t)), \ldots, (h(f^{N-1}(t)), s(f^{N-1}(t))w(f^{N-1}(t))), f^{N-1}(t)\right) \\
&\overset{(2)}{=} \left((p_{\mathtt{T}}^N(h(t)), s(h(t))w(t)), \ldots, (p_{\mathtt{T}}^N(h(f^{N-1}(t))), s(f^{N-1}(t))w(f^{N-1}(t)))\right) \\
&\overset{(3)}{=} ((\![v_1]\!], [\![w_1]\!]), \ldots, ([\![v_N]\!], [\![w_N]\!]))
\end{aligned}$$

其中，(1) 遵循与映射的 Case 分支中相同的推导，但使用了上述 δ 的定义；(2) 根据 F 和 $p_{\mathtt{T}\times\mathbb{R}^+}^N$ 的定义成立；(3) 则是将归纳假设应用于重新加权规则的 N 前提所得。

（ix）采样器乘积。证明与映射和重新加权的证明完全相同。

（x）thin。证明与映射和重新加权的证明完全相同。

（xi）伪随机数生成器。考虑项 prng(s,t)： $\Sigma\,\mathtt{T}$。使用

$$\delta \triangleq \langle \mathrm{id}_{[\![\mathtt{T}]\!]}, 1, [\![s]\!] \rangle$$

以及 $b =$ beh(δ)，按照 map 和重新加权 的情况所采用的相同步骤可得

$$\begin{aligned}
p_{\Sigma\,\mathtt{T}}^N([\![\mathtt{prng}(s,t)]\!]) &\triangleq \pi_{1:N} \circ (p_{\mathtt{T}\times\mathbb{R}^+}^N)^\omega([\![\mathtt{prng}(s,t]\!]) \\
&= \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^N \circ F^{N-1}\pi_1 \circ F^{N-1} b((t,1),(s(t),1),\ldots,s^{N-1}(t),1),s^N(t)) \\
&= ((p_{\mathtt{T}}^N(t),1),(p_{\mathtt{T}}^N(s(t)),1),\ldots,(p_{\mathtt{T}}^N(s^{N-1}(t)),1)) \\
&= [\![(v_1,1),(v_2,1),\ldots,(v_N,1))]\!]
\end{aligned}$$

(xii) 头部。对于某个 $t:\ \Sigma\mathsf{T}$，考虑项 hd(t)。使用上述相同的记号

$$\begin{aligned}
p_{\mathtt{T}}^N \circ [\![\mathtt{hd}(t)]\!] &\triangleq p_{\mathtt{T}}^N \circ \pi_1 \circ \gamma([\![t]\!]) \\
&= \pi_1 \circ \pi_1 \circ \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^N \circ \pi_{1:N}([\![t]\!]) \\
&= \pi_1 \circ \pi_1\, [\![(v_1,w_1),\ldots,(v_N,w_N)]\!] && \text{归纳假设} \\
&= [\![v_1]\!]
\end{aligned}$$

(xiii) 权重。对于某个 $t:\ \Sigma\mathsf{T}$，考虑项 wt(t)。证明与上述 :

$$\begin{aligned}
p_{\mathtt{T}}^N \circ [\![\mathtt{wt}(t)]\!] &\triangleq p_{\mathtt{T}}^N \circ \pi_2 \circ \gamma([\![t]\!]) \\
&= \pi_2 \circ \pi_1 \circ \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^N \circ \pi_{1:N}([\![t]\!]) \\
&= \pi_2 \circ \pi_1\, [\![(v_1,w_1),\ldots,(v_N,w_N)]\!] && \text{归纳假设} \\
&= [\![w_1]\!]
\end{aligned}$$

(xiv) 尾部。对于某个 $t:\ \Sigma\mathsf{T}$，考虑项 tl(t) f。显然

$$\begin{aligned}
p_{\Sigma\,\mathtt{T}}^N \circ [\![\mathtt{tl}(t)]\!] &\triangleq \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^N \circ \pi_{1:N}(\pi_3 \circ \gamma([\![t]\!])) \\
&= \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^N \circ \pi_{2:N+1}([\![t]\!]) \\
&= \pi_{2:N+1} \circ \left(p_{\mathtt{T}\times\mathbb{R}^+}^N\right)^{N+1} \circ \pi_{1:N+1}([\![t]\!]) \\
&= \pi_{2:N+1}\, [\![(v_1,w_1),\ldots,(v_{N+1},w_{N+1})]\!] && \text{归纳假设} \\
&= [\![(v_2,w_2),\ldots,(v_{N+1},w_{N+1})]\!]
\end{aligned}$$

⇒) 根据类型归纳 t的证明。注意，对于任意项 $t:\mathsf{T}$，根据 $p_{\mathtt{T}}^N$ 的定义， $p_{\mathtt{T}}^N\,[\![t]\!]$ 必然是一个值。

基例。唯一能够在 0 步内进行类型检查的程序是常量。由于所有常量都是值，且值在操作语义下会求值为自身，因此基例显然成立。

归纳情形。证明是常规的，我们只展示几个情形。假设类型检查 t 时所应用规则的最后一步是

（i）乘积。假设 $\vdash (s,t) : \mathtt{S} \times \mathtt{T}$，并且对于某个值 v，有 $p^N_{\mathtt{S}\times\mathtt{T}}(\llbracket (s,t) \rrbracket) = \llbracket v \rrbracket$。由于最后应用的规则的前提为 $\vdash s : \mathtt{S}$ 和 $\vdash t : \mathtt{T}$，我们有

$$\begin{aligned}
\llbracket v \rrbracket &= p^N_{\mathtt{S}\times\mathtt{T}}(\llbracket s \otimes t \rrbracket) \\
&= p^N_{\mathtt{S}} \times p^N_{\mathtt{T}} \langle \llbracket s \rrbracket, \llbracket t \rrbracket \rangle && p^N_{\mathtt{S}\times\mathtt{T}} \text{ 的归纳定义} \\
&= (p^N_{\mathtt{S}}\ \llbracket s \rrbracket, p^N_{\mathtt{T}}\ \llbracket t \rrbracket) \\
&= (\llbracket v_1 \rrbracket, \llbracket v_2 \rrbracket)
\end{aligned}$$

对于某些值 v_1、v_2。因此，根据归纳假设，对于任意 $N \in \mathbb{N}$，有 $(s,N) \to v_1$ 且 $(t,N) \to v_2$，从而根据归约关系 $\to$ 的定义可知 $(s \otimes t, N) \to (v_1, v_2)$。

（ii） λ-抽象。如果 $\vdash \lambda x : \mathtt{S}.\ t : \mathtt{S} \to \mathtt{T}$，那么项 $\lambda x : \mathtt{S}.\ t$ 是一个值，因此 $(\lambda x :\ \mathtt{S}.\ t,\ \ N) \to \lambda x :\ \mathtt{S}.\ t$ 显然成立。

（iii）头部。假设 $\vdash \mathsf{hd}(t) : \mathtt{T}$，并且 $p^N_{\mathtt{T}}(\llbracket \mathsf{hd}(t) \rrbracket) = \llbracket v_1 \rrbracket$ 对于某个值 v_1成立。由于最后应用的规则具有前提 $\vdash t :\ \Sigma\,\mathtt{T}$，并根据 hd 的语义，对于任意 $N \geq 1$，$\left(p^N_{\mathtt{T}\times\mathtt{R}^+}\right)^N \circ \pi_{1:N}\ t = \llbracket\ v_1\ w_1\ \ldots\ v_N\ w_N\ \rrbracket$ $v_i\ w_i$ ( ) ((,) , (,)), 必然存在某些值 , 。根据归纳假设，必然有 $(t,N) \to ((v_1,w_1)\ldots,(v_N,w_N))$，因此 $(\mathsf{hd}(t)\ N) \to v_1$。

（iv）权重。该证明与 hd 的证明相同。假设 $\vdash \mathsf{wt}(t) : \mathtt{T}$，并且 $p^N_{\mathtt{T}}(\llbracket \mathsf{wt}(t) \rrbracket) = \llbracket w_1 \rrbracket$ 对于某个权重$w_1 \geq 0$成立。由于最后应用的规则具有前提 $\vdash t :\ \Sigma\,\mathtt{T}$，并根据 wt 的语义，对于任意 $N \geq 1$, $\left(p^N_{\mathtt{T}\times\mathtt{R}^+}\right)^N \circ \pi_{1:N}(t) = \llbracket ((v_1,w_1)\ldots,(v_N, w_N)) \rrbracket$，必然存在某些值 v_i, w_i。根据归纳假设，必然有 $(t,N) \to ((v_1,w_1)\ldots,(v_N,w_N))$，因此 $(\mathsf{wt}(t)\ N) \to w_1$。

(v) 推前。假设 $\vdash \mathsf{map}(t,s) :\ \Sigma\,\mathtt{T}$，且 $p^N_{\Sigma\mathtt{T}}(\llbracket \mathsf{map}(t,s) \rrbracket) = \llbracket ((v_1,w_1)\ldots,(v_n,w_n)) \rrbracket$。最后应用的规则的前提必定是 $\vdash s :\ \Sigma\mathtt{S}$ 和 $\vdash t : \mathtt{S} \to \mathtt{T}$。根据 map 的语义可得 $\llbracket v_i \rrbracket = p^N_{\mathtt{T}}\ \llbracket t\ \ {}^{i-1}\ s\ \ \rrbracket$ $\llbracket w_i \rrbracket = \llbracket\ \ {}^{i-1}\ s\ \ \rrbracket$ $t\ \ {}^{i-1}\ s\ \ N \to (\mathsf{hd}(\mathsf{tl}\,(\,)))$ $v_i\ \ wt\ {}^{i-1}\ s\ \ N \to w_i$ $\to$ $t\ s\ \ N \to\ \ v_1\ w_1\ \ldots\ \ v_n\ w_n$) and wt(tl ( ))。根据归纳假设可得 ((hd(tl ( ))) and ((tl ( )))，因此根据 的定义，我们有 (map(,)) ((,) , (,)。

□

定理 IV.1 的证明。

标准规则：

1) β-和 η-等价性。

$$\begin{aligned}
\llbracket \Gamma \vdash (\lambda x : \mathtt{S}.t)(s) : \mathtt{T} \rrbracket &= \llbracket \Gamma \vdash t[x \leftarrow s] : T \rrbracket, \\
\llbracket \Gamma \vdash \lambda x : \mathtt{S}.t(x) : \mathtt{S} \to \mathtt{T} \rrbracket &= \llbracket \Gamma \vdash t : \mathtt{S} \to \mathtt{T} \rrbracket
\end{aligned}$$

β-和 η-等价性的可靠性是众所周知的，并可由指数对象的性质直接得出。2) let 归约。

$$\llbracket \Gamma, s : \mathtt{S} \vdash \mathtt{let}\ x = s\ \mathtt{in}\ t : \mathtt{T} \rrbracket = \llbracket \Gamma, s : \mathtt{S} \vdash (\lambda x : \mathtt{S}.t)(s) : \mathtt{T} \rrbracket$$

根据 let 的指称语义定义，这显然成立。

3) 投影。

$$\begin{aligned}
\llbracket \Gamma \vdash \mathtt{fst}((s,t)) : \mathtt{S} \rrbracket &= \llbracket \Gamma \vdash s : \mathtt{S} \rrbracket, \\
\llbracket \Gamma \vdash \mathtt{snd}((s,t)) : \mathtt{T} \rrbracket &= \llbracket \Gamma \vdash t : \mathtt{T} \rrbracket
\end{aligned}$$

这可由笛卡尔积的性质直接得出。

同余规则：在指称语义中显然成立：如果 $\llbracket \Gamma \vdash s : \mathtt{S} \rrbracket = \llbracket \Gamma \vdash s' : \mathtt{S} \rrbracket$ 具有相同的语义，那么对于任意内置操作 $\mathsf{op} : \mathtt{S} \to \mathtt{T}$，显然有 $\llbracket \Gamma \vdash \mathsf{op}(s) : \mathtt{T} \rrbracket = \llbracket \Gamma \vdash \mathsf{op}(s') : \mathtt{T} \rrbracket$；同样适用于 n 元操作。余归纳定义：

1) 映射。

$$\begin{aligned}
\llbracket \Gamma \vdash \mathsf{hd}(\mathsf{map}(s,t)) : \mathtt{T} \rrbracket &= \llbracket \Gamma \vdash s(\mathsf{hd}(t)) : \mathtt{T} \rrbracket, \\
\left\llbracket \Gamma \vdash \mathsf{wt}(\mathsf{map}(s,t)) : \mathtt{R}^+ \right\rrbracket &= \left\llbracket \Gamma \vdash s(\mathsf{wt}(t)) : \mathtt{R}^+ \right\rrbracket, \\
\llbracket \Gamma \vdash \mathsf{tl}(\mathsf{map}(s,t)) : \Sigma\,\mathtt{T} \rrbracket &= \llbracket \Gamma \vdash \mathsf{map}(\mathsf{tl}(t)) : \Sigma\,\mathtt{T} \rrbracket
\end{aligned}$$

由映射的余归纳定义可直接得到。

2）乘积。

$$\begin{gathered}[\![\Gamma \vdash (\mathtt{hd}(s), \mathtt{hd}(t)) : \mathtt{S} \times \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{hd}(s \otimes t) : \mathtt{S} \times \mathtt{T}]\!], \\ [\![\Gamma \vdash \mathtt{wt}(s) * \mathtt{wt}(t) : \mathtt{R}^+]\!] = [\![\Gamma \vdash \mathtt{wt}(s \otimes t) : \mathtt{R}^+]\!], \\ [\![\Gamma \vdash \mathtt{tl}(s) \otimes \mathtt{tl}(t) : \Sigma\,(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{tl}(s \otimes t) : \Sigma\,(\mathtt{S} \times \mathtt{T})]\!]\end{gathered}$$

直接由 $\otimes$ 的余归纳定义可得。

3）稀疏化。

$$\begin{gathered}[\![\Gamma \vdash \mathtt{hd}(\mathtt{thin}(n, t)) : \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{hd}(t) : \mathtt{T}]\!], \\ [\![\Gamma \vdash \mathtt{wt}(\mathtt{thin}(n, t)) : \mathtt{R}^+]\!] = [\![\Gamma \vdash \mathtt{wt}(t) : \mathtt{R}^+]\!], \\ \forall n \in \mathbb{N}, [\![\Gamma \vdash \mathtt{tl}(\mathtt{thin}(n, t)) : \Sigma\,\mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{tl}^n(t)) : \Sigma\,\mathtt{T}]\!], \\ [\![\Gamma \vdash \mathtt{thin}(1, t) : \Sigma\,\mathtt{T}]\!] = [\![\Gamma \vdash t : \Sigma\,\mathtt{T}]\!]\end{gathered}$$

直接由 thin 的余归纳定义可得。

4）伪随机数生成器。

$$\begin{gathered}[\![\Gamma \vdash \mathtt{hd}(\mathtt{prng}(s, t)) : \mathtt{T}]\!] = [\![\Gamma \vdash t : \mathtt{T}]\!], \\ [\![\Gamma \vdash \mathtt{wt}(\mathtt{prng}(s, t)) : \mathtt{R}^+]\!] = [\![\Gamma \vdash 1 : \mathtt{R}^+]\!], \\ [\![\Gamma \vdash \mathtt{tl}(\mathtt{prng}(s, t)) : \Sigma\,\mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{prng}(s, s(t)) : \Sigma\,\mathtt{T}]\!]\end{gathered}$$

直接由伪随机数生成器的余归纳定义可得。

5）重新加权。

$$\begin{gathered}[\![\Gamma \vdash \mathtt{hd}(\mathtt{reweight}(s, t)) : \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{hd}(t) : \mathtt{T}]\!], \\ [\![\Gamma \vdash \mathtt{wt}(\mathtt{reweight}(s, t)) : \mathtt{R}^+]\!] = [\![\Gamma \vdash s(\mathtt{hd}(t)) * \mathtt{wt}(t) : \mathtt{R}^+]\!], \\ [\![\Gamma \vdash \mathtt{tl}(\mathtt{reweight}(s, t)) : \Sigma\,\mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{reweight}(s, \mathtt{tl}(t)) : \Sigma\,\mathtt{T}]\!]\end{gathered}$$

由重新加权的余归纳定义可立即得到。

复合规则：

1) 在稀疏化之上进行稀疏化。

$$[\![\Gamma \vdash \mathtt{thin}(n, \mathtt{thin}(m, t)) : \Sigma\,\mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{thin}(n * m, t) : \Sigma\,\mathtt{T}]\!]$$

对于上下文 $\gamma \in [\![\Gamma]\!]$ 的任意可能值，我们归纳地证明 $[\![\Gamma \vdash \mathtt{thin}(n, \mathtt{thin}(m, t)) : \Sigma\,\mathtt{T}]\!](\gamma)$ 中的元素与 $[\![\Gamma \vdash \mathtt{thin}(n * m, t) : \Sigma\,\mathtt{T}]\!](\gamma)$ 中的元素相等。由于我们将提出的所有论证都具有完全相同的结构，因此仅在此证明中完整给出该结构；对于其余证明，我们只展示能够得到结论的双模拟。

我们通过为每个 $\gamma \in [\![\Gamma]\!]$ 构造一个双模拟 $R(\gamma) \subseteq [\![\Sigma\, T]\!] \times [\![\Sigma\, T]\!]$ 来证明这一结果。这是一个满足以下三个性质的采样器集合：

a) $\forall (s, t) \in R(\gamma)\ \pi_1(\mathrm{unfold}_{\mathrm{T}}(s)) = \pi_1(\mathrm{unfold}_{\mathrm{T}}(t))$；也就是说，$s$ 的头部和 t 的头部相同 b) $\forall (s, t) \in R(\gamma)\ \pi_2(\mathrm{unfold}_{\mathrm{T}}(s)) = \pi_2(\mathrm{unfold}_{\mathrm{T}}(t))$；也就是说，$s$ 的第一个权重和 t 的第一个权重相同 c) $\forall (s, t) \in R(\gamma)$，$(\pi_3(\mathrm{unfold}_{\mathrm{T}}(s)) \ \pi_3(\mathrm{unfold}_{\mathrm{T}}(t))) \in R(\gamma)$；也就是说，对双模拟中的两个采样器应用 tl，会得到双模拟中的另外两个采样器。

这种双模拟 $R(\gamma)$ 的结构通常通过对我们希望证明的等价性两侧应用 tl，然后应用我们之前展示过的规则（在本例中通常是每个运算的余归纳的定义，即 thin）来化简所得结果。例如，在本例中，我们可以化简

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{thin}(n, \mathtt{thin}(m, t))) : \Sigma\,\mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{thin}(m, \mathtt{tl}^{m*n}(t))) : \Sigma\,\mathtt{T}]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{thin}(n * m, t)) : \Sigma\,\mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{thin}(n * m, \mathtt{tl}^{n*m}(t)) : \Sigma\,\mathtt{T}]\!].$$

这表明可以将以下集合作为双模拟：

$$R(\gamma) = \left\{ [\![(\Gamma \vdash \mathtt{thin}(n, \mathtt{thin}(m, \mathtt{tl}^k(t))) : \Sigma\,\mathtt{T}]\!](\gamma), [\![\Gamma \vdash \mathtt{thin}(n * m, \mathtt{tl}^k(t)) : \Sigma\,\mathtt{T})]\!](\gamma) \mid k \in \mathbb{N} \right\}.$$

今后，鉴于这个表达式相当繁琐，我们将省略对 γ 的依赖。

现在我们必须证明这是一个有效的双模拟。首先，我们必须证明对这些程序中的每一个应用 hd 和 wt 所得到的结果都相同，这一点总是显而易见的。在本例中，应用我们之前称为 thin 的余归纳定义的规则可得

$$[\![\Gamma \vdash \mathtt{hd}(\mathtt{thin}(n, \mathtt{thin}(m, \mathtt{tl}^k(t)))) : \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{hd}(\mathtt{thin}(m, \mathtt{tl}^k(t))) : \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{hd}(\mathtt{tl}^k(t)) : \mathtt{T}]\!] :$$

and

$$[\![\Gamma \vdash \mathtt{hd}(\mathtt{thin}(n * m, \mathtt{tl}^k(t))) : \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{hd}(\mathtt{tl}^k(t)) : \mathtt{T}]\!] ;$$

同样的论证完全适用于 wt:

$$[\![\Gamma \vdash \mathtt{wt}(\mathtt{thin}(n, \mathtt{thin}(m, \mathtt{tl}^k(t)))) : \mathtt{R}^+]\!] = [\![\Gamma \vdash \mathtt{wt}(\mathtt{thin}(m, \mathtt{tl}^k(t))) : \mathtt{R}^+]\!] = [\![\Gamma \vdash \mathtt{wt}(\mathtt{tl}^k(t)) : \mathtt{R}^+]\!]$$

and

$$[\![\Gamma \vdash \mathtt{wt}(\mathtt{thin}(n * m, \mathtt{tl}^k(t))) : \mathtt{R}^+]\!] = [\![\Gamma \vdash \mathtt{wt}(\mathtt{tl}^k(t)) : \mathtt{R}^+]\!] ;$$

最后，我们必须证明，对这些表达式逐一应用 tl，会得到双模拟中的另一个元素。这本质上与引出双模拟 R 的论证相同:

$$\begin{aligned} &[\![\Gamma \vdash \mathtt{tl}(\mathtt{thin}(n, \mathtt{thin}(m, \mathtt{tl}^k(t)))) : \Sigma\, \mathtt{T}]\!] \\ &= [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{tl}^n(\mathtt{thin}(m, \mathtt{tl}^k(t)))) : \Sigma\, \mathtt{T}]\!] \\ &= [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{thin}(m, \mathtt{tl}^{n*m+k}(t))) : \Sigma\, \mathtt{T}]\!] \end{aligned}$$

and

$$\begin{aligned} &[\![\Gamma \vdash \mathtt{tl}(\mathtt{thin}(n * m, \mathtt{tl}^k(t))) : \Sigma\, \mathtt{T}]\!] = \\ &[\![\Gamma \vdash \mathtt{thin}(n * m, \mathtt{tl}^{n*m+k}(t)) : \Sigma\, \mathtt{T}]\!] \end{aligned}$$

因此，对于任意 γ，应用 tl 都会得到我们双模拟中的另一个元素，由此证明完毕。以此证明为参考，我们将在附录中简略处理其余的双模拟证明，因为每个论证的结构都相同。

2）映射上的映射。

$$[\![\Gamma \vdash \mathtt{map}(g, \mathtt{map}(f, t)) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{map}(g \circ f, t) : \Sigma\, \mathtt{T}]\!]$$

应用 map 的余归纳定义，我们很容易看出

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{map}(g, \mathtt{map}(f, t))) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{map}(g, \mathtt{map}(f, \mathtt{tl}(t))) : \Sigma\, \mathtt{T}]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{map}(g \circ f, t)) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{map}(g \circ f, \mathtt{tl}(t)) : \Sigma\, \mathtt{T}]\!] ;$$

这表明双模拟

$$R = \{([\![\Gamma \vdash \mathtt{map}(g, \mathtt{map}(f, \mathtt{tl}^n(t))) : \Sigma\, \mathtt{T}]\!], [\![\Gamma \vdash \mathtt{map}(g \circ f, \mathtt{tl}^n(t)) : \Sigma\, \mathtt{T})]\!] \mid n \in \mathbb{N}\} ,$$

很容易验证这一双模拟：只需对两边应用 hd，并通过应用 map 的余归纳定义进行化简，即可看到我们得到两个相等的表达式；对两边应用 wt，并通过应用 map 的余归纳定义进行化简，也会得到两个相等的表达式；最后对两边应用 tl，并通过应用 map 的余归纳定义进行化简，可以看到所得的有序对也包含在这一双模拟中。

3）重新加权之上的重新加权。

$$[\![\Gamma \vdash \mathtt{reweight}(g, \mathtt{reweight}(f, t)) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{reweight}(f \cdot g, t) : \Sigma\, \mathtt{T}]\!]$$

对两边应用 tl，并使用前面关于 tl 与重新加权之间关系的规则，我们很容易得到

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{reweight}(g, \mathtt{reweight}(f, t))) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{reweight}(g, \mathtt{reweight}(f, \mathtt{tl}(t))) : \Sigma\, \mathtt{T}]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{reweight}(g \cdot f, t)) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{reweight}(g \circ f, \mathtt{tl}(t)) : \Sigma\, \mathtt{T}]\!] .$$

于是，等价性由该双模拟得出

$$R = \{([\![\Gamma \vdash \mathtt{reweight}(g, \mathtt{reweight}(f, \mathtt{tl}^m(t))) : \Sigma\, \mathtt{T}]\!], [\![\Gamma \vdash \mathtt{reweight}(g \circ f, \mathtt{tl}^m(t)) : \Sigma\, \mathtt{T})]\!] \mid m \in \mathbb{N}\}$$

这一点很容易验证，从而得到所需的等式。

4）在伪随机数生成器上进行稀疏化。

$$\forall n \in \mathbb{N}, [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{prng}(s,t)) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{prng}(s^n, t) : \Sigma\, \mathtt{T}]\!]$$

对每个表达式的两边应用 tl，并利用 thin 和伪随机数生成器的余归纳定义进行化简，可得

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{thin}(n, \mathtt{map}(s,t))) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{map}(s, \mathtt{tl}^n(t))) : \Sigma\, \mathtt{T}]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{map}(s, \mathtt{thin}(n,t))) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{map}(s, \mathtt{thin}(n, \mathtt{tl}^n(t))) : \Sigma\, \mathtt{T}]\!].$$

这表明应选择双模拟

$$R = \{([\![\Gamma \vdash \mathtt{thin}(n, \mathtt{map}(s, \mathtt{tl}^m(t))) : \Sigma\, \mathtt{T}]\!], [\![\Gamma \vdash \mathtt{map}(s, \mathtt{thin}(n, \mathtt{tl}^m(t))) : \Sigma\, \mathtt{T})]\!]) \mid m \in \mathbb{N}\}$$

它很容易得到验证，并给出我们的结果。

5）在映射上进行稀疏化。

$$[\![\Gamma \vdash \mathtt{thin}(\mathtt{n}, \mathtt{map}(\mathtt{s}, \mathtt{t})) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{map}(\mathtt{s}, \mathtt{thin}(\mathtt{n}, \mathtt{t})) : \Sigma\, \mathtt{T}]\!]$$

利用映射和 thin 的余归纳定义，我们得到

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{thin}(n, \mathtt{map}(s,t)) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{map}(s, \mathtt{tl}^n(t))) : \Sigma\, \mathtt{T}]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{map}(s, \mathtt{thin}(n,t))) : \Sigma\, \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{map}(s, \mathtt{thin}(n, \mathtt{tl}^n(t))) : \Sigma\, \mathtt{T}]\!].$$

所需结果由此得出

$$R = \{([\![\Gamma \vdash \mathtt{thin}(n, \mathtt{map}(s, \mathtt{tl}^m(t))) : \Sigma\, \mathtt{T}]\!], [\![\Gamma \vdash \mathtt{map}(s, \mathtt{thin}(n, \mathtt{tl}^m(t))) : \Sigma\, \mathtt{T}]\!]) \mid m \in \mathbb{N}\}$$

很容易看出这是一个有效的双模拟。

乘积规则:

1) 稀疏化。

$$[\![\Gamma \vdash \mathtt{thin}(n,s) \otimes \mathtt{thin}(n,t) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{thin}(n, s \otimes t) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!]$$

使用 thin 的余归纳定义来证明

$$[\![\Gamma \vdash \mathrm{tl}(\mathrm{t\ \ hin}(n,s) \otimes \mathrm{thin}(n,t)) \colon\ \ \Sigma\, (\mathrm{S\ \ } \times \mathrm{T})]\!] = [\![\Gamma \vdash \mathrm{thin}(n, \mathrm{tl}^n(s)) \otimes \mathrm{thin}(n, \mathrm{tl}^n(t)) \colon\ \ \Sigma\, (\mathrm{S\ \ } \times \mathrm{T})]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{thin}(n, s \otimes t)) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{tl}^n(s) \otimes \mathtt{tl}^n(t)) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!],$$

这表明

$$\begin{aligned} R = \{([\![\Gamma \vdash \mathtt{thin}(n, \mathtt{tl}^m(s)) \otimes \mathtt{thin}(n, \mathtt{tl}^m(t)) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!], \\ [\![\Gamma \vdash \mathtt{thin}(n, \mathtt{tl}^m(s \otimes t)) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!]) : m \in \mathbb{N}\} \end{aligned}$$

构成一个双模拟。

2）映射。

$$[\![\Gamma \vdash s \otimes \mathtt{map}(g, t') : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{map}(id_{\mathtt{S}} \times g, s \otimes t') : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!]$$

对每个表达式应用 tl 可得

$$[\![\Gamma \vdash \mathtt{tl}(s \otimes \mathtt{map}(g, t')) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{tl}(s) \otimes \mathtt{map}(g, \mathtt{tl}(t')) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{map}(id_{\mathtt{S}} \times g, s \otimes t')) : \Sigma\, (\mathtt{S} \times \mathtt{T}]\!] = [\![\Gamma \vdash \mathtt{map}(id_{\mathtt{S}} \times g, \mathtt{tl}(s) \otimes \mathtt{tl}(t')) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!],$$

表明

$$\begin{aligned} R = \{([\![\Gamma \vdash \mathtt{tl}^m(s) \otimes \mathtt{map}(g, \mathtt{tl}^m(s')) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!], \\ [\![\Gamma \vdash \mathtt{map}(id_{S'} \times g, \mathtt{tl}^m(s), \mathtt{tl}^m(s))) : \Sigma\, (\mathtt{S} \times \mathtt{T})]\!]) \mid m \in \mathbb{N}\} \end{aligned}$$

作为双模拟。

$$[\![\Gamma \vdash \text{映射}(f,t) \otimes s' : \ \Sigma(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \text{映射}(f \times id_T, t \otimes s') : \ \Sigma(\mathtt{S} \times \mathtt{T})]\!]$$

证明与前文相同。

3）重新加权。

$$[\![\Gamma \vdash s \otimes \mathtt{reweight}(g,t') : \Sigma(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{reweight}(1_S \cdot g, s \otimes t') : \Sigma(\mathtt{S} \times \mathtt{T})]\!]$$

对两边应用 tl，并使用重新加权的余归纳定义进行化简，可得

$$[\![\Gamma \vdash \mathtt{tl}(s \otimes \mathtt{reweight}(g,t')) : \Sigma(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{tl}(s) \otimes \mathtt{reweight}(g, \mathtt{tl}(t')) : \Sigma(\mathtt{S} \times \mathtt{T})]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{reweight}(1_{\mathtt{S}} \cdot g, s \otimes t')) : \Sigma(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{reweight}(1_{\mathtt{S}} \cdot g, \mathtt{tl}(s) \otimes \mathtt{tl}(t')) : \Sigma(\mathtt{S} \times \mathtt{T})]\!].$$

选择双模拟

$$\begin{aligned} R = \{([\![\Gamma \vdash \mathtt{tl}^m(s) \otimes \mathtt{reweight}(g, \mathtt{tl}^m(t')) : \Sigma(\mathtt{S} \times \mathtt{T})]\!], \\ [\![\Gamma \vdash \mathtt{reweight}(1_{\mathtt{S}} \cdot g, \mathtt{tl}^m(s) \otimes \mathtt{tl}^m(t')) : \Sigma(\mathtt{S} \times \mathtt{T})]\!]) \mid m \in \mathbb{N}\}, \end{aligned}$$

即可得到我们的结果。

$$[\![\Gamma \vdash \mathtt{reweight}(f,t) \otimes s' : \Sigma(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{reweight}(f \cdot 1_{\mathtt{T}}, s' \otimes t) : \Sigma(\mathtt{S} \times \mathtt{T})]\!]$$

证明与前述相同。

4）伪随机数生成器。

$$[\![\Gamma \vdash \mathtt{prng}(f,a) \otimes \mathtt{prng}(g,b) : \Sigma(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{prng}(f \times g, (a,b)) : \Sigma(\mathtt{S} \times \mathtt{T})]\!]$$

利用伪随机数生成器的余归纳定义，我们很快得到

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{prng}(f,a) \otimes \mathtt{prng}(g,b)) : \Sigma(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{prng}(f, f(a)) \otimes \mathtt{prng}(g, g(b)) : \Sigma(\mathtt{S} \times \mathtt{T})]\!]$$

and

$$[\![\Gamma \vdash \mathtt{tl}(\mathtt{prng}(f \times g, (a,b))) : \Sigma(\mathtt{S} \times \mathtt{T})]\!] = [\![\Gamma \vdash \mathtt{prng}(f \times g, (f \times g)(a,b)) : \Sigma(\mathtt{S} \times \mathtt{T})]\!],$$

这表明存在双模拟

$$\begin{aligned} R = \{([\![\Gamma \vdash \mathtt{prng}(f, f^m(a)) \otimes \mathtt{prng}(g, g^m(b)) : \Sigma(\mathtt{S} \times \mathtt{T})]\!], \\ [\![\Gamma \vdash \mathtt{prng}(f \times g, (f \times g)^m(a,b)) : \Sigma(\mathtt{S} \times \mathtt{T})]\!]) \mid m \in \mathbb{N}\} \end{aligned}$$

由此得到我们所需的结果。

□

命题 IV.1 的证明。

展开式（3），对于任意类型正确的采样器 $\Gamma \vdash s : \Sigma\mathtt{S}$，嵌套自乘积 $(s^m)^n$ 定义为

$$\mathtt{thin}(n, \mathtt{thin}(m, s \otimes \mathtt{tl}(s) \otimes \cdots \otimes \mathtt{tl}^{m-1}(s)) \otimes \cdots \otimes \mathtt{tl}^{n-1}(\mathtt{thin}(m, s \otimes \mathtt{tl}(s) \otimes \cdots \otimes \mathtt{tl}^{m-1}(s)))).$$

对最内层表达式应用表 IV 中的规则 $\Gamma \vdash \mathrm{tl}(\mathrm{thin}(m,t)) \approx \mathrm{thin}(m, \mathrm{tl}^m(t)) : \Sigma\mathrm{T}$，可得该程序在上下文 Γ 中等价于

$$\mathtt{thin}(n, \mathtt{thin}(m, \mathtt{tl}^0(s) \otimes \cdots \otimes \mathtt{tl}^{m-1}(s)) \otimes \cdots \otimes \mathtt{thin}(m, \mathtt{tl}^{mn-m}(s) \otimes \cdots \otimes \mathtt{tl}^{mn-1}(s))).$$

接下来，应用规则 $\Gamma \vdash \mathrm{thin}(m, s \otimes t) \approx \mathrm{thin}(m,s) \otimes \mathrm{thin}(m,t) : \Sigma(\mathrm{S} \times \mathrm{T})$，我们看到嵌套自乘积等价于

$$\mathtt{thin}(n, \mathtt{thin}(m, \mathtt{tl}^0(s) \otimes \mathtt{tl}^1(s) \otimes \cdots \otimes \mathtt{tl}^{mn-1}(s))).$$

Ap应用用于 thin 的复合的规则 $\Gamma \vdash \mathrm{thin}(n, \mathrm{thin}(m,t)) \approx \mathrm{thin}(mn,t) : \Sigma\mathrm{T}$，可得 s

$$\mathtt{thin}(mn, \mathtt{tl}^0(s) \otimes \mathtt{tl}^1(s) \otimes \cdots \otimes \mathtt{tl}^{mn-1}(s)),$$

而上述内容正是自乘积 s^{mn}的定义。 □

命题 IV.3 的证明。

- 映射：应用自乘积公式（3）的定义，语法 $\mathsf{map}(f,s)^n$ 是采样器的简写

$$\mathtt{thin}(n,\mathtt{map}(f,s)\otimes\mathtt{tl}(\mathtt{map}(f,s))\otimes\cdots\otimes\mathtt{tl}^{n-1}(\mathtt{map}(f,s)))$$

  在此采样器类型正确的上下文 Γ 中，应用规则 $\Gamma\vdash\mathsf{tl}(\mathsf{map}(f,s))\approx\mathsf{map}(f,\mathsf{tl}(s)):\Sigma\,\mathsf{T}$，可知上述采样器等价于

$$\mathtt{thin}(n,\mathtt{map}(f,\mathtt{tl}^0(s))\otimes\cdots\otimes\mathtt{map}(f,\mathtt{tl}^{n-1}(s))).$$

  在此证明中，将程序 $f:\mathsf{S}\to\mathsf{T}$ 的 n 重笛卡尔积简记为 $f^{\times n}:\mathsf{S}^n\to\mathsf{T}^n$。应用规则 $\Gamma\vdash\mathsf{map}(f,s)\otimes\mathsf{map}(g,t)\approx\mathsf{map}(f\times g,(s,t)):\Sigma(\mathsf{S}\times\mathsf{T})$，该采样器也可以写成等价形式

$$\mathtt{thin}(n,\mathtt{map}(f^{\times n},\mathtt{tl}^0(s)\otimes\cdots\otimes\mathtt{tl}^{n-1}(s))).$$

  最后，应用规则 $\Gamma\vdash\mathsf{thin}(n,\mathsf{map}(f,s))\approx\mathsf{map}(f,\mathsf{thin}(n,s)):\Sigma\,\mathsf{S}$，得到

$$\mathtt{map}(f^{\times n},\mathtt{thin}(n,\mathtt{tl}^0(s)\otimes\cdots\otimes\mathtt{tl}^{n-1}(s))))$$

  根据自乘积的定义，这就是我们所需的结果 $\mathsf{map}(f^{\times n},s^n)$。
- 重新加权：该证明与上文相同，只需将 map 替换为 reweight，并将笛卡尔积$\times$ 替换为逐点乘积 $\cdot$；不过，我们仍将完整证明。根据自乘积的定义式 (3)，语法 $\mathsf{reweight}(f,s)^n$ 是以下内容的简写

$$\mathtt{thin}(n,\mathtt{reweight}(f,s)\otimes\mathtt{tl}(\mathtt{reweight}(f,s))\otimes\cdots\otimes\mathtt{tl}^{n-1}(\mathtt{reweight}(f,s)))$$

  在 Γ 该采样器类型正确的上下文中，应用规则 $\Gamma\vdash\mathsf{tl}(\mathsf{reweight}(f,s))\approx\mathsf{reweight}(f,\mathsf{tl}(s)):\Sigma\,\mathsf{T}$ 可知，上述采样器等价于

$$\mathtt{thin}(n,\mathtt{reweight}(f,\mathtt{tl}^0(s))\otimes\cdots\otimes\mathtt{reweight}(f,\mathtt{tl}^{n-1}(s))).$$

  为便于证明，将程序 $f:\mathsf{S}\to\mathsf{R}$ 的 n重逐点乘积简写为 $f^{\cdot n}:\mathsf{S}^n\to\mathsf{R}$。应用规则 $\Gamma\vdash\mathsf{reweight}(f,s)\otimes\mathsf{reweight}(g,t)\approx\mathsf{reweight}(f\cdot g,(s,t)):\Sigma(\mathsf{S}\times\mathsf{T})$，该采样器也可以写成等价形式

$$\mathtt{thin}(n,\mathtt{reweight}(f^{\cdot n},\mathtt{tl}^0(s)\otimes\cdots\otimes\mathtt{tl}^{n-1}(s))).$$

  Fina同样地，应用规则 $\Gamma\vdash\mathsf{thin}(n,\mathsf{reweight}(f,s))\approx\mathsf{reweight}(f,\mathsf{thin}(n,s)):\Sigma\,\mathsf{S}$，可得

$$\mathtt{reweight}(f^{\cdot n},\mathtt{thin}(n,\mathtt{tl}^0(s)\otimes\cdots\otimes\mathtt{tl}^{n-1}(s))))$$

  根据自乘积的定义，这就是我们所需的结果 $\mathsf{reweight}(f^{\cdot n},s^n)$。

□

命题 V.1 的证明。

设 $g\colon Y\to\mathbb{R}$ 为一个有界连续函数。则 $g\circ f\colon X\to\mathbb{R}$ 也是有界连续的，由弱收敛和 ε 的定义可知

$$\begin{aligned}\lim_{n\to\infty}\int_X g\;d\widehat{f\circ\sigma}_n &= \lim_{n\to\infty}\int_X g\circ f\;d\hat{\sigma}_n && \text{根据定义}\\ &= \int_X g\circ f\;d\mu && \text{由于 } \varepsilon_X(\sigma)=\mu\\ &= \int_X g\;df_*\mu && \text{变量替换}\end{aligned}$$

⌢因此 $f\circ\sigma_n\longrightarrow f_*\mu$ 弱地，即 $\varepsilon(f\circ\sigma)=f_*(\mu)$。

□

定理 V.1 的证明。

对该推导进行归纳。

(i) 内置采样器。这些公理根据假设成立。(ii) 等价性。等价项指向同一测度这一事实，是指向的定义以及定理 IV.1 的直接推论。

(iii) 尾部。采样器 σ 的尾部与 σ 指向同一测度这一事实，是根据极限定义的指向的直接推论。

(iv) 推前。如果 $[\![f]\!]$ 在类型系统的标准拓扑下连续，则该规则可直接由命题 V.1 得出。如果 $[\![f]\!]$ 在类型系统的标准拓扑下不连续，则有以下两种情况：(a) 测度 $\mu(\gamma)$ 在划分中某个元素的边界上赋予了正质量，使得 $[\![f]\!]$ 分段连续；此时 $\varepsilon[\![s]\!](\gamma)$ 不会收敛到 $\mu(\gamma)$，因此该规则的前提不成立；或者 (b) 没有测度 $\mu(\gamma)$ 在划分中任何元素的边界上赋予质量，使得 $[\![f]\!]$ 连续；此时 $\varepsilon[\![s]\!](\gamma) = \mu(\gamma)$，结论再次可由命题 V.1 得出。

(v) 重新加权。该规则仅仅编码了重要性采样的有效性。为使记号简洁，省略 γ 中的依赖关系，并令 $\nu = f \cdot \mu$ 为重新加权后的测度；该规则的前提与附加条件合起来表明，存在 $\alpha \in \mathbb{R}^+$ 使得 $f = \alpha \frac{d\mu}{d\nu}$，且 f 在 μ 的支撑集上有界。由于 μ 是一个概率分布，我们有

$$\int f \, d\mu = \alpha \int \frac{d\mu}{d\nu} \, d\nu = \alpha \int d\nu = \alpha$$

此外，由于 s 趋于 μ 且 f 是有界连续的，通过写成 x_i， $\pi_1(\pi_i([\![s]\!])$ 和 w_i， $\pi_2(\pi_i([\![s]\!])$，我们得到

$$\alpha = \int f \, d\mu = \lim_{N\to\infty} \frac{1}{N} \frac{\sum_{i=1}^N f(x_i) w_i}{\sum_{k=1}^N w_k} \tag{5}$$

令 g 为任意有界连续函数 $[\![\mathrm{S}]\!] \to \mathbb{R}$，并注意到逐点乘积 $g.f$ 在 μ 和 ν 的支撑集上是有界连续的，我们有

$$\begin{aligned}
\int g \, d\nu &= \frac{1}{\alpha} \int g.f \, d\mu \\
&= \frac{1}{\alpha} \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^N g(x_i) f(x_i) \frac{w_i}{\sum_{k=1}^N w_k} && \text{由于 } s \text{ 趋于 } \mu \\
&= \left(\lim_{N\to\infty} \frac{1}{N} \frac{\sum_{i=1}^N f(x_i) w_i}{\sum_{k=1}^N w_k} \right)^{-1} \left(\lim_{N\to\infty} \frac{1}{N} \frac{\sum_{i=1}^N g(x_i) f(x_i) w_i}{\sum_{k=1}^N w_k} \right) && \text{By (5)} \\
&= \lim_{N\to\infty} \frac{1}{N} \frac{\sum_{i=1}^N g(x_i) f(x_i) w_i}{\sum_{i=1}^N f(x_i) w_i} \\
&\triangleq \lim_{N\to\infty} \int g \, d\widehat{\texttt{reweight}}(f, s)_n
\end{aligned}$$

换言之，重新加权(f, s) 趋于 ν。

(vi) 伪随机数生成器。伪随机数生成器规则只是对众所周知的遍历定理 [19,定理 9.6] 的重新表述。

□

命题 V.2 的证明。

如果能够证明 $\mathrm{S} \triangleleft \mathrm{T}$ 蕴含 $[\![\mathrm{S}]\!]$ 和 $[\![\mathrm{T}]\!]$ 是同一个可测空间，那么证明的第一部分将立即成立；这将通过对子类型推导进行归纳来证明。我们首先证明函子 $\mathbf{Borel} : \mathbf{Top} \to \mathbf{Meas}$ 与余积相容。这将证明基例、余积规则以及图 4b 中最后两个规则。

设 X、Y 为两个拓扑空间（我们将对拓扑空间及其拓扑，以及可测空间及其 σ-代数使用相同的名称）。我们使用 π-λ 引理来证明 $\mathbf{Borel}(X+Y) = \mathbf{Borel}(X) + \mathbf{Borel}(Y)$。首先注意，根据定义，$\mathbf{Borel}(X+Y) = \sigma(X+Y)$。由于 $X+Y$ 是一个拓扑，因此显然也是一个 π-系统；又由于 $\mathbf{Borel}(X)+\mathbf{Borel}(Y)$ 是一个 σ-代数，因此显然也是一个 λ-系统。根据定义，每个开集 U 在 $X+Y$ 中都具有如下性质：$U = X \cap U$ 在 X 中是开集，因而是 $\mathbf{Borel}(X)$ 的元素。同样，$Y \cap U$ 在 Y 中是开集，因而属于 $\mathbf{Borel}(Y)$。根据 Meas 中余积的定义，$U = (U \cap X) \uplus (U \cap Y)$ 属于 $\mathbf{Borel}(X) + \mathbf{Borel}(Y)$。于是，由 π-λ 引理可得 $\mathbf{Borel}(X+Y) \subseteq \mathbf{Borel}(X) + \mathbf{Borel}(Y)$ 的包含关系。

反之，$\mathbf{Borel}(X)$ 中的每个可测 $A + \mathbf{Borel}(Y)$，根据定义，都具有 $(A \cap X) \uplus (A \cap Y)$ 的形式，其中 $(A \cap X) \in \mathbf{Borel}(X)$ 且 $(A \cap Y) \in \mathbf{Borel}(Y)$。利用 π-λ 引理，很容易证明 $\mathbf{Borel}(X) \subseteq \mathbf{Borel}(X+Y)$ 且 $\mathbf{Borel}(Y) \subseteq \mathbf{Borel}(X+Y)$；因此，由于 $\mathbf{Borel}(X+Y)$ 对并集运算封闭，可得 $A = (A \cap X) \uplus (A \cap Y) \in \mathbf{Borel}(X+Y)$，这就证明了 $\mathbf{Borel}(X+Y) \supseteq \mathbf{Borel}(X) + \mathbf{Borel}(Y)$。

为了证明函子 $\mathbf{Borel}: \mathbf{Top} \to \mathbf{Meas}$ 与乘积可交换，我们需要额外假设这些空间是第二可数的。随后可在例如 [3, p244] 中找到证明。

对于证明的第二部分，设 U 属于 $[\![\mathrm{S}]\!]$ 的拓扑，但不属于 $[\![\mathrm{T}]\!]$ 的拓扑。这意味着 $\partial_{\mathrm{T}}(U) = U \cap \mathrm{int}_{\mathrm{T}}(U) \neq$ 在 $[\![\mathrm{S}]\!]$ 中是开集（因为它是 $[\![\mathrm{S}]\!]$ 中两个开集的交集）。特别地，它是 $[\![\mathrm{S}]\!]$ 中的连续性集（由于它是开集，其内部是空集，因此不可能具有任何 μ-质量）。根据 Portmanteau 引理（由于假设这些空间是可度量化的，该引理适用），我们必有

$$\lim_{n\to\infty} \widehat{[\![s]\!]}_n \left(\partial_{\mathrm{T}}(U)\right) = \mu(\partial_{\mathrm{T}}(U)) > 0$$

尤其是，如果 $[\![s]\!]$ 仅有限次访问 $\partial_{\mathrm{T}}(U)$，这显然是不可能的。 □