Mathematical Structures in Computer Science (2021), 1–49
doi:10.1017/S0960129521000396

CAMBRIDGE UNIVERSITY PRESS

PAPER

The geometry of Bayesian programming

Ugo Dal Lago[1,*] and Naohiko Hoshino[2]

[1]Department of Computer Science and Engineering, University of Bologna, Bologna, Italy and [2]Department of Computer and Information Sciences, Sojo University, Kumamoto, Japan

[*]Corresponding author. Email: ugo.dallago@unibo.it

(Received 4 February 2020; revised 3 September 2021; accepted 29 October 2021)

Abstract

We give two geometry of interaction models for a typed λ-calculus with recursion endowed with operators for sampling from a continuous uniform distribution and soft conditioning, namely a paradigmatic calculus for higher-order Bayesian programming. The models are based on the category of measurable spaces and partial measurable functions, and the category of measurable spaces and s-finite kernels, respectively. The former is proved adequate with respect to both a distribution-based and a sampling-based operational semantics, while the latter is proved adequate with respect to a sampling-based operational semantics.

Keywords: Probabilistic lambda-calculus; conditioning; Bayesian programming; geometry of interaction

1. Introduction

Randomisation provides the most efficient algorithmic solutions, at least concretely, in many different contexts. A typical example is the one of primality testing, where the Miller–Rabin test (Miller 1976; Rabin 1980) remains the preferred choice despite polynomial time deterministic algorithms are available from many years now (Agrawal et al. 2002). Probability theory can be exploited even more fundamentally in programming, by way of the so-called probabilistic (or, more specifically, Bayesian) programming, as popularised by languages like, among others, **ANGLICAN** (Wood et al. 2014) or **CHURCH** (Goodman et al. 2008). This has stimulated research about probabilistic programming languages and their semantics (Danos and Harmer 2002; Ehrhard et al. 2018*a*; Jones 1990), together with type systems (Breuvart and Dal Lago 2018; Dal Lago and Grellois 2017), equivalence methodologies (Crubillé and Dal Lago 2014; Dal Lago et al. 2014) and verification techniques (Sato et al. 2019).

Giving a satisfactory denotational semantics to higher-order functional languages is already problematic in presence of probabilistic choice (Jones 1990; Jung and Tix 1998) and becomes even more challenging in presence of continuous distributions and scoring. Recently, quasi-Borel spaces (Heunen et al. 2017) have been proposed as a way to give semantics to calculi with all these features, and only very recently (Vákár et al. 2019) this framework has been shown to be adaptable to a fully fledged calculus for probabilistic programming, in which continuous distributions *and* soft conditioning are present. Probabilistic coherent spaces (Danos and Ehrhard 2011) are fully abstract (Ehrhard et al. 2018*a*) for λ-calculi with discrete probabilistic choice, and can, with some effort, be adapted to calculi with sampling from continuous distributions (Ehrhard et al. 2018*b*), although without scoring.

A research path which has been studied only marginally, so far, consists in giving semantics to Bayesian higher-order programming languages through *interactive forms* of semantics, for example, game semantics (Abramsky et al. 2000; Hyland and Ong 2000) or the geometry of interaction (GOI) (Girard 1989). One of the very first models for higher-order calculi with discrete probabilistic choice was in fact a game model, proved fully abstract for a probabilistic calculus with global ground references (Danos and Harmer 2002). After more than 10 years, a parallel form of GoI and some game models have been introduced for λ-calculi with probabilistic choice (Castellan et al. 2018; Clairambault and Paquet 2018; Dal Lago et al. 2017), but in all these cases only discrete probabilistic choice can be handled, with the exception of a recent work on concurrent games and continuous distributions (Paquet and Winskel 2018).

In this paper, we will report on some results about two GoI models of higher-order Bayesian languages: one is a GoI model closer to the standard GoI semantics presented in terms of so-called token machines and the other is a GoI model based on s-finite kernels (Staton 2017). The distinguishing features of the introduced GoI models can be summarised as follows:

- **Simplicity**. The categories on which the models are defined are the category of measurable spaces and *partial* measurable functions and the category of measurable spaces and s-finite kernels, respectively. As such they are completely standard from a measure-theoretic perspective.
- **Expressivity**. As is well known, the GoI construction (Abramsky et al. 2002; Joyal et al. 1996) allows to give semantics to calculi featuring higher-order functions and recursion. Indeed, our GoI model can be proved adequate for **PCFSS**, a fully fledged calculus for probabilistic programming.
- **Flexibility**. The model we present is quite flexible, in the sense of being able to reflect the operational behaviour of programs as captured by *both* the distribution-based and the sampling-based semantics (Borgström et al. 2016).
- **Intuitiveness**. GoI visualises the structure of programs in terms of graphs, from which dependencies between subprograms can be analysed. Adequacy of our model provides some diagrammatic reasoning principles about observational equivalence of **PCFSS**.

This paper's contributions, beside the model's definition, are adequacy results which precisely relate our GoI model to the operational semantics both in the *distribution* and in *sampling* styles. As applications of our adequacy results, we show that integrating over the sampling-based operational semantics, one obtains precisely the distribution-based operational semantics, and we also observe commutativity of let-bindings in our language.

Turning Measurable Spaces into a GoI Model Before delving into the details of our model, it is worthwhile to give some hints about how the proposed model is obtained, and why it differs from similar GoI models from the literature.

The thread of work the proposed model stems from is the one of so-called memoryful GOI (Hoshino et al. 2014; Muroya et al. 2016). The underlying idea of this paper is very similar: program execution is modelled as an interaction between the program and its environment, and memoisation takes place inside the program as a result of the interaction. In the previous work on memoryful GoI by the second author with Hasuo and Muroya, the goal consisted in modelling a λ-calculus with *algebraic* effects. Starting from a monad together with some algebraic effects, they gave an adequate GoI model for such a calculus, which is applicable to a wide range of algebraic effects. In principle, then, their recipe could be applicable to **PCFSS**, since sampling-based operational semantics enables us to see scoring and sampling as algebraic effects acting on global states. However, that would not work for **PCFSS**, since the category **Meas** of measurable spaces is not cartesian closed, and we thus *cannot* define a state monad by way of the exponential $S \Rightarrow S \times (-)$.

In this paper, we side step this issue by a series of translations, to be described in Section 4 below. Instead of looking for a state monad on **Meas**, we embed **Meas** into the category of **Int**-objects and Mealy machines (Section 5) and use a state monad *on this* category. This is doable because the category of **Int**-objects and Mealy machines is a *compact closed category* obtained by applying the **Int**-*construction* (Abramsky et al. 2002) to the traced symmetric monoidal category of measurable spaces and partial measurable functions. For more detail on categorical aspect of our semantics, see Dal Lago and Hoshino (2019*a*). The use of such compact closed categories (or, more generally, of traced monoidal categories) is the way GoI models higher-order functions.

Outline The rest of the paper is organised as follows. After giving some necessary measure-theoretic preliminaries in Section 2 below, we introduce in Section 3 the language **PCFSS**, together with the two kinds of operational semantics we were referring to above. In Section 4, we introduce one of our GoI models, which is based on partial measurable functions, informally. In Sections 5 and 6, a more rigorous treatment of the involved concepts is given, together with the adequacy results. We discuss in Sections 7 and 8 an alternative way of giving a GoI semantics to **PCFSS** based on s-finite kernels, and as an application, we prove commutativity of let-bindings. We conclude in Section 9. This paper is a revised and extended version of our conference paper (Dal Lago and Hoshino 2019*b*).

2. Measure-Theoretic Preliminaries

In this section, we recall some basic notions in measure theory that will be needed in the following. We also fix some useful notations. For more about measure theory, see standard textbooks such as Billingsley (1986).

A σ*-algebra* on a set X is a family Σ consisting of subsets of X such that $\emptyset \in \Sigma$; and if $A \in \Sigma$, then the complement $X \setminus A$ is in Σ; and for any family $\{A_n \in \Sigma\}_{n\in\mathbb{N}}$, the intersection $\bigcap_{n\in\mathbb{N}} A_n$ is in Σ. A *measurable space* X is a set $|X|$ equipped with a σ-algebra Σ_X on $|X|$. We often confuse a measurable space X with its underlying set $|X|$. For example, we simply write $x \in X$ instead of $x \in |X|$. For measurable spaces X and Y, we say that a partial function $f\colon X \to Y$ (in this paper, we use $\to$ for both partial functions and total functions) is *measurable* when for all $A \in \Sigma_Y$, the inverse image:

$$\{x \in X \mid f(x) \text{ is defined and is equal to an element of } A\}$$

is in Σ_X. A *measurable function* is a totally defined partial measurable function. A partial measurable function $f\colon X \to Y$ is *invertible* when there is a (partial) measurable function $g\colon Y \to X$ such that $g \circ f$ and $f \circ g$ are identities. In this case, we say that f is an *isomorphism* from X to Y and say that X is *isomorphic* to Y.

We denote a singleton set $\{*\}$ by $\mathbf{1}$, and we regard the set $\mathbf{1}$ as a measurable space by endowing it with the trivial σ-algebra. We also regard the empty set $\emptyset$ as a measurable space in the obvious way. In this paper, $\mathbb{N}$ denotes the measurable set of all non-negative integers equipped with the σ-algebra consisting of *all subsets* of $\mathbb{N}$, and $\mathbb{R}$ denotes the measurable set of all real numbers equipped with the σ-algebra consisting of *Borel sets*, that is, the least σ-algebra that contains all open subsets of $\mathbb{R}$ (with respect to the standard topology).

When Y is a subset of the underlying set of a measurable space X, we can equip Y with a σ-algebra $\Sigma_Y = \{A \cap Y \mid A \in \Sigma_X\}$. This way, we regard the unit interval and the set of all non-negative real numbers as measurable spaces and indicate them as follows:

$$\mathbb{R}_{[0,1]} = \{a \in \mathbb{R} \mid 0 \le a \le 1\}, \qquad \mathbb{R}_{\ge 0} = \{a \in \mathbb{R} \mid a \ge 0\}.$$

For measurable spaces X and Y, we define the product measurable space $X \times Y$ and the coproduct measurable space $X + Y$ as follows:

$$|X \times Y| = |X| \times |Y|, \qquad |X + Y| = \{(\bullet, x) \mid x \in X\} \cup \{(\circ, y) \mid y \in Y\},$$

where the underlying σ-algebras are

$$\Sigma_{X\times Y} = \text{the least } \sigma\text{-algebra such that } A\times B\in\Sigma_{X\times Y} \text{ for all } A\in\Sigma_X \text{ and } B\in\Sigma_Y,$$
$$\Sigma_{X+Y} = \{\{\bullet\}\times A\cup\{\circ\}\times B \mid A\in\Sigma_X \text{ and } B\in\Sigma_Y\}.$$

For a finite family of measurable spaces $\{X_i\}_{1\le i\le n}$, we write $\sum_{1\le i\le n} X_i$ for $X_1+\cdots+X_n = (\cdots(X_1+X_2)+\cdots)+X_n$. When $n=0$, we define $\sum_{1\le i\le n} X_i$ to be $\emptyset$. We assume that $\times$ has higher precedence than $+$, that is, we write $X+Y\times Z$ for $X+(Y\times Z)$. In this paper, we always regard finite products $\mathbb{R}^n$ as the product measurable space on $\mathbb{R}$. It is well known that the σ-algebra $\Sigma_{\mathbb{R}^n}$ is the set of all Borel sets, that is, $\Sigma_{\mathbb{R}^n}$ is the least σ-algebra that contains all open subsets of $\mathbb{R}^n$. Partial measurable functions are closed under compositions, products and coproducts.

Let X be a measurable space. A *measure* μ on X is a function from Σ_X to $[0,\infty]$, that is the set of all non-negative real numbers extended with ∞, such that:

- $\mu(\emptyset)=0$; and
- for any mutually disjoint family $\{A_n\in\Sigma_X\}_{n\in\mathbb{N}}$, we have $\sum_{n\in\mathbb{N}}\mu(A_n)=\mu\left(\bigcup_{n\in\mathbb{N}}A_n\right)$.

We say that a measure μ on X is *finite* when $\mu(X)<\infty$. For a measurable space X, we write $\varnothing_X$ for a measure on X given by $\varnothing_X(A)=0$ for all $A\in\Sigma_X$. If μ is a measure on a measurable space X, then for any non-negative real number a, the function $(a\,\mu)(A)=a(\mu(A))$ is also a measure on X. The *Borel measure* μ_{Borel} on $\mathbb{R}^n$ is the unique measure that satisfies

$$\mu_{\text{Borel}}([a_1,b_1]\times\cdots\times[a_n,b_n]) = \prod_{1\le i\le n}|a_i-b_i|.$$

We define the Borel measure μ_{Borel} on $\mathbf{1}=\{*\}$ by $\mu_{\text{Borel}}(\mathbf{1})=1$. For a measurable function $f\colon\mathbb{R}^n\to\mathbb{R}$ and a measurable subset $X\subseteq\mathbb{R}^n$, we denote the integral of f with respect to the Borel measure restricted to X by:

$$\int_X f(x)\,\mathrm{d}x.$$

For a measurable space X and for an element $x\in X$, a *Dirac measure* δ_x on X is given by:

$$\delta_x(A)=[x\in A]=\begin{cases}1, & \text{if } x\in A;\\ 0, & \text{if } x\notin A.\end{cases}$$

The square bracket notation in the right-hand side is called Iverson's bracket. In general, for a proposition P, we have $[P]=1$ when P is true and $[P]=0$ when P is false.

Finally, let us recall the notion of a kernel, which is a well-known concept in the theory of stochastic processes. For measurable spaces X and Y, a *kernel* from X to Y is a function $k\colon X\times\Sigma_Y\to[0,\infty]$ such that for any $x\in X$, the function $k(x,-)$ is a measure on Y, and for any $A\in\Sigma_Y$, the function $k(-,A)$ is measurable. When k is a kernel from X to Y, we write $k\colon X\rightsquigarrow Y$. If there is $r>0$ such that $k(x,Y)<r$ for all $x\in X$, we say that k is a *finite* kernel. Those kernels which can be expressed as the sum of countably many finite kernels are said to be *s-finite* (Staton 2017). We use kernels to give semantics for our probabilistic programming language, to be defined in Section 7. Kernels can be composed as follows, by way of a number of constructions.

- The pointwise addition of a countable family of s-finite kernels $\{k_i\colon X\rightsquigarrow Y\}_{i\in I}$ is an s-finite kernel and is denoted by $\sum_{i\in I}k_i\colon X\rightsquigarrow Y$ or $k_0+k_1+\cdots$ when $I=\{0,1,\ldots\}$.
- Every partial measurable function $f\colon X\to Y$ gives rise to an s-finite kernel $f^{\#}\colon X\rightsquigarrow Y$ given by $f^{\#}(x,A)=[f(x)\in A]$.

- For s-finite kernels $k\colon X \rightsquigarrow Y$ and $h\colon Y \rightsquigarrow Z$, we define an s-finite kernel $h \circ k\colon X \rightsquigarrow Z$ by:

$$(h \circ k)(x, C) = \int h(y, C)\, k(x, \mathrm{d}y).$$

  The composition of s-finite kernels is associative (Staton 2017, Lemma 3) and satisfies the unit laws, namely, we have $k \circ \mathrm{id}_X^{\#} = k$ and $\mathrm{id}_X^{\#} \circ k = k$.
- Let $k\colon X \rightsquigarrow Y$ and $h\colon Z \rightsquigarrow W$ be s-finite kernels. We define s-finite kernels $k \otimes h\colon X \times Z \rightsquigarrow Y \times W$ and $k \oplus h\colon X + Z \rightsquigarrow Y + W$ by:

$$(k \otimes h)((x, z), A) = \int k(x, \mathrm{d}y) \int h(z, \mathrm{d}w)\, [(y, w) \in A],$$

$$(k \oplus h)(w, A) = \begin{cases} k(x, A_Y), & \text{if } w = (\bullet, x), \\ k(y, A_W), & \text{if } w = (\circ, y), \end{cases}$$

  where $A_Y = \{y \in Y \mid (\bullet, y) \in A\}$ and $A_W = \{w \in W \mid (\circ, w) \in A\}$. In the same way, for a family of s-finite kernels $\{k_n\colon X \rightsquigarrow Y\}_{n\in\mathbb{N}}$, we define an s-finite kernel $\bigoplus_{n\in\mathbb{N}} k_n\colon \mathbb{N} \times X \rightsquigarrow \mathbb{N} \times Y$ by:

$$\left(\bigoplus_{n\in\mathbb{N}} k_n\right)((n, x), A) = k_n(x, A_n) \text{ where } A_n = \{y \in Y \mid (n, y) \in A\}.$$

The tensor product and the coproduct of s-finite kernels are functorial, that is, these constructions are compatible with the composition and preserves identities. For functoriality of the tensor product, see Staton (2017, Proposition 5).

3. Syntax and Operational Semantics

3.1 Syntax and type system

Our language **PCFSS** for higher-order Bayesian programming can be seen as Plotkin's **PCF** endowed with real numbers, measurable functions, sampling from the uniform distribution on $\mathbb{R}_{[0,1]}$, and soft conditioning. We define types $\mathtt{A}, \mathtt{B}, \ldots$ values $\mathtt{V}, \mathtt{W}, \ldots$ and terms $\mathtt{M}, \mathtt{N}, \ldots$ as follows:

$$\mathtt{A}, \mathtt{B} ::= \mathtt{Unit} \mid \mathtt{Real} \mid \mathtt{A} \to \mathtt{B},$$

$$\mathtt{V}, \mathtt{W} ::= \mathtt{skip} \mid \mathtt{x} \mid \lambda \mathtt{x}^{\mathtt{A}}.\, \mathtt{M} \mid \mathtt{r}_a \mid \mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f}, \mathtt{x}, \mathtt{M}),$$

$$\mathtt{M}, \mathtt{N} ::= \mathtt{V} \mid \mathtt{V}\,\mathtt{W} \mid \mathtt{let\ x\ be\ M\ in\ N} \mid \mathtt{ifz}(\mathtt{V}, \mathtt{M}, \mathtt{N}) \mid \mathtt{F}(\mathtt{V}_1, \ldots, \mathtt{V}_{|\mathtt{F}|}) \mid \mathtt{sample} \mid \mathtt{score}(\mathtt{V}).$$

Here, $\mathtt{x}$ varies over a countably infinite set of variable symbols and a varies over the set $\mathbb{R}$ of all real numbers. Each *function identifier* $\mathtt{F}$ is supposed to have an arity $|\mathtt{F}|$ and is associated with an $|\mathtt{F}|$-ary measurable function $\overline{\mathtt{F}}$ from $\mathbb{R}^{|\mathtt{F}|}$ to $\mathbb{R}$. For terms $\mathtt{M}$ and $\mathtt{N}$, we write $\mathtt{M}\{\mathtt{N}/\mathtt{x}\}$ for the capture-avoiding substitution of $\mathtt{x}$ in $\mathtt{M}$ by $\mathtt{N}$. The recursively defined function $\mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f}, \mathtt{x}, \mathtt{M})$ is a value as in the standard call-by-value **PCF**.

Terms in **PCFSS** are restricted to be *A-normal forms* in order to make some of the arguments about our semantics simpler. This restriction is harmless to the language expressive power, thanks to the presence of `let`-bindings. For example, term application $\mathtt{M}\,\mathtt{N}$ can be defined to be $\mathtt{let\ x\ be\ M\ in\ (let\ y\ be\ N\ in\ (x\ y))}$.

The term constructor `score` and the constant `sample` enable probabilistic programming in **PCFSS**. Evaluation of $\mathtt{score}(\mathtt{r}_a)$ has the effect of multiplying the weight of the current probabilistic branch by $|a|$, this way enabling a form of *soft conditioning*. The constant `sample` generates a real number randomly drawn from the uniform distribution on $\mathbb{R}_{[0,1]}$. Only one sampling

$$\frac{\mathtt{x}:\mathtt{A}\in\Delta}{\Delta\vdash\mathtt{x}:\mathtt{A}} \qquad \frac{a\in\mathbb{R}}{\Delta\vdash \mathtt{r}_a:\mathtt{Real}} \qquad \frac{\Delta\vdash \mathtt{V}_1:\mathtt{Real} \quad \ldots \quad \Delta\vdash \mathtt{V}_{|\mathtt{F}|}:\mathtt{Real}}{\Delta\vdash \mathtt{F}(\mathtt{V}_1,\ldots,\mathtt{V}_{|\mathtt{F}|}):\mathtt{Real}}$$

$$\frac{\Delta\vdash \mathtt{V}:\mathtt{A}\to\mathtt{B} \quad \Delta\vdash \mathtt{W}:\mathtt{A}}{\Delta\vdash \mathtt{V}\,\mathtt{W}:\mathtt{B}} \qquad \frac{\Delta\vdash \mathtt{M}:\mathtt{B} \quad \Delta,\mathtt{x}:\mathtt{B}\vdash \mathtt{N}:\mathtt{A}}{\Delta\vdash \mathtt{let}\,\mathtt{x}\,\mathtt{be}\,\mathtt{M}\,\mathtt{in}\,\mathtt{N}:\mathtt{A}} \qquad \frac{\Delta,\ \mathtt{x}:\mathtt{A}\vdash \mathtt{M}:\mathtt{B}}{\Delta\vdash \lambda\mathtt{x}^{\mathtt{A}}.\mathtt{M}:\mathtt{A}\to\mathtt{B}}$$

$$\frac{\Delta,\ \mathtt{f}:\mathtt{A}\to\mathtt{B},\ \mathtt{x}:\mathtt{A}\vdash \mathtt{M}:\mathtt{B}}{\Delta\vdash \mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f},\mathtt{x},\mathtt{M}):\mathtt{A}\to\mathtt{B}} \qquad \frac{\Delta\vdash \mathtt{V}:\mathtt{Real} \quad \Delta\vdash \mathtt{M}:\mathtt{A} \quad \Delta\vdash \mathtt{N}:\mathtt{A}}{\Delta\vdash \mathtt{ifz}(\mathtt{V},\mathtt{M},\mathtt{N}):\mathtt{A}}$$

$$\frac{}{\Delta\vdash \mathtt{skip}:\mathtt{Unit}} \qquad \frac{}{\Delta\vdash \mathtt{sample}:\mathtt{Real}} \qquad \frac{\Delta\vdash \mathtt{V}:\mathtt{Real}}{\Delta\vdash \mathtt{score}(\mathtt{V}):\mathtt{Unit}}$$

Figure 1. Typing rules.

mechanism is sufficient, because we can model sampling from other standard distributions by composing `sample` with measurable functions (Wand et al. 2018).

Terms can be typed in a natural way. A *type environment* Δ is a finite sequence consisting of pairs of a variable and a type such that every variable appears in Δ at most once. A *type judgement* is a triple $\Delta\vdash \mathtt{M}:\mathtt{A}$ consisting of a type environment Δ, a term $\mathtt{M}$ and a type $\mathtt{A}$. We say that a type judgement $\Delta\vdash \mathtt{M}:\mathtt{A}$ is *derivable* when we can derive $\Delta\vdash \mathtt{M}:\mathtt{A}$ from the typing rules in Figure 1. Here, the type of `sample` is `Real`, and the type of `score`$(\mathtt{V})$ is `Unit` because `sample` randomly returns a real number, and what matters about scoring is its side effect. In the sequel, we only consider derivable type judgements and *typable* closed terms, that is, closed terms $\mathtt{M}$ such that $\vdash \mathtt{M}:\mathtt{A}$ is derivable for some type $\mathtt{A}$.

3.2 Distribution-based operational semantics

We define distribution-based operational semantics for **PCFSS** following Borgström et al. (2016) where, however, a σ-algebra on the set of terms is necessary so as to define evaluation results of terms to be distributions (i.e., measures) over values. In this paper, we only consider evaluation of terms of type `Real` and avoid introducing σ-algebras on sets of closed terms, thus greatly simplifying the overall development.

Distribution-based operational semantics sends a closed term $\mathtt{M}:\mathtt{Real}$ to a measure μ on $\mathbb{R}$. Because of the presence of `score`, the measure μ may not be a probabilistic measure, that is, $\mu(\mathbb{R})$ may be larger than 1, but the idea of distribution-based operational semantics is precisely that of associating each closed term of type `Real` with a probabilistic measure over $\mathbb{R}$.

As common in call-by-value programming languages, evaluation is defined by way of *evaluation contexts*:

$$\mathtt{E}[\,-\,] ::= [\,-\,] \;\mid\; \mathtt{let}\,\mathtt{x}\,\mathtt{be}\,\mathtt{E}[\,-\,]\,\mathtt{in}\,\mathtt{M}.$$

The distribution-based operational semantics of **PCFSS** is a family of binary relations $\{\Rightarrow_n\}_{n\in\mathbb{N}}$ between closed terms of type `Real` and measures on $\mathbb{R}$ inductively defined by the evaluation rules in Figure 3, where the evaluation rule for `score` is inspired from the one in Staton (2017). The binary relation $\xrightarrow{\text{red}}$ in the precondition of the third rule in Figure 3 is called *deterministic reduction* and is defined in Figure 2.

The last evaluation rule in Figure 3 makes sense because k in the precondition is a kernel from $\mathbb{R}_{[0,1]}$ to $\mathbb{R}$ as the following lemma shows.

Lemma 1. *For any $n\in\mathbb{N}$ and for any term:*

$$\mathtt{x}_1:\mathtt{Real},\ldots,\mathtt{x}_m:\mathtt{Real}\vdash \mathtt{M}:\mathtt{Real},$$

$$(\lambda \mathtt{x}^{\mathtt{A}}.\,\mathtt{M})\,\mathtt{V} \xrightarrow{\text{red}} \mathtt{M}\{\mathtt{V}/\mathtt{x}\} \qquad \mathtt{ifz}(\mathtt{r}_a, \mathtt{M}, \mathtt{N}) \xrightarrow{\text{red}} \begin{cases} \mathtt{M}, & \text{if } a = 0 \\ \mathtt{N}, & \text{if } a \neq 0 \end{cases}$$

$$\mathtt{let\ x\ be\ V\ in\ M} \xrightarrow{\text{red}} \mathtt{M}\{\mathtt{V}/\mathtt{x}\} \qquad \mathtt{F}(\mathtt{r}_a, \ldots \mathtt{r}_b) \xrightarrow{\text{red}} \mathtt{r}_{\overline{\mathtt{F}}(a,\ldots,b)}$$

$$\mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f},\mathtt{x},\mathtt{M})\,\mathtt{V} \xrightarrow{\text{red}} \mathtt{M}\{\mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f},\mathtt{x},\mathtt{M})/\mathtt{f}, \mathtt{V}/\mathtt{x}\}$$

Figure 2. Deterministic reduction.

$$\frac{}{\mathtt{M} \Rightarrow_0 \varnothing_{\mathbb{R}}} \qquad \frac{n > 0}{\mathtt{r}_a \Rightarrow_n \delta_a} \qquad \frac{\mathtt{M} \xrightarrow{\text{red}} \mathtt{N} \quad \mathtt{E[N]} \Rightarrow_n \mu}{\mathtt{E[M]} \Rightarrow_{n+1} \mu}$$

$$\frac{\mathtt{E[skip]} \Rightarrow_n \mu}{\mathtt{E[score(r}_a\mathtt{)]} \Rightarrow_{n+1} |a|\,\mu} \qquad \frac{\mathtt{E[r}_a\mathtt{]} \Rightarrow_n k(a,-) \quad \text{for all } a \in \mathbb{R}_{[0,1]}}{\mathtt{E[sample]} \Rightarrow_{n+1} \int_{\mathbb{R}_{[0,1]}} k(a,-)\,\mathrm{d}a}$$

Figure 3. Evaluation rules of distribution-based operational semantics.

there is a unique finite kernel k from $\mathbb{R}^m$ to $\mathbb{R}$ such that for any $(a_1, \ldots, a_m) \in \mathbb{R}^m$,

$$\mathtt{M}\{\mathtt{r}_{a_1}/\mathtt{x}_1, \ldots, \mathtt{r}_{a_m}/\mathtt{x}_m\} \Rightarrow_n k((a_1, \ldots, a_m), -).$$

Proof. The statement can be checked by induction on n. □

Lemma 1 implies that the step-indexed distribution-based operational semantics $\Rightarrow_n$ can be seen as inducing an $\mathbb{N}$-indexed family of functions from the set of closed terms of type `Real` to the set of finite measures on $\mathbb{R}$, approximating the evaluation of closed terms by restricting the number of reduction steps. Thus, the limit of the step-indexed distribution-based operational semantics represents the 'true' result of evaluating the underlying term:

Definition 2. *For a closed term $\vdash \mathtt{M} : \mathtt{Real}$ and a measure μ on $\mathbb{R}$, we write $\mathtt{M} \Rightarrow_\infty \mu$ when there is a family of measures $\{\mu_n\}_{n\in\mathbb{N}}$ on $\mathbb{R}$ such that $\mathtt{M} \Rightarrow_n \mu_n$ and for all $A \in \Sigma_{\mathbb{R}}$,*

$$\mu(A) = \sup_{n\in\mathbb{N}} \mu_n(A).$$

The binary relation $\Rightarrow_\infty$ is a function from the set of closed terms of type `Real` to the set of measures on $\mathbb{R}$. This follows from Lemma 1 and from the easy observation that the family of measures $\{\mu_n\}_{n\in\mathbb{N}}$ on $\mathbb{R}$ such that $\mathtt{M} \Rightarrow_n \mu_n$ forms an ascending chain $\mu_0 \leq \mu_1 \leq \cdots$ with respect to the pointwise order. Moreover, it can be proved that for any $\mathtt{x}_1 : \mathtt{Real}, \ldots, \mathtt{x}_m : \mathtt{Real} \vdash \mathtt{M} : \mathtt{Real}$, there is an s-finite kernel k given by $\mathtt{M}\{\mathtt{r}_{a_1}/\mathtt{x}_1, \ldots, \mathtt{r}_{a_m}/\mathtt{x}_m\} \Rightarrow_\infty k((a_1, \ldots, a_m), -)$.

3.3 Sampling-based operational semantics

PCFSS can be endowed with another form of operational semantics, closer in spirit to the way inference algorithms see probabilistic programs, called the *sampling-based* operational semantics. The way we formulate it is deeply inspired from the one in Borgström et al. (2016).

The idea behind sampling-based operational semantics is to give the evaluation result of each probabilistic branch somehow independently. We specify each probabilistic branch by two parameters: one is a sequence of random draws, which will be consumed by `sample`, and the other is a likelihood measure called weight, which will be modified by `score`.

Definition 3. *A* configuration *is a triple $(\mathtt{M}, a, u)$ consisting of a closed term $\mathtt{M} : \mathtt{Real}$, a real number $a \geq 0$ called the configuration's* weight *and a finite sequence u of real numbers in $\mathbb{R}_{[0,1]}$, called its* trace.

$$\frac{\mathtt{M} \xrightarrow{\text{red}} \mathtt{N}}{(\mathtt{M}, b, u) \to (\mathtt{N}, b, u)}$$

$$(\mathtt{E}[\mathtt{score}(\mathtt{r}_a)], b, u) \to (\mathtt{E}[\mathtt{skip}], |a|\, b, u) \qquad (\mathtt{E}[\mathtt{sample}], a, b :: u) \to (\mathtt{E}[\mathtt{r}_b], a, u)$$

Figure 4. Evaluation rules of sampling-based operational semantics.

Below, we write ε for the empty sequence. For a real number a and a finite sequence u consisting of real numbers, we write $a :: u$ for the finite sequence obtained by putting a on the head of u. In Figure 4, we give the evaluation rules of sampling-based operational semantics that is a binary relation between configurations. In the definition, $\xrightarrow{\text{red}}$ is the deterministic reduction relation introduced in Figure 2. We denote the reflective transitive closure of $\to$ by $\to^*$. Intuitively, $(\mathtt{M}, 1, u) \to^* (\mathtt{r}_a, b, \varepsilon)$ means that by evaluating $\mathtt{M}$, we get the real number a with *weight* b consuming *all* the random draws in u.

4. Towards Mealy Machine Semantics

In this section, we give some intuitions about our GoI model based on partial measurable functions, which we also call *Mealy machine semantics*. Schematically, Mealy machine semantics for **PCFSS** translates terms in **PCFSS** into Mealy machines in the following way:

$$\begin{array}{c}\textbf{PCFSS}\\ \downarrow \text{Moggi's translation}\\ \text{Moggi's meta-language} + \mathtt{sample} + \mathtt{score}\\ \downarrow \text{Girard translation}\\ \text{the linear } \lambda\text{-calculus} + \mathtt{sample} + \mathtt{score}\\ \downarrow\\ \text{proof structures} + \mathtt{sample} + \mathtt{score}\\ \downarrow \text{GoI}\\ \text{Mealy machines}\end{array}$$

In Section 4.1, we explain the first three steps. The last step deserves to be explained in more detail, which we do in Section 4.2. For the sake of simplicity, we ignore the translation of conditional branching and the fixed point operator. A more technical presentation of the GoI model is deferred to Section 5.

4.1 *From* PCFSS *to proof structures*

4.1.1 *Moggi's translation*

In the first step, we translate **PCFSS** into an extension of the Moggi's meta-language by Moggi's translation (Moggi 1991), whose only type constructor is the function type. Here, in order to translate scoring and sampling in **PCFSS**, we equip Moggi's meta-language with base types $\mathtt{Unit}$ and $\mathtt{Real}$ and the following term constructors:

$$\frac{a \in \mathbb{R}}{\Delta \vdash \mathtt{r}_a : \mathtt{Real}}, \qquad \frac{\Delta \vdash \mathtt{M} : \mathtt{Real}}{\Delta \vdash \mathtt{score}(\mathtt{M}) : \mathtt{T}\,\mathtt{Unit}}, \qquad \frac{}{\Delta \vdash \mathtt{sample} : \mathtt{T}\,\mathtt{Real}}$$

where $\mathtt{T}$ is the monad of Moggi's meta-language. Any type $\mathtt{A}$ of **PCFSS** is translated into the type $\mathtt{A}^\sharp$ defined as follows:

$$\mathtt{Unit}^\sharp = \mathtt{Unit}, \qquad \mathtt{Real}^\sharp = \mathtt{Real}, \qquad (\mathtt{A} \to \mathtt{B})^\sharp = \mathtt{A}^\sharp \to \mathtt{T}\,\mathtt{B}^\sharp.$$

Terms $\mathtt{score}(-)$ and $\mathtt{sample}$ in **PCFSS** are translated into the eponymous $\mathtt{score}(-)$ and $\mathtt{sample}$ in Moggi's meta-language, respectively. See Moggi (1991) for more detail about Moggi's translation.

4.1.2 Girard translation

We next translate the extended Moggi's meta-language into an extension of the linear λ-calculus, by way of the so-called Girard translation (Girard 1987). Types are given by:

$$\mathtt{A}, \mathtt{B} ::= \mathtt{Unit} \mid \mathtt{Real} \mid \mathtt{State} \mid \mathtt{A}^{\perp} \mid \mathtt{A} \otimes \mathtt{B} \mid \mathtt{A} \parr \mathtt{B} \mid \,!\mathtt{A}$$

where $\mathtt{Unit}$, $\mathtt{Real}$ and $\mathtt{State}$ are base types, and terms are generated by the standard term constructors of the linear λ-calculus, plus the following rules:

$$\frac{a \in \mathbb{R}}{\vdash \mathtt{r}_a : \mathtt{Real}}, \quad \frac{\Delta \vdash \mathtt{M} : \mathtt{Real}}{\Delta \vdash \mathtt{score}(\mathtt{M}) : \mathtt{State} \multimap \mathtt{State} \otimes\, !\mathtt{Unit}}, \quad \frac{}{\vdash \mathtt{sample} : \mathtt{State} \multimap \mathtt{State} \otimes\, !\mathtt{Real}}.$$

(As customary in linear logic, $\mathtt{A} \multimap \mathtt{B}$ is an abbreviation of $\mathtt{A}^{\perp} \parr \mathtt{B}$.) These typing rules are derived from the following translation $(-)^\flat$ of types of the extended Moggi's meta-language into types of the extended linear λ-calculus:

$$\mathtt{Unit}^\flat = \mathtt{Unit}, \quad \mathtt{Real}^\flat = \mathtt{Real}, \quad (\mathtt{A} \to \mathtt{B})^\flat = \,!\mathtt{A}^\flat \multimap \mathtt{B}^\flat, \quad (\mathtt{T}\,\mathtt{A})^\flat = \mathtt{State} \multimap \mathtt{State} \otimes\, !\mathtt{A}^\flat.$$

The definition of $(\mathtt{T}\,\mathtt{A})^\flat$, which lift the monad T to a state passing construct, is motivated by the following categorical observation: let $\mathbf{L}$ be the syntactic category of the extended linear λ-calculus, which is a symmetric monoidal closed category endowed with a comonad $!\colon \mathbf{L} \to \mathbf{L}$ with certain conditions (see e.g., Hyland and Schalk (2003)), and let $\mathbf{L}_!$ be the coKleisli category of the comonad $!$. Then, by composing the adjunction between $\mathbf{L}$ and $\mathbf{L}_!$ with a state monad $\mathtt{State} \multimap \mathtt{State} \otimes (-)$ on $\mathbf{L}$, we obtain a monad on $\mathbf{L}_!$:

$$\mathtt{State} \multimap \mathtt{State} \otimes (-) \;\circlearrowright\; \mathbf{L} \;\underset{\curvearrowleft}{\overset{\curvearrowright}{\top}}\; \mathbf{L}_!\,,$$

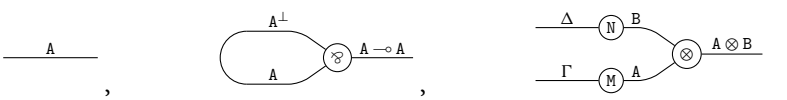
which sends an object $\mathtt{A} \in \mathbf{L}_!$ to $\mathtt{State} \multimap \mathtt{State} \otimes\, !\mathtt{A}$. This use of the state monad is motivated by sampling-based operational semantics: we can regard **PCFSS** as a call-by-value λ-calculus with global states consisting of pairs of a non-negative real number and a finite sequence of real numbers, and we can regard $\mathtt{score}$ and $\mathtt{sample}$ as effectful operations interacting with those states. Following this line, we model call-by-value evaluation by the state monad encoded in the linear type system.

4.1.3 The third step

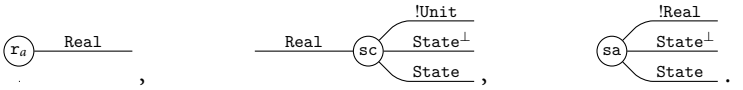
We translate terms in the extended linear λ-calculus into (an extension of) proof structures (Lafont 1995), which are graphical presentations of type derivation trees of linear λ-terms. We can also understand proof structures as string diagrams for compact closed categories (Selinger 2011). Operators of the pure linear λ-calculus can be translated as usual (Lafont 1995). For example, type derivation trees:

$$\frac{}{\mathtt{x} : \mathtt{A} \vdash \mathtt{x} : \mathtt{A}}, \qquad \frac{\dfrac{}{\mathtt{x} : \mathtt{A} \vdash \mathtt{x} : \mathtt{A}}}{\vdash \lambda \mathtt{x}^{\mathtt{A}}.\, \mathtt{x} : \mathtt{A} \multimap \mathtt{A}}, \qquad \frac{\Gamma \vdash \mathtt{M} : \mathtt{A} \quad \Delta \vdash \mathtt{N} : \mathtt{B}}{\Gamma, \Delta \vdash \mathtt{M} \otimes \mathtt{N} : \mathtt{A} \otimes \mathtt{B}}$$

are translated into proof structures:



respectively. Terms of the form $\mathtt{r}_a$, $\mathtt{score}(\mathtt{M})$ and $\mathtt{sample}$ require new kinds of nodes:

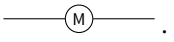


This does not directly reflect typing rules for `score` and `sample` in the linear λ-calculus, but the correspondence can be recovered by way of multiplicative nodes ⊗ and ⅋.

4.2 From proof structures to Mealy machines

The series of translations from **PCFSS** to proof structures is agnostic as for the computational meaning of `score` and `sample` in **PCFSS**, because `score` and `sample` introduced in these translations are just constant symbols. In other words, the translation from **PCFSS** to the extended proof structures is not sound with respect to either form of operational semantics for **PCFSS**. In the last translation step, we assign proof structures a computational meaning, respecting the operational semantics of the underlying **PCFSS** term.

We do this by associating proof structures with Mealy machines. A *Mealy machine* is an input/output machine whose evolution may depend on its current state. In this paper, for the sake of supporting intuition and of enabling graphical reasoning, we depict a Mealy machine M as a node with some wires:

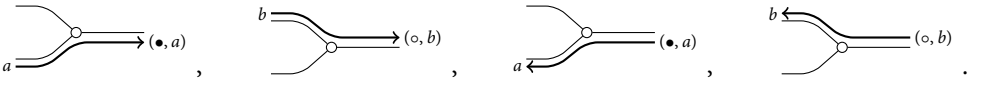


Inputs to this Mealy machine are given through one of its wires, and to each input, the Mealy machine gives an output through one of its wires. We indicate how the Mealy machine handles inputs by thick arrows like

For example, for the case of the left diagram, the thick arrow indicates that if the current state is s and the given input is x, then the Mealy machine outputs y and changes its state to t.

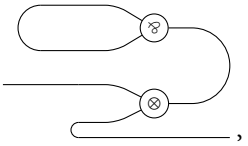
For the standard proof structures, we can follow Laurent (2001) where Mealy machines associated with proof structures are built up from Mealy machines associated with each nodes. For example, the ⊗-node and the ⅋-node:

are both associated with a one-state Mealy machine that behaves in the following manner:

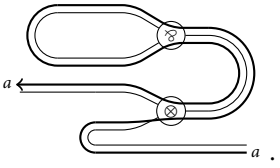


Namely, the Mealy machine forwards each input from the left-hand side to the right-hand side endowing it with a tag that tells where the input came from. The Mealy machine handles inputs from the right-hand side in the reverse way.

Soundness of Mealy machine semantics states that if two (pure) linear λ-terms are β-equivalent, then the behaviours of the Mealy machines associated with these terms are the same. As an example, let us consider a β-reduction step $(\lambda \mathtt{x}.\, \mathtt{x})\, \mathtt{y} \to \mathtt{y}$. The proof structure associated with $(\lambda \mathtt{x}.\, \mathtt{x})\, \mathtt{y}$ is



and the following thick arrow illustrates a trace of a run of this Mealy machine for an input a from the right wire:



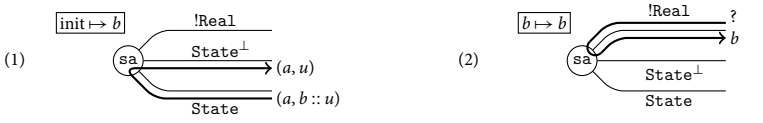
In general, this Mealy machine forwards any input from the right-hand side to the left-hand side as indicated by the thick arrow, and it also forwards any input from the left-hand side to the right-hand side. Hence, the behaviour of this Mealy machine is equivalent to the behaviour of the following trivial Mealy machine:

which is the interpretation of y. This is in fact a symptom of a general phenomenon: Mealy machine semantics for the linear λ-calculus captures β-reduction.

But how can we extend this Mealy machine semantics to `score` and `sample`? Here, we borrow an idea from Game semantics (Abramsky and McCusker 1996), which models computation in terms of interaction between programs and environments. For scoring and sampling, we can infer how they interact with the environment from sampling-based operational semantics. For scoring, we associate `score` with a Mealy machine that has the following transitions:

where u is a finite sequence of real numbers in $\mathbb{R}_{[0,1]}$ and a and b are real numbers such that $a \geq 0$. We can read these transitions as follows: (1) in the initial state init, for each 'configuration' (a, u), the Mealy machine sends a query ? to environment in order to know the value of its argument and memorises the configuration (a, u) by changing its state from init to (a, u); (2) if the environment answers that the value is b, that is, if the Mealy machine receives b, then it outputs $(|b|\, a, u)$. This process corresponds to the evaluation rule $(\texttt{score}(\texttt{r}_b), a, u) \to (\texttt{skip}, |b|\, a, u)$.

The following diagrams explain our idea of modelling `sample`:

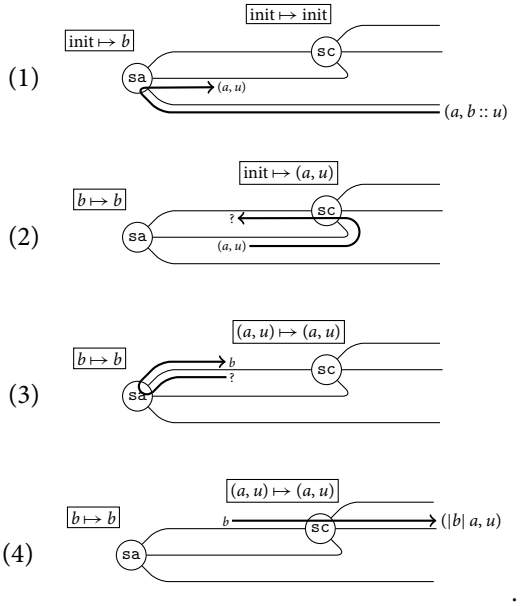


where $b :: u$ is a finite sequence of real numbers in $\mathbb{R}_{[0,1]}$ and a is a non-negative real number. (To be precise, this is slightly different from how we model `sample`. See Section 3.3.) The first transition (1) means that in the initial state init, given a 'configuration' $(a, b :: u)$, the Mealy machine

pops b and memorises the value b by changing its state from init to b. In the second transition (2), for any query ? asking the result of sampling, it answers the value memorised in the first transition. For example, the execution:

$$(\texttt{let x be sample in score(x)}, a, b :: u) \to^* (\texttt{score}(\texttt{r}_b), a, u) \to^* (\texttt{skip}, |b|\ a, u)$$

is modelled as the following sequential process:



Here, the boxes next to sa tells how sa changes its state, and the boxes next to sc tells how sc changes its state. This is why we have two 'initial states' in the first stage. The above sequential process can be explained as follows: (1) given a 'configuration' $(a, b :: u)$, the node sa memorises the value b and forwards the rest (a, u) to sc. (2) Then sc requests the value of its argument, and (3) sa returns b to the request. (4) Finally, sc returns $(|b|\ a, u)$ to environment. (Formally, this Mealy machine is slightly different from the Mealy machine given by our denotational semantics. But the difference is just a matter of bureaucracy.) In this interaction process, the memoisation mechanism of the sa-node is necessary, otherwise the sa-node can not tell the sc-node that the sampling result is b.

Remark 4. Two notions of *state* (the one coming from sc and the other coming from sa) are used for different purpose here: the first notion is needed to model the call-by-value evaluation strategy where we need to store intermediate effects that are invoked during the evaluation. The second notion of state is needed to model history-dependent computation. More concretely, for the case of sampling, the Mealy machine sa needs to remember the already sampled values in the current probabilistic branch.

5. Mealy Machines and their Compositions

After having described Mealy machine semantics briefly and informally, it is now time to get more precise. In this section, we introduce the notion of a Mealy machine and some constructions on Mealy machines. We also introduce a way of diagrammatically presenting Mealy machines which is behaviourally sound.

5.1 Mealy machines, formally

In this paper, we call a pair of measurable spaces an **Int**-*object*. We use sans-serif capital letters $\mathsf{X}, \mathsf{Y}, \mathsf{Z}, \ldots$ to denote **Int**-objects, and we denote the positive/negative part of an **Int**-object by the same italic letter superscripted by $+/-$. For example, X denotes an **Int**-object (X^+, X^-) consisting of two measurable spaces X^+ and X^-. The name '**Int**-object' comes from the so-called **Int**-*construction* (Joyal et al. 1996).

Definition 5. *For* **Int**-*objects* X *and* Y*, a* Mealy machine M *from* X *to* Y *consists of*

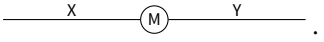
- *a measurable space* S_{M} *called the* state space *of* M*;*
- *an element* $\mathrm{init}_{\mathsf{M}} \in S_{\mathsf{M}}$ *called the* initial state *of* M*;*
- *a partial measurable function:*

$$\tau_{\mathsf{M}} \colon (X^+ + Y^-) \times S_{\mathsf{M}} \to (X^- + Y^+) \times S_{\mathsf{M}}$$

called the transition function *of* M.

If M *is a Mealy machine from* X *to* Y*, we write* $\mathsf{M} \colon \mathsf{X} \multimap \mathsf{Y}$.

The transition function τ_{M} of a Mealy machine $\mathsf{M} \colon \mathsf{X} \multimap \mathsf{Y}$ describes a mapping between inputs and outputs which can also alter the underlying state. For an input $x \in X^+ + Y^-$ and a state $s \in S_{\mathsf{M}}$, a transition $\tau_{\mathsf{M}}(x, s) = (y, t)$ means that when the current state of M is s, given an input x, there is an output y and the next state is t. Note that we have X^- in the target and Y^- in the source of τ_{M}. In short, this is because we are interested in Mealy machines that handle bidirectional computation. The diagrammatic presentation of Mealy machines clarifies the meaning of 'bidirectional'. Let $\mathsf{M} \colon \mathsf{X} \multimap \mathsf{Y}$ be a Mealy machine. In this paper, we depict M as follows:



In the GoI jargon, data travelling along wires of proof structures are often called *tokens*. Intuitively, each label on a wire indicates the type of tokens travelling along the wire. Namely, on the X-wire (on the Y-wire), tokens in X^+ (in Y^+) go from left to right, and tokens in X^- (in Y^-) go from right to left. For example, we depict the following transitions:

$$\tau_{\mathsf{M}}((\circ, y), s_0) = ((\bullet, x), s_1), \qquad \tau_{\mathsf{M}}((\circ, y'), s_0) = ((\circ, y''), s_2)$$

for some $y, y' \in Y^-$, $x \in X^-$, $y'' \in Y^+$ and $s_0, s_1, s_2 \in S_{\mathsf{M}}$ as the following thick arrows:

(Recall that the tag $\bullet/\circ$ indicates the left/right part of the disjoint sum.) The expressions $s_0 \mapsto s_1$ and $s_0 \mapsto s_2$ on the Mealy machine M stand for transitions of states. We omit states transitions when they can be inferred from the context.

We will give some Mealy machines whose state spaces are trivial, namely $\mathbf{1}$. We call such a Mealy machine *token machine*. Our usage of the term token machine is along the lines of that in other papers on GoI such as Mackie (1995) and Laurent (2001). In order to specify a token machine, it is enough to give a partial measurable function of the following form:

$$X^+ + Y^- \cong (X^+ + Y^-) \times \mathbf{1} \longrightarrow (X^- + Y^+) \times \mathbf{1} \cong X^- + Y^+.$$

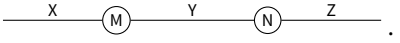
Therefore, in the sequel, we define a token machine $\mathsf{M} \colon \mathsf{X} \multimap \mathsf{Y}$ by giving a partial measurable function from $X^+ + Y^-$ to $Y^+ + X^-$, and we also call this partial measurable function the *transition function* of M. Abusing notation, we write τ_{M} for this transition function.

5.2 Constructions on mealy machines

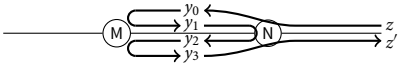
It is now time to give some constructions which are the basic building blocks of our Mealy machine semantics. This section consists of three parts. The first part (from Sections 5.2.1 to 5.2.5) is related to the linear λ-calculus and serves to model the purely functional features of **PCFSS**. In the second part (from Sections 5.2.6 to 5.2.8), we give Mealy machines modelling real numbers, measurable functions and conditional branching. In the last part (Sections 5.2.9 and 5.2.10), we introduce Mealy machines modelling `score` and `sample`.

5.2.1 Composition

Let X, Y and Z be **Int**-objects, and let $\mathsf{M}\colon \mathsf{X} \multimap \mathsf{Y}$, $\mathsf{N}\colon \mathsf{Y} \multimap \mathsf{Z}$ be Mealy machines. We can now define their composition $\mathsf{N} \circ \mathsf{M}\colon \mathsf{X} \multimap \mathsf{Z}$. Before giving a precise definition, some intuitive explanation about $\mathsf{N} \circ \mathsf{M}$ is in order. The main idea is to define $\mathsf{N} \circ \mathsf{M}$ as a Mealy machine obtained by connecting N and M in the following manner:



The series of thick arrows in the following diagram:



illustrates an execution of the obtained Mealy machine. Given an input from a wire, M and N engage in some interactive communication, and at some point, some output is produced. Because $\mathsf{N} \circ \mathsf{M}$ performs 'parallel composition plus connecting', the state space of $\mathsf{N} \circ \mathsf{M}$ should be $S_{\mathsf{N}} \times S_{\mathsf{M}}$, and the initial state should be $(\mathrm{init}_{\mathsf{N}}, \mathrm{init}_{\mathsf{M}})$. The transition function of $\mathsf{N} \circ \mathsf{M}$ should be given by the collection of all possible interaction paths between M and N.

Let us give a precise definition. For Mealy machines $\mathsf{M}\colon \mathsf{X} \multimap \mathsf{Y}$ and $\mathsf{N}\colon \mathsf{Y} \multimap \mathsf{Z}$, we define the state space and the initial states of $\mathsf{N} \circ \mathsf{M}$ by:

$$S_{\mathsf{N}\circ\mathsf{M}} = S_{\mathsf{N}} \times S_{\mathsf{M}}, \qquad \mathrm{init}_{\mathsf{N}\circ\mathsf{M}} = (\mathrm{init}_{\mathsf{N}}, \mathrm{init}_{\mathsf{M}})$$

and we define the transition function $\tau_{\mathsf{N}\circ\mathsf{M}}$ by:

$$\tau_{\mathsf{N}\circ\mathsf{M}} = f_{X^+,Z^-,X^-,Z^+} \vee \bigvee_{n\in\mathbb{N}} f_{Y^+,Y^-,X^-,Z^+} \circ f^n_{Y^+,Y^-,Y^+,Y^-} \circ f_{X^+,Z^-,Y^+,Y^-}$$

where the measurable functions $f_{A,B,C,D}\colon (A+B) \times S_{\mathsf{N}\circ\mathsf{M}} \to (C+D) \times S_{\mathsf{N}\circ\mathsf{M}}$ are restrictions of the following partial measurable function:

$$\begin{array}{c}
(X^+ + Z^- + Y^+ + Y^-) \times S_{\mathsf{N}\circ\mathsf{M}} \\
\downarrow\cong \\
(X^+ + Y^-) \times S_{\mathsf{M}} \times S_{\mathsf{N}} + (Y^+ + Z^-) \times S_{\mathsf{N}} \times S_{\mathsf{M}} \\
\downarrow \tau_{\mathsf{M}} \times S_{\mathsf{N}} + \tau_{\mathsf{N}} \times S_{\mathsf{M}} \\
(X^- + Y^+) \times S_{\mathsf{M}} \times S_{\mathsf{N}} + (Y^- + Z^+) \times S_{\mathsf{N}} \times S_{\mathsf{M}} \\
\downarrow\cong \\
(X^- + Z^+ + Y^+ + Y^-) \times S_{\mathsf{N}\circ\mathsf{M}}\ ,
\end{array}$$

and the join in the definition of $\tau_{\mathsf{N}\circ\mathsf{M}}$ is with respect to the inclusion order between graph relations. It is tedious but doable to check that the above join defines a partial measurable function.

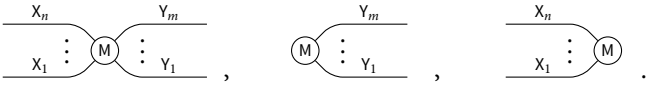
5.2.2 Monoidal products

Monoidal Products of Int-objects We introduce monoidal products of **Int**-objects and their diagrammatic presentation. For **Int**-objects X and Y, we define an **Int**-object $\mathsf{X} \otimes \mathsf{Y}$ by:

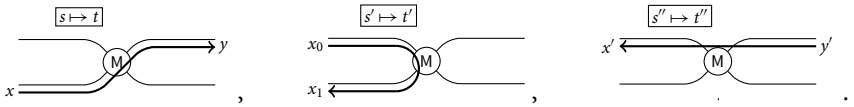
$$\mathsf{X} \otimes \mathsf{Y} = (X^+ + Y^+, X^- + Y^-).$$

We define an **Int**-object I to be $(\emptyset, \emptyset)$. We write $\mathsf{X} \otimes \mathsf{Y} \otimes \mathsf{Z} \otimes \cdots$ for $(\cdots((\mathsf{X} \otimes \mathsf{Y}) \otimes \mathsf{Z}) \otimes \cdots)$. For a finite family of **Int**-objects $\{\mathsf{X}_i\}_{1 \leq i \leq n}$, we write $\bigotimes_{1 \leq i \leq n} \mathsf{X}_i$ for $\mathsf{X}_1 \otimes \mathsf{X}_2 \otimes \cdots \otimes \mathsf{X}_n$. When $n = 0$, we define $\bigotimes_{1 \leq i \leq n} \mathsf{X}_i$ to be I.

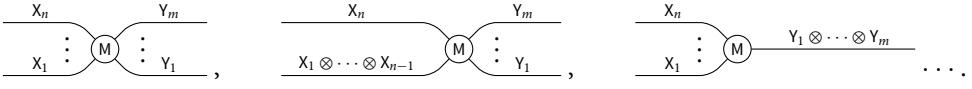
Let $\mathsf{X}_1, \ldots, \mathsf{X}_n, \mathsf{Y}_1, \ldots, \mathsf{Y}_m$ be **Int**-objects. We depict a Mealy machine M from $\mathsf{X}_1 \otimes \cdots \otimes \mathsf{X}_n$ to $\mathsf{Y}_1 \otimes \cdots \otimes \mathsf{Y}_m$ as a node with wires labelled by $\mathsf{X}_1, \ldots, \mathsf{X}_n$ on the left-hand side and wires labelled by $\mathsf{Y}_1, \ldots, \mathsf{Y}_m$ on the right-hand side, and we sometimes omit wires on the left-/right-hand side when the domain/codomain of M is I:



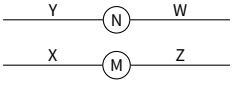
The diagrammatic presentation of monoidal products allows for intuitive description of transition functions:



For example, the first transition corresponds to $\tau_{\mathsf{M}}((\bullet, (\bullet, x)), s) = ((\circ, (\circ, y)), t)$. We note that there are several ways to present a Mealy machine $\mathsf{M} \colon \mathsf{X}_1 \otimes \cdots \otimes \mathsf{X}_n \multimap \mathsf{Y}_1 \otimes \cdots \otimes \mathsf{Y}_m$ such as

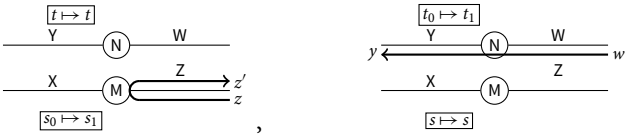


Monoidal Product of Mealy Machines We give monoidal products of Mealy machines. Intuitively, the monoidal product $\mathsf{M} \otimes \mathsf{N} \colon \mathsf{X} \otimes \mathsf{Y} \multimap \mathsf{Z} \otimes \mathsf{W}$ of Mealy machines $\mathsf{M} \colon \mathsf{X} \multimap \mathsf{Z}$ and $\mathsf{N} \colon \mathsf{Y} \multimap \mathsf{W}$ is given by parallel composition:



consisting of two sub-machines M and N working independently. For example, if we have

then $\mathsf{M} \otimes \mathsf{N}$ has the following transitions for any $t \in S_{\mathsf{N}}$ and any $s \in S_{\mathsf{M}}$:



Formally, we define $\mathsf{M} \otimes \mathsf{N} : \mathsf{X} \otimes \mathsf{Y} \multimap \mathsf{Z} \otimes \mathsf{W}$ by:

$$S_{\mathsf{M}\otimes\mathsf{N}} = S_{\mathsf{M}} \times S_{\mathsf{N}}, \qquad \mathrm{init}_{\mathsf{M}\otimes\mathsf{N}} = (\mathrm{init}_{\mathsf{M}}, \mathrm{init}_{\mathsf{N}})$$

and

$$\tau_{\mathsf{M}\otimes\mathsf{N}} = \begin{array}{c} ((X^+ + Y^+) + (Z^- + W^-)) \times S_{\mathsf{M}\otimes\mathsf{N}} \\ \downarrow\cong \\ (X^+ + Z^-) \times S_{\mathsf{M}} \times S_{\mathsf{N}} + (Y^+ + W^-) \times S_{\mathsf{N}} \times S_{\mathsf{M}} \\ \downarrow\tau_{\mathsf{M}} \times S_{\mathsf{N}} + \tau_{\mathsf{N}} \times S_{\mathsf{M}} \\ (X^- + Z^+) \times S_{\mathsf{M}} \times S_{\mathsf{N}} + (Y^- + W^+) \times S_{\mathsf{N}} \times S_{\mathsf{M}} \\ \downarrow\cong \\ ((X^- + Y^-) + (Z^+ + W^+)) \times S_{\mathsf{M}\otimes\mathsf{N}} \ . \end{array}$$

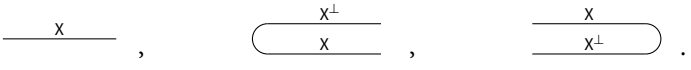
Below, for brevity, we sometimes identity **Int**-objects $\mathsf{X} \otimes (\mathsf{Y} \otimes \mathsf{Z})$ with $(\mathsf{X} \otimes \mathsf{Y}) \otimes \mathsf{Z}$ by the canonical isomorphism $A + (B + C) \cong (A + B) + C$ and identify $\mathsf{I} \otimes \mathsf{X}$ and $\mathsf{X} \otimes \mathsf{I}$ with X by the unit laws $A + \emptyset \cong A$ and $\emptyset + A \cong A$.

5.2.3 Identity, axiom link and cut link

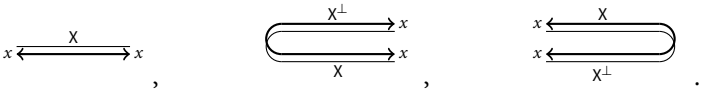
For an **Int**-object X, let an **Int**-object $\mathsf{X}^\perp$ be (X^-, X^+). We define token machines $\mathsf{id}_{\mathsf{X}} : \mathsf{X} \multimap \mathsf{X}$, $\mathsf{ax}_{\mathsf{X}} : \mathsf{I} \multimap \mathsf{X} \otimes \mathsf{X}^\perp$ and $\mathsf{cut}_{\mathsf{X}} : \mathsf{X}^\perp \otimes \mathsf{X} \multimap \mathsf{I}$ by:

$$\begin{array}{lll} \tau_{\mathsf{id}_{\mathsf{X}}}(\bullet, x) = (\circ, x), & \tau_{\mathsf{ax}_{\mathsf{X}}}(\circ, (\bullet, x)) = (\circ, (\circ, x)), & \tau_{\mathsf{cut}_{\mathsf{X}}}(\bullet, (\bullet, x)) = (\bullet, (\circ, x)), \\ \tau_{\mathsf{id}_{\mathsf{X}}}(\circ, x) = (\bullet, x), & \tau_{\mathsf{ax}_{\mathsf{X}}}(\circ, (\circ, x)) = (\circ, (\bullet, x)), & \tau_{\mathsf{cut}_{\mathsf{X}}}(\bullet, (\circ, x)) = (\bullet, (\bullet, x)). \end{array}$$

Note that $\tau_{\mathsf{ax}_{\mathsf{X}}}$ and $\tau_{\mathsf{cut}_{\mathsf{X}}}$ are partial measurable functions from $\emptyset + (X^+ + X^-)$ to $\emptyset + (X^- + X^+)$ and from $(X^- + X^+) + \emptyset$ to $(X^+ + X^-) + \emptyset$, respectively. We depict these Mealy machines by single wires:

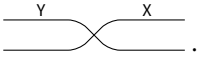


This is compatible with behaviour of these Mealy machines. As the following thick arrows indicate, all tokens travel along wires:



5.2.4 Symmetry

Let X and Y be **Int**-objects. We define a token machine $\mathsf{sym}_{\mathsf{X},\mathsf{Y}} : \mathsf{X} \otimes \mathsf{Y} \multimap \mathsf{Y} \otimes \mathsf{X}$ by letting its transition function be the canonical isomorphism between $(X^+ + Y^+) + (Y^- + X^-)$ and $(X^- + Y^-) + (Y^+ + X^+)$. We depict $\mathsf{sym}_{\mathsf{X},\mathsf{Y}}$ by a crossing:



As the diagram indicates, given an input token from an wire in the one side, $\mathsf{sym}_{\mathsf{X},\mathsf{Y}}$ outputs the same value to the corresponding wire on the other side:

5.2.5 A resource modality

We give a constructor on Mealy machines that corresponds to the resource modality in linear logic. The purpose of the resource modality is to enable discarding and duplicating resources. Here, resources are **Int**-objects.

For an **Int**-object X, we define an **Int**-object $!\mathsf{X}$ by:

$$!\mathsf{X} = (\mathbb{N} \times X^+, \mathbb{N} \times X^-)$$

where $\mathbb{N}$ is equipped with the discrete σ-algebra. Informally, we can regard $!\mathsf{X}$ as a countable monoidal product $\mathsf{X} \otimes \mathsf{X} \otimes \cdots \approx (X^+ + X^+ + \cdots, X^- + X^- + \cdots)$. Following this intuition, we extend the action of $!(-)$ to Mealy machines. Let $\mathsf{M}\colon \bigotimes_{1\le i\le n} \mathsf{X}_i \multimap \mathsf{Y}$ be a Mealy machine. We define $!_n\mathsf{M}\colon \bigotimes_{1\le i\le n} !\mathsf{X}_i \multimap !\mathsf{Y}$ to be a Mealy machine consisting of

$$S_{!_n\mathsf{M}} = S_{\mathsf{M}}^{\mathbb{N}}, \qquad \mathrm{init}_{!_n\mathsf{M}} = (\mathrm{init}_{\mathsf{M}}, \mathrm{init}_{\mathsf{M}}, \dots)$$

and

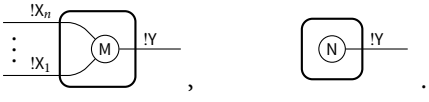
$$\tau_{!_n\mathsf{M}}\colon \left(\sum_{1\le i\le n} \mathbb{N} \times X_i^+ + \mathbb{N} \times Y^-\right) \times S_{\mathsf{M}}^{\mathbb{N}} \to \left(\sum_{1\le i\le n} \mathbb{N} \times X_i^- + \mathbb{N} \times Y^+\right) \times S_{\mathsf{M}}^{\mathbb{N}}$$

given by

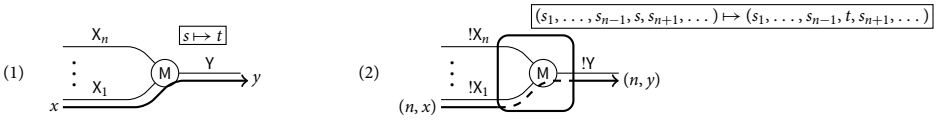
$$\tau_{!_n\mathsf{M}}(j@z, (s_1, s_2 \dots)) = \begin{cases} (j@w, (s_1, \dots, s_{j-1}, t, s_{j+1}, \dots)), & \text{if } \tau_{\mathsf{M}}(z, s_j) = (w, t), \\ \text{undefined}, & \text{if } \tau_{\mathsf{M}}(z, s_j) \text{ is undefined.} \end{cases}$$

Here, for $j \in \mathbb{N}$ and $a \in A + \cdots + B$, we write $j@a$ for the element in $\mathbb{N} \times A + \cdots + \mathbb{N} \times B$ obtained by the canonical isomorphism between $\mathbb{N} \times A + \cdots + \mathbb{N} \times B$ and $\mathbb{N} \times (A + \cdots + B)$.

As for diagrammatic presentation, we follow the way of proof-nets. Given a Mealy machine $\mathsf{M}\colon \mathsf{X}_1 \otimes \cdots \otimes \mathsf{X}_n \multimap \mathsf{Y}$ (a Mealy machine $\mathsf{N}\colon \mathsf{I} \multimap \mathsf{Y}$), we depict $!_n\mathsf{M}\colon !\mathsf{X}_1 \otimes \cdots \otimes !\mathsf{X}_n \multimap !\mathsf{Y}$ (depict $!_0\mathsf{M}\colon \mathsf{I} \multimap !\mathsf{Y}$) by surrounding M by a thick line box:



The Mealy machine $!_n\mathsf{M}$ behaves as a parallel composition of countably infinite copies of M. For example, if M has a transition (1) below, then $!_n\mathsf{M}$ has a transition (2) below for all $n \in \mathbb{N}$ and $s_1, s_2, \dots \in S_{\mathsf{M}}$:



In other words, given an input whose first entry is n, then the nth copy of M handles the input, and there is no side effect to the other copies of M. In the sequel, we omit the subscript of $!_n\mathsf{M}$ when $n = 1$.

Dereliction, Digging, Contraction and Weakening For natural numbers $n, m \in \mathbb{N}$, we write $\langle n, m\rangle$ for the Cantor pairing $m + (n + m)(n + m + 1)/2$. For **Int**-objects X and Y, we define $\mathsf{d}_{\mathsf{X}}\colon !\mathsf{X} \multimap \mathsf{X}$, $\mathsf{g}_{\mathsf{X}}\colon !\mathsf{X} \multimap !!\mathsf{X}$, $\mathsf{c}_{\mathsf{X}}\colon !\mathsf{X} \multimap !\mathsf{X} \otimes !\mathsf{X}$ and $\mathsf{w}_{\mathsf{X}}\colon \mathsf{X} \multimap \mathsf{I}$ to be token machines whose transition functions:

$$\tau_{\mathsf{d}_{\mathsf{X}}}\colon \mathbb{N} \times X^+ + X^- \to \mathbb{N} \times X^- + X^+,$$

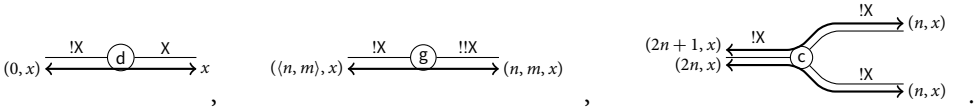
$$\tau_{\mathsf{g}_{\mathsf{X}}}\colon \mathbb{N} \times X^+ + \mathbb{N} \times \mathbb{N} \times X^- \to \mathbb{N} \times X^- + \mathbb{N} \times \mathbb{N} \times X^+,$$

$$\tau_{\mathsf{c}_X} : \mathbb{N} \times X^+ + (\mathbb{N} \times X^- + \mathbb{N} \times X^-) \to \mathbb{N} \times X^- + (\mathbb{N} \times X^+ + \mathbb{N} \times X^+),$$
$$\tau_{\mathsf{w}_X} : X^+ + \emptyset \to X^- + \emptyset$$

are given by:

$$\tau_{\mathsf{d}_X}(\bullet, (n, x)) = (\circ, x), \qquad \tau_{\mathsf{d}_X}(\circ, x) = (\bullet, (0, x)),$$
$$\tau_{\mathsf{g}_X}(\bullet, (\langle n, m\rangle, x)) = (\circ, (n, m, x)), \qquad \tau_{\mathsf{g}_X}(\circ, (n, m, x)) = (\bullet, (\langle n, m\rangle, x)),$$
$$\tau_{\mathsf{c}_X}(\bullet, (2n, x)) = (\circ, (\bullet, (n, x))), \qquad \tau_{\mathsf{c}_X}(\circ, (\bullet, (n, x))) = (\bullet, (2n, x)),$$
$$\tau_{\mathsf{c}_X}(\bullet, (2n+1, x)) = (\circ, (\circ, (n, x))), \qquad \tau_{\mathsf{c}_X}(\circ, (\circ, (n, x))) = (\bullet, (2n+1, x)),$$
$$\tau_{\mathsf{w}_X} = \text{the empty partial measurable function.}$$

These token machines d_X, g_X and c_X behave as follows:



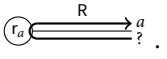
The token machine w_X never interacts with outside. As we did here, we often omit subscripts of d_X, g_X, c_X and w_X in diagrams.

5.2.6 Real numbers

We define an **Int**-object R to be $(\mathbb{R}, \{?\})$. For $a \in \mathbb{R}$, we define a token machine $\mathsf{r}_a : \mathsf{I} \multimap \mathsf{R}$ by:

$$\tau_{\mathsf{r}_a}(\circ, ?) = (\circ, a).$$

Note that the transition function τ_{r_a} is from $\emptyset + \{?\}$ to $\emptyset + \mathbb{R}$. The transition function of r_a means that given a query ? from environment, r_a answers its value a:

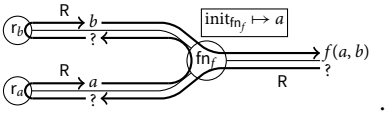


5.2.7 Measurable functions

We associate a measurable function $f : \mathbb{R}^n \to \mathbb{R}$ with a Mealy machine $\mathsf{fn}_f : \mathsf{R}^{\otimes n} \multimap \mathsf{R}$. For simplicity, we suppose $n = 2$. The state space S_{fn_f} is defined to be $\mathbb{R}$, and the initial state $\mathrm{init}_{\mathsf{fn}_f}$ is 0. The transition function $\tau_{\mathsf{fn}_f} : ((\mathbb{R} + \mathbb{R}) + \{?\}) \times S_{\mathsf{fn}_f} \to (((\{?\} + \{?\}) + \mathbb{R}) \times S_{\mathsf{fn}_f}$ is given by:

$$\tau_{\mathsf{fn}_f}((\bullet, (\bullet, a)), s) = ((\bullet, (\circ, ?)), a),$$
$$\tau_{\mathsf{fn}_f}((\bullet, (\circ, a)), s) = ((\circ, f(s, a)), s),$$
$$\tau_{\mathsf{fn}_f}((\circ, ?), s) = ((\bullet, (\bullet, ?)), s).$$

For real numbers $a, b \in \mathbb{R}$, the Mealy machine $\mathsf{fn}_f \circ (\mathsf{r}_a \otimes \mathsf{r}_b)$ behaves as follows:



Namely, given a query ? from the right R-wire, fn_f first sends a query ? to the lower R-wire in order to obtain the value of its first argument. The Mealy machine fn_f memorises the return value

a from r_a, and another query ? is sent to the upper R-wire. Then r_b returns b. Now, fn_f sees that its first argument is a and its second argument is b. Finally, fn_f outputs $f(a, b)$. We note that fn_f is always used in this way in our denotational semantics, and therefore, the definition of $\mathrm{init}_{\mathsf{fn}_f}$ is not essential to simulate the measurable function f.

5.2.8 Conditional branching

For an **Int**-object X, we define

$$\mathsf{cd}_\mathsf{X} \colon \mathsf{R} \otimes \mathsf{X} \otimes \mathsf{X} \to \mathsf{X}$$

to be a Mealy machine consisting of

$$S_{\mathsf{cd}_\mathsf{X}} = \{*\} \cup X^-, \qquad \mathrm{init}_{\mathsf{cd}_\mathsf{X}} = *$$

and

$$\tau_{\mathsf{cd}_\mathsf{X}} \colon (((\mathbb{R} + X^+) + X^+) + X^-) \times S_{\mathsf{cd}_\mathsf{X}} \to ((((\{?\} + X^-) + X^-) + X^+) \times S_{\mathsf{cd}_\mathsf{X}}$$

given by:

$$\tau_{\mathsf{cd}_\mathsf{X}}((\bullet, (\bullet, (\bullet, a))), s) = \begin{cases} ((\bullet, (\bullet, (\circ, s))), s), & \text{if } a = 0 \text{ and } s \in X^-, \\ ((\bullet, (\circ, s)), s), & \text{if } a \neq 0 \text{ and } s \in X^-, \\ \text{undefined}, & \text{otherwise}, \end{cases}$$

$$\tau_{\mathsf{cd}_\mathsf{X}}((\bullet, (\bullet, (\circ, x))), s) = ((\circ, x), s),$$

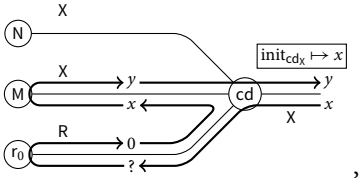
$$\tau_{\mathsf{cd}_\mathsf{X}}((\bullet, (\circ, x)), s) = ((\circ, x), s),$$

$$\tau_{\mathsf{cd}_\mathsf{X}}((\circ, x), s) = ((\bullet, (\bullet, (\bullet, ?))), x).$$

Here, $\Sigma_{S_{\mathsf{cd}_\mathsf{X}}}$ is the σ-algebra generated by $\{\{*\}\} \cup \Sigma_{X^-}$.

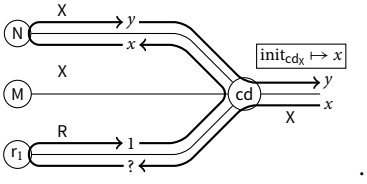
For a real number $a \in \mathbb{R}$ and Mealy machines $\mathsf{M}, \mathsf{N} \colon \mathsf{I} \multimap \mathsf{X}$, the Mealy machine $\mathsf{cd}_\mathsf{X} \circ (\mathsf{r}_a \otimes \mathsf{M} \otimes \mathsf{N})$ behaves as follows:

$(a = 0)$



,

$(a = 1)$



.

Namely, given an input $x \in X^-$, then cd_X memorises x and throws a query to the R-wire. There are two cases: (i) if a is 0 (the left diagram), then r_0 returns 0, and cd_X sends the memorised value x to the middle X-wire and (ii) if a is not 0, say 1 (the right diagram), then r_1 returns 1, and cd_X sends x to the upper X-wire. In both cases, any output from M (or N) is sent to the X-wire in the right-hand side. In this way, $\mathsf{cd}_\mathsf{X} \circ (\mathsf{r}_a \otimes \mathsf{M} \otimes \mathsf{N})$ simulates M when $a = 0$ and simulates N when $a \neq 0$.

5.2.9 Scoring

Let $\mathbb{R}^*_{[0,1]}$ be the measurable space of finite sequences of real numbers in the unit interval $[0, 1]$ endowed with the following σ-algebra:

$$A \in \Sigma_{\mathbb{R}^*_{[0,1]}} \iff A \cap \mathbb{R}^n_{[0,1]} \in \Sigma_{\mathbb{R}^n} \text{ for all } n \in \mathbb{N}.$$

We define S to be an **Int**-object given by $(\emptyset, \mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]})$. Below, we identify $\mathsf{S} \otimes \mathsf{S}^\perp$ with an **Int**-object $(\mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]}, \mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]})$ by the obvious manner. Then we define $\mathsf{sc} \colon \mathsf{R} \multimap \mathsf{S} \otimes \mathsf{S}^\perp \otimes !\mathsf{I}$ to

be a Mealy machine consisting of

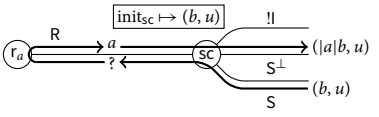
$$S_{\mathsf{sc}} = \mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]}, \qquad \mathrm{init}_{\mathsf{sc}} = (0, \varepsilon)$$

and $\tau_{\mathsf{sc}} : (\mathbb{R} + (\mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]} + \emptyset)) \times S_{\mathsf{sc}} \to (\{?\} + (\mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]} + \emptyset)) \times S_{\mathsf{sc}}$ given by

$$\tau_{\mathsf{sc}}((\bullet, a), (b, v)) = ((\circ, (\bullet, (|a|\ b, v))), (b, v)),$$

$$\tau_{\mathsf{sc}}((\circ, (\bullet, (a, u))), (b, v)) = ((\bullet, ?), (a, u)).$$

The Mealy machine simulates scoring $(\mathtt{score}(\mathtt{r}_a), b, u) \to (\mathtt{skip}, |a|\ b, u)$ in the following way:



.

Namely, in the first transition, sc memorises the 'configuration' (b, u), which is used to give the 'configuration' $(|a|\ b, u)$ in the final transition. We note that sc is always used in this way in our denotational semantics, and therefore, the definition of $\mathrm{init}_{\mathsf{sc}}$ is not essential to simulate scoring.

5.2.10 Sampling

We define $\mathsf{sa} : \mathsf{I} \multimap \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes !\mathsf{R}$ to be a Mealy machine consisting of

$$S_{\mathsf{sa}} = \mathbb{R}_{[0,1]}, \qquad \mathrm{init}_{\mathsf{sa}} = 0$$

and

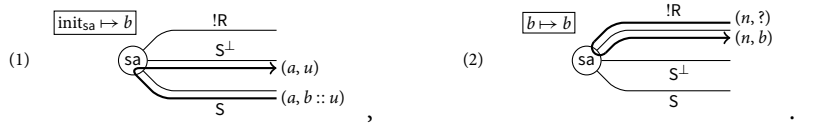
$$\tau_{\mathsf{sa}} : (\emptyset + (\mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]} + \mathbb{N} \times \{?\})) \times S_{\mathsf{sa}} \to (\emptyset + (\mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]} + \mathbb{N} \times \mathbb{R})) \times S_{\mathsf{sa}}$$

given by:

$$\tau_{\mathsf{sa}}((\circ, (\bullet, (a, u))), s) = \begin{cases} ((\circ, (\bullet, (a, v))), b), & \text{if } u = b :: v, \\ \text{undefined}, & \text{if } u = \varepsilon, \end{cases}$$

$$\tau_{\mathsf{sa}}((\circ, (\circ, (n, ?))), s) = ((\circ, (\circ, (n, s))), s).$$

The Mealy machine sa simulates sampling $(\mathtt{sample}, a, b :: u) \to (b, a, u)$:



,

.

Namely, (1) sa pops b and memorises the value b; (2) whenever the sampling result is required, sa returns the memorised value. We note that sa is always used in this way in our denotational semantics, and therefore, the definition of $\mathrm{init}_{\mathsf{sa}}$ is not essential to simulate sampling.

5.3 Behavioural equivalence and diagrammatic reasoning

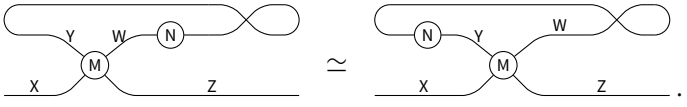
We now give a brief remark on diagrammatic presentation of Mealy machines. The diagrammatic presentation of a Mealy machine is not only for intuitive explanation, but also for rigorous reasoning about *behavioural equivalence*, which identifies Mealy machines *behaving the same way*. Identifying Mealy machines in terms of their behaviour is important to reason about composition of Mealy machines in the coming sections. Here, we are inspired by behavioural equivalence from coalgebraic theory of modelling transition systems (Jacobs 2016).

Let M and N be Mealy machines from X to Y. For a measurable function $f\colon S_{\mathsf{M}} \to S_{\mathsf{N}}$, we write $f \Vdash \mathsf{M} \preceq \mathsf{N}$ when we have $f(\mathrm{init}_{\mathsf{M}}) = \mathrm{init}_{\mathsf{N}}$ and

$$\begin{array}{ccc} (X^+ + Y^-) \times S_{\mathsf{M}} & \xrightarrow{\mathrm{id}\times f} & (X^+ + Y^-) \times S_{\mathsf{N}} \\ {\scriptstyle \tau_{\mathsf{M}}}\downarrow & & \downarrow{\scriptstyle \tau_{\mathsf{N}}} \\ (Y^+ + X^-) \times S_{\mathsf{M}} & \xrightarrow{\mathrm{id}\times f} & (Y^+ + X^-) \times S_{\mathsf{N}} \end{array},$$

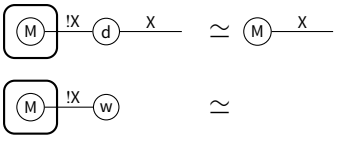
and we write $\mathsf{M} \preceq \mathsf{N}$ when there is a measurable function $f\colon S_{\mathsf{M}} \to S_{\mathsf{N}}$ such that $f \Vdash \mathsf{M} \preceq \mathsf{N}$. The definition means that if we have $\mathsf{M} \preceq \mathsf{N}$, then no observer can distinguish between M and N from their input/output behaviour, although their internal structure can be quite different. We define an equivalence relation $\simeq$ to be the reflective, symmetric and transitive closure of $\preceq$. For Mealy machines $\mathsf{M}, \mathsf{N}\colon \mathsf{X} \multimap \mathsf{Y}$, we say that M is *behaviourally equivalent* to N when $\mathsf{M} \simeq \mathsf{N}$. Below, we list some important behavioural equivalences in terms of diagrams.

- If two Mealy machines have the same diagrammatic presentation modulo some rearrangement of wires and nodes, then they are behaviourally equivalent. This is because rearrangement of wires and nodes has nothing to do with how tokens flow along wires. For example, for all Mealy machines $\mathsf{M}\colon \mathsf{X} \otimes \mathsf{Y} \multimap \mathsf{Z} \otimes \mathsf{W}$ and $\mathsf{N}\colon \mathsf{W} \multimap \mathsf{Y}$, we have



- For any Mealy machine $\mathsf{M}\colon \mathsf{I} \multimap \mathsf{X}$, we have the following behavioural equivalences:

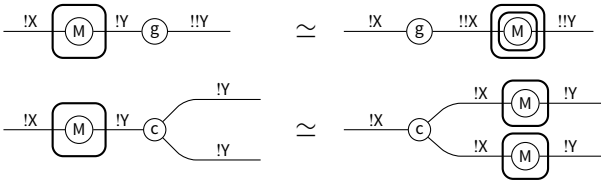
$$\mathsf{d}_{\mathsf{X}} \circ !_0\mathsf{M} \simeq \mathsf{M} \qquad \mathsf{w}_{!\mathsf{X}} \circ !_0\mathsf{M} \simeq \mathsf{id}_{\mathsf{I}}$$



The second behavioural equivalence means that we can always remove $\mathsf{w}_{\mathsf{X}} \circ !_0\mathsf{M}$ from diagrams without changing their behaviour.

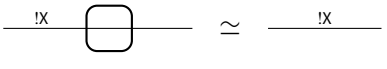
- For any Mealy machine $\mathsf{M}\colon \mathsf{X} \multimap \mathsf{Y}$, we have the following behavioural equivalences:

$$\mathsf{g}_{\mathsf{Y}} \circ !\mathsf{M} \simeq (!!\mathsf{M}) \circ \mathsf{g}_{\mathsf{X}} \qquad \mathsf{c}_{\mathsf{Y}} \circ !\mathsf{M} \simeq (!\mathsf{M} \otimes !\mathsf{M}) \circ \mathsf{c}_{\mathsf{X}}$$



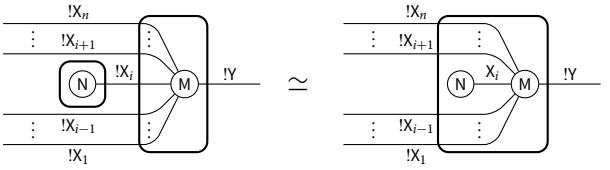
- We can always remove a thick box surrounding a single wire:

$$!\mathsf{id}_{\mathsf{X}} \simeq \mathsf{id}_{!\mathsf{X}}$$



- For any Mealy machine $\mathsf{M}\colon \bigotimes_{1\le i\le n} \mathsf{X}_i \multimap \mathsf{Y}$, and for any Mealy machine $\mathsf{N}\colon \mathsf{I} \multimap \mathsf{X}_i$, we have

$$!_n\mathsf{M} \circ (\mathsf{id}_{!\mathsf{X}_1} \otimes \cdots \otimes \mathsf{id}_{!\mathsf{X}_{i-1}} \otimes !_0\mathsf{N} \otimes \mathsf{id}_{!\mathsf{X}_{i+1}} \otimes \cdots \otimes \mathsf{id}_{!\mathsf{X}_n}) \simeq !_{n-1}(\mathsf{M} \circ (\mathsf{id}_{\mathsf{X}_1} \otimes \cdots \otimes \mathsf{id}_{\mathsf{X}_{i-1}} \otimes \mathsf{N} \otimes \mathsf{id}_{\mathsf{X}_{i+1}} \otimes \cdots \otimes \mathsf{id}_{\mathsf{X}_n})).$$



- For all Mealy machines $\mathsf{M}, \mathsf{N} : \mathsf{I} \multimap \mathsf{X}$, we have $\mathsf{cd}_{\mathsf{X}} \circ (\mathsf{r}_0 \otimes \mathsf{M} \otimes \mathsf{N}) \simeq \mathsf{M}$ and $\mathsf{cd}_{\mathsf{X}} \circ (\mathsf{r}_a \otimes \mathsf{M} \otimes \mathsf{N}) \simeq \mathsf{N}$ for all real numbers $a \neq 0$:

Furthermore, we can compositionally apply these behavioural equivalences, that is, we can replace a subdiagram of a diagram with another behaviourally equivalent diagram. This is because the composition, the monoidal product and the resource modality ! are compatible with behavioural equivalence.

- For all Mealy machines $\mathsf{M} \simeq \mathsf{M}' : \mathsf{X} \multimap \mathsf{Y}$ and $\mathsf{N} \simeq \mathsf{N}' : \mathsf{Y} \multimap \mathsf{Z}$, we have $\mathsf{N} \circ \mathsf{M} \simeq \mathsf{N}' \circ \mathsf{M}'$.
- For all Mealy machines $\mathsf{M} \simeq \mathsf{M}' : \mathsf{X} \multimap \mathsf{Y}$ and $\mathsf{N} \simeq \mathsf{N}' : \mathsf{Z} \multimap \mathsf{W}$, we have $\mathsf{N} \otimes \mathsf{M} \simeq \mathsf{N}' \otimes \mathsf{M}'$.
- For all Mealy machines $\mathsf{M} \simeq \mathsf{M}' : \bigotimes_{1 \leq i \leq n} \mathsf{X}_i \multimap \mathsf{Y}$, we have $!_n\mathsf{M} \simeq !_n\mathsf{M}'$.

6. Mealy Machine Semantics and Adequacy Theorems

6.1 Mealy machine semantics

We describe our denotational semantics for **PCFSS** based on Mealy machines. We interpret a type $\mathtt{A}$ as the **Int**-object $[\![\mathtt{A}]\!]$ given by:

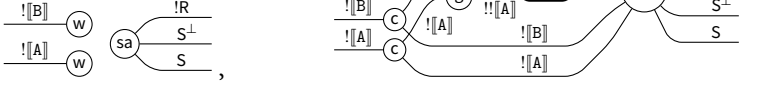
$$[\![\mathtt{Unit}]\!] = \mathsf{I}, \quad [\![\mathtt{Real}]\!] = \mathsf{R}, \quad [\![\mathtt{A} \to \mathtt{B}]\!] = \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes !\,[\![\mathtt{B}]\!] \otimes !\,[\![\mathtt{A}]\!]^{\perp},$$

and we interpret terms $\mathtt{x}:\mathtt{A}, \ldots, \mathtt{y}:\mathtt{B} \vdash \mathtt{M}:\mathtt{C}$ and values $\mathtt{x}:\mathtt{A}, \ldots, \mathtt{y}:\mathtt{B} \vdash \mathtt{V}:\mathtt{C}$ by Mealy machines:

$$[\![\mathtt{x}:\mathtt{A}, \ldots, \mathtt{y}:\mathtt{B} \vdash \mathtt{M}:\mathtt{C}]\!] : !\,[\![\mathtt{A}]\!] \otimes \cdots \otimes !\,[\![\mathtt{B}]\!] \multimap \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes !\,[\![\mathtt{C}]\!],$$

$$[\![\mathtt{x}:\mathtt{A}, \ldots, \mathtt{y}:\mathtt{B} \vdash \mathtt{V}:\mathtt{C}]\!]_{\mathrm{v}} : !\,[\![\mathtt{A}]\!] \otimes \cdots \otimes !\,[\![\mathtt{B}]\!] \multimap [\![\mathtt{C}]\!]$$

inductively defined by diagrams in Figure 5. In these definitions, we simply write $[\![\mathtt{M}]\!]$ and $[\![\mathtt{V}]\!]_{\mathrm{v}}$ for $[\![\Delta \vdash \mathtt{M}:\mathtt{A}]\!]$ and $[\![\Delta \vdash \mathtt{V}:\mathtt{A}]\!]_{\mathrm{v}}$, respectively, and we suppose that Δ is of the form $(\mathtt{x}:\mathtt{A})$ for some variable $\mathtt{x}$ and some type $\mathtt{A}$. In general, when $\Delta = (\mathtt{x}:\mathtt{A}, \ldots, \mathtt{y}:\mathtt{B})$, we need to replace $!\,[\![\Delta]\!]$ in Figure 5 with $!\,[\![\mathtt{A}]\!] \otimes \cdots \otimes !\,[\![\mathtt{B}]\!]$ and appropriately duplicate $\mathtt{g}$, $\mathtt{c}$ and $\mathtt{w}$. For example, interpretations $[\![\mathtt{x}:\mathtt{A}, \mathtt{y}:\mathtt{B} \vdash \mathtt{sample}:\mathtt{Real}]\!]$ and $[\![\mathtt{x}:\mathtt{A}, \mathtt{y}:\mathtt{B} \vdash \mathtt{V}\,\mathtt{W}:\mathtt{C}]\!]$ are defined to be



respectively. We note that $[\![\vdash \mathtt{M}:\mathtt{A}]\!]$ and $[\![\vdash \mathtt{V}:\mathtt{A}]\!]_{\mathrm{v}}$ are defined to be Mealy machines from I.

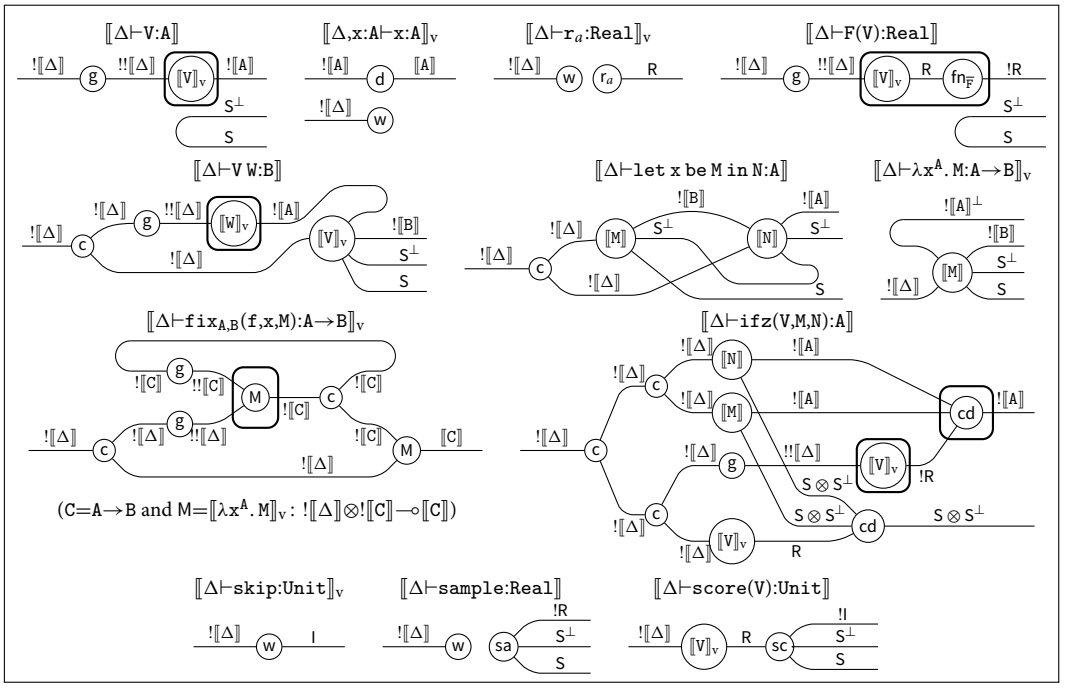


Figure 5. Interpretation of terms and values.

6.2 Two adequacy theorems

For a Mealy machine $\mathsf{M} : \mathsf{I} \multimap \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes !\mathsf{R}$, we define measurable functions $\mathsf{weight}_{\mathsf{M}} : \mathbb{R}_{\geq 0} \times \mathbb{R}^{*}_{[0,1]} \to \mathbb{R}_{\geq 0}$ and $\mathsf{value}_{\mathsf{M}} : \mathbb{R}_{\geq 0} \times \mathbb{R}^{*}_{[0,1]} \to \mathbb{R}$ as follows.

- For $(a, u) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{*}_{[0,1]}$, if there are $s, s' \in S_{\mathsf{M}}$, $a' \in \mathbb{R}_{\geq 0}$ and $b \in \mathbb{R}$ such that

$$\tau_{\mathsf{M}}((\circ, (\bullet, (a, u))), \mathrm{init}_{\mathsf{M}}) = ((\circ, (\bullet, (a', \varepsilon))), s), \tag{1}$$

$$\tau_{\mathsf{M}}((\circ, (\circ, (0, ?))), s) = (\circ, (\circ, (0, b)), s'), \tag{2}$$

  that is, if we have the following transitions:

  then we define $\mathsf{weight}_{\mathsf{M}}(a, u)$ and $\mathsf{value}_{\mathsf{M}}(a, u)$ by:

$$\mathsf{weight}_{\mathsf{M}}(a, u) = a', \qquad \mathsf{value}_{\mathsf{M}}(a, u) = b.$$

- Otherwise, both $\mathsf{weight}_{\mathsf{M}}(a, u)$ and $\mathsf{value}_{\mathsf{M}}(a, u)$ are undefined.

Abusing notation, for a closed term $\vdash \mathtt{M} : \mathtt{Real}$, we write $\mathsf{weight}_{\mathtt{M}}$ and $\mathsf{value}_{\mathtt{M}}$ for $\mathsf{weight}_{[\![\mathtt{M}]\!]}$ and $\mathsf{value}_{[\![\mathtt{M}]\!]}$, respectively.

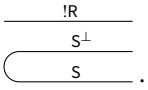
We give some intuitions about the measurable functions weight and value.

- The measurable function $\mathsf{weight}_{\mathsf{M}}(a, u)$ has two purposes: one is to determine sampling results of sas in M using u and the other is to collect scoring results by multiplying a with weights given by scs in M.
- The purpose of $\mathsf{value}_{\mathsf{M}}(a, u)$ is to calculate 'execution result' of M following the sampling results determined by $\mathsf{weight}_{\mathsf{M}}(a, u)$.

Theorem 6. (Adequacy). *For any configuration* $(\mathtt{M}, a, u)$ *and for any* $a' \in \mathbb{R}_{\geq 0}$ *and* $b \in \mathbb{R}$,

$$(\mathsf{weight}_{\mathsf{M}}(a, u) = a' \,\&\, \mathsf{value}_{\mathtt{M}}(a, u) = b) \iff (\mathtt{M}, a, u) \to^* (\mathtt{r}_b, a', \varepsilon).$$

Proof. The left-to-right implication follows from Proposition 18. For the other direction, it follows from Lemma 20 that $(\mathtt{M}, [\![\mathtt{M}]\!]) \in \overline{R}_{\mathtt{Real}}$. Let E be a Mealy machine given by $\mathsf{ax}_{\mathsf{S}} \otimes \mathsf{id}_{!\mathsf{R}}$, which is depicted as follows:



Because $([\,-\,], \mathsf{E})$ is an element of $R^{\top}_{\mathtt{Real}}$, we see that $(\mathtt{M}, \mathsf{E} \star [\![\mathtt{M}]\!]) = (\mathtt{M}, [\![\mathtt{M}]\!])$ is an element of Ω. By the definition of Ω, we obtain the implication from left to right. □

For a Mealy machine $\mathsf{M} : \mathsf{I} \multimap \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes !\mathsf{R}$ and $a \in \mathbb{R}_{\geq 0}$, we define a measure $\mu_{\mathsf{M},a}$ on $\mathbb{R}$ by:

$$\mu_{\mathsf{M},a}(A) = \sum_{n \in \mathbb{N}} \int_{\mathbb{R}^n} \underline{\mathsf{weight}}_{\mathsf{M}}(a, u)\, [\underline{\mathsf{value}}_{\mathsf{M}}(a, u) \in A]\, \mathrm{d}u$$

where $\underline{\mathsf{weight}}_{\mathsf{M}}$ and $\underline{\mathsf{value}}_{\mathsf{M}}$ are measurable functions from $\mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]}$ to $\mathbb{R}$ given by:

$$\underline{\mathsf{weight}}_{\mathsf{M}}(a, u) = \begin{cases} \mathsf{weight}_{\mathsf{M}}(a, u), & \text{if } \mathsf{weight}_{\mathsf{M}}(a, u) \text{ is defined,} \\ 0, & \text{otherwise,} \end{cases}$$

$$\underline{\mathsf{value}}_{\mathsf{M}}(a, u) = \begin{cases} \mathsf{value}_{\mathsf{M}}(a, u), & \text{if } \mathsf{value}_{\mathsf{M}}(a, u) \text{ is defined,} \\ 0, & \text{otherwise.} \end{cases}$$

We note that the value of $\underline{\mathsf{value}}_{\mathsf{M}}(a, u)$ in the otherwise case has nothing to do with the value of $\mu_{\mathsf{M},a}(A)$. Abusing notation, for a closed term $\vdash \mathtt{M} : \mathtt{Real}$ and for $a \in \mathbb{R}$, we write $\mu_{\mathtt{M},a}$ for $\mu_{[\![\mathtt{M}]\!],a}$.

Theorem 7. (Adequacy). *For any closed term* $\vdash \mathtt{M} : \mathtt{Real}$, *we have* $\mathtt{M} \Rightarrow_{\infty} \mu_{\mathtt{M},1}$.

Proof. Let μ be the unique measure such that $\mathtt{M} \Rightarrow_{\infty} \mu$. It follows from Proposition 21 that $\mu \leq \mu_{\mathtt{M},1}$. This is because μ is the sup of the family of measures $\{\mu_n\}_{n \in \mathbb{N}}$ given by $\mathtt{M} \Rightarrow_n \mu_n$. It remains to check $\mu_{\mathtt{M},1} \leq \mu$. It follows from Lemma 23 that $(\mathtt{M}, [\![\mathtt{M}]\!]) \in \overline{Q}_{\mathtt{Real}}$. Let E be the Mealy machine given in the proof of Theorem 6. Because $([\,-\,], \mathsf{E})$ is an element of $Q^{\top}_{\mathtt{Real}}$, we see that $(\mathtt{M}, \mathsf{E} \star [\![\mathtt{M}]\!]) = (\mathtt{M}, [\![\mathtt{M}]\!])$ is an element of Σ. By the definition of Σ, we obtain the claim. □

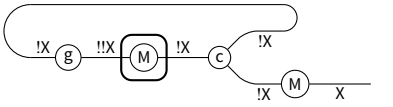
It follows from Theorems 6 and 7 that sampling-based operational semantics is coherent with distribution-based operational semantics. In fact, the latter can be derived from the former. An analogous result has already been proved by way of a purely operational (and quite laborious) argument in an untyped setting in which, however, `score` is not available in its full generality (Borgström et al. 2016). Here, it is just an easy corollary of our adequacy theorems.

The rest of this section is devoted to proving the above two theorems, which constitute the main result of this paper, together with Theorem 25.

6.3 On interpretation of the fixed point operator

We give some observations on the interpretation of the fixed point operator $\mathtt{fix}_{\mathtt{A},\mathtt{B}}(-,-,-)$, which will be used in Section 6.4 and Section 6.5. Let $\mathsf{M}\colon !\mathsf{X} \multimap \mathsf{X}$ be a Mealy machine. We show that a Mealy machine $\mathsf{M}^\dagger\colon \mathsf{I} \multimap \mathsf{X}$ given by:

$$\mathsf{M}^\dagger = (\mathsf{M} \otimes \mathsf{cut}_{!\mathsf{X}\perp}) \circ ((\mathsf{c}_\mathsf{X} \circ !\mathsf{M} \circ \mathsf{g}_\mathsf{X}) \otimes \mathsf{id}_{!\mathsf{X}\perp}) \circ \mathsf{ax}_{!\mathsf{X}} =$$



is a 'least' fixed point of M. This construction is used in the interpretation of the fixed point operator. In fact, for a term $\mathtt{f}:\mathtt{A}\to\mathtt{B}$, $\mathtt{x}:\mathtt{A}\vdash\mathtt{M}:\mathtt{B}$, we have $[\![\mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f},\mathtt{x},\mathtt{M})]\!] = [\![\lambda\mathtt{x}^{\mathtt{A}}.\,\mathtt{M}]\!]^\dagger$.

Parametrised Resource Modality and Parametrised Loop Operator. To see that $\mathsf{M}^\dagger$ is a 'least fixed point' of M, we introduce parametrisation of the resource modality ! and the loop operator $(-)^\dagger$. For a subset $\alpha \subseteq \mathbb{N}$, and for a Mealy machine $\mathsf{M}\colon \mathsf{X} \multimap \mathsf{Y}$, we define $!_\alpha\mathsf{M}\colon !\mathsf{X} \multimap !\mathsf{Y}$ to be a Mealy machine consisting of

$$S_{!_\alpha\mathsf{M}} = S_{!\mathsf{M}} = S_\mathsf{M}^\mathbb{N}, \qquad \mathrm{init}_{!_\alpha\mathsf{M}} = \mathrm{init}_{!\mathsf{M}} = (\mathrm{init}_\mathsf{M}, \mathrm{init}_\mathsf{M}, \dots)$$

and $\tau_{!_\alpha\mathsf{M}}\colon (\mathbb{N}\times X^+ + \mathbb{N}\times Y^-)\times S_\mathsf{M}^\mathbb{N} \to (\mathbb{N}\times X^- + \mathbb{N}\times Y^+)\times S_\mathsf{M}^\mathbb{N}$ given by:

$$\tau_{!_\alpha\mathsf{M}}(n@z, (s_i)_{i\in\mathbb{N}}) = \begin{cases} (n@w, (s_0,\dots,s_{n-1},t,s_{n+1},\dots)), & \text{if } \tau_\mathsf{M}(z,s_n) = (w,t) \text{ and } n\in\alpha, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Recall that for $n\in\mathbb{N}$ and $a\in A+B$, we write $n@a$ for the element in $\mathbb{N}\times A+\mathbb{N}\times B$ obtained by the canonical isomorphism between $\mathbb{N}\times A+\mathbb{N}\times B$ and $\mathbb{N}\times(A+B)$. Then for a Mealy machine $\mathsf{M}\colon !\mathsf{X}\multimap\mathsf{X}$, we define a Mealy machine $\mathsf{M}^{\dagger,\alpha}\colon \mathsf{I}\multimap!\mathsf{X}$ by:

$$\mathsf{M}^{\dagger,\alpha} = (\mathsf{M}\otimes\mathsf{cut}_{!\mathsf{X}\perp})\circ((\mathsf{c}_\mathsf{X}\circ!_\alpha\mathsf{M}\circ\mathsf{g}_\mathsf{X})\otimes\mathsf{id}_{!\mathsf{X}\perp})\circ\mathsf{ax}_{!\mathsf{X}}.$$

We note that we have $!_\mathbb{N}\mathsf{M} = !\mathsf{M}$ and $\mathsf{M}^{\dagger,\mathbb{N}} = \mathsf{M}^\dagger$.

Let $\alpha_n, \beta_n \subseteq \mathbb{N}$ be subsets of $\mathbb{N}$ given by:

$$\alpha_0 = \emptyset, \qquad \beta_n = \{\langle i,j\rangle \mid i\in\alpha_n \text{ and } j\in\mathbb{N}\}, \qquad \alpha_{n+1} = \{2i \mid i\in\mathbb{N}\}\cup\{2i+1 \mid i\in\beta_n\}.$$

(Recall that $\langle n,m\rangle = m + (n+m)(n+m+1)/2$.) The definitions of α_n and β_n are motivated by the following lemma.

Lemma 8. *For any $n\in\mathbb{N}$ and for any Mealy machine $\mathsf{M}\colon \mathsf{X}\multimap\mathsf{Y}$, we have*

$$(\mathbf{1}\times S_\mathsf{M}^\mathbb{N} \cong S_\mathsf{M}^\mathbb{N} \xrightarrow{f} S_\mathsf{M}^\mathbb{N}\times S_\mathsf{M}^\mathbb{N} \cong S_\mathsf{M}^\mathbb{N}\times S_\mathsf{M}^\mathbb{N}\times\mathbf{1}) \Vdash \mathsf{c}_\mathsf{Y}\circ!_{\alpha_{n+1}}\mathsf{M} \preceq (!\mathsf{M}\otimes!_{\beta_n}\mathsf{M})\circ\mathsf{c}_\mathsf{X},$$

$$(\mathbf{1}\times S_\mathsf{M}^\mathbb{N} \cong S_\mathsf{M}^\mathbb{N} \xrightarrow{g} (S_\mathsf{M}^\mathbb{N})^\mathbb{N} \cong (S_\mathsf{M}^\mathbb{N})^\mathbb{N}\times\mathbf{1}) \Vdash \mathsf{g}_\mathsf{Y}\circ!_{\beta_n}\mathsf{M} \preceq !_{\alpha_n}!\mathsf{M}\circ\mathsf{g}_\mathsf{X}$$

where $f(s_n)_{n\in\mathbb{N}} = ((s_{2n})_{n\in\mathbb{N}}, (s_{2n+1})_{n\in\mathbb{N}})$ and $g(s_n)_{n\in\mathbb{N}} = ((s_{\langle i,j\rangle})_{j\in\mathbb{N}})_{i\in\mathbb{N}}$.

Proof. By the definition of α_n and β_n. □

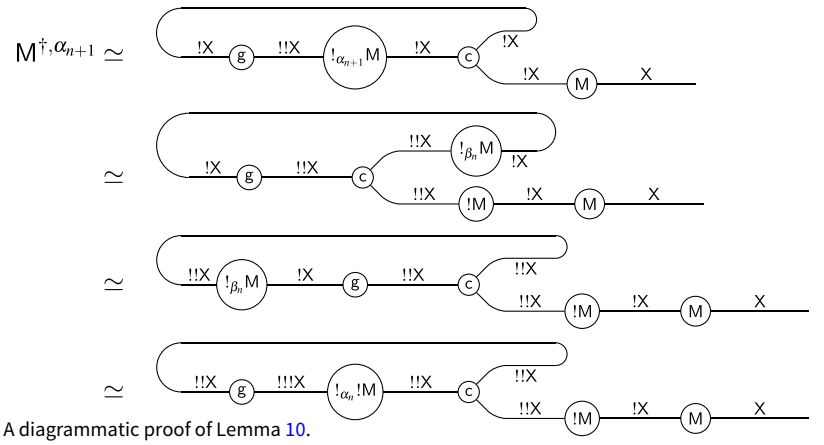


Figure 6. A diagrammatic proof of Lemma 10.

Lemma 9. *Let X be a measurable space, and let $u_X \colon X^{\mathbb{N}} \to X^{\mathbb{N}} \times (X^{\mathbb{N}})^{\mathbb{N}}$ be a measurable isomorphism given by:*

$$u_X(x_n)_{n\in\mathbb{N}} = ((x_{2n})_{n\in\mathbb{N}}, ((x_{2\langle m,l\rangle+1})_{m\in\mathbb{N}})_{l\in\mathbb{N}}).$$

Then there is a family of measurable functions $\{\phi_X \colon (X^{\mathbb{N}})^{\mathbb{N}} \to (X^{\mathbb{N}})^{\mathbb{N}}\}_{X:\text{measurable space}}$ such that

$$u_X^{\mathbb{N}} \circ \phi_X = (X^{\mathbb{N}})^{\mathbb{N}} \xrightarrow{u_{X^{\mathbb{N}}}} (X^{\mathbb{N}})^{\mathbb{N}} \times ((X^{\mathbb{N}})^{\mathbb{N}})^{\mathbb{N}} \xrightarrow{(X^{\mathbb{N}})^{\mathbb{N}} \times \phi_{X^{\mathbb{N}}}} (X^{\mathbb{N}})^{\mathbb{N}} \times ((X^{\mathbb{N}})^{\mathbb{N}})^{\mathbb{N}} \xrightarrow{\cong} (X^{\mathbb{N}} \times (X^{\mathbb{N}})^{\mathbb{N}})^{\mathbb{N}}$$

where the last isomorphism is given by $(((x_{i,j})_i)_j, (((y_{i,k,j})_i)_k)_j) \mapsto ((x_{i,j})_i, ((y_{i,k,j})_i)_k)_j$.

Proof. For sets $I, J, \ldots, K$, we regard elements in $(((X^I)^J)\cdots)^K$ as functions from $I \times J \times \cdots \times K$ to X. For $x \in (X^{\mathbb{N}})^{\mathbb{N}}$ and $(a, b) \in \mathbb{N}$, we define $(\phi_X(x))(a, b)$ by induction on a:

$$(\phi_X(x))(a,b) = \begin{cases} x(a', 2b), & \text{if } a = 2a', \\ (\phi_{X^{\mathbb{N}}}(x'))(a_0, a_1, b), & \text{if } a = 2\langle a_0, a_1\rangle + 1 \end{cases}$$

where $x' \in ((X^{\mathbb{N}})^{\mathbb{N}})^{\mathbb{N}}$ is given by:

$$x'(n, m, l) = x(n, 2\langle m, l\rangle + 1).$$

Here, $\phi_{X^{\mathbb{N}}}(x')$ takes three arguments because $\phi_{X^{\mathbb{N}}}(x')$ is an element of $((X^{\mathbb{N}})^{\mathbb{N}})^{\mathbb{N}}$. We note that the inductive definition of ϕ_X makes sense because $a = 2\langle a_0, a_1\rangle + 1$ implies $a_1 < a$. It is straightforward to check that the family ϕ_X satisfies the condition. □

Lemma 10. *For any $n \in \mathbb{N}$, and for any Mealy machine $\mathsf{M} \colon !\mathsf{X} \multimap \mathsf{X}$, we have*

$$u'_{\mathsf{M}} \Vdash \mathsf{M}^{\dagger,\alpha_{n+1}} \preceq \mathsf{M} \circ (!\mathsf{M})^{\dagger,\alpha_n}$$

where u'_{M} is a measurable function given by:

$$S_{\mathsf{M}^{\dagger,\alpha_{n+1}}} \cong S_{\mathsf{M}} \times S_{\mathsf{M}}^{\mathbb{N}} \xrightarrow{\mathrm{id}_{S_{\mathsf{M}}} \times u_{S_{\mathsf{M}}}} S_{\mathsf{M}} \times S_{\mathsf{M}}^{\mathbb{N}} \times (S_{\mathsf{M}}^{\mathbb{N}})^{\mathbb{N}} \cong S_{\mathsf{M} \circ (!\mathsf{M})^{\dagger,\alpha_n}}.$$

Here, $u_{(-)}$ is the isomorphism given in Lemma 9 and the first and the last isomorphisms are obtained by applying canonical isomorphisms $\mathbf{1} \times (-) \cong (-)$ and $(-) \times \mathbf{1} \cong (-)$.

Proof. See Figure 6. By tracking how states in those Mealy machines are related, we obtain $u'_{\mathsf{M}} \Vdash \mathsf{M}^{\dagger,\alpha_{n+1}} \preceq \mathsf{M} \circ (!\mathsf{M})^{\dagger,\alpha_n}$. □

Lemma 11. *For any $n \in \mathbb{N}$, and for any Mealy machine $\mathsf{M} \colon !\mathsf{X} \multimap \mathsf{X}$, we have*

$$\phi'_{\mathsf{M}} \Vdash (!\mathsf{M})^{\dagger,\alpha_n} \preceq !_0(\mathsf{M}^{\dagger,\alpha_n})$$

where ϕ'_{M} is a measurable function given by:

$$S_{(!\mathsf{M})^{\dagger,\alpha_n}} \cong S_{\mathsf{M}}^{\mathbb{N}} \times (S_{\mathsf{M}}^{\mathbb{N}})^{\mathbb{N}} \xrightarrow{\mathrm{id}_{S_{\mathsf{M}}^{\mathbb{N}}} \times \phi_{S_{\mathsf{M}}}} S_{\mathsf{M}}^{\mathbb{N}} \times (S_{\mathsf{M}}^{\mathbb{N}})^{\mathbb{N}} \xrightarrow{\nu} (S_{\mathsf{M}} \times S_{\mathsf{M}}^{\mathbb{N}})^{\mathbb{N}} \cong S_{!_0(\mathsf{M}^{\dagger,\alpha_n})}.$$

Here, $\phi_{(-)}$ is the measurable function given in Lemma 9, and ν is a measurable function given by $\nu((s_i)_i, ((t_{j,i})_j)_i) = (s_i, (t_{j,i})_j)_i$, and the first and the last isomorphisms are obtained by applying canonical isomorphisms $\mathbf{1} \times (-) \cong (-)$ and $(-) \times \mathbf{1} \cong (-)$.

Proof. We prove the statement by induction on n. (Base case)

$$(!\mathsf{M})^{\dagger,\alpha_0} = (!\mathsf{M} \otimes \mathsf{cut}_{!!\mathsf{X}^\perp}) \circ ((\mathsf{c}_{!\mathsf{X}} \circ !_{\alpha_0} !\mathsf{M} \circ \mathsf{g}_{!\mathsf{X}}) \otimes \mathsf{id}_{!!\mathsf{X}^\perp}) \circ \mathsf{ax}_{!!\mathsf{X}}$$
$$\simeq !\mathsf{M} \circ ! \left((\mathsf{id}_{!\mathsf{X}} \otimes \mathsf{cut}_{!\mathsf{X}^\perp}) \circ ((\mathsf{c}_{\mathsf{X}} \circ !_{\alpha_0} \mathsf{M} \circ \mathsf{g}_{\mathsf{X}}) \otimes \mathsf{id}_{!\mathsf{X}^\perp}) \circ \mathsf{ax}_{!\mathsf{X}}\right) \simeq !_0(\mathsf{M}^{\dagger,\alpha_0}).$$

(Induction step)

$$(!\mathsf{M})^{\dagger,\alpha_{n+1}} \overset{\text{Lemma 10}}{\simeq} !\mathsf{M} \circ (!!\mathsf{M})^{\dagger,\alpha_n} \overset{\text{Induction hypothesis}}{\simeq} !\mathsf{M} \circ !_0(!\mathsf{M})^{\dagger,\alpha_n} \simeq !_0(\mathsf{M} \circ (!\mathsf{M})^{\dagger,\alpha_n}) \overset{\text{Lemma 10}}{\simeq} !_0(\mathsf{M}^{\dagger,\alpha_{n+1}}).$$

By tracking how states are related in these behavioural equivalences, we obtain $\phi'_{\mathsf{M}} \Vdash (!\mathsf{M})^{\dagger,\alpha_n} \preceq !_0(\mathsf{M}^{\dagger,\alpha_n})$. For the induction step, we can use Lemma 9. □

For $n \in \mathbb{N}$, and for a Mealy machine $\mathsf{M} \colon !\mathsf{X} \multimap \mathsf{X}$, we inductively define $\mathsf{iter}_n(\mathsf{M}) \colon \mathsf{I} \multimap \mathsf{X}$ by:

$$\mathsf{iter}_0(\mathsf{M}) = \bot_{\mathsf{X}}, \qquad \mathsf{iter}_{n+1}(\mathsf{M}) = \mathsf{M} \circ !_0(\mathsf{iter}_n(\mathsf{M})).$$

Proposition 12. *For all $n \in \mathbb{N}$, we have*

$$\mathsf{M}^{\dagger,\alpha_n} \simeq \mathsf{iter}_{n+1}(\mathsf{M}), \qquad \tau_{\mathsf{M}^{\dagger,\alpha_0}} \leq \tau_{\mathsf{M}^{\dagger,\alpha_1}} \leq \tau_{\mathsf{M}^{\dagger,\alpha_2}} \leq \cdots$$

and

$$S_{\mathsf{M}^\dagger} = S_{\mathsf{M}^{\dagger,\alpha_n}}, \qquad \mathrm{init}_{\mathsf{M}^\dagger} = \mathrm{init}_{\mathsf{M}^{\dagger,\alpha_n}}, \qquad \tau_{\mathsf{M}^\dagger} = \bigvee_{n \geq 0} \tau_{\mathsf{M}^{\dagger,\alpha_n}}.$$

Proof. For all $n \in \mathbb{N}$, by the definition of $\mathsf{M}^{\dagger,\alpha_n}$, we have $S_{\mathsf{M}^\dagger} = S_{\mathsf{M}^{\dagger,\alpha_n}}$, $\mathrm{init}_{\mathsf{M}^\dagger} = \mathrm{init}_{\mathsf{M}^{\dagger,\alpha_n}}$ and

$$\tau_{\mathsf{M}^{\dagger,\alpha_0}} \leq \tau_{\mathsf{M}^{\dagger,\alpha_1}} \leq \tau_{\mathsf{M}^{\dagger,\alpha_2}} \leq \cdots, \qquad \tau_{\mathsf{M}^\dagger} = \bigvee_{n \geq 0} \tau_{\mathsf{M}^{\dagger,\alpha_n}}.$$

By induction on $n \in \mathbb{N}$, we check $\mathsf{M}^{\dagger,\alpha_n} \simeq \mathsf{iter}_{n+1}(\mathsf{M})$. For the base case, we have $\mathsf{M}^{\dagger,\emptyset} \simeq \mathsf{iter}_1(\mathsf{M})$ because $(\mathsf{id}_{!\mathsf{X}} \otimes \mathsf{cut}_{!\mathsf{X}^\perp}) \circ ((\mathsf{c}_{\mathsf{X}} \circ !_{\alpha_0} \mathsf{M} \circ \mathsf{g}_{\mathsf{X}}) \otimes \mathsf{id}_{!\mathsf{X}}) \circ \mathsf{ax}_{!\mathsf{X}}$ is behaviourally equivalent to $!_0 \bot_{\mathsf{X}}$. The induction step follows from Lemmas 10 and 11. □

Proposition 13. *For any Mealy machine $\mathsf{M} \colon !\mathsf{X} \multimap \mathsf{X}$,*

$$\mathsf{M} \circ !_0 \mathsf{M}^\dagger \simeq \mathsf{M}^\dagger.$$

Proof. By Lemmas 10 and 11, there is a measurable function h from $S_{\mathsf{M}^\dagger}$ to $S_{\mathsf{M} \circ !_0 \mathsf{M}^\dagger}$ that preserves the initial states and makes the following diagram commute for all $n \in \mathbb{N}$:

$$\begin{array}{ccc} (\emptyset + X^-) \times S_{\mathsf{M}^\dagger} & \xrightarrow{(\emptyset+X^-)\times h} & (\emptyset + X^-) \times S_{\mathsf{M} \circ !_0 \mathsf{M}^\dagger} \\ \downarrow{\scriptstyle \tau_{\mathsf{M}^\dagger, \alpha_{n+1}}} & & \downarrow{\scriptstyle \tau_{\mathsf{M} \circ !_0 \mathsf{M}^\dagger, \alpha_n}} \\ (\emptyset + X^+) \times S_{\mathsf{M}^\dagger} & \xrightarrow[(\emptyset+X^+)\times h]{} & (\emptyset + X^+) \times S_{\mathsf{M} \circ !_0 \mathsf{M}^\dagger} \ . \end{array}$$

Because $\tau_{\mathsf{M}^\dagger} = \bigvee_{n \in \mathbb{N}} \tau_{\mathsf{M}^\dagger, \alpha_{n+1}}$ and $\tau_{\mathsf{M} \circ !_0 \mathsf{M}^\dagger} = \bigvee_{n \in \mathbb{N}} \tau_{\mathsf{M} \circ !_0 \mathsf{M}^\dagger, \alpha_n}$, we obtain $\mathsf{M} \circ !_0 \mathsf{M}^\dagger \simeq \mathsf{M}^\dagger$. □

6.4 Proof of Theorem 6

Lemma 14. (Substitution Lemma). *Let* $\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n \vdash \mathtt{M} : \mathtt{B}$ *be term, and let* $\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n \vdash \mathtt{V} : \mathtt{B}$ *be a value. For any closed value* $\vdash \mathtt{U} : \mathtt{A}_i$,

$$[\![\mathtt{M}]\!] \circ (\mathsf{id}_{![\![\mathtt{A}_1]\!]} \otimes \cdots \otimes \mathsf{id}_{![\![\mathtt{A}_{i-1}]\!]} \otimes !_0 [\![\mathtt{U}]\!]_{\mathrm{v}} \otimes \mathsf{id}_{![\![\mathtt{A}_{i+1}]\!]} \otimes \cdots \otimes \mathsf{id}_{![\![\mathtt{A}_n]\!]}) \simeq [\![\mathtt{M}\{\mathtt{U}/\mathtt{x}_i\}]\!],$$

$$[\![\mathtt{V}]\!]_{\mathrm{v}} \circ (\mathsf{id}_{![\![\mathtt{A}_1]\!]} \otimes \cdots \otimes \mathsf{id}_{![\![\mathtt{A}_{i-1}]\!]} \otimes !_0 [\![\mathtt{U}]\!]_{\mathrm{v}} \otimes \mathsf{id}_{![\![\mathtt{A}_{i+1}]\!]} \otimes \cdots \otimes \mathsf{id}_{![\![\mathtt{A}_n]\!]}) \simeq [\![\mathtt{V}\{\mathtt{U}/\mathtt{x}_i\}]\!]_{\mathrm{v}}.$$

Proof. By induction on M and V. Induction steps can be checked by diagrammatic reasoning. □

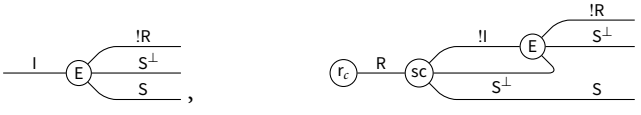
Lemma 15. *For Mealy machines* $\mathsf{M} \simeq \mathsf{N} \colon \mathsf{X} \multimap \mathsf{Y}$, *we have* $\mathsf{weight}_{\mathsf{M}} = \mathsf{weight}_{\mathsf{N}}$ *and* $\mathsf{value}_{\mathsf{M}} = \mathsf{value}_{\mathsf{N}}$.

Proof. By the definition of behavioural equivalence, if $\mathsf{M} \preceq \mathsf{N}$, then we have $\mathsf{weight}_{\mathsf{M}} = \mathsf{weight}_{\mathsf{N}}$ and $\mathsf{value}_{\mathsf{M}} = \mathsf{value}_{\mathsf{N}}$. Because $\simeq$ is the symmetric transitive closure of $\preceq$, we obtain the statement. □

Lemma 16. *Let* $\vdash \mathtt{M} : \mathtt{Real}$ *be a closed term of the form* $\mathtt{E}[\mathtt{score}(\mathtt{r}_a)]$ *for some evaluation context* $\mathtt{E}[\,-\,]$ *and* $a \in \mathbb{R}$. *Then for any* $(b, u) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]}$,

- *both* $\mathsf{weight}_{\mathtt{M}}(a, u)$ *and* $\mathsf{value}_{\mathtt{M}}(a, u)$ *are defined if and only if both* $\mathsf{weight}_{\mathtt{E}[\mathtt{skip}]}(|b|\, a, u)$ *and* $\mathsf{value}_{\mathtt{E}[\mathtt{skip}]}(|b|\, a, u)$ *are defined; and*
- *if both* $\mathsf{weight}_{\mathtt{M}}(a, u)$ *and* $\mathsf{value}_{\mathtt{M}}(a, u)$ *are defined, then* $\mathsf{weight}_{\mathtt{M}}(a, u) = \mathsf{weight}_{\mathtt{E}[\mathtt{skip}]}(|b|\, a, u)$ *and* $\mathsf{value}_{\mathtt{M}}(a, u) = \mathsf{value}_{\mathtt{E}[\mathtt{skip}]}(|b|\, a, u)$.

Proof. By induction on E with some diagrammatic reasoning, we can show that there is a Mealy machine $\mathtt{E} \colon \mathsf{I} \multimap \mathsf{S} \otimes \mathsf{S}^\perp \otimes !\mathsf{R}$ such that $[\![\mathtt{E}[\mathtt{skip}]]\!]$ and $[\![\mathtt{M}]\!]$ are behaviourally equivalent to



respectively. (Note that !I is equal to I.) By tracking how tokens travel, we obtain the statement. □

Lemma 17. *Let* $\vdash \mathtt{M} : \mathtt{Real}$ *be a closed term of the form* $\mathtt{E}[\mathtt{sample}]$ *for some evaluation context* $\mathtt{E}[\,-\,]$.

- *For any* $a \in \mathbb{R}_{\geq 0}$, *both* $\mathsf{weight}_{\mathtt{M}}(a, \varepsilon)$ *and* $\mathsf{value}_{\mathtt{M}}(a, \varepsilon)$ *are undefined.*
- *For any* $a \in \mathbb{R}_{\geq 0}$ *and for any* $b :: v \in \mathbb{R}^*_{[0,1]}$, *both* $\mathsf{weight}_{\mathtt{M}}(a, b :: v)$ *and* $\mathsf{value}_{\mathtt{M}}(a, b :: v)$ *are defined if and only if both* $\mathsf{weight}_{\mathtt{E}[\mathtt{r}_b]}(a, v)$ *and* $\mathsf{value}_{\mathtt{E}[\mathtt{r}_b]}(a, v)$ *are defined, and if all of them are defined, then* $\mathsf{weight}_{\mathtt{M}}(a, b :: v) = \mathsf{weight}_{\mathtt{E}[\mathtt{r}_b]}(a, v)$ *and* $\mathsf{value}_{\mathtt{M}}(a, b :: v) = \mathsf{value}_{\mathtt{E}[\mathtt{r}_b]}(a, v)$.

Proof. By induction on E with some diagrammatic reasoning, we can show that there is a Mealy machine $\mathsf{E} : !\mathsf{R} \multimap \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes !\mathsf{R}$ such that $[\![\mathtt{E}[\mathtt{r}_c]]\!]$ and $[\![\mathtt{M}]\!]$ are behaviourally equivalent to

respectively. By tracking how tokens travel, we obtain the statement. □

We first prove soundness.

Proposition 18. (Soundness). *For any configuration* $(\mathtt{M}, a, u)$ *and for any* $a' \in \mathbb{R}_{\geq 0}$ *and* $b \in \mathbb{R}$*, if* $(\mathtt{M}, a, u) \to^* (\mathtt{r}_b, a', \varepsilon)$*, then* $\mathsf{weight}_{\mathtt{M}}(a, u) = a'$ *and* $\mathsf{value}_{\mathtt{M}}(a, u) = b$.

Proof. By induction on the length of $\to^*$. (Base case) Easy. (Induction step) By case analysis of the first evaluation step of $(\mathtt{M}, a, u) \to^* (\mathtt{r}_b, a', \varepsilon)$.

- If the first evaluation step is of the form $(\mathtt{E}[\mathtt{N}], a, u) \to (\mathtt{E}[\mathtt{L}], a, u)$ for some $\mathtt{N} \xrightarrow{\text{red}} \mathtt{L}$, then by Lemma 14 and by Proposition 13 with some diagrammatic reasoning, we can check $[\![\mathtt{E}[\mathtt{N}]]\!] \simeq [\![\mathtt{E}[\mathtt{L}]]\!]$. Because $(\mathtt{E}[\mathtt{L}], a, u) \to^* (\mathtt{r}_b, a', \varepsilon)$, it follows from induction hypothesis and Lemma 15 that $\mathsf{weight}_{\mathtt{E}[\mathtt{N}]}(a, u) = a'$ and $\mathsf{value}_{\mathtt{E}[\mathtt{N}]}(a, u) = b$.
- If the first evaluation step is of the form $(\mathtt{E}[\mathtt{score}(\mathtt{r}_c)], a, u) \to (\mathtt{E}[\mathtt{skip}], |c|\, a, u)$, then by induction hypothesis, we have $\mathsf{weight}_{\mathtt{E}[\mathtt{skip}]}(|c|\, a, u) = a'$ and $\mathsf{value}_{\mathtt{E}[\mathtt{skip}]}(|c|\, a, u) = b$. By Lemma 16, we obtain $\mathsf{weight}_{\mathtt{M}}(a, u) = a'$ and $\mathsf{value}_{\mathtt{M}}(a, u) = b$.
- If the first evaluation step is of the form $(\mathtt{E}[\mathtt{sample}], a, c :: u) \to (\mathtt{E}[\mathtt{r}_c], a, u)$, then by induction hypothesis, $\mathsf{weight}_{\mathtt{E}[\mathtt{r}_c]}(a, u) = a'$ and $\mathsf{value}_{\mathtt{E}[\mathtt{r}_c]}(a, u) = b$. By Lemma 17, we obtain $\mathsf{weight}_{\mathtt{M}}(a, c :: u) = a'$ and $\mathsf{value}_{\mathtt{M}}(a, c :: u) = b$.

□

We next show that if $\mathsf{weight}_{\mathtt{M}}(a, u) = a'$ and $\mathsf{value}_{\mathtt{M}}(a, u) = b$, then $(\mathtt{M}, a, u) \to^* (\mathtt{r}_b, a', \varepsilon)$. To prove this, we use logical relations. We define a binary relation Ω between closed terms of type `Real` and Mealy machines from I to $\mathsf{S} \otimes \mathsf{S}^{\perp} \otimes !\mathsf{R}$ by:

$$(\mathtt{M}, \mathsf{M}) \in \Omega \iff \text{for all } (a, u) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]} \text{ and for all } a' \in \mathbb{R}_{\geq 0} \text{ and } b \in \mathbb{R},$$
$$\text{if } \mathsf{weight}_{\mathtt{M}}(a, u) = a' \text{ and } \mathsf{value}_{\mathtt{M}}(a, u) = b, \text{ then } (\mathtt{M}, a, u) \to^* (\mathtt{r}_b, a', \varepsilon).$$

We then inductively define binary relations:

$$R_{\mathtt{A}} \subseteq \{\text{closed values of type } \mathtt{A}\} \times \{\text{Mealy machines from I to } [\![\mathtt{A}]\!]\}$$

$$R^{\top}_{\mathtt{A}} \subseteq \{\text{evaluation contexts } \mathtt{x} : \mathtt{A} \vdash \mathtt{E}[\mathtt{x}] : \mathtt{Real}\} \times \{\text{Mealy machines from } ![\![\mathtt{A}]\!] \text{ to } \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes !\mathsf{R}\}$$

$$\overline{R}_{\mathtt{A}} \subseteq \{\text{closed terms of type } \mathtt{A}\} \times \{\text{Mealy machines from I to } \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes ![\![\mathtt{A}]\!]\}$$

by:

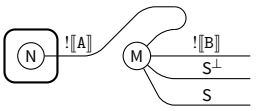
$$R_{\mathtt{Real}} = \{(\mathtt{r}_a, \mathsf{r}_a) \mid a \in \mathbb{R}\},$$
$$R_{\mathtt{Unit}} = \{(\mathtt{skip}, \mathsf{id}_{\mathsf{I}})\},$$
$$R_{\mathtt{A} \to \mathtt{B}} = \{(\mathtt{V}, \mathsf{M}) \mid \forall (\mathtt{W}, \mathsf{N}) \in R_{\mathtt{A}},\ (\mathtt{V}\,\mathtt{W}, \mathsf{M} \bullet \mathsf{N}) \in \overline{R}_{\mathtt{B}}\}$$

and

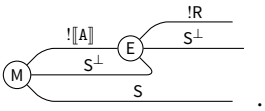
$$R_{\mathtt{A}}^{\top} = \{(\mathtt{E}[\,-\,], \mathtt{E}) \mid \forall (\mathtt{V}, \mathtt{M}) \in R_{\mathtt{A}},\ (\mathtt{E}[\mathtt{V}], \mathtt{E} \circ !_0 \mathtt{M}) \in \Omega\},$$

$$\overline{R}_{\mathtt{A}} = \{(\mathtt{M}, \mathtt{M}) \mid \forall (\mathtt{E}[\,-\,], \mathtt{E}) \in R_{\mathtt{A}}^{\top},\ (\mathtt{E}[\mathtt{M}], \mathtt{E} \star \mathtt{M}) \in \Omega\}$$

where Mealy machines $\mathtt{M} \bullet \mathtt{N} \colon \mathtt{I} \multimap \mathtt{S} \otimes \mathtt{S}^{\perp} \otimes !\llbracket \mathtt{B} \rrbracket$ and $\mathtt{E} \star \mathtt{M} \colon \mathtt{I} \multimap \mathtt{S} \otimes \mathtt{S}^{\perp} \otimes !\mathtt{R}$ are given by:



,



.

Lemma 19. *Let* $\mathtt{A}$ *be a type.*

(1) If $(\mathtt{V}, \mathtt{M}) \in R_{\mathtt{A}}$, *then* $(\mathtt{V}, \mathsf{ax}_{\mathtt{S}} \otimes !_0 \mathtt{M}) \in \overline{R}_{\mathtt{A}}$.
(2) If $\mathtt{M} \xrightarrow{\mathrm{red}} \mathtt{N}$, *then* $(\mathtt{M}, \mathtt{M}) \in \overline{R}_{\mathtt{A}}$ *if and only if* $(\mathtt{N}, \mathtt{M}) \in \overline{R}_{\mathtt{A}}$.
(3) If $(\mathtt{M}, \mathtt{M}) \in \overline{R}_{\mathtt{A}}$ *and* $\mathtt{M} \simeq \mathtt{N}$, *then* $(\mathtt{M}, \mathtt{N}) \in \overline{R}_{\mathtt{A}}$.
(4) For any closed term $\mathtt{M} \colon \mathtt{A}$, $(\mathtt{M}, \perp_{\mathtt{S} \otimes \mathtt{S}^{\perp} \otimes !\llbracket \mathtt{A} \rrbracket}) \in \overline{R}_{\mathtt{A}}$ *where* $\perp_{\mathtt{X}} \colon \mathtt{I} \multimap \mathtt{X}$ *is the token machine whose transition function is the empty partial measurable function.*
(5) For any closed value $\mathtt{V} \colon \mathtt{A} \to \mathtt{B}$, $(\mathtt{V}, \perp_{\llbracket \mathtt{A} \to \mathtt{B} \rrbracket}) \in R_{\mathtt{A} \to \mathtt{B}}$.
(6) If $(\mathtt{M}, \mathtt{M}_i) \in \overline{R}_{\mathtt{A}}$ *for Mealy machines* $(\mathtt{M}_i)_{i \in \mathbb{N}}$ *such that* $S_{\mathtt{M}_1} = S_{\mathtt{M}_2} = \cdots$ *and* $\mathrm{init}_{\mathtt{M}_1} = \mathrm{init}_{\mathtt{M}_2} = \cdots$ *and* $\tau_{\mathtt{M}_1} \leq \tau_{\mathtt{M}_2} \leq \cdots$, *then* $(\mathtt{M}, \mathtt{N}) \in \overline{R}_{\mathtt{A}}$ *where* $\mathtt{N}$ *is given by* $S_{\mathtt{N}} = S_{\mathtt{M}_1}$, $\mathrm{init}_{\mathtt{N}} = \mathrm{init}_{\mathtt{M}_1}$ *and* $\tau_{\mathtt{N}} = \bigvee_{n \in \mathbb{N}} \tau_{\mathtt{M}_n}$. *Here, the lub is the union of graph relations.*

Proof. (1) and (2) follow from the definition of $\overline{R}_{\mathtt{A}}$. (3) follows from Lemma 15. (4) holds because $\mathsf{weight}_{\mathtt{E} \star \perp_{\mathtt{S} \otimes \mathtt{S}^{\perp} \otimes !\llbracket \mathtt{A} \rrbracket}}$ is the empty partial measurable function for any $\mathtt{E} \colon !\llbracket \mathtt{A} \rrbracket \multimap \mathtt{S} \otimes \mathtt{S}^{\perp} \otimes !\mathtt{R}$. (5) follows from (4) and $\perp_{\llbracket \mathtt{A} \to \mathtt{B} \rrbracket} \bullet \mathtt{N} \simeq \perp_{\mathtt{S} \otimes \mathtt{S}^{\perp} \otimes !\llbracket \mathtt{B} \rrbracket}$ for any Mealy machine $\mathtt{N} \colon \mathtt{I} \multimap \llbracket \mathtt{A} \rrbracket$. (6) follows from the definition of $\mathsf{weight}_{(-)}$ and $\mathsf{value}_{(-)}$. □

Lemma 20. (Basic Lemma). *Let* $\Delta = (\mathtt{x} : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n)$ *be a type environment.*

- *For any term* $\Delta \vdash \mathtt{M} : \mathtt{A}$ *and for any* $(\mathtt{V}_i, \mathtt{N}_i) \in R_{\mathtt{A}_i}$ *for* $i = 1, 2, \ldots, n$, *we have*
$$\big(\mathtt{M}\{\mathtt{V}_1/\mathtt{x}_1, \ldots, \mathtt{V}_n/\mathtt{x}_n\}, \llbracket \mathtt{M} \rrbracket \circ (!_0 \mathtt{N}_1 \otimes \cdots \otimes !_0 \mathtt{N}_n)\big) \in \overline{R}_{\mathtt{A}}.$$
- *For any value* $\Delta \vdash \mathtt{V} : \mathtt{A}$ *and for any* $(\mathtt{V}_i, \mathtt{N}_i) \in R_{\mathtt{A}_i}$ *for* $i = 1, 2, \ldots, n$, *we have*
$$\big(\mathtt{V}\{\mathtt{V}_1/\mathtt{x}_1, \ldots, \mathtt{V}_n/\mathtt{x}_n\}, \llbracket \mathtt{V} \rrbracket_{\mathrm{v}} \circ (!_0 \mathtt{N}_1 \otimes \cdots \otimes !_0 \mathtt{N}_n)\big) \in R_{\mathtt{A}}.$$

Proof. By induction on $\mathtt{M}$ and $\mathtt{V}$. Most cases follow from Lemma 19 with some diagrammatic reasoning. Here, we only check a few induction steps.

- When $\mathtt{M} = \mathtt{score}(\mathtt{V})$, it is enough to check $(\mathtt{score}(\mathtt{r}_c), \mathsf{sc} \circ \mathtt{r}_c) \in \overline{R}_{\mathtt{Unit}}$ for all $c \in \mathbb{R}$. Let $(\mathtt{E}, \mathtt{E})$ be a pair in $R_{\mathtt{Unit}}^{\top}$. If $\mathsf{weight}_{\mathtt{E} \star (\mathsf{sc} \circ \mathtt{r}_c)}(a, u) = a'$ and $\mathsf{value}_{\mathtt{E} \star (\mathsf{sc} \circ \mathtt{r}_c)}(a, u) = b$, then by the definition of sc, we see that $\mathtt{E}$ satisfies $\mathsf{weight}_{\mathtt{E} \circ !_0 \mathsf{id}_{\mathrm{I}}}(|c|\ a, u) = a'$ and $\mathsf{value}_{\mathtt{E} \circ !_0 \mathsf{id}_{\mathrm{I}}}(|c|\ a, u) = b$. Because $(\mathtt{E}, \mathtt{E}) \in R_{\mathtt{Unit}}^{\top}$, we obtain $(\mathtt{E}[\mathtt{score}(\mathtt{r}_c)], a, u) \to (\mathtt{E}[\mathtt{skip}], |c|\ a, u) \to^* (\mathtt{r}_b, a', \varepsilon)$.
- When $\mathtt{M} = \mathtt{sample}$, we show that $(\mathtt{sample}, \mathsf{sa})$ is an element of $\overline{R}_{\mathtt{Real}}$. For any $(\mathtt{E}, \mathtt{E})$ in $R_{\mathtt{Real}}^{\top}$, if $\mathsf{weight}_{\mathtt{E} \star \mathsf{sa}}(a, u) = a'$ and $\mathsf{value}_{\mathtt{E} \star \mathsf{sa}}(a, u) = b$, then by the definition of sa, we see that u must be $c :: v$ for some $c \in \mathbb{R}_{[0,1]}$ and $v \in \mathbb{R}_{[0,1]}^*$ such that $\mathsf{weight}_{\mathtt{E} \circ !_0 \mathtt{r}_c}(a, v) = a'$ and $\mathsf{value}_{\mathtt{E} \circ !_0 \mathtt{r}_c}(a, v) = b$. Because $(\mathtt{E}, \mathtt{E}) \in R_{\mathtt{Real}}^{\top}$, we obtain $(\mathtt{E}[\mathtt{sample}], a, u) \to (\mathtt{E}[\mathtt{r}_c], a, v) \to^* (\mathtt{r}_b, a', \varepsilon)$.

- When $\mathtt{V} = \mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f}, \mathtt{x}, \mathtt{N})$, for simplicity, we suppose that $\mathtt{V}$ is a closed value. By induction hypothesis and (2) in Lemma 19, we can check that for all $k \in \mathbb{N}$,

$$(\mathtt{V}, [\![\lambda \mathtt{x}^{\mathtt{A}}.\, \mathtt{N}]\!] \circ \overbrace{!_0([\![\lambda \mathtt{x}^{\mathtt{A}}.\, \mathtt{N}]\!] \circ \cdots \circ !_0([\![\lambda \mathtt{x}^{\mathtt{A}}.\, \mathtt{N}]\!]}^{k} \circ !_0 \bot_{[\![\mathtt{A}\to\mathtt{B}]\!]}) \cdots)) \in R_{\mathtt{A}\to\mathtt{B}}.$$

  By Proposition 12 and by (6) in Lemma 19, we obtain $(\mathtt{V}, [\![\mathtt{V}]\!]_{\mathtt{v}}) \in R_{\mathtt{A}\to\mathtt{B}}$. □

6.5 *Proof of Theorem 7*

We first prove soundness. The following proposition means that $\mu_{\mathtt{M},1}$ takes all terminating probabilistic branches into account.

Proposition 21. (Soundness). *For all* $\vdash \mathtt{M} : \mathtt{Real}$ *and all* $n \in \mathbb{N}$, *if* $\mathtt{M} \Rightarrow_n \mu$ *for a measure* μ *on* $\mathbb{R}$, *then* $a\,\mu \le \mu_{\mathtt{M},a}$ *for any* $a \in \mathbb{R}_{\ge 0}$.

Proof. By induction on n. (Base case) Easy. (Induction step) By case analysis.

- If $\mathtt{M} = \mathtt{E}[\mathtt{N}] \Rightarrow_{n+1} \mu$ and $\mathtt{N} \xrightarrow{\mathrm{red}} \mathtt{L}$ for some term $\mathtt{L}$, then we can check $[\![\mathtt{E}[\mathtt{N}]]\!] \simeq [\![\mathtt{E}[\mathtt{L}]]\!]$ by Lemma 14 and by Proposition 13 with some diagrammatic reasoning. Because $\mathtt{E}[\mathtt{L}] \Rightarrow_n \mu$, it follows from induction hypothesis and Lemma 15 that $a\,\mu \le \mu_{\mathtt{M},a}$.
- If $\mathtt{M} = \mathtt{r}_b$, then $\mu_{\mathtt{M},a} = a\,\delta_b$.
- If $\mathtt{M} = \mathtt{E}[\mathtt{score}(\mathtt{r}_b)]$ and $\mathtt{E}[\mathtt{skip}] \Rightarrow_n \mu$, then by Lemma 16 and induction hypothesis, we obtain $\mu_{\mathtt{M},a} \ge |b|\, a\,\mu$.
- If $\mathtt{M} = \mathtt{E}[\mathtt{sample}]$, then

$$\begin{aligned}
\mu_{\mathtt{M},a}(A) &= \sum_{n\in\mathbb{N}} \int_{\mathbb{R}^n_{[0,1]}} \underline{\mathsf{weight}}_{\mathtt{M}}(a,u)\,[\underline{\mathsf{value}}_{\mathtt{M}}(a,u) \in A]\,\mathrm{d}u \\
&\overset{\text{Lem 17}}{=} \sum_{n\in\mathbb{N}} \int_{\mathbb{R}^{n+1}_{[0,1]}} \underline{\mathsf{weight}}_{\mathtt{E}[\mathtt{r}_b]}(a,v)[\underline{\mathsf{value}}_{\mathtt{E}[\mathtt{r}_b]}(a,v) \in A]\,\mathrm{d}(b,v) \\
&= \int_{\mathbb{R}_{[0,1]}} \left(\sum_{n\in\mathbb{N}} \int_{\mathbb{R}^n_{[0,1]}} \underline{\mathsf{weight}}_{\mathtt{E}[\mathtt{r}_b]}(a,v)[\underline{\mathsf{value}}_{\mathtt{E}[\mathtt{r}_b]}(a,v)]\,\mathrm{d}v \right) \mathrm{d}b.
\end{aligned}$$

  Let $k\colon \mathbb{R} \rightsquigarrow \mathbb{R}$ be a finite kernel such that $\mathtt{E}[\mathtt{r}_b] \Rightarrow_n k(b,-)$ for any $b \in \mathbb{R}$. Then by induction hypothesis, $\mu_{\mathtt{M},a}(A) \ge a \int_{\mathbb{R}_{[0,1]}} k(b,-)\,\mathrm{d}b$.

□

We next show that $\mathtt{M} \Rightarrow_\infty \mu$ implies $\mu_{\mathtt{M},a} \le a\,\mu$ by means of logical relations. We define a binary relation Σ between closed terms of type $\mathtt{Real}$ and Mealy machines from I to $\mathsf{S} \otimes \mathsf{S}^{\perp} \otimes\, !\mathsf{R}$ by:

$$(\mathtt{M}, \mathsf{M}) \in \Sigma \iff \text{if } \mathtt{M} \Rightarrow_\infty \mu, \text{ then } \mu_{\mathtt{M},a} \le a\,\mu \text{ for any } a \in \mathbb{R}_{\ge 0}.$$

We then inductively define binary relations:

$$\begin{aligned}
Q_{\mathtt{A}} &\subseteq \{\text{closed values of type } \mathtt{A}\} \times \{\text{Mealy machines from } \mathsf{I} \text{ to } [\![\mathtt{A}]\!]\} \\
Q^{\top}_{\mathtt{A}} &\subseteq \{\text{evaluation contexts } \mathtt{x} : \mathtt{A} \vdash \mathtt{E}[\mathtt{x}] : \mathtt{Real}\} \times \{\text{Mealy machines from } !{[\![\mathtt{A}]\!]} \text{ to } \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes\, !\mathsf{R}\} \\
\overline{Q}_{\mathtt{A}} &\subseteq \{\text{closed terms of type } \mathtt{A}\} \times \{\text{Mealy machines from } \mathsf{I} \text{ to } \mathsf{S} \otimes \mathsf{S}^{\perp} \otimes\, !{[\![\mathtt{A}]\!]}\}
\end{aligned}$$

by:

$$Q_{\mathtt{Real}} = \{(\mathtt{r}_a, \mathtt{r}_a) \mid a \in \mathbb{R}\},$$
$$Q_{\mathtt{Unit}} = \{(\mathtt{skip}, \mathtt{id}_{\mathtt{I}})\},$$
$$Q_{\mathtt{A}\to\mathtt{B}} = \{(\mathtt{V}, \mathtt{M}) \mid \forall (\mathtt{W}, \mathtt{N}) \in Q_{\mathtt{A}},\ (\mathtt{V}\,\mathtt{W}, \mathtt{M} \bullet \mathtt{N}) \in \overline{Q}_{\mathtt{B}}\}$$

and

$$Q^{\top}_{\mathtt{A}} = \{(\mathtt{E}[-], \mathtt{E}) \mid \forall (\mathtt{V}, \mathtt{M}) \in Q_{\mathtt{A}},\ (\mathtt{E}[\mathtt{V}], \mathtt{E} \circ !_0 \mathtt{M}) \in \Sigma\},$$
$$\overline{Q}_{\mathtt{A}} = \{(\mathtt{M}, \mathtt{M}) \mid \forall (\mathtt{E}[-], \mathtt{E}) \in Q^{\top}_{\mathtt{A}},\ (\mathtt{E}[\mathtt{M}], \mathtt{E} \star \mathtt{M}) \in \Sigma\}.$$

Lemma 22. *Let* $\mathtt{A}$ *be a type.*

(1) If $(\mathtt{V}, \mathtt{M}) \in Q_{\mathtt{A}}$, *then* $(\mathtt{V}, \mathsf{ax}_{\mathtt{S}} \otimes !_0 \mathtt{M}) \in \overline{Q}_{\mathtt{A}}$.
(2) If $\mathtt{M} \xrightarrow{\mathrm{red}} \mathtt{N}$, *then* $(\mathtt{M}, \mathtt{M}) \in \overline{Q}_{\mathtt{A}}$ *if and only if* $(\mathtt{N}, \mathtt{M}) \in \overline{Q}_{\mathtt{A}}$.
(3) If $(\mathtt{M}, \mathtt{M}) \in \overline{Q}_{\mathtt{A}}$ *and* $\mathtt{M} \simeq \mathtt{N}$, *then* $(\mathtt{M}, \mathtt{N}) \in \overline{Q}_{\mathtt{A}}$.
(4) For any closed value $\mathtt{V} : \mathtt{A} \to \mathtt{B}$, $(\mathtt{V}, \bot_{[\![\mathtt{A}\to\mathtt{B}]\!]}) \in Q_{\mathtt{A}\to\mathtt{B}}$.
(5) If $(\mathtt{M}, \mathtt{M}_i) \in \overline{Q}_{\mathtt{A}}$ *for Mealy machines* $(\mathtt{M}_i)_{i\in\mathbb{N}}$ *such that* $S_{\mathtt{M}_1} = S_{\mathtt{M}_2} = \cdots$ *and* $\mathrm{init}_{\mathtt{M}_1} = \mathrm{init}_{\mathtt{M}_2} = \cdots$ *and* $\tau_{\mathtt{M}_1} \le \tau_{\mathtt{M}_2} \le \cdots$, *then* $(\mathtt{M}, \mathtt{N}) \in \overline{Q}_{\mathtt{A}}$ *where* $\mathtt{N}$ *is given by* $S_{\mathtt{N}} = S_{\mathtt{M}_1}$, $\mathrm{init}_{\mathtt{N}} = \mathrm{init}_{\mathtt{M}_1}$ *and* $\tau_{\mathtt{N}} = \bigvee_{n\in\mathbb{N}} \tau_{\mathtt{M}_n}$. *Here, the lub is the union of graph relations.*

Proof. Similar to the proof of Lemma 19. □

Lemma 23. (Basic Lemma). *Let* $\Delta = (\mathtt{x} : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n)$ *be a type environment.*

- *For any term* $\Delta \vdash \mathtt{M} : \mathtt{A}$ *and for any* $(\mathtt{V}_i, \mathtt{N}_i) \in Q_{\mathtt{A}_i}$ *for* $i = 1, 2, \ldots, n$, *we have*
$$\big(\mathtt{M}\{\mathtt{V}_1/\mathtt{x}_1, \ldots, \mathtt{V}_n/\mathtt{x}_n\}, [\![\mathtt{M}]\!] \circ (!_0\mathtt{N}_1 \otimes \cdots \otimes !_0\mathtt{N}_n)\big) \in \overline{Q}_{\mathtt{A}}.$$
- *For any value* $\Delta \vdash \mathtt{V} : \mathtt{A}$ *and for any* $(\mathtt{V}_i, \mathtt{N}_i) \in Q_{\mathtt{A}_i}$ *for* $i = 1, 2, \ldots, n$, *we have*
$$\big(\mathtt{V}\{\mathtt{V}_1/\mathtt{x}_1, \ldots, \mathtt{V}_n/\mathtt{x}_n\}, [\![\mathtt{V}]\!]_{\mathrm{v}} \circ (!_0\mathtt{N}_1 \otimes \cdots \otimes !_0\mathtt{N}_n)\big) \in Q_{\mathtt{A}}.$$

Proof. By induction on $\mathtt{M}$ and $\mathtt{V}$. Most cases follow from Lemma 22 with some diagrammatic reasoning. Here, we only check a few induction steps.

- When $\mathtt{M} = \mathtt{score}(\mathtt{V})$, it is enough to show that $(\mathtt{score}(\mathtt{r}_a), \mathsf{sc} \circ \mathtt{r}_a)$ is an element of $\overline{Q}_{\mathtt{Unit}}$ for any $a \in \mathbb{R}$. Let $(\mathtt{E}, \mathtt{E})$ be a pair in $Q^{\top}_{\mathtt{Unit}}$, and let μ be the measure such that $\mathtt{E}[\mathtt{skip}] \Rightarrow_{\infty} \mu$. By the definition of sc and induction hypothesis, for any $b \in \mathbb{R}_{\ge 0}$, we obtain $\mu_{\mathtt{E}\star(\mathsf{sc}\circ \mathtt{r}_a),b} = \mu_{\mathtt{E}\circ !_0 \mathtt{id}_{\mathtt{I}},|a|\,b} \ge |a|\,b\,\mu$.
- When $\mathtt{M} = \mathtt{sample}$, we show that $(\mathtt{sample}, \mathsf{sa})$ is an element of $\overline{Q}_{\mathtt{Real}}$. Let $k \colon \mathbb{R} \rightsquigarrow \mathbb{R}$ be an s-finite kernel such that $\mathtt{E}[\mathtt{r}_a] \Rightarrow_{\infty} k(a, -)$. Then for any $(\mathtt{E}, \mathtt{E}) \in Q^{\top}_{\mathtt{Real}}$, and for any $a \in \mathbb{R}_{\ge 0}$,

$$\begin{aligned}
\mu_{\mathtt{E}\star\mathsf{sa},a}(A) &= \sum_{n\in\mathbb{N}} \int_{\mathbb{R}^n_{[0,1]}} \underline{\mathsf{weight}}_{\mathtt{E}\star\mathsf{sa}}(a, u)\, [\underline{\mathsf{value}}_{\mathtt{E}\star\mathsf{sa}}(a, u) \in A]\, \mathrm{d}u \\
&= \sum_{n\in\mathbb{N}} \int_{\mathbb{R}^{n+1}_{[0,1]}} \underline{\mathsf{weight}}_{\mathtt{E}\circ !_0 \mathtt{r}_b}(a, v) [\underline{\mathsf{value}}_{\mathtt{E}\circ !_0 \mathtt{r}_b}(a, v) \in A]\, \mathrm{d}(b, v) \\
&= \int_{\mathbb{R}_{[0,1]}} \left(\sum_{n\in\mathbb{N}} \int_{\mathbb{R}^n_{[0,1]}} \underline{\mathsf{weight}}_{\mathtt{E}\circ !_0 \mathtt{r}_b}(a, v) [\underline{\mathsf{value}}_{\mathtt{E}\circ !_0 \mathtt{r}_b}(a, v) \in A]\, \mathrm{d}v \right) \mathrm{d}b.
\end{aligned}$$

  Hence, by induction hypothesis, we obtain $\mu_{\mathtt{E}\star\mathsf{sa},a}(A) \le a \int_{\mathbb{R}_{[0,1]}} k(b, A)$.

- When $\mathtt{V} = \mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f}, \mathtt{x}, \mathtt{N})$, for simplicity, we suppose that $\mathtt{V}$ is a closed term. By induction hypothesis and (4) in Lemma 22, we can check that for all $k \in \mathbb{N}$,

$$(\mathtt{V}, [\![\lambda \mathtt{x}^{\mathtt{A}}.\, \mathtt{N}]\!] \circ \overbrace{!_0([\![\lambda \mathtt{x}^{\mathtt{A}}.\, \mathtt{N}]\!] \circ \cdots \circ !_0([\![\lambda \mathtt{x}^{\mathtt{A}}.\, \mathtt{N}]\!]}^{k} \circ !_0 \bot_{[\![\mathtt{A} \to \mathtt{B}]\!]}) \cdots)) \in Q_{\mathtt{A} \to \mathtt{B}}.$$

  By Proposition 12 and by (5) in Lemma 22, we obtain $(\mathtt{V}, [\![\mathtt{V}]\!]_{\mathrm{v}}) \in Q_{\mathtt{A} \to \mathtt{B}}$. □

7. How About S-Finite Kernels?

The reader experienced with the semantics of probabilistic programming languages have probably already wondered whether a GoI model for **PCFSS** could be given out of s-finite kernels instead of measurable functions, following Staton's work on the semantics of a first-order probabilistic programming language (Staton 2017).

The answer is indeed positive: the kind of construction we have presented in Section 5 can in fact be adapted to the category of measurable spaces *and s-finite kernels*. The latter, being traced monoidal, has all the necessary structure one needs (Abramsky et al. 2002). What one obtains proceeding this way is indeed a GoI model, but adequate only for the distribution-based operational semantics.

The interpretation of any program in this alternative GoI can be seen as structurally identical to the one from Section 5 once the sample and score operators are interpreted as usual, namely as those s-finite kernels which actually perform sampling and scoring *internally*. Below, we introduce *probabilistic* Mealy machines whose transitions are described in terms of an s-finite kernel, and we give some basic probabilistic Mealy machines. Finally, we give an adequate GoI model for the distribution-based operational semantics, and we apply the GoI model to prove commutativity of let-bindings.

Being adequate for the distribution-based semantics directly (and not by way of integration as in Theorem 7) has the pleasant consequence of validating a number of useful program transformations, and in particular commutation of sampling and scoring effects, see Borgström et al. (2016) for a thorough discussion about this topic, and about how s-finite kernels are a particularly nice way of achieving commutativity in presence of scoring.

7.1 Probabilistic Mealy machine

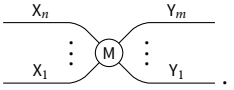
A probabilistic Mealy machine is a Mealy machine whose transition function is given by an s-finite kernel. Let X and Y be **Int**-objects.

Definition 24. *A* probabilistic Mealy machine M *from* X *to* Y *consists of*

- *a measurable space* S_{M} *called the* state space *of* M*;*
- *an element* $\mathrm{init}_{\mathsf{M}} \in S_{\mathsf{M}}$ *called the* initial state *of* M*;*
- *an s-finite kernel* $\tau_{\mathsf{M}} \colon (X^+ + Y^-) \times S_{\mathsf{M}} \rightsquigarrow (X^- + Y^+) \times S_{\mathsf{M}}$ *called the* transition kernel *of* M.

When M *is a probabilistic Mealy machine from* X *to* Y*, we write* $\mathsf{M} \colon \mathsf{X} \multimap \mathsf{Y}$.

We can regard a Mealy machine $\mathsf{M} \colon \mathsf{X} \multimap \mathsf{Y}$ (as we defined in Section 5.1) as a probabilistic Mealy machine from X to Y by identifying the transition function τ_{M} with the corresponding s-finite kernel $\tau_{\mathsf{M}}^{\#}$. We write $\mathsf{M}^{\#}$ for this probabilistic Mealy machine. Just like Mealy machines, we depict a probabilistic Mealy machine $\mathsf{M} \colon \otimes_{1 \le i \le n} \mathsf{X}_i \multimap \otimes_{1 \le i \le m} \mathsf{Y}_i$ as a node with labelled wires:



In order to informally explain behaviour of probabilistic Mealy machines, we sometimes use thick arrows to present how tokens are handled.

7.2 Construction of probabilistic Mealy machines

We introduce probabilistic Mealy machines and their constructions that are building blocks of our denotational semantics based on s-finite kernels. Most of them are adaptations of Mealy machines in Section 5.2, and we just give their formal definitions.

7.2.1 Composition

For probabilistic Mealy machines $\mathsf{M}\colon \mathsf{X} \multimap \mathsf{Y}$ and $\mathsf{N}\colon \mathsf{Y} \multimap \mathsf{Z}$, we define a probabilistic Mealy machine $\mathsf{N} \circ \mathsf{M}\colon \mathsf{X} \multimap \mathsf{Z}$ by:

$$S_{\mathsf{N}\circ\mathsf{M}} = S_{\mathsf{N}} \times S_{\mathsf{M}}, \qquad \mathrm{init}_{\mathsf{N}\circ\mathsf{M}} = (\mathrm{init}_{\mathsf{N}}, \mathrm{init}_{\mathsf{M}})$$

and

$$\tau_{\mathsf{N}\circ\mathsf{M}} = k_{X^+,Z^-,X^-,Z^+} + \sum_{n\in\mathbb{N}} k_{Y^+,Y^-,X^-,Z^+} \circ k^n_{Y^+,Y^-,Y^+,Y^-} \circ k_{X^-,Z^+,Y^+,Y^-}$$

where $k_{A,B,C,D}\colon (A+B)\times S_{\mathsf{N}\circ\mathsf{M}} \rightsquigarrow (C+D)\times S_{\mathsf{N}\circ\mathsf{M}}$ are the restrictions of an s-finite kernel:

$$h = \begin{array}{c} (X^+ + Z^- + Y^+ + Y^-)\times S_{\mathsf{N}\circ\mathsf{M}} \\ \wr \\ (X^+ + Y^-)\times S_{\mathsf{M}}\times S_{\mathsf{N}} + (Y^+ + Z^-)\times S_{\mathsf{N}}\times S_{\mathsf{M}} \\ \wr (\tau_{\mathsf{M}} \otimes \mathrm{id}^{\#}_{S_{\mathsf{N}}}) \oplus (\tau_{\mathsf{N}} \otimes \mathrm{id}^{\#}_{S_{\mathsf{M}}}) \\ (X^- + Y^+)\times S_{\mathsf{M}}\times S_{\mathsf{N}} + (Y^- + Z^+)\times S_{\mathsf{N}}\times S_{\mathsf{M}} \\ \wr \\ (X^- + Z^+ + Y^+ + Y^-)\times S_{\mathsf{N}\circ\mathsf{M}} \end{array}.$$

The first arrow and the last arrow in the definition of h are induced by canonical measurable isomorphisms. For example, $k_{X^+,Z^-,X^-,Z^+}\colon (X^+ + Z^-)\times S_{\mathsf{N}\circ\mathsf{M}} \rightsquigarrow (X^- + Z^+)\times S_{\mathsf{N}\circ\mathsf{M}}$ is given by:

$$k_{X^+,Z^-,X^-,Z^+}((w,s,t),A) = h((u(w),s,t),(v\times \mathrm{id}_{S_{\mathsf{N}\circ\mathsf{M}}})(A))$$

where $u\colon X^+ + Z^- \to X^+ + Z^- + Y^+ + Y^-$ and $v\colon X^- + Z^+ \to X^- + Z^+ + Y^+ + Y^-$ are the canonical injections.

7.2.2 Monoidal products

For probabilistic Mealy machines $\mathsf{M}\colon \mathsf{X} \multimap \mathsf{Z}$ and $\mathsf{N}\colon \mathsf{Y} \multimap \mathsf{W}$, we define a probabilistic Mealy machine $\mathsf{M}\otimes\mathsf{N}\colon \mathsf{X}\otimes\mathsf{Y} \multimap \mathsf{Z}\otimes\mathsf{W}$ by:

$$S_{\mathsf{M}\otimes\mathsf{N}} = S_{\mathsf{M}} \times S_{\mathsf{N}}, \qquad \mathrm{init}_{\mathsf{M}\otimes\mathsf{N}} = (\mathrm{init}_{\mathsf{M}}, \mathrm{init}_{\mathsf{N}})$$

and

$$\tau_{\mathsf{M}\otimes\mathsf{N}} = \begin{array}{c} ((X^+ + Y^+) + (Z^- + W^-)) \times S_{\mathsf{M}\otimes\mathsf{N}} \\ \wr \\ (X^+ + Z^-) \times S_{\mathsf{M}} \times S_{\mathsf{N}} + (Y^+ + W^-) \times S_{\mathsf{N}} \times S_{\mathsf{M}} \\ \wr (\tau_{\mathsf{M}} \otimes \mathrm{id}^{\#}_{S_{\mathsf{N}}}) \oplus (\tau_{\mathsf{N}} \otimes \mathrm{id}^{\#}_{S_{\mathsf{M}}}) \\ (X^- + Z^+) \times S_{\mathsf{M}} \times S_{\mathsf{N}} + (Y^- + W^+) \times S_{\mathsf{N}} \times S_{\mathsf{M}} \\ \wr \\ ((X^- + Y^-) + (Z^+ + W^+)) \times S_{\mathsf{M}\otimes\mathsf{N}} \end{array}$$

where the first and the last arrows are induced by canonical measurable isomorphisms.

7.2.3 A resource modality

Let $\mathsf{M}\colon \bigotimes_{1\le i\le n} \mathsf{X}_i \multimap \mathsf{Y}$ be a probabilistic Mealy machine. We define $!_n\mathsf{M}\colon \bigotimes_{1\le i\le n} !\mathsf{X}_i \multimap !\mathsf{Y}$ to be a probabilistic Mealy machine consisting of:

$$S_{!_n\mathsf{M}} = S_{\mathsf{M}}^{\mathbb{N}}, \qquad \mathrm{init}_{!_n\mathsf{M}} = (\mathrm{init}_{\mathsf{M}}, \mathrm{init}_{\mathsf{M}}, \dots)$$

and

$$\tau_{!_n\mathsf{M}} = \begin{array}{c} (\sum_{1\le i\le n} \mathbb{N} \times X_i^+ + \mathbb{N} \times Y^-) \times S_{\mathsf{M}}^{\mathbb{N}} \\ \wr g^{\#} \\ \mathbb{N} \times (\sum_{1\le i\le n} X_i^+ + Y^-) \times S_{\mathsf{M}} \times S_{\mathsf{M}}^{\mathbb{N}} \\ \wr \bigoplus_{n\in\mathbb{N}} \tau_{\mathsf{M}} \otimes \mathrm{id}^{\#}_{S_{\mathsf{M}}^{\mathbb{N}}} \\ \mathbb{N} \times (\sum_{1\le i\le n} X_i^- + Y^+) \times S_{\mathsf{M}} \times S_{\mathsf{M}}^{\mathbb{N}} \\ \wr (h^{-1})^{\#} \\ (\sum_{1\le i\le n} \mathbb{N} \times X_i^- + \mathbb{N} \times Y^+) \times S_{\mathsf{M}}^{\mathbb{N}} \end{array}$$

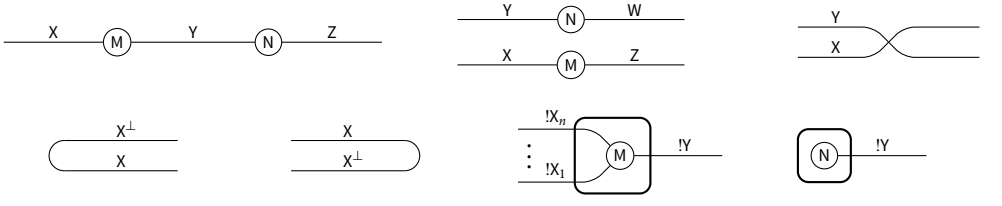
where g is a measurable isomorphism given by:

$$g(j@z, (s_1, s_2, \dots)) = (j, z, s_j, (s_1, s_2, \dots, s_{j-1}, s_{j+1}, s_{j+2}, \dots))$$

for $j \in \mathbb{N}$, $z \in \sum_{1\le i\le n} \mathbb{N} \times X_i^+ + \mathbb{N} \times Y^-$ and $(s_1, s_2, \dots) \in S_{\mathsf{M}}^{\mathbb{N}}$; and h is a measurable isomorphism given in the same way. (The operator $(-)@(-)$ is given in Section 5.2.5.) Below, we omit the subscript of $!_n\mathsf{M}$ when $n = 1$.

7.2.4 Diagrammatic presentation

We adopt the same diagrammatic presentation: we depict the composition, the monoidal products, $\mathsf{sym}^{\#}_{\mathsf{X},\mathsf{Y}}$, $\mathsf{ax}^{\#}_{\mathsf{X}}$, $\mathsf{cut}^{\#}_{\mathsf{X}}$, and the resource modality $!_n(-)$ as follows:



7.2.5 Scoring

We define an **Int**-object J by $(\emptyset, \mathbf{1})$, and we often identify $\mathsf{J} \otimes \mathsf{J}^{\perp}$ with $(\mathbf{1}, \mathbf{1})$. We use J as the 'state' object in the denotational semantics based on s-finite kernels. Here, we use $\mathsf{J} = (\mathbf{1}, \emptyset)$ rather

than $\mathsf{S} = (\mathbb{R}_{\geq 0} \times \mathbb{R}^*_{[0,1]}, \emptyset)$ as in the denotational semantics based on partial measurable functions because s-finite kernels can manage these informations internally.

We define $\mathsf{Sc} \colon \mathsf{R} \multimap \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !\mathsf{I}$ to be a probabilistic Mealy machine consisting of

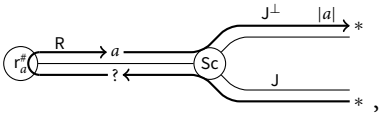
$$S_{\mathsf{Sc}} = \mathbf{1}, \qquad \mathrm{init}_{\mathsf{Sc}} = *$$

and $\tau_{\mathsf{Sc}} \colon (\mathbb{R} + (\mathbf{1} + \emptyset)) \times S_{\mathsf{Sc}} \rightsquigarrow (\{?\} + (\mathbf{1} + \emptyset)) \times S_{\mathsf{Sc}}$ given by:

$$\tau_{\mathsf{Sc}}(((\bullet, a), \mathrm{init}_{\mathsf{Sc}}), A) = |a|\,[((\circ, (\bullet, *)), \mathrm{init}_{\mathsf{Sc}}) \in A],$$

$$\tau_{\mathsf{Sc}}(((\circ, (\bullet, *)), \mathrm{init}_{\mathsf{Sc}}), A) = [((\bullet, ?), \mathrm{init}_{\mathsf{Sc}}) \in A].$$

The probabilistic Mealy machine simulates scoring $\mathtt{score}(\mathtt{r}_a)$ as follows:



that is, Sc first throws a query to its argument, and to each answer a from its argument, Sc outputs $*$ with 'likelihood' $|a|$.

7.2.6 Sampling

We define $\mathsf{Sa} \colon \mathsf{I} \multimap \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !\mathsf{R}$ to be a probabilistic Mealy machine consisting of

$$S_{\mathsf{Sa}} = \mathbb{R}_{[0,1]}, \qquad \mathrm{init}_{\mathsf{Sa}} = 0$$

and

$$\tau_{\mathsf{Sa}} \colon (\emptyset + (\mathbf{1} + \mathbb{N} \times \{?\})) \times S_{\mathsf{Sa}} \rightsquigarrow (\emptyset + (\mathbf{1} + \mathbb{N} \times \mathbb{R})) \times S_{\mathsf{Sa}}$$

given by:

$$\tau_{\mathsf{Sa}}(((\circ, (\bullet, *)), s), A) = \mu_{\mathrm{Borel}}(\{a \in \mathbb{R}_{[0,1]} \mid ((\circ, (\bullet, *)), a) \in A\}),$$

$$\tau_{\mathsf{Sa}}(((\circ, (\circ, (n, ?))), s), A) = [((\circ, (\circ, (n, s))), s) \in A].$$

The probabilistic Mealy machine behaves as follows:

1. In the initial state $\mathrm{init}_{\mathsf{Sa}}$, given $*$ from the J-wire, Sa draws a real number from the Borel measure and memorises the real number. For example, the probability of the state being a real number in $[0.1, 0.3]$ after this transition is 0.2.
2. If Sa drew a in (1), then for every 'query' $(n, ?)$, Sa answers (n, a).

We note that we always use Sa in this way in our denotational semantics, and therefore, the definition of $\mathrm{init}_{\mathsf{Sa}}$ is not essential.

7.3 Diagrammatic reasoning on probabilistic Mealy machines

We extend the notion of behavioural equivalence to probabilistic Mealy machines. Let $\mathsf{M}, \mathsf{N} \colon \mathsf{X} \multimap \mathsf{Y}$ be probabilistic Mealy machines. We write $\mathsf{M} \preceq \mathsf{N}$ when there is a measurable function $f \colon S_{\mathsf{M}} \to S_{\mathsf{N}}$ such that $f(\mathrm{init}_{\mathsf{M}}) = \mathrm{init}_{\mathsf{N}}$ and the following diagram commutes

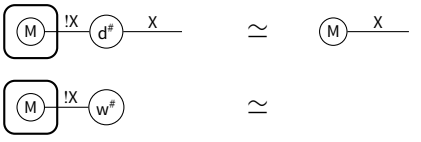
$$\begin{array}{ccc} (X^+ + Y^-) \times S_{\mathsf{M}} & \rightsquigarrow^{(X^+ + Y^-) \otimes f^\#} & (X^+ + Y^-) \times S_{\mathsf{N}} \\ \downarrow \tau_{\mathsf{M}} & & \downarrow \tau_{\mathsf{N}} \\ (X^- + Y^+) \times S_{\mathsf{M}} & \rightsquigarrow^{(X^- + Y^+) \otimes f^\#} & (X^- + Y^+) \times S_{\mathsf{N}}. \end{array}$$

We define an equivalence relation $\simeq$ to be the symmetric transitive closure of $\preceq$ and say that probabilistic Mealy machines $\mathsf{M}, \mathsf{N} \colon \mathsf{X} \nrightarrow \mathsf{Y}$ are *behaviourally equivalent* when we have $\mathsf{M} \simeq \mathsf{N}$.

Behavioural equivalences given in Section 5.3 are valid for probabilistic Mealy machines. This is the reason why we adopt the same diagrammatic presentation.

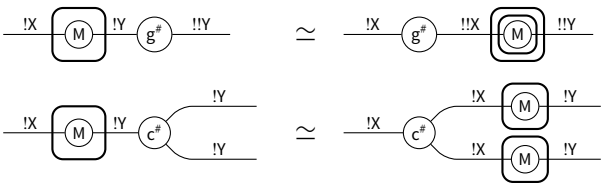
- If two probabilistic Mealy machines have the same diagrammatic presentation modulo some rearrangement of wires and nodes, then they are behaviourally equivalent.
- For any probabilistic Mealy machine $\mathsf{M} \colon \mathsf{I} \nrightarrow \mathsf{X}$, the following behavioural equivalences hold

$$\mathsf{d}^\#_{\mathsf{X}} \circ !_0 \mathsf{M} \simeq \mathsf{M} \qquad \mathsf{w}^\#_{!\mathsf{X}} \circ !_0 \mathsf{M} \simeq \mathsf{id}_{\mathsf{I}}$$



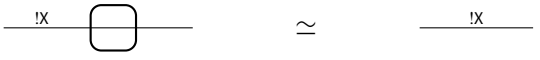
- For any probabilistic Mealy machine $\mathsf{M} \colon \mathsf{X} \nrightarrow \mathsf{Y}$, the following behavioural equivalences hold

$$\mathsf{g}^\#_{\mathsf{Y}} \circ !\mathsf{M} \simeq !!\mathsf{M} \circ \mathsf{g}^\#_{\mathsf{X}} \qquad \mathsf{c}^\#_{\mathsf{Y}} \circ !\mathsf{M} \simeq (!\mathsf{M} \otimes !\mathsf{M}) \circ \mathsf{c}^\#_{X}$$



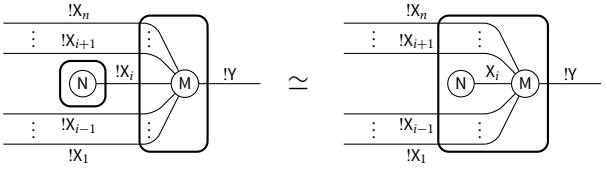
- We can always remove a thick box surrounding a single wire:

$$!\mathsf{id}_{\mathsf{X}} \simeq \mathsf{id}_{!\mathsf{X}}$$



- For any probabilistic Mealy machine $\mathsf{M} \colon \bigotimes_{1 \le i \le n} \mathsf{X}_i \nrightarrow \mathsf{Y}$, and for any probabilistic Mealy machine $\mathsf{N} \colon \mathsf{I} \nrightarrow \mathsf{X}_i$, we have

$$!_n \mathsf{M} \circ (\mathsf{id}^\#_{!\mathsf{X}_1} \otimes \cdots \otimes \mathsf{id}^\#_{!\mathsf{X}_{i-1}} \otimes !_0 \mathsf{N} \otimes \mathsf{id}^\#_{!\mathsf{X}_{i+1}} \otimes \cdots \otimes \mathsf{id}^\#_{!\mathsf{X}_n}) \simeq !_{n-1}(\mathsf{M} \circ (\mathsf{id}^\#_{\mathsf{X}_1} \otimes \cdots \otimes \mathsf{id}^\#_{\mathsf{X}_{i-1}} \otimes \mathsf{N} \otimes \mathsf{id}^\#_{\mathsf{X}_{i+1}} \otimes \cdots \otimes \mathsf{id}^\#_{\mathsf{X}_n})).$$



- For all probabilistic Mealy machines $\mathsf{M}, \mathsf{N} \colon \mathsf{I} \nrightarrow \mathsf{X}$, we have $\mathsf{cd}^\#_{\mathsf{X}} \circ (\mathsf{r}^\#_0 \otimes \mathsf{M} \otimes \mathsf{N}) \simeq \mathsf{M}$ and $\mathsf{cd}^\#_{\mathsf{X}} \circ (\mathsf{r}^\#_a \otimes \mathsf{M} \otimes \mathsf{N}) \simeq \mathsf{N}$ for all real numbers $a \neq 0$.

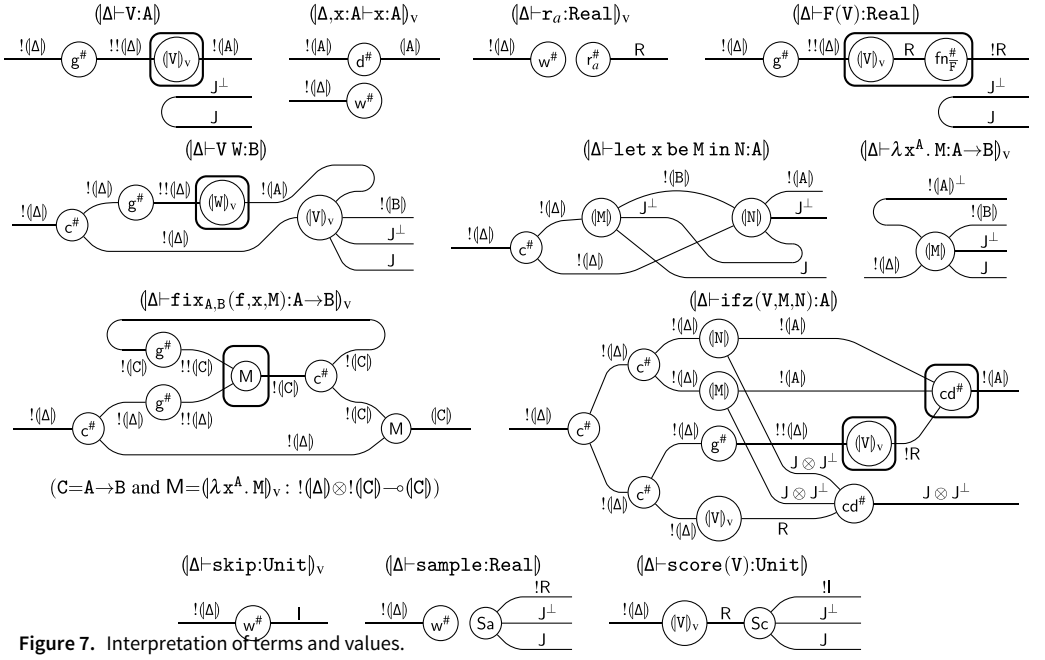


Figure 7. Interpretation of terms and values.

- For all probabilistic Mealy machines $\mathsf{M} \simeq \mathsf{M}' \colon \mathsf{X} \multimap \mathsf{Y}$ and $\mathsf{N} \simeq \mathsf{N}' \colon \mathsf{Y} \multimap \mathsf{Z}$, we have $\mathsf{N} \circ \mathsf{M} \simeq \mathsf{N}' \circ \mathsf{M}'$.
- For all probabilistic Mealy machines $\mathsf{M} \simeq \mathsf{M}' \colon \mathsf{X} \multimap \mathsf{Y}$ and $\mathsf{N} \simeq \mathsf{N}' \colon \mathsf{Z} \multimap \mathsf{W}$, we have $\mathsf{N} \otimes \mathsf{M} \simeq \mathsf{N}' \otimes \mathsf{M}'$.
- For all probabilistic Mealy machines $\mathsf{M} \simeq \mathsf{M}' \colon \bigotimes_{1 \le i \le n} \mathsf{X}_i \multimap \mathsf{Y}$, we have $!_n\mathsf{M} \simeq !_n\mathsf{M}'$.

8. Probabilistic Mealy Machine Semantics and Adequacy Theorem

8.1 Probabilistic Mealy machine semantics

We interpret a type $\mathtt{A}$ as the **Int**-object $(\!|\mathtt{A}|\!)$ given by:

$$(\!|\mathtt{Unit}|\!) = \mathsf{I}, \qquad (\!|\mathtt{Real}|\!) = \mathsf{R}, \qquad (\!|\mathtt{A} \to \mathtt{B}|\!) = \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes\, !(\!|\mathtt{B}|\!) \otimes\, !(\!|\mathtt{A}|\!)^{\perp},$$

and we interpret terms $\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n \vdash \mathtt{M} : \mathtt{B}$ and values $\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n \vdash \mathtt{V} : \mathtt{B}$ by probabilistic Mealy machines:

$$(\!|\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n \vdash \mathtt{M} : \mathtt{B}|\!) \colon !(\!|\mathtt{A}_1|\!) \otimes \cdots \otimes\, !(\!|\mathtt{A}_n|\!) \multimap \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes\, !(\!|\mathtt{B}|\!),$$
$$(\!|\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n \vdash \mathtt{V} : \mathtt{B}|\!)_{\mathrm{v}} \colon !(\!|\mathtt{A}_1|\!) \otimes \cdots \otimes\, !(\!|\mathtt{A}_n|\!) \multimap (\!|\mathtt{B}|\!)$$

that are inductively defined by diagrams in Figure 7. Here, as we did in Figure 5, we suppose that Δ is of the form $(\mathtt{x} : \mathtt{A})$ for some variable $\mathtt{x}$ and some type $\mathtt{A}$.

8.2 Adequacy for distribution-based operational semantics

Let $\mathsf{M} \colon \mathsf{I} \multimap \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes\, !\mathsf{R}$ be a probabilistic Mealy machine. We define s-finite kernels $t_0^{\mathsf{M}} \colon \mathbf{1} \rightsquigarrow S_{\mathsf{M}}$ and $t_1^{\mathsf{M}} \colon S_{\mathsf{M}} \rightsquigarrow \mathbb{R}$ by:

$$t_0^{\mathsf{M}}(*, A) = \tau_{\mathsf{M}}(((\circ, (\bullet, *)), \mathrm{init}_{\mathsf{M}}), \{((\circ, (\bullet, *)), s) \mid s \in A\}),$$
$$t_1^{\mathsf{M}}(s, A) = \tau_{\mathsf{M}}(((\circ, (\circ, (0, ?))), s), \{((\circ, (\circ, (0, a))), s') \mid a \in A \text{ and } s' \in S_{\mathsf{M}}\}).$$

Then we define a measure $\mathsf{obs}_{\mathsf{M}}$ on $\mathbb{R}$ to be $t_1^{\mathsf{M}} \circ t_0^{\mathsf{M}}(*, -)$. Intuitively, for a measurable set $A \in \Sigma_{\mathbb{R}}$, the value $\mathsf{obs}_{\mathsf{M}}(A)$ is the probability of getting $a \in A$ as the result of the following process:

Namely,

(1) we first input $*$ to the J-wire;
(2) after M outputs $*$ to the J-wire, we input $(0, ?)$ to the !R-wire. Outputs $(0, a)$ for some $a \in \mathbb{R}$ from the !R-wire are regarded as outputs of this whole process.

Abusing notation, for a closed term $\mathtt{M} : \mathtt{Real}$, we write $\mathsf{obs}_{\mathtt{M}}$ for $\mathsf{obs}_{(\!|\mathtt{M}|\!)}$.

The intuition behind the definition of $\mathsf{obs}_{\mathsf{M}}$ is almost the same as that of $\mathsf{weight}_{\mathsf{M}}$ and $\mathsf{value}_{\mathsf{M}}$ in Section 6.2. First, in the run (1), the token $*$ traverses M and fires sampling and scoring in an appropriate order. Information provided by sampling and scoring is internally collected by the s-finite kernel t_0^{M}. Then the run (2) gives the execution result of M associated with the sampling results obtained in the first stage. Finally, the composition $\mathsf{obs}_{\mathsf{M}} = t_1^{\mathsf{M}} \circ t_0^{\mathsf{M}}(*, -)$ integrates sampling and scoring results in the first stage with associated execution results in the second stage.

Theorem 25. (Adequacy). *For any closed term* $\vdash \mathtt{M} : \mathtt{Real}$, *if* $\mathtt{M} \Rightarrow_{\infty} \mu$, *then* $\mathsf{obs}_{\mathtt{M}} = \mu$.

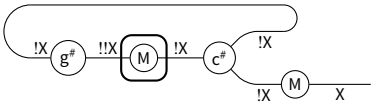
Proof. Let μ be the measure such that $\mathtt{M} \Rightarrow_{\infty} \mu$. It follows from Proposition 30 that $\mu \leq \mathsf{obs}_{\mathtt{M}}$. This is because μ is the sup of the family of measures $\{\mu_n\}_{n \in \mathbb{N}}$ given by $\mathtt{M} \Rightarrow_n \mu_n$. It remains to check $\mathsf{obs}_{\mathtt{M}} \leq \mu$. It follows from Lemma 32 that $(\mathtt{M}, [\![\mathtt{M}]\!]) \in \overline{P}_{\mathtt{Real}}$. Let E be the Mealy machine given in the proof of Theorem 6. Because $([-], \mathsf{E})$ is an element of $P^{\top}_{\mathtt{Real}}$, we see that $(\mathtt{M}, \mathsf{E} \star [\![\mathtt{M}]\!]) = (\mathtt{M}, [\![\mathtt{M}]\!])$ is an element of Ω_{d}. By the definition of Ω_{d}, we obtain the implication from right to left. □

In Section 8.3, we prepare two propositions that are used in the proof of Proposition 30 and Lemma 32, which are proved in Section 8.4.

8.3 On interpretation of the fixed point operator

Let $\mathsf{M} : !\mathsf{X} \nrightarrow \mathsf{X}$ be a probabilistic Mealy machine. In this section, we show that a probabilistic Mealy machine $\mathsf{M}^{\dagger} : \mathsf{I} \to \mathsf{X}$ given by:

$$\mathsf{M}^{\dagger} = (\mathsf{M} \otimes \mathsf{cut}^{\#}_{!\mathsf{X} \perp}) \circ ((\mathsf{c}^{\#}_{\mathsf{X}} \circ !\mathsf{M} \circ \mathsf{g}^{\#}_{\mathsf{X}}) \otimes \mathsf{id}^{\#}_{!\mathsf{X}}) \circ \mathsf{ax}^{\#}_{!\mathsf{X}} =$$



is a 'least' fixed point of M. The argument in this section is essentially equivalent to the argument in Section 6.3, and we only give definitions and statements.

Parametrised Modal Operator and Parametrised Loop Operator. For a subset $\alpha \subseteq \mathbb{N}$, and for a Mealy machine $\mathsf{M} : \mathsf{X} \multimap \mathsf{Y}$, we define $!_{\alpha}\mathsf{M} : !\mathsf{X} \multimap !\mathsf{Y}$ to be a Mealy machine consisting of

$$S_{!_{\alpha}\mathsf{M}} = S_{!\mathsf{M}} = S_{\mathsf{M}}^{\mathbb{N}}, \qquad \mathrm{init}_{!_{\alpha}\mathsf{M}} = \mathrm{init}_{!\mathsf{M}}$$

and

$$\tau_{!_\alpha \mathsf{M}} = \begin{array}{c} (\mathbb{N} \times X^+ + \mathbb{N} \times Y^-) \times S_{\mathsf{M}}^{\mathbb{N}} \\ \wr g^\# \\ \mathbb{N} \times (X^+ + Y^-) \times S_{\mathsf{M}} \times S_{\mathsf{M}}^{\mathbb{N}} \\ \wr \bigoplus_{n \in \mathbb{N}} k_n \otimes \mathrm{id}^\#_{S_{\mathsf{M}}^{\mathbb{N}}} \\ \mathbb{N} \times (X^- + Y^+) \times S_{\mathsf{M}} \times S_{\mathsf{M}}^{\mathbb{N}} \\ \wr (h^{-1})^\# \\ (\mathbb{N} \times X^- + \mathbb{N} \times Y^+) \times S_{\mathsf{M}}^{\mathbb{N}} \end{array}$$

where $g^\#$ and $(h^{-1})^\#$ are measurable functions given in Section 7.2.3, and $k_n \colon (X^+ + Y^-) \times S_{\mathsf{M}} \rightsquigarrow (X^- + Y^+) \times S_{\mathsf{M}}$ is an s-finite kernel given by:

$$k_n = \begin{cases} \tau_{\mathsf{M}}, & \text{if } n \in \alpha, \\ \emptyset, & \text{if } n \notin \alpha. \end{cases}$$

Then for a probabilistic Mealy machine $\mathsf{M} \colon !\mathsf{X} \nrightarrow \mathsf{X}$, we define a probabilistic Mealy machine $\mathsf{M}^{\dagger,\alpha} \colon \mathsf{I} \nrightarrow !\mathsf{X}$ by:

$$\mathsf{M}^{\dagger,\alpha} = (\mathsf{M} \otimes \mathsf{cut}^\#_{!\mathsf{X}\perp}) \circ ((\mathsf{c}^\#_{\mathsf{X}} \circ !_\alpha \mathsf{M} \circ \mathsf{g}^\#_{\mathsf{X}}) \otimes \mathsf{id}^\#_{!\mathsf{X}}) \circ \mathsf{ax}^\#_{!\mathsf{X}}.$$

For a probabilistic Mealy machine $\mathsf{M} \colon !\mathsf{X} \nrightarrow \mathsf{X}$, we inductively define $\mathsf{iter}_n(\mathsf{M}) \colon \mathsf{I} \nrightarrow \mathsf{X}$ by:

$$\mathsf{iter}_0(\mathsf{M}) = \perp^\#_{\mathsf{X}}, \qquad \mathsf{iter}_{n+1}(\mathsf{M}) = \mathsf{M} \circ !_0(\mathsf{iter}_n(\mathsf{M})).$$

Proposition 26. *For all $n \in \mathbb{N}$, we have*

$$\mathsf{M}^{\dagger,\alpha_n} \simeq \mathsf{iter}_{n+1}(\mathsf{M}), \qquad \tau_{\mathsf{M}^{\dagger,\alpha_0}} \le \tau_{\mathsf{M}^{\dagger,\alpha_1}} \le \tau_{\mathsf{M}^{\dagger,\alpha_2}} \le \cdots, \qquad \tau_{\mathsf{M}^\dagger} = \bigvee_{n \ge 0} \tau_{\mathsf{M}^{\dagger,\alpha_n}}$$

where α_n are subsets of $\mathbb{N}$ given in Section 6.3.

Proposition 27. *For any Mealy machine $\mathsf{M} \colon !\mathsf{X} \nrightarrow \mathsf{X}$,*

$$\mathsf{M} \circ !_0 \mathsf{M}^\dagger \simeq \mathsf{M}^\dagger.$$

8.4 Proof of Theorem 25

Lemma 28. (Substitution Lemma). *Let $\mathtt{M}$ be a term $\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n \vdash \mathtt{M} : \mathtt{B}$, and let $\mathtt{V}$ be a value $\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n \vdash \mathtt{V} : \mathtt{B}$. For any closed value $\vdash \mathtt{U} : \mathtt{A}_i$,*

$$(\!|\mathtt{M}|\!) \circ (\mathsf{id}^\#_{!(\!|\mathtt{A}_1|\!)} \otimes \cdots \otimes \mathsf{id}^\#_{!(\!|\mathtt{A}_{i-1}|\!)} \otimes !_0 (\!|\mathtt{U}|\!)_{\mathrm{v}} \otimes \mathsf{id}^\#_{!(\!|\mathtt{A}_{i+1}|\!)} \otimes \cdots \otimes \mathsf{id}^\#_{!(\!|\mathtt{A}_n|\!)}) \simeq (\!|\mathtt{M}\{\mathtt{U}/\mathtt{x}_i\}|\!),$$

$$(\!|\mathtt{V}|\!)_{\mathrm{v}} \circ (\mathsf{id}^\#_{!(\!|\mathtt{A}_1|\!)} \otimes \cdots \otimes \mathsf{id}^\#_{!(\!|\mathtt{A}_{i-1}|\!)} \otimes !_0 (\!|\mathtt{U}|\!)_{\mathrm{v}} \otimes \mathsf{id}^\#_{!(\!|\mathtt{A}_{i+1}|\!)} \otimes \cdots \otimes \mathsf{id}^\#_{!(\!|\mathtt{A}_n|\!)}) \simeq (\!|\mathtt{V}\{\mathtt{U}/\mathtt{x}_i\}|\!)_{\mathrm{v}}.$$

Proof. By induction on $\mathtt{M}$ and $\mathtt{V}$. Induction steps can be checked by diagrammatic reasoning. □

Lemma 29. *For probabilistic Mealy machines $\mathsf{M} \simeq \mathsf{N} \colon \mathsf{X} \nrightarrow \mathsf{Y}$, we have $\mathsf{obs}_{\mathsf{M}} = \mathsf{obs}_{\mathsf{N}}$.*

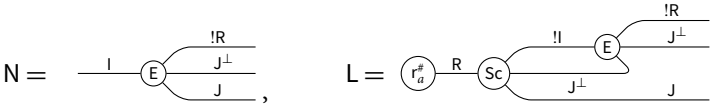
Proof. By the definition of behavioural equivalence, if $\mathsf{M} \preceq \mathsf{N}$, then we have $\mathsf{obs}_{\mathsf{M}} = \mathsf{obs}_{\mathsf{N}}$. Because $\simeq$ is the symmetric transitive closure of $\preceq$, we obtain the statement. □

We first prove soundness.

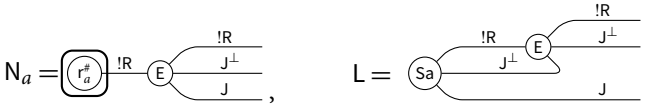
Proposition 30. (Soundness). *For any closed term $\mathtt{M} : \mathtt{Real}$ and for any $n \in \mathbb{N}$, if $\mathtt{M} \Rightarrow_n \mu$ for a measure μ on $\mathbb{R}$, then $\mu \preceq \mathsf{obs}_{\mathtt{M}}$.*

Proof. By induction on n. (Base case) Easy. (Induction step) By case analysis.

- If $\mathtt{M} = \mathtt{E}[\mathtt{N}] \Rightarrow_{n+1} \mu$ and $\mathtt{N} \xrightarrow{\text{red}} \mathtt{L}$ for some term $\mathtt{L}$, then we can check $(\!|\mathtt{E}[\mathtt{N}]|\!) \simeq (\!|\mathtt{E}[\mathtt{L}]|\!)$ by using Lemma 28, Proposition 27 and diagrammatic reasoning. Because $\mathtt{E}[\mathtt{L}] \Rightarrow_n \mu$, it follows from induction hypothesis and Lemma 29 that $\mu \leq \mathsf{obs}_{\mathtt{E}[\mathtt{N}]}$.
- If $\mathtt{M} = \mathtt{r}_a$, then $\mathsf{obs}_{\mathtt{M}} = \delta_a$.
- If $\mathtt{M} = \mathtt{E}[\mathtt{score}(\mathtt{r}_a)]$ and $\mathtt{E}[\mathtt{skip}] \Rightarrow_n \mu$, then by induction hypothesis, we have $\mu \leq \mathsf{obs}_{\mathtt{E}[\mathtt{skip}]}$. By induction on $\mathtt{E}$ with some diagrammatic reasoning, we can show that there is a probabilistic Mealy machine $\mathtt{E} \colon \mathsf{I} \nrightarrow \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !\mathsf{R}$ such that $(\!|\mathtt{E}[\mathtt{skip}]|\!)$ and $(\!|\mathtt{M}|\!)$ are behaviourally equivalent to Mealy machines:



  respectively. (Note that !I is equal to I.) Let us identify S_{L} and S_{N} in the obvious way. Then $t_0^{\mathsf{L}}(*, A)$ is equal to $|a|\, t_0^{\mathsf{N}}(*, A)$, and t_1^{L} is equal to t_1^{N}. Hence, $\mathsf{obs}_{\mathtt{M}} = \mathsf{obs}_{\mathsf{L}} = |a|\, \mathsf{obs}_{\mathsf{N}} = |a|\, \mathsf{obs}_{\mathtt{E}[\mathtt{skip}]} \geq |a|\, \mu$.
- If $\mathtt{M} = \mathtt{E}[\mathtt{sample}]$ and $\mathtt{E}[\mathtt{r}_a] \Rightarrow_n k(a, -)$ for some finite kernel k, then by induction on $\mathtt{E}$, we can show that there is a probabilistic Mealy machine $\mathtt{E} \colon !\mathsf{R} \nrightarrow \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !\mathsf{R}$ such that $(\!|\mathtt{E}[\mathtt{r}_a]|\!)$ and $(\!|\mathtt{M}|\!)$ are behaviourally equivalent to Mealy machines:



  respectively. There is an s-finite kernel $h \colon \mathbb{R}_{[0,1]} \rightsquigarrow S_{\mathsf{E}} \times S_{\mathsf{Sa}}$ such that

$$t_0^{\mathsf{L}} = 1 \xrightsquigarrow{(*, A) \mapsto \mu_{\mathrm{Borel}(A)}} \mathbb{R}_{[0,1]} \xrightsquigarrow{h} S_{\mathsf{E}} \times S_{\mathsf{Sa}}$$

  and $h(a, A) = t_0^{\mathsf{N}_a}(*, \{s \in S_{\mathsf{E}} \mid (s, a) \in A\})$ for all $a \in \mathbb{R}$ and $A \in \Sigma_{S_{\mathsf{E}} \times S_{\mathsf{Sa}}}$. (Here, we identified S_{N_a} with S_{E} and identified S_{L} with $S_{\mathsf{E}} \times S_{\mathsf{Sa}}$ in the obvious way.) We also have

$$t_1^{\mathsf{L}}((s, a), A) = t_1^{\mathsf{N}_a}(s, A)$$

  for all $s \in S_{\mathsf{E}}$, $a \in \mathbb{R}_{[0,1]}$ and $A \in \Sigma_{\mathbb{R}}$. Hence, we obtain

$$\begin{aligned} \mathsf{obs}_{\mathtt{M}}(A) = \mathsf{obs}_{\mathsf{L}}(A) = (t_1^{\mathsf{L}} \circ t_0^{\mathsf{L}})(*, A) &= \int_{\mathbb{R}_{[0,1]}} (t_1^{\mathsf{L}} \circ h)(a, A)\, \mathrm{d}a \\ = \int_{\mathbb{R}_{[0,1]}} (t_1^{\mathsf{N}_a} \circ t_0^{\mathsf{N}_a})(*, A)\, \mathrm{d}a = \int_{\mathbb{R}_{[0,1]}} \mathsf{obs}_{\mathsf{N}_a}(*, A)\, \mathrm{d}a &= \int_{\mathbb{R}_{[0,1]}} \mathsf{obs}_{\mathtt{E}[\mathtt{r}_a]}(*, A)\, \mathrm{d}a. \end{aligned}$$

  By induction hypothesis, $\mathsf{obs}_{\mathtt{M}}(A) \geq \int_{\mathbb{R}_{[0,1]}} k(a, A)\, \mathrm{d}a$. □

We prove adequacy by means of logical relations. We define a binary relation Ω_{d} between closed terms of type `Real` and probabilistic Mealy machines from I to $\mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !\mathsf{R}$ by:

$$(\mathtt{M}, \mathsf{M}) \in \Omega_{\mathrm{d}} \iff \text{if } \mathtt{M} \Rightarrow_{\infty} \mu \text{ then } \mathsf{obs}_{\mathsf{M}} \leq \mu.$$

We then inductively define binary relations:

$$P_{\mathtt{A}} \subseteq \{\text{closed values of type } \mathtt{A}\} \times \{\text{probabilistic Mealy machines from } \mathsf{I} \text{ to } (\!|\mathtt{A}|\!)\}$$

$$P_{\mathtt{A}}^{\top} \subseteq \{\text{evaluation contexts } \mathtt{x} : \mathtt{A} \vdash \mathtt{E}[\mathtt{x}] : \mathtt{Real}\} \times \{\text{probabilistic Mealy machines from } !(\!|\mathtt{A}|\!) \text{ to } \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !\mathsf{R}\}$$

$$\overline{P}_{\mathtt{A}} \subseteq \{\text{closed terms of type } \mathtt{A}\} \times \{\text{probabilistic Mealy machines from } \mathsf{I} \text{ to } \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !(\!|\mathtt{A}|\!)\}$$

by:

$$P_{\mathtt{Real}} = \{(\mathtt{r}_a, \mathsf{r}_a^{\#}) \mid a \in \mathbb{R}\},$$

$$P_{\mathtt{Unit}} = \{(\mathtt{skip}, \mathsf{id}_{\mathsf{I}}^{\#})\},$$

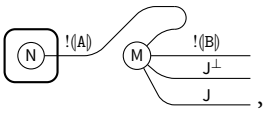
$$P_{\mathtt{A}\to\mathtt{B}} = \{(\mathtt{V}, \mathsf{M}) \mid \forall (\mathtt{W}, \mathsf{N}) \in P_{\mathtt{A}},\ (\mathtt{V}\,\mathtt{W}, \mathsf{M} \bullet \mathsf{N}) \in \overline{P}_{\mathtt{B}}\}$$

and

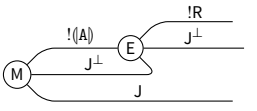
$$P_{\mathtt{A}}^{\top} = \{(\mathtt{E}[-], \mathsf{E}) \mid \forall (\mathtt{V}, \mathsf{M}) \in P_{\mathtt{A}},\ (\mathtt{E}[\mathtt{V}], \mathsf{E} \circ !_0 \mathsf{M}) \in \Omega_{\mathrm{d}}\},$$

$$\overline{P}_{\mathtt{A}} = \{(\mathtt{M}, \mathsf{M}) \mid \forall (\mathtt{E}[-], \mathsf{E}) \in P_{\mathtt{A}}^{\top},\ (\mathtt{E}[\mathtt{M}], \mathsf{E} \star \mathsf{M}) \in \Omega_{\mathrm{d}}\}$$

where Mealy machines $\mathsf{M} \bullet \mathsf{N} \colon \mathsf{I} \multimap \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !(\!|\mathtt{B}|\!)$ and $\mathsf{E} \star \mathsf{M} \colon \mathsf{I} \multimap \mathsf{J} \otimes \mathsf{J}^{\perp} \otimes !\mathsf{R}$ are given by:



,



respectively.

Lemma 31. *Let* $\mathtt{A}$ *be a type.*

(1) If $(\mathtt{V}, \mathsf{M}) \in P_{\mathtt{A}}$, *then* $(\mathtt{V}, \mathsf{ax}_{\mathsf{J}}^{\#} \otimes !_0 \mathsf{M}) \in \overline{P}_{\mathtt{A}}$.

(2) If $\mathtt{N} \xrightarrow{\mathrm{red}} \mathtt{M}$, *then* $(\mathtt{M}, \mathsf{M}) \in \overline{P}_{\mathtt{A}}$ *if and only if* $(\mathtt{N}, \mathsf{M}) \in \overline{P}_{\mathtt{A}}$.

(3) If $(\mathtt{M}, \mathsf{M}) \in \overline{P}_{\mathtt{A}}$ *and* $\mathsf{M} \simeq \mathsf{N}$, *then* $(\mathtt{M}, \mathsf{N}) \in \overline{P}_{\mathtt{A}}$.

(4) For any closed value $\mathtt{V} : \mathtt{A} \to \mathtt{B}$, $(\mathtt{V}, \bot_{(\!|\mathtt{A}\to\mathtt{B}|\!)}^{\#}) \in P_{\mathtt{A}\to\mathtt{B}}$.

(5) If $(\mathtt{M}, \mathsf{M}_i) \in \overline{P}_{\mathtt{A}}$ *for probabilistic Mealy machines* $(\mathsf{M}_i)_{i\in\mathbb{N}}$ *such that* $S_{\mathsf{M}_1} = S_{\mathsf{M}_2} = \cdots$ *and* $\mathrm{init}_{\mathsf{M}_1} = \mathrm{init}_{\mathsf{M}_2} = \cdots$ *and* $\tau_{\mathsf{M}_1} \le \tau_{\mathsf{M}_2} \le \cdots$, *then* $(\mathtt{M}, \mathsf{N}) \in \overline{P}_{\mathtt{A}}$ *where the probabilistic Mealy machine* N *is given by* $S_{\mathsf{N}} = S_{\mathsf{M}_1}$, $\mathrm{init}_{\mathsf{N}} = \mathrm{init}_{\mathsf{M}_1}$ *and* $\tau_{\mathsf{N}} = \bigvee_{n\in\mathbb{N}} \tau_{\mathsf{M}_n}$. *Here, the order between s-finite kernels is given by the pointwise manner.*

Proof. Similar to the proof of Lemma 19. □

Lemma 32. (Basic Lemma). *Let* $\Delta = (\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n)$ *be a type environment.*

- *For any term* $\Delta \vdash \mathtt{M} : \mathtt{A}$ *and for any* $(\mathtt{V}_i, \mathsf{N}_i) \in P_{\mathtt{A}_i}$ *for* $i = 1, 2, \ldots, n$, *we have*
$$\left(\mathtt{M}\{\mathtt{V}_1/\mathtt{x}_1, \ldots, \mathtt{V}_n/\mathtt{x}_n\}, (\!|\mathtt{M}|\!) \circ (!_0\mathsf{N}_1 \otimes \cdots \otimes !_0\mathsf{N}_n)\right) \in \overline{P}_{\mathtt{A}}.$$
- *For any value* $\Delta \vdash \mathtt{V} : \mathtt{A}$ *and for any* $(\mathtt{V}_i, \mathsf{N}_i) \in P_{\mathtt{A}_i}$ *for* $i = 1, 2, \ldots, n$, *we have*
$$\left(\mathtt{V}\{\mathtt{V}_1/\mathtt{x}_1, \ldots, \mathtt{V}_n/\mathtt{x}_n\}, (\!|\mathtt{V}|\!)_{\mathrm{v}} \circ (!_0\mathsf{N}_1 \otimes \cdots \otimes !_0\mathsf{N}_n)\right) \in P_{\mathtt{A}}.$$

Proof. By induction on $\mathtt{M}$ and $\mathtt{V}$. Most cases follow from Lemma 31 with some diagrammatic reasoning. Here, we check the statement for $\mathtt{M} = \mathtt{score}(\mathtt{V})$, $\mathtt{M} = \mathtt{sample}$ and $\mathtt{V} = \mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f}, \mathtt{x}, \mathtt{N})$.

- When $\mathtt{M} = \mathtt{score}(\mathtt{V})$, it is enough to show that $(\mathtt{score}(\mathtt{r}_a), \mathsf{Sc} \circ \mathsf{r}_a^{\#})$ is an element of $\overline{P_{\mathtt{Unit}}}$ for any $a \in \mathbb{R}$. To see this, we show that $\mathtt{E}[\mathtt{skip}] \Rightarrow_\infty \mu$ implies $\mathsf{obs}_{\mathtt{E}\star(\mathsf{Sc}\circ\mathsf{r}_a^{\#})} \leq |a|\,\mu$ for each $(\mathtt{E}, \mathtt{E})$ in $P^{\top}_{\mathtt{Unit}}$. Below, we identify $S_{\mathtt{E}\star(\mathsf{Sc}\circ\mathsf{r}_a^{\#})}$ with $S_{\mathtt{E}\circ!_0\mathsf{id}_!^{\#}}$ in the obvious way. Because $t_0^{\mathtt{E}\star(\mathsf{Sc}\circ\mathsf{r}_a^{\#})}(*, A) = |a|\, t_0^{\mathtt{E}\circ!_0\mathsf{id}_!^{\#}}(*, A)$ and $t_1^{\mathtt{E}\star(\mathsf{Sc}\circ\mathsf{r}_a^{\#})} = t_1^{\mathtt{E}\circ!_0\mathsf{id}_!^{\#}}$, we have $\mathsf{obs}_{\mathtt{E}\star(\mathsf{Sc}\circ\mathsf{r}_a^{\#})} = |a|\,\mathsf{obs}_{\mathtt{E}\circ!_0\mathsf{id}_!^{\#}}$. Hence, by induction hypothesis, $\mathsf{obs}_{\mathtt{E}\star(\mathsf{Sc}\circ\mathsf{r}_a^{\#})} = |a|\,\mathsf{obs}_{\mathtt{E}\circ!_0\mathsf{id}_!^{\#}} \leq |a|\,\mu$.
- When $\mathtt{M} = \mathtt{sample}$, we show that $(\mathtt{sample}, \mathsf{Sa})$ is an element of $\overline{P_{\mathtt{Real}}}$. Let $(\mathtt{E}, \mathtt{E})$ be a pair in $P^{\top}_{\mathtt{Real}}$. Then there is an s-finite kernel $h\colon \mathbb{R}_{[0,1]} \rightsquigarrow S_{\mathtt{E}} \times S_{\mathsf{Sa}}$ such that

$$t_0^{\mathtt{E}\star\mathsf{Sa}} = 1 \overset{(*,A)\mapsto \mu_{\mathrm{Borel}}(A)}{\rightsquigarrow} \mathbb{R}_{[0,1]} \overset{h}{\rightsquigarrow} S_{\mathtt{E}} \times S_{\mathsf{Sa}}\,, \qquad h(a, A) = t_0^{\mathtt{E}\circ!_0\mathsf{r}_a^{\#}}(*, \{s \in S_{\mathtt{E}} \mid (s, a) \in A\})$$

  for all $a \in \mathbb{R}$ and $A \in \Sigma_{S_{\mathtt{E}} \times S_{\mathsf{Sa}}}$. (Here, we identified $S_{\mathtt{E}\circ!_0\mathsf{r}_a^{\#}}$ with $S_{\mathtt{E}}$ and identified $S_{\mathtt{E}\star\mathsf{Sa}}$ with $S_{\mathtt{E}} \times S_{\mathsf{Sa}}$ in the obvious way.) We also have

$$t_1^{\mathtt{E}\star\mathsf{Sa}}((s, a), A) = t_1^{\mathtt{E}\circ!_0\mathsf{r}_a^{\#}}(s, A)$$

  for all $s \in S_{\mathtt{E}}$, $a \in \mathbb{R}_{[0,1]}$ and $A \in \Sigma_{\mathbb{R}}$. Hence,

$$\begin{aligned}\mathsf{obs}_{\mathtt{E}\star\mathsf{Sa}}(A) = (t_1^{\mathtt{E}\star\mathsf{Sa}} \circ t_0^{\mathtt{E}\star\mathsf{Sa}})(*, A) &= \int_{\mathbb{R}_{[0,1]}} (t_1^{\mathtt{E}\star\mathsf{Sa}} \circ h)(a, A)\,\mathrm{d}a \\ &= \int_{\mathbb{R}_{[0,1]}} (t_1^{\mathtt{E}\circ!_0\mathsf{r}_a^{\#}} \circ t_0^{\mathtt{E}\circ!_0\mathsf{r}_a^{\#}})(*, A)\,\mathrm{d}a \\ &= \int_{\mathbb{R}_{[0,1]}} \mathsf{obs}_{\mathtt{E}\circ!_0\mathsf{r}_a^{\#}}(A)\,\mathrm{d}a \leq \int_{\mathbb{R}_{[0,1]}} k(a, A)\,\mathrm{d}a\end{aligned}$$

  where $k\colon \mathbb{R}_{[0,1]} \rightsquigarrow \mathbb{R}$ is the s-finite kernel such that $\mathtt{E}[\mathsf{r}_a^{\#}] \Rightarrow_\infty k(a, -)$.
- When $\mathtt{V} = \mathtt{fix}_{\mathtt{A},\mathtt{B}}(\mathtt{f}, \mathtt{x}, \mathtt{N})$, for simplicity, we suppose that $\mathtt{V}$ is a closed term. By induction hypothesis and (2) in Lemma 31, we can check

$$(\mathtt{V}, (\!|\lambda \mathtt{x}^{\mathtt{A}}.\,\mathtt{N}|\!)_{\mathrm{v}} \circ \overbrace{!_0((\!|\lambda \mathtt{x}^{\mathtt{A}}.\,\mathtt{N}|\!)_{\mathrm{v}} \circ \cdots \circ !_0((\!|\lambda \mathtt{x}^{\mathtt{A}}.\,\mathtt{N}|\!)_{\mathrm{v}}}^{n} \circ !_0 \perp^{\#}_{(\!|\mathtt{A}\to\mathtt{B}|\!)}) \cdots)) \in P_{\mathtt{A}\to\mathtt{B}}$$

  by induction on n. By Lemma 31 and by Proposition 26, we obtain $(\mathtt{V}, (\!|\mathtt{V}|\!)_{\mathrm{v}}) \in P_{\mathtt{A}\to\mathtt{B}}$. □

8.5 Commutativity modulo observational equivalence

Finally, as an application of our GoI semantics, we show commutativity of let-bindings modulo observational equivalence. For type environments Δ and Δ', and for types $\mathtt{A}$ and $\mathtt{B}$, when a context $\mathtt{C}[\,-\,]$ satisfies $\Delta \vdash \mathtt{C}[\mathtt{M}] : \mathtt{A}$ for any term $\Delta' \vdash \mathtt{M} : \mathtt{B}$, we write $\mathtt{C} : (\Delta', \mathtt{B}) \to (\Delta, \mathtt{A})$.

Definition 33. *For terms $\Delta \vdash \mathtt{M}, \mathtt{N} : \mathtt{A}$, we say that $\mathtt{M}$ is* observationally equivalent *to $\mathtt{N}$ when for any measure μ on $\mathbb{R}$ and for any context $\mathtt{C}[\,-\,] : (\Delta, \mathtt{A}) \to (\varnothing, \mathtt{Real})$, we have $\mathtt{C}[\mathtt{M}] \Rightarrow_\infty \mu$ if and only if $\mathtt{C}[\mathtt{N}] \Rightarrow_\infty \mu$. We write $\mathtt{M} \simeq_{\mathrm{obs}} \mathtt{N}$ when $\mathtt{M}$ is observationally equivalent to $\mathtt{N}$.*

Formally, our goal is to prove that for all terms $\Delta \vdash \mathtt{M} : \mathtt{A}$, $\Delta \vdash \mathtt{N} : \mathtt{B}$ and $\Delta, \mathtt{x} : \mathtt{A}, \mathtt{y} : \mathtt{B} \vdash \mathtt{L} : \mathtt{C}$, we have

$$\mathtt{let\ x\ be\ M\ in\ (let\ y\ be\ N\ in\ L)} \simeq_{\mathrm{obs}} \mathtt{let\ y\ be\ N\ in\ (let\ x\ be\ M\ in\ L)}.$$

Below, we first show commutativity of let-bindings in our GoI semantics based on probabilistic Mealy machines, and then, we deduce the above observational equivalence using adequacy.

Lemma 34. *For any term* $\mathtt{x}:\mathtt{A}_1,\dots\mathtt{y}:\mathtt{A}_n\vdash\mathtt{M}:\mathtt{B}$*, there are s-finite kernels:*

$$k\colon (\mathbb{N}\times(\!|\mathtt{A}_1|\!)^+ +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^+ +\mathbf{1})\times S_{(\!|\mathtt{M}|\!)} \rightsquigarrow (\mathbb{N}\times(\!|\mathtt{A}_1|\!)^- +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^- +\mathbf{1})\times S_{(\!|\mathtt{M}|\!)}$$

$$\ell_n\colon (\mathbb{N}\times(\!|\mathtt{A}_1|\!)^+ +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^+ +(\!|\mathtt{B}|\!)^-)\times S_{(\!|\mathtt{M}|\!)} \rightsquigarrow (\mathbb{N}\times(\!|\mathtt{A}_1|\!)^- +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^- +(\!|\mathtt{B}|\!)^+)\times S_{(\!|\mathtt{M}|\!)} \qquad (n\in\mathbb{N})$$

and bijections $r,s\colon\mathbb{N}\to\mathbb{N}$*, which are necessarily measurable isomorphisms, such that the transition kernel* $(\!|\mathtt{M}|\!)$ *is equal to*

$$\begin{aligned}
&((\mathbb{N}\times(\!|\mathtt{A}_1|\!)^+ +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^+)+(\mathbf{1}+\mathbb{N}\times(\!|\mathtt{B}|\!)^-))\times S_{(\!|\mathtt{M}|\!)}\\
&\qquad \S^{((r\times \mathrm{id}_{(\!|\mathtt{A}_1|\!)^+}+\cdots+r\times \mathrm{id}_{(\!|\mathtt{A}_n|\!)^+})+(\mathrm{id}_{\mathbf{1}}+s\times \mathrm{id}_{(\!|\mathtt{B}|\!)^-}))\times \mathrm{id}_{S_{(\!|\mathtt{M}|\!)}})^{\#}}\\
&((\mathbb{N}\times(\!|\mathtt{A}_1|\!)^+ +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^+)+(\mathbf{1}+\mathbb{N}\times(\!|\mathtt{B}|\!)^-))\times S_{(\!|\mathtt{M}|\!)}\\
&\qquad \S^{(g^{-1})^{\#}}\\
&(\mathbb{N}\times(\!|\mathtt{A}_1|\!)^+ +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^+ +\mathbf{1})\times S_{(\!|\mathtt{M}|\!)}+\mathbb{N}\times(\mathbb{N}\times(\!|\mathtt{A}_1|\!)^+ +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^+ +(\!|\mathtt{B}|\!)^-)\times S_{(\!|\mathtt{M}|\!)}\\
&\qquad \S^{k\oplus\bigoplus_{n\in\mathbb{N}}\ell_n}\\
&(\mathbb{N}\times(\!|\mathtt{A}_1|\!)^- +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^- +\mathbf{1})\times S_{(\!|\mathtt{M}|\!)}+\mathbb{N}\times(\mathbb{N}\times(\!|\mathtt{A}_1|\!)^- +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^- +(\!|\mathtt{B}|\!)^+)\times S_{(\!|\mathtt{M}|\!)}\\
&\qquad \S^{h^{\#}}\\
&(\mathbb{N}\times(\!|\mathtt{A}_1|\!)^- +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^-)+(\mathbf{1}+\mathbb{N}\times(\!|\mathtt{B}|\!)^+))\times S_{(\!|\mathtt{M}|\!)}\\
&\qquad \S^{((r^{-1}\times \mathrm{id}_{(\!|\mathtt{A}_1|\!)^-}+\cdots+r^{-1}\times \mathrm{id}_{(\!|\mathtt{A}_n|\!)^-})+(\mathrm{id}_{\mathbf{1}}+s^{-1}\times \mathrm{id}_{(\!|\mathtt{B}|\!)^+}))\times \mathrm{id}_{S_{(\!|\mathtt{M}|\!)}})^{\#}}\\
&((\mathbb{N}\times(\!|\mathtt{A}_1|\!)^- +\cdots+\mathbb{N}\times(\!|\mathtt{A}_n|\!)^-)+(\mathbf{1}+\mathbb{N}\times(\!|\mathtt{B}|\!)^+))\times S_{(\!|\mathtt{M}|\!)}
\end{aligned}$$

where g *is a measurable isomorphism given by:*

$$g(\bullet,((\bullet,i@a),s))=((\bullet,(2i)@a),s),\qquad g(\bullet,((\circ,*),s))=((\circ,(\bullet,*)),s),$$

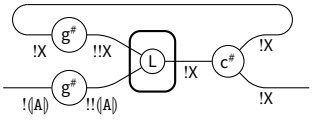
$$g(\circ,(i,(\bullet,j@a),s))=((\bullet,(2\langle i,j\rangle+1)@a),s),\qquad g(\circ,(i,(\circ,b),s))=((\circ,(\circ,(i,b))),s)$$

for $i,j\in\mathbb{N}$*,* $a\in(\!|\mathtt{A}_1|\!)^+ +\cdots+(\!|\mathtt{A}_n|\!)^+$ *and* $b\in(\!|\mathtt{B}|\!)^-$*; h is a measurable isomorphism given in the same way.*

Proof. By induction on $\mathtt{M}$. Here, we only sketch the proof of the statement for $\mathtt{M}=\mathtt{V}\,\mathtt{W}$. The statement for other cases can be derived from induction hypothesis and the definition of $(\!|\mathtt{M}|\!)$. When $\mathtt{M}=\mathtt{V}\,\mathtt{W}$, we need case analysis: (1) $\mathtt{V}=\mathtt{x}$ for some variable $\mathtt{x}$, (2) $\mathtt{V}=\lambda\mathtt{x}^{\mathtt{A}}.\,\mathtt{N}$, (3) $\mathtt{V}=\mathtt{fix}_{\mathtt{C},\mathtt{C}'}(\mathtt{f},\mathtt{x},\mathtt{N})$. For the case (1), the statement follows from some diagrammatic reasoning. For the case (2), the statement follows from induction hypothesis. For the case (3), for simplicity, we suppose that $n=1$ and write $\mathtt{A}$ for $\mathtt{A}_1$. Then the statement for this case follows from induction hypothesis and that for any **Int**-object $\mathtt{X}$ and for any probabilistic Mealy machine $\mathtt{L}\colon\,!(\!|\mathtt{A}|\!)\otimes!\mathtt{X}\multimap\mathtt{X}$, there are s-finite kernels:

$$\ell_n\colon(\mathbb{N}\times(\!|\mathtt{A}|\!)^+ +X^-)\times S_{\mathsf{M}}\rightsquigarrow(\mathbb{N}\times(\!|\mathtt{A}|\!)^- +X^+)\times S_{\mathsf{M}}\qquad(n\in\mathbb{N})$$

and measurable isomorphisms $r\colon\mathbb{N}\to\mathbb{N}\times\mathbb{N}$ and $s\colon\mathbb{N}\to\mathbb{N}$ such that the transition kernel of

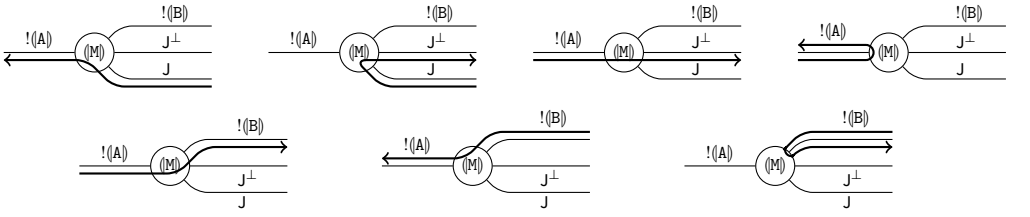


is equal to

$$
\begin{array}{c}
(\mathbb{N} \times (\!|\mathtt{A}|\!)^+ + \mathbb{N} \times X^-) \times S_{\mathsf{L}} \\
\big\downarrow (r \times \mathrm{id}_{(\!|\mathtt{A}|\!)^+} + s \times \mathrm{id}_{X^-})^{\#} \\
(\mathbb{N} \times \mathbb{N} \times (\!|\mathtt{A}|\!)^+ + \mathbb{N} \times X^-) \times S_{\mathsf{L}} \\
\big\downarrow (1) \\
\mathbb{N} \times (\mathbb{N} \times (\!|\mathtt{A}|\!)^+ + X^-) \times S_{\mathsf{L}} \\
\big\downarrow \bigoplus_{n \in \mathbb{N}} \ell_n \\
\mathbb{N} \times (\mathbb{N} \times (\!|\mathtt{A}|\!)^- + X^+) \times S_{\mathsf{L}} \\
\big\downarrow (2) \\
(\mathbb{N} \times \mathbb{N} \times (\!|\mathtt{A}|\!)^- + \mathbb{N} \times X^+) \times S_{\mathsf{L}} \\
\big\downarrow (r^{-1} \times \mathrm{id}_{(\!|\mathtt{A}|\!)^+} + s^{-1} \times \mathrm{id}_{X^-})^{\#} \\
(\mathbb{N} \times (\!|\mathtt{A}|\!)^- + \mathbb{N} \times X^+) \times S_{\mathsf{L}}
\end{array}
$$

where the arrows (1) and (2) are s-finite kernels induced by distributivity $\mathbb{N} \times (Y + Z) \cong \mathbb{N} \times Y + \mathbb{N} \times Z$. This decomposition of the transition kernel can be checked by direct calculation. □

Intuitively, it follows from Lemma 34 that for any term $\mathtt{x} : \mathtt{A} \vdash \mathtt{M} : \mathtt{B}$, the probabilistic Mealy machine $(\!|\mathtt{M}|\!)$ handles tokens along the following thick arrows:



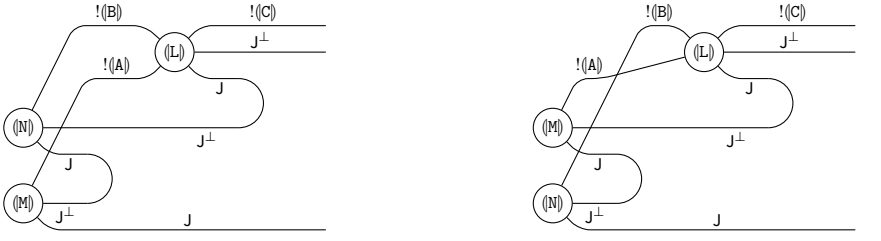
and $(\!|\mathtt{M}|\!)$ never handles tokens as follows:

From this observation, we can show commutativity of let-bindings in the GoI semantics based on probabilistic Mealy machines.

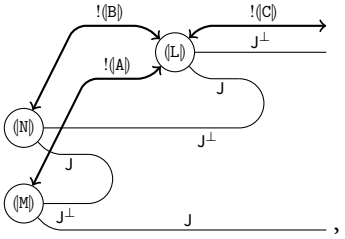
Proposition 35. *For all terms* $\vdash \mathtt{M} : \mathtt{A}$, $\vdash \mathtt{N} : \mathtt{B}$ *and* $\mathtt{x} : \mathtt{A}, \mathtt{y} : \mathtt{B} \vdash \mathtt{L} : \mathtt{C}$, *we have* $(\!|\mathtt{K}|\!) \simeq (\!|\mathtt{K}'|\!)$ *where*

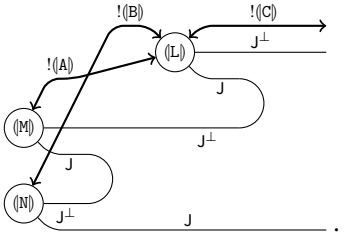
$$\mathtt{K} = \mathtt{let\ x\ be\ M\ in\ let\ y\ be\ N\ in\ L}, \qquad \mathtt{K}' = \mathtt{let\ y\ be\ N\ in\ let\ x\ be\ M\ in\ L}.$$

Proof. We only give an informal proof. Let us describe how the probabilistic Mealy machines $(\!|\mathtt{K}|\!)$ and $(\!|\mathtt{K}'|\!)$ behave. First, $(\!|\mathtt{K}|\!)$ and $(\!|\mathtt{K}'|\!)$ are diagrammatically presented as follows:



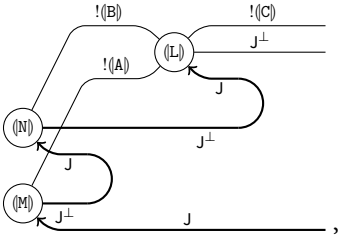
As for input tokens from the $!(\!|\mathtt{C}|\!)$-wire, it follows from Lemma 34 that tokens run back and forth along the following thick wires until they go out from the $!(\!|\mathtt{C}|\!)$-wire:

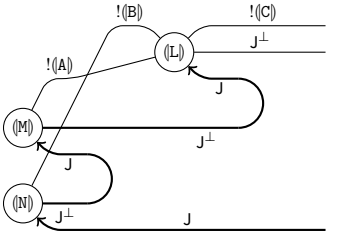


,


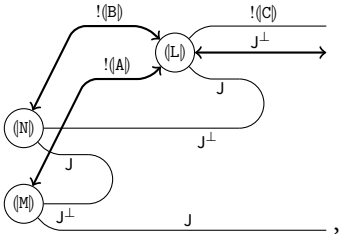
.

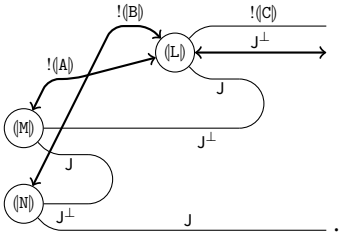
Therefore, there is no behavioural difference between $(\!|\mathtt{K}|\!)$ and $(\!|\mathtt{K}'|\!)$ for any input token from the $!(\!|\mathtt{C}|\!)$-wire. It remains to check that there is no behavioural difference between $(\!|\mathtt{K}|\!)$ and $(\!|\mathtt{K}'|\!)$ for any input token from the $\mathtt{J}$-wire. As for input tokens from the $\mathtt{J}$-wire, it follows from Lemma 34 that tokens first run along the following thick arrows:



,


and after these transitions, tokens run back and forth along the following thick arrows until they go out from the $\mathtt{J}^{\perp}$-wire:



,


.

Hence, the difference between $(\!|\mathtt{K}|\!)$ and $(\!|\mathtt{K}'|\!)$ is the first two transitions for input tokens from the $\mathtt{J}$-wire. There are s-finite kernels $k\colon S_{\mathsf{M}} \rightsquigarrow S_{\mathsf{M}}$ and $h\colon S_{\mathsf{N}} \rightsquigarrow S_{\mathsf{N}}$ such that

$$\mathbf{1} \times S_{\mathsf{M}} \times S_{\mathsf{N}} \overset{\mathrm{id}_{\mathbf{1}}^{\#} \otimes k \otimes \mathrm{id}_{S_{\mathsf{N}}}^{\#}}{\rightsquigarrow} \mathbf{1} \times S_{\mathsf{M}} \times S_{\mathsf{N}} \overset{\mathrm{id}_{\mathbf{1}}^{\#} \otimes \mathrm{id}_{S_{\mathsf{M}}}^{\#} \otimes h}{\rightsquigarrow} \mathbf{1} \times S_{\mathsf{M}} \times S_{\mathsf{N}}$$

describes the first two transitions in $(\!|\mathtt{K}|\!)$, and

$$\mathbf{1} \times S_{\mathsf{M}} \times S_{\mathsf{N}} \overset{\mathrm{id}_{\mathbf{1}}^{\#} \otimes \mathrm{id}_{S_{\mathsf{M}}}^{\#} \otimes h}{\rightsquigarrow} \mathbf{1} \times S_{\mathsf{M}} \times S_{\mathsf{N}} \overset{\mathrm{id}_{\mathbf{1}}^{\#} \otimes k \otimes \mathrm{id}_{S_{\mathsf{N}}}^{\#}}{\rightsquigarrow} \mathbf{1} \times S_{\mathsf{M}} \times S_{\mathsf{N}}$$

describes the first two transitions in $(\!|\mathtt{K}'|\!)$. It follows from commutativity of the tensor product of s-finite kernels that they are the same. □

Lemma 36. *Let* Δ *be* $(\mathtt{x}_1 : \mathtt{A}_1, \ldots, \mathtt{x}_n : \mathtt{A}_n)$*, and let* Δ' *be* $(\mathtt{x}'_1 : \mathtt{A}'_1, \ldots, \mathtt{x}'_m : \mathtt{A}'_m)$*. Let* $\mathtt{B}$ *and* $\mathtt{B}'$ *be types, and let* $\mathtt{C}[\,-\,] : (\Delta, \mathtt{B}) \to (\Delta', \mathtt{B}')$ *be a context. For terms* $\Delta \vdash \mathtt{M} : \mathtt{B}$ *and* $\Delta \vdash \mathtt{N} : \mathtt{B}$*, if for all* $\{(\mathtt{V}_i, \mathtt{L}_i) \in P_{\mathtt{A}_i}\}_{1 \le i \le n}$*, we have*

$$(\mathtt{M}\{\mathtt{V}_1/\mathtt{x}_1, \ldots, \mathtt{V}_n/\mathtt{x}_n\}, (\!|\mathtt{N}|\!) \circ (!_0\mathtt{L}_1 \otimes \cdots \otimes !_0\mathtt{L}_n)) \in \overline{P}_{\mathtt{B}},$$

then for all $\{(\mathtt{V}'_i, \mathtt{L}'_i) \in P_{\mathtt{A}_i}\}_{1 \le i \le m}$, *we have*

$$(\mathtt{C}[\mathtt{M}]\{\mathtt{V}'_1/\mathtt{x}'_1, \ldots, \mathtt{V}'_m/\mathtt{x}'_m\}, (\!|\mathtt{C}[\mathtt{N}]|\!) \circ (!_0\mathtt{L}'_1 \otimes \cdots \otimes !_0\mathtt{L}'_m)) \in \overline{P}_{\mathtt{B}'}.$$

Proof. By induction on $\mathtt{C}[-]$. To check induction steps, we can use Theorem 25 and Proposition 26 and some diagrammatic reasoning. □

Proposition 37. *For all terms* $\Delta \vdash \mathtt{M} : \mathtt{A}_1$, $\Delta \vdash \mathtt{N} : \mathtt{A}_2$ *and* $\Delta, \mathtt{x} : \mathtt{A}_1, \mathtt{y} : \mathtt{A}_2 \vdash \mathtt{L} : \mathtt{B}$, *we have*

$$\mathtt{let\ x\ be\ M\ in\ let\ y\ be\ N\ in\ L} \simeq_{\mathrm{obs}} \mathtt{let\ y\ be\ N\ in\ let\ x\ be\ M\ in\ L}.$$

Proof. For simplicity, we suppose that $\Delta = (\mathtt{z} : \mathtt{D})$. Let $\mathtt{C}[-] : (\Delta, \mathtt{B}) \to ((), \mathtt{Real})$ be a context, and let μ and μ' be the measures such that $\mathtt{C}[\mathtt{let\ x\ be\ M\ in\ let\ y\ be\ N\ in\ L}] \Rightarrow_\infty \mu$ and $\mathtt{C}[\mathtt{let\ y\ be\ N\ in\ let\ x\ be\ M\ in\ L}] \Rightarrow_\infty \mu'$, respectively. We show that $\mu = \mu'$. By Lemma 32, for any $(\mathtt{V}, \mathtt{N}) \in Q_{\mathtt{D}}$:

$$(\mathtt{let\ x\ be\ M\{V/z\}\ in\ let\ y\ be\ N\{V/z\}\ in\ L\{V/z\}}, (\!|\mathtt{let\ x\ be\ M\ in\ let\ y\ be\ N\ in\ L}|\!) \circ !_0\mathtt{N}) \in \overline{P}_{\mathtt{B}}.$$

For any measure ν on $\mathbb{R}$, and for any evaluation context $\mathtt{E}$, it follows from Proposition 37, Theorem 25 and compositionality of our denotational semantics that

$$\mathtt{E}[\mathtt{let\ x\ be\ M\{V/z\}\ in\ let\ y\ be\ N\{V/z\}\ in\ L\{V/z\}}] \Rightarrow_\infty \nu \iff \mathtt{E}[\mathtt{let\ y\ be\ N\{V/z\}\ in\ let\ x\ be\ M\{V/z\}\ in\ L\{V/z\}}] \Rightarrow_\infty \nu.$$

Hence, by the definition of $\overline{P_{\mathtt{B}}}$, we obtain

$$(\mathtt{let\ y\ be\ N\{V/z\}\ in\ let\ x\ be\ M\{V/z\}\ in\ L\{V/z\}}, (\!|\mathtt{let\ x\ be\ M\ in\ let\ y\ be\ N\ in\ L}|\!) \circ !_0\mathtt{N}) \in \overline{P}_{\mathtt{B}}.$$

Then by Lemma 36,

$$(\mathtt{C}[\mathtt{let\ y\ be\ N\ in\ let\ x\ be\ M\ in\ L}], (\!|\mathtt{C}[\mathtt{let\ x\ be\ M\ in\ let\ y\ be\ N\ in\ L}]|\!)) \in \overline{P}_{\mathtt{Real}}.$$

Hence, by Theorem 25, $\mu \le \mu'$. We can similarly check $\mu' \le \mu$. □

9. Conclusion

We introduced a denotational semantics for **PCFSS**, a higher-order functional language with sampling from a uniform continuous distribution and scoring. Following Borgström et al. (2016), we considered two operational semantics, namely a *distribution-based* operational semantics, which associates terms with distributions over real numbers, and a *sampling-based* operational semantics, which associates each term with a weight along every probabilistic branch. Our main results are adequacy theorems for both kinds of operational semantics. From these theorems, it follows that sampling-based operational semantics is essentially equivalent to distribution-based operational semantics. Another consequence of adequacy theorems is the possibility of reasoning for observational equivalence of programs diagrammatically. It follows from the observation in Section 5.3 and the adequacy theorems that diagrammatic equivalence implies observational equivalence. It would be interesting to explore possible connections between our work and other works on diagrammatic reasoning for probabilistic computation, such as Cho and Jacobs (2017); Jacobs et al. (2018).

At this point, our language does not support any normalisation mechanism as a first-class operator. However, capturing inference algorithms such as the Metropolis–Hastings algorithm (Hastings 1970; Metropolis et al. 1953), which is structured around a number of interactions between programs and their environment, seems plausible. Exploring the relationships between 'idealised' normalisation mechanisms and such 'approximating' normalisation mechanisms from the point of view of GoI is an interesting topic for future work.

Acknowledgements. The authors are partially supported by the INRIA/JSPS project 'CRECOGI' and would like to thank Michele Pagani for many fruitful discussions about an earlier version of this work. Ugo Dal Lago is supported by the ERC CoG DIAPASoN (Grant Agreement No. 818616). Naohiko Hoshino is supported by JST ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603).

References

Abramsky, S., Haghverdi, E. and Scott, P. (2002). Geometry of interaction and linear combinatory algebras. *Mathematical. Structures in Compter Science* **12** (5) 625–665.

Abramsky, S., Jagadeesan, R. and Malacaria, P. (2000). Full abstraction for PCF. *Information and Computation* **163** (2) 409–470.

Abramsky, S. and McCusker, G. (1996). Linearity, sharing and state: a fully abstract game semantics for idealized algol with active expressions: Extended abstract. *Electronic Notes in Theoretical Computer Science* **3** 2–14.

Agrawal, M., Kayal, N. and Saxena, N. (2002). PRIMES is in P. *Annals of Mathematics* **2** 781–793.

Billingsley, P. (1986). *Probability and Measure*, second edition. John Wiley and Sons.

Borgström, J., Dal Lago, U., Gordon, A. D. and Szymczak, M. (2016). A lambda-calculus foundation for universal probabilistic programming. In: *Proceedings of ICFP 2016*, 33–46.

Breuvart, F. and Dal Lago, U. (2018). On intersection types and probabilistic lambda calculi. In: *Proceedings of PPDP 2018*, 8:1–8:13.

Castellan, S., Clairambault, P., Paquet, H. and Winskel, G. (2018). The concurrent game semantics of probabilistic PCF. In: *Proceedings of LICS 2018*, 215–224.

Cho, K. and Jacobs, B. (2017). Disintegration and bayesian inversion, both abstractly and concretely. to appear in Mathematical Structures in Computer Science.

Clairambault, P. and Paquet, H. (2018). Fully abstract models of the probabilistic lambda-calculus. In: *Proceedings of CSL 2018*, 16:1–16:17.

Crubillé, R. and Dal Lago, U. (2014). On probabilistic applicative bisimulation and call-by-value λ-calculi. In: *Proceedings of ESOP 2014*, 209–228.

Dal Lago, U., Faggian, C., Valiron, B. and Yoshimizu, A. (2017). The geometry of parallelism: classical, probabilistic, and quantum effects. In: *Proceedings of POPL 2017*, 833–845.

Dal Lago, U. and Grellois, C. (2017). Probabilistic termination by monadic affine sized typing. In: *Proceedings of ESOP 2017*, 393–419.

Dal Lago, U. and Hoshino, N. (2019a). The geometry of Bayesian programming. *CoRR*, abs/1904.07425.

Dal Lago, U. and Hoshino, N. (2019b). The geometry of Bayesian programming. In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 1–13.

Dal Lago, U., Sangiorgi, D. and Alberti, M. (2014). On coinductive equivalences for higher-order probabilistic functional programs. In: *Proceedings of POPL 2014*, 297–308.

Danos, V. and Ehrhard, T. (2011). Probabilistic coherence spaces as a model of higher-order probabilistic computation. *Information and Computation* **209** (6) 966–991.

Danos, V. and Harmer, R. (2002). Probabilistic game semantics. *ACM Transactions on Computational Logic* **3** (3), 359–382.

Ehrhard, T., Pagani, M. and Tasson, C. (2018a). Full abstraction for probabilistic PCF. *Journal of the ACM* **65** (4) 23:1–23:44.

Ehrhard, T., Pagani, M. and Tasson, C. (2018b). Measurable cones and stable, measurable functions: a model for probabilistic higher-order programming. *PACMPL* **2**(POPL) 59:1–59:28.

Girard, J.-Y. (1987). Linear logic. *Theoretical Computer Science* **50** (1) 1–101.

Girard, J.-Y. (1989). Geometry of interaction 1: Interpretation of system F. In: Ferro, R., Bonotto, C., Valentini, S. and Zanardo, A. (eds.), *Logic Colloquium 1988*, vol. 127. Studies in Logic and the Foundations of Mathematics. Elsevier, 221–260.

Goodman, N. D., Mansinghka, V. K., Roy, D. M., Bonawitz, K. and Tenenbaum, J. B. (2008). Church: A language for generative models. In: *Proceedings of UAI 2008*, 220–229.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57** (1) 97–109.

Heunen, C., Kammar, O., Staton, S. and Yang, H. (2017). A convenient category for higher-order probability theory. In: *Proceedings of LICS 2017*. IEEE Computer Society, 1–12.

Hoshino, N., Muroya, K. and Hasuo, I. (2014). Memoryful geometry of interaction: From coalgebraic components to algebraic effects. In: *Proceedings of CSL-LICS 2014*.

Hyland, J. M. E. and Ong, C. L. (2000). On full abstraction for PCF: i, ii, and III. *Information and Computation* **163** (2) 285–408.

Hyland, M. and Schalk, A. (2003). Glueing and orthogonality for models of linear logic. *Theoretical Computer Science* **294** (1) 183–231. Category Theory and Computer Science.

Jacobs, B. (2016). *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Jacobs, B., Zanasi, F. and Kissinger, A. (2018). Causal inference by string diagram surgery. arXiv:1811.08338 [cs.LO].
Jones, C. (1990). *Probabilistic Non-Determinism*. PhD thesis, University of Edinburgh.
Joyal, A., Street, R. and Verity, D. (1996). Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society* **119** (3) 447–468.
Jung, A. and Tix, R. (1998). The troublesome probabilistic powerdomain. *Electronic Notes in Theoretical Computer Science* **13** 70–91.
Lafont, Y. (1995). *From Proof Nets to Interaction Nets*. London Mathematical Society Lecture Note Series. Cambridge University Press, 225–248.
Laurent, O. (2001). A token machine for full geometry of interaction. In: *Proceedings of TLCA 2001*, 283–297.
Mackie, I. (1995). The geometry of interaction machine. In: *Proceedings of POPL 1995*, 198–208.
Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21** (6) 1087–1092.
Miller, G. L. (1976). Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences* **13** (3) 300–317.
Moggi, E. (1991). Notions of computation and monads. *Information and Computation* **93** (1) 55–92. Selections from 1989 IEEE Symposium on Logic in Computer Science.
Muroya, K., Hoshino, N. and Hasuo, I. (2016). Memoryful geometry of interaction ii: Recursion and adequacy. In: *Proceedings of POPL 2016*, 748–760.
Paquet, H. and Winskel, G. (2018). Continuous probability distributions in concurrent games. *Electronic Notes in Theoretical Computer Science* **341** 321–344.
Rabin, M. O. (1980). Probabilistic algorithm for testing primality. *Journal of Number Theory* **12** (1) 128–138.
Sato, T., Aguirre, A., Barthe, G., Gaboardi, M., Garg, D. and Hsu, J. (2019). Formal verification of higher-order probabilistic programs: Reasoning about approximation, convergence, Bayesian inference, and optimization. *PACMPL* **3** 38:1–38:30.
Selinger, P. (2011). *A Survey of Graphical Languages for Monoidal Categories*. Springer Berlin Heidelberg, Berlin, Heidelberg, 289–355.
Staton, S. (2017). Commutative semantics for probabilistic programming. In: *Proceedings of ESOP 2017*, 855–879.
Vákár, M., Kammar, O. and Staton, S. (2019). A domain theory for statistical probabilistic programming. *Proceedings of the ACM on Programming Languages* **3** (POPL) 36:1–36:29.
Wand, M., Culpepper, R., Giannakopoulos, T. and Cobb, A. (2018). Contextual equivalence for a probabilistic language with continuous random variables and recursion. *Proceedings of the ACM on Programming Languages* **2** (ICFP) 87:1–87:30.
Wood, F. D., van de Meent, J. and Mansinghka, V. (2014). A new approach to probabilistic programming inference. In: *Proceedings of AISTATS 2014*, 1024–1032.

Cite this article: Dal Lago U and Hoshino N. (2021). The geometry of Bayesian programming. *Mathematical Structures in Computer Science*. https://doi.org/10.1017/S0960129521000396