A Model of Stochastic Memoization and Name Generation

Younesse Kaddar

University of Oxford
younesse.kaddar@cs.ox.ac.uk

Contents

1 A simple typed language

Deterministic memoization of a function f is the process of systematically storing the result of applying f to every argument, so that we can reuse it later when the same call is made [Mic68]. Since we only need to compute such a result once, this leads to a speed-up of the program, but it does not change its semantics.

However, in the presence of probabilistic side-effects, memoizing a stochastic function f is no longer just an optimization technique. It does change the semantics [Roy+08; Sta21; Woo+09], enabling us to define (possibly infinite) random sequences, which are of paramount importance in probability and statistics.

Categorically, such a stochastic function is interpreted as a probabilistic kernel $f\colon X \to PY$ in a suitable cartesian closed category[1] – *e.g.* the category of Quasi-Borel spaces [Heu+17] – where P is a probability monad. Memoization is then internally expressed as a morphism

$$\mathrm{mem}_{X,Y}\colon (PY)^X \longrightarrow P(Y^X)$$

converting a probabilistic kernel (which associates, for every given input in X, a random output in Y) into a random function $X \to Y$ (randomly choosing all the outputs for all the possible inputs at once). If f is written as a lambda-abstraction $\lambda x.\, u\colon X \to PY$, $\mathrm{mem}_{X,Y}(f)$ will be denoted by[2] $\lambda_{\text{מ}} x.\, u$.

We consider a small simply typed language to shed light on three features that we model semantically:

- **name generation**: we can generate fresh names (referred to as *atomic* names or *atoms*, in the sense of Pitt's nominal set theory [Pit13]) with constructs such as $\mathrm{let}\, x = \mathsf{fresh}()\, \mathrm{in}\, \cdots$.
- basic **probabilistic effects**: for illustrative purposes, the only distribution we consider, as a first step, is the Bernoulli distribution with bias $p = 1/2$. Constructs like $\mathrm{let}\, b = \mathsf{flip}()\, \mathrm{in}\, \cdots$ amount to flipping a fair coin and storing its result in a variable b.
- **stochastic memoization**: if a stochastic function f – memoized with the new $\lambda_{\text{מ}}$ operator – is called twice on the same argument, it should return the same result.

We have the following base types: bool (booleans), $\mathbb{A}$ (atomic names), and $\mathbb{F}$ (intended for memoized functions $\mathbb{A} \to \mathsf{bool}$). For the sake of simplicity, we do not have arbitrary function types.

[1] cartesian closedness enables us to model higher-order functions

[2] borrowing Melliès' use of the Hebrew letter "mem" [Mel14], to mean "memoization" here

$$A, B \ ::= \ \mathsf{bool} \mid \mathbb{A} \mid \mathbb{F} \mid A \times B$$

In fine-grained call-by-value fashion [Lev06], there are two kinds of judgements: typed values, and typed computations.

Values:

$$\frac{-}{\Gamma, x : A \vdash^{\mathsf{v}} x : A} \qquad \frac{\Gamma \vdash^{\mathsf{v}} v : A \qquad \Gamma \vdash^{\mathsf{v}} w : B}{\Gamma \vdash^{\mathsf{v}} (v, w) : A \times B} \qquad \frac{-}{\Gamma \vdash^{\mathsf{v}} \mathsf{true} : \mathsf{bool}} \qquad \frac{-}{\Gamma \vdash^{\mathsf{v}} \mathsf{false} : \mathsf{bool}}$$

Computations:

$$\frac{\Gamma \vdash^{\mathsf{v}} v : A}{\Gamma \vdash^{\mathsf{c}} \mathsf{return}(v) : A} \qquad \frac{\Gamma \vdash^{\mathsf{c}} u : A \qquad \Gamma, x : A \vdash^{\mathsf{c}} t : B}{\Gamma \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \ \leftarrow \ u\ \mathsf{in}\ t : B}$$

Matching:

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \mathsf{bool} \qquad \Gamma \vdash^{\mathsf{c}} u : A \qquad \Gamma \vdash^{\mathsf{c}} t : A}{\Gamma \vdash^{\mathsf{c}} \mathsf{if}\ v\ \mathsf{then}\ u\ \mathsf{else}\ t : A} \qquad \frac{\Gamma \vdash^{\mathsf{v}} v : A \times B \qquad \Gamma, x : A, y : B \vdash^{\mathsf{c}} t : C}{\Gamma \vdash^{\mathsf{c}} \mathsf{match}\ v\ \mathsf{as}\ (x, y)\ \mathsf{in}\ t : C}$$

Language-specific commands:

$$\frac{-}{\Gamma \vdash^{\mathsf{c}} \mathsf{flip}() : \mathsf{bool}} \qquad \frac{-}{\Gamma \vdash^{\mathsf{c}} \mathsf{fresh}() : \mathbb{A}} \qquad \frac{\Gamma \vdash^{\mathsf{v}} v : \mathbb{A} \qquad \Gamma \vdash^{\mathsf{v}} w : \mathbb{A}}{\Gamma \vdash^{\mathsf{c}} (v = w) : \mathsf{bool}}$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \mathbb{F} \qquad \Gamma \vdash^{\mathsf{v}} w : \mathbb{A}}{\Gamma \vdash^{\mathsf{c}} (v @ w) : \mathsf{bool}} \qquad \frac{\Gamma, x : \mathbb{A} \vdash^{\mathsf{c}} u : \mathsf{bool}}{\Gamma \vdash^{\mathsf{c}} \lambda_{\text{ת}} x.\ u : \mathbb{F}}$$

Remark 1.1. Note that the only way to produce a term of type $\mathbb{A}$ or $\mathbb{F}$ is by a typed computation. In derivation rules such as the ones of $\Gamma \vdash^{\mathsf{c}} (v @ w) : \mathsf{bool}$ and $\Gamma \vdash^{\mathsf{c}} (v = w) : \mathsf{bool}$, where v, w are assumed to be *values* of type $\mathbb{A}$ or $\mathbb{F}$, this implies that v, w are bound in the overall program to variables by a let-binding (*e.g.* $\mathsf{let\ val}\ v \ \leftarrow \ \mathsf{fresh}()\ \mathsf{in\ let\ val}\ w \ \leftarrow \ \mathsf{fresh}()\ \mathsf{in}\ (v = w)$).

Notation 1.2. We denote by Contexts and Values the sets of contexts and values.

2 Denotational semantics

We work in the (cartesian closed) category of covariant presheaves on the category

$$\mathbf{BiGrph}_{emb} \;:=\; (\mathbf{Grph}_{emb} \hookrightarrow \mathbf{Grph}) \downarrow (\mathbf{Grph} \xleftarrow{\Delta_{K_2}} *)$$

of finite bipartite graphs (henceforth called *bigraphs*) and embeddings (that do not add or remove edges).

Notation 2.1. For a bigraph g, we denote by g_L (resp. g_R) and E^g its set of left (resp. right) nodes and its edge relation.

The denotation of basic types is given by:

$$[\![\mathbb{F}]\!] = \mathbf{BiGrph}_{emb}(\circ, -) \qquad [\![\mathbb{A}]\!] = \mathbf{BiGrph}_{emb}(\bullet, -)$$

where $\circ$ and $\bullet$ are the one-vertex left and right graphs respectively. The denotation of the type of booleans is the constant presheaf $2 \cong 1 + 1$, as usual.

For a bigraph g and a presheaf $X = [\![\mathcal{X}]\!]$, $X(g)$ is thought of as the set of generative models/programs of type $\mathcal{X}$ that may use the bigraph g, in the following sense: probabilistic function (that we want to memoize) and atom labels are stored as left and right nodes respectively. The presence (resp. absence) of an edge between a given left and right node memoizes the fact that a probabilistic call of the corresponding function on the corresponding atom has resulted in true (resp. false).

For every embedding $\iota \colon g \hookrightarrow g'$, the function $X\iota \colon X(g) \to X(g')$ models substitution in the programs in $X(g)$ according to ι.

Remark 2.2. The model will soon be refined to the subtopos of sheaves over $\mathbf{BiGrph}_{emb}$ for the atomic Grothendieck topology. The sheaf condition expresses the fact that if two bigraphs g_1, g_2 have a common intersection g and $x \in X(g_1) \cap X(g_2)$, then the program x can be regarded as only using g, so that $x \in X(g)$ [Sta].

Moreover, in a standard way, a value judgement $\Gamma \vdash^{\mathrm{v}} v : A$ is interpreted as a natural transformation $[\![\Gamma]\!] \to [\![A]\!]$: the variable rule being a projection, the product type constructor a categorical pairing, and the bool constructors (true, false) the two canonical coprojections.

Let P_{f} be the finite probability monad on $[\mathbf{BiGrph}_{emb}, \mathbf{Set}]$:

$$P_{\mathrm{f}}(X)(g) = \left\{ p : X(g) \to [0,1] \;\middle|\; \mathsf{supp}(p) \text{ finite and } \sum_x p(x) = 1 \right\}$$

By considering the following 'node-generation' monad N on $[\mathbf{BiGrph}_{emb}, \mathbf{Set}]$:

$$N(X)(g) = \mathrm{colim}_{g \to h}\, X(h)$$

one could be tempted to think that modeling name generation and stochastic memoization is a matter of composing these two monads. But this is not quite enough.

We also need to remember, in the monadic computations, the probability of a function returning true for a fresh, unseen atom. To do so, analogously to Plotkin and Power's local state monad [Mel14; PP02; Sta10], we introduce a probabilistic local state monad on $[\mathbf{BiGrph}_{emb}, \mathbf{Set}]$.

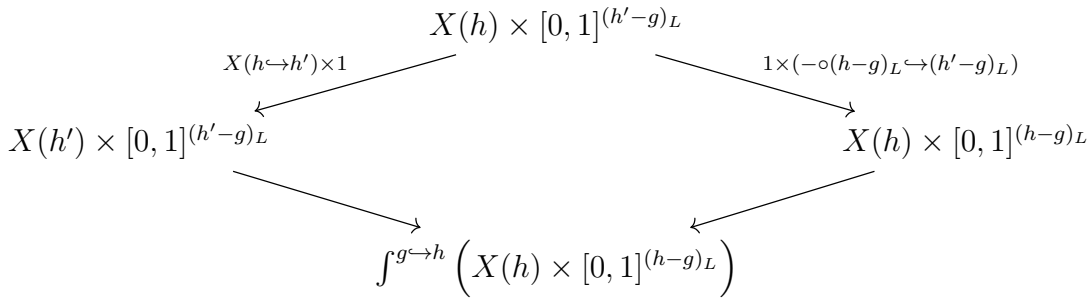
2.1 Probabilistic local state monad

Definition 2.3 (Probabilistic local state monad)**.** For all covariant presheaf $X : \mathbf{BiGrph}_{emb} \to \mathbf{Set}$ and bigraph $g \in \mathbf{BiGrph}_{emb}$:

$$T(X)(g) \;:=\; \left(P_{\mathrm{f}} \int^{g \hookrightarrow h} \left(X(h) \times [0,1]^{(h-g)_L}\right)\right)^{[0,1]^{g_L}}$$

It is similar to the read-only state monad, except that any fresh node can be initialized. Every $\lambda \in [0,1]^{g_L}$ is thought of as the probability of the corresponding function/left node being true on a new fresh atom. We will refer to such a λ as a *state of biases*.

Remark 2.4. The coend takes care of garbage collection. Indeed, if g is a subbigraph of h, h', and h is a subbigraph of h', by universal property of the coend, we have the following commutative diagram:



This means that if the state of biases is defined over h' but $h' - h$ is unused (since the program comes from an element $X(h)$), we might as well discard it and restrict ourselves to h (*i.e.* $h' - h$ is garbage collected).

In the following subsections, we will show that:

Theorem 2.5. *The assignment T can be extended to an affine strong commutative monad on* $\mathbf{BiGrph}_{emb}$.

2.2 Functoriality

In the following, $X, Y, Z \colon \mathbf{BiGrph}_{emb} \to \mathbf{Set}$ denote presheaves, $g = (g_L, g_R, E^g), g', h, h' \in \mathbf{BiGrph}_{emb}$ bigraphs, $\iota, \iota' \colon g \hookrightarrow g'$ bigraph embeddings, and $\lambda, \lambda' \colon [0,1]^{g_L}$ states of biases. We will omit subscripts when they are clear from the context.

Notation 2.6. Equivalence classes in $\int^{g \hookrightarrow h} X(h) \times [0,1]^{(h-g)_L}$ are written $[x_h, \lambda^h]_g$. We use Dirac's bra-ket notation $\left|[x_h, \lambda^h]_g\right\rangle_h$ to denote a formal column vector of equivalence classes ranging over a finite set of h's. As such, a formal convex sum $\sum_i p_i [x_{h_i}, \lambda^{h_i}]_g \in P_{\mathrm{f}} \int^{g \hookrightarrow h} X(h) \times [0,1]^{(h-g)_L}$ will be concisely denoted by $\left\langle \overrightarrow{p} \;\middle|\; [x_h, \lambda^h]_g \right\rangle_h$.

Definition 2.7 (Action of $T(X)$ on morphisms)**.**

$$T(X)(g \overset{\iota}{\hookrightarrow} g') \colon \begin{cases} \left(P_{\mathrm{f}} \int^{g \hookrightarrow h} X(h) \times [0,1]^{(h-g)_L}\right)^{[0,1]^{g_L}} \longrightarrow \left(P_{\mathrm{f}} \int^{g' \hookrightarrow h'} X(h') \times [0,1]^{(h'-g')_L}\right)^{[0,1]^{g'_L}} \\ \vartheta \mapsto [0,1]^{g'_L} \xrightarrow{- \circ \iota_L} [0,1]^{g_L} \xrightarrow{\vartheta} P_{\mathrm{f}} \int^{g \hookrightarrow h} X(h) \times [0,1]^{(h-g)_L} \\ \qquad\qquad \xrightarrow{P_{\mathrm{f}} \psi_{g,g'}} P_{\mathrm{f}} \int^{g' \hookrightarrow h'} X(h') \times [0,1]^{(h'-g')_L} \end{cases}$$

where

- the map $\int^{g \hookrightarrow h} X(h) \times [0,1]^{(h-g)_L} \xrightarrow{\psi_{g,g'}} \int^{g' \hookrightarrow h'} X(h') \times [0,1]^{(h'-g')_L}$ is given by:

$$\frac{\begin{cases} X(h) \times [0,1]^{(h-g)_L} \to X(h \coprod_g g') \times [0,1]^{(h \coprod_g g' - g')_L} \to \int^{g' \hookrightarrow h'} X(h') \times [0,1]^{(h'-g')_L} \\ (x_h,\, \lambda^h) \longmapsto (X(h \hookrightarrow h \coprod_g g')(x_h),\, \lambda^h) \longmapsto [X(h \hookrightarrow h \coprod_g g')(x_h),\, \lambda^h]_{g'} \end{cases}}{\int^{g \hookrightarrow h} X(h) \times [0,1]^{(h-g)_L} \xrightarrow{\psi_{g,g'}} \int^{g' \hookrightarrow h'} X(h') \times [0,1]^{(h'-g')_L}} \text{ extranat. in } h$$

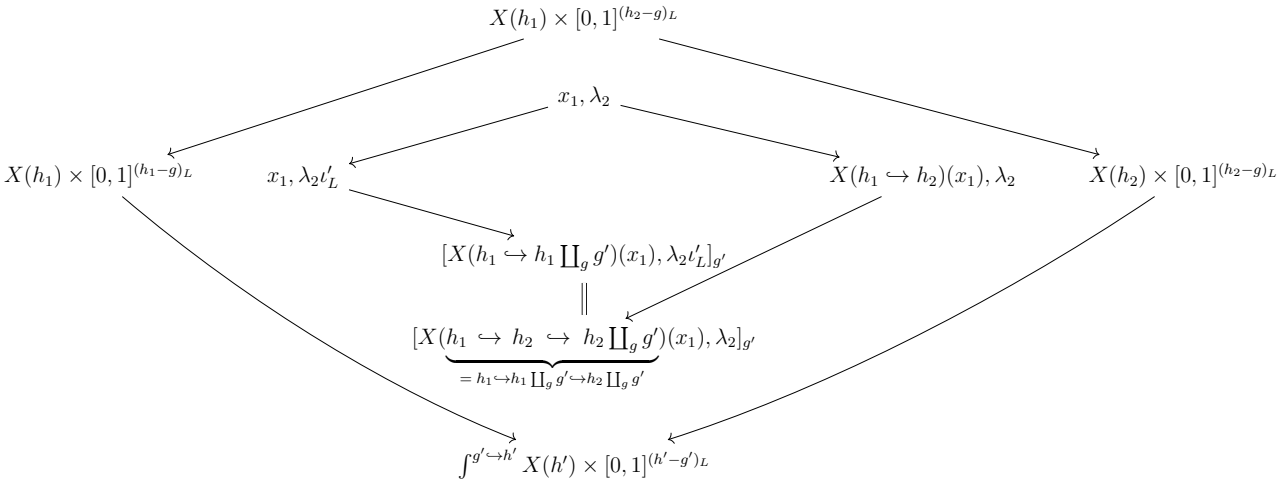
- $\iota_L \colon g_L \hookrightarrow g'_L$ is the embedding restricted to left nodes.
- $h \coprod_g g'$ is the pushout in the category of graphs regarded as an object of $\mathbf{BiGrph}_{emb}$:

$$\begin{array}{ccc} g & \hookrightarrow & g' \\ \downarrow & \ulcorner & \downarrow \\ h & \hookrightarrow & h \coprod_g g' \end{array}$$

Notation 2.8. More concretely, with Dirac's convenient bra-ket notation, $T(X)(g \overset{\iota}{\hookrightarrow} g')$ can be written as:

$$T(X)(\iota) = \begin{cases} \left(P_{\mathrm{f}} \int^{g \hookrightarrow h} X(h) \times [0,1]^{(h-g)_L}\right)^{[0,1]^{g_L}} \longrightarrow \left(P_{\mathrm{f}} \int^{g' \hookrightarrow h'} X(h') \times [0,1]^{(h'-g')_L}\right)^{[0,1]^{g'_L}} \\ \vartheta \longmapsto \lambda' \mapsto \text{let } \vartheta(\lambda' \iota_L) \;=\; \left\langle \overrightarrow{p} \;\middle|\; [x_h, \lambda^h]_g \right\rangle_h \text{ in } \left\langle \overrightarrow{p} \;\middle|\; [X(h \hookrightarrow h \coprod_g g')(x_h), \lambda^h]_{g'} \right\rangle_h \end{cases}$$

In the definition of $\psi_{g,g'}$, extranaturality in h does indeed hold, as for every $g \hookrightarrow h_1, h_2$ and $h_1 \hookrightarrow h_2$:



Proposition 2.9 (Functoriality). *T is an endofunctor on* $\mathbf{BiGrph}_{emb}$.

Proof. For $\iota \colon g \hookrightarrow g = \mathrm{id}$, $\left\langle \overrightarrow{p} \;\middle|\; [X(h \hookrightarrow \underbrace{h \coprod_g g}_{=h})(x_h), \lambda^h]_g \right\rangle_h = \left\langle \overrightarrow{p} \;\middle|\; [x_h, \lambda^h]_{g'} \right\rangle_h$ and $\lambda' \iota_L = \lambda'$ so $T(X)(g \overset{\iota}{\hookrightarrow} g) = \mathrm{id}$.

For every $g \hookrightarrow g' \hookrightarrow g''$:

$$\sum_i p_i \Big[X(h_i \hookrightarrow h_i \coprod_g g'')(x_{h_i}), \lambda^{h_i} \Big]_{g''} \\ = \sum_i p_i \Big[X(h_i \hookrightarrow h_i \coprod_g g' \hookrightarrow (h_i \coprod_g g') \coprod_{g'} g'')(x_{h_i}), \lambda^{h_i \coprod_g g'} \Big]_{g''}$$

so $T(X)(g \hookrightarrow g' \hookrightarrow g'') = T(X)(g \hookrightarrow g') \,;\, T(X)(g' \hookrightarrow g'')$. □

2.3 Monad structure

T can be endowed with the structure of a $\mathrm{Psh}(\mathbf{BiGrph}^{\mathrm{op}}_{emb})$-enriched monad, that is, since $\mathrm{Psh}(\mathbf{BiGrph}^{\mathrm{op}}_{emb})$ is a (cartesian) monoidal closed category, a strong monad. Its unit $\eta_X \colon 1 \to TX^X$ and bind $(-)^* \colon TY^X \to TY^{TX}$ are described in the following section.

Reminder. Recall that, in the category $\mathrm{Psh}(\mathbf{BiGrph}^{\mathrm{op}}_{emb}) := [\mathbf{BiGrph}_{emb}, \mathbf{Set}]$ of covariant presheaves over $\mathbf{BiGrph}_{emb}$, the exponential object Y^X is given by $g \mapsto [X \times \text{よ}(g), Y]$, where

- $\text{よ} : \mathbf{BiGrph}^{\mathrm{op}}_{emb} \to [\mathbf{BiGrph}_{emb}, \mathbf{Set}]$ is the covariant Yoneda embedding.
- $[X', Y'] = \mathrm{Nat}\,(X', Y')$ is the set of natural transformations between two presheaves X' and Y'.

2.3.1 Enriched Bind and Unit

Enriched unit $\eta_X \colon 1 \to TX^X$:

$$\eta_X(g) \colon \begin{cases} \{*\} & \longrightarrow [X \times \text{ﻉ}(g), TX] \\ * & \longmapsto \mathfrak{y}g' \colon \begin{cases} X(g') \times \text{ﻉ}(g)(g') & \longrightarrow TX(g') \\ x_{g'},\ \_ & \longmapsto \Big([0,1]^{g'_L} \ni \lambda \mapsto 1 \cdot |[x_{g'}, !]_{g'}\rangle \Big) \end{cases} \end{cases}$$

Enriched bind $(-)^* \colon TY^X \to TY^{TX}$:

$$(-)^*_g \colon \begin{cases} TY^X(g) \longrightarrow \big[TX \times \text{ﻉ}(g),\ TY\big] \\ \varphi \longmapsto \varphi^* \end{cases}$$

where

$$\varphi^*_{g'} \colon \begin{cases} \Big(P_{\mathrm{f}} \displaystyle\int^{g' \hookrightarrow h} X(h) \times [0,1]^{(h-g')_L}\Big)^{[0,1]^{g'_L}} \times \text{ﻉ}(g)(g') \xrightarrow{\varphi^*_{g'}} \Big(P_{\mathrm{f}} \int^{g' \hookrightarrow h'} Y(h') \times [0,1]^{(h'-g')_L}\Big)^{[0,1]^{g'_L}} \\ (\vartheta,\ g \xhookrightarrow{\iota} g') \xmapsto{\varphi^*_{g'}} \lambda' \mapsto & \text{let } \vartheta(\lambda') \ = \ \big\langle \overrightarrow{p} \,\big|\, [x_h, \lambda^h]_{g'} \big\rangle_{h \in H_{g'}} \text{ in} \\ & \text{for each } h \in H_{g'}, \\ & \quad \text{let } \varphi_h(x_h, g \xhookrightarrow{\iota} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \ = \ \big\langle \overrightarrow{q_h} \,\big|\, [y'_h, \gamma^{h'}]_h \big\rangle_{h' \in H_h} \text{ in} \\ & \big\langle \overrightarrow{p} \,\big|\, Q \,\big|\, [y_{h'}, \gamma^{h'} \sqcup \lambda^h]_{g'} \big\rangle_{h' \in \bigcup\limits_{h \in H_{g'}} H_h} \end{cases}$$

and

$$Q \ := \ \begin{pmatrix} \overrightarrow{q_{h_1}} \\ \vdots \\ \overrightarrow{q_{h_n}} \end{pmatrix}_{h_i \in H_{g'}} = \left(\begin{array}{c|c|c} q_{h_1,h'_1} & \cdots & q_{h_1,h'_m} \\ \vdots & & \vdots \\ q_{h_n,h'_1} & & q_{h_n,h'_m} \end{array}\right)_{\substack{h_i \in H_{g'} \\ h'_j \in \bigcup\limits_{h \in H_{g'}} H_h}}$$

Remark 2.10. In the definition of Q, we assume that each $\overrightarrow{q_h}$ (for $h \in H_{g'}$) has been 0-padded accordingly.

Reminder. Internal composition $Z^Y \times Y^X \xrightarrow{\circ} Z^X$ in the category of presheaves is given by:

$$\forall g, \begin{cases} [Y \times \text{ﻉ}(g), Z] \times [X \times \text{ﻉ}(g), Y] & \longrightarrow [X \times \text{ﻉ}(g), Z] \\ \psi, \varphi \longmapsto X \times \text{ﻉ}(g) \xrightarrow{1 \times \Delta} X \times \text{ﻉ}(g) \times \text{ﻉ}(g) \xrightarrow{\varphi \times 1} Y \times \text{ﻉ}(g) \xrightarrow{\psi} Z \end{cases}$$

where Δ is the diagonal functor. In particular, $Z^Y \times Y \xrightarrow{\mathrm{ev} = \circ} Z$ reduces to:

$$\forall g, \begin{cases} [Y \times \text{ﻉ}(g), Z] \times Y(g) & \longrightarrow Z(g) \\ \psi, y & \longmapsto \psi_g((y, \mathrm{id}_g)) \end{cases}$$

Notation 2.11. Morphisms $f \colon X \to TY$ in $[\mathbf{BiGrph}_{emb}, \mathbf{Set}]$ will sometimes, by abuse of notation, be identified with their exponential transpose $\lambda_X.f \colon 1 \to TY^X$. Moreover:

$$f^* := 1 \xrightarrow{f} TY^X \xrightarrow{(-)^*} TY^{TX}$$

2.3.2 Associativity and unit laws

Right unit law $\Phi^* \circ \eta = \Phi$: Let us show that

$$\begin{array}{ccccc} 1 & \xrightarrow{\Phi} & TY^X & \xrightarrow{(-)^*} & TY^{TX} \\ \times & & \times & \xrightarrow{\quad\circ\quad} & TY^X \\ 1 & \xrightarrow[\eta]{\qquad\qquad} & TX^X & & \end{array} \qquad = \qquad 1 \xrightarrow{\Phi} TY^X$$

First:

$$\Phi_g^* = \begin{cases} 1(g) \xrightarrow{\Phi_g} [X \times \text{ﻉ}(g), TY] \xrightarrow{(-)_g^*} TY^{TX}(g) \\ * \longmapsto \varphi \quad \longmapsto (\vartheta, g \xhookrightarrow{\iota} g') \\ \qquad\qquad \xrightarrow{\varphi_{g'}^*} \lambda' \mapsto \text{let } \vartheta(\lambda') \;=\; \left\langle \overrightarrow{p} \;\middle|\; [x_h, \lambda^h]_{g'} \right\rangle_h \text{ in} \\ \qquad\qquad\quad \text{for each } h, \\ \qquad\qquad\qquad \text{let } \varphi_h(x_h, g \xhookrightarrow{\iota} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \;=\; \left\langle \overrightarrow{q_h} \;\middle|\; [y_{h'}, \gamma^{h'}]_h \right\rangle_{h'} \text{ in} \\ \qquad\qquad\quad \left\langle \overrightarrow{p} \;\middle|\; Q \;\middle|\; [y_{h'}, \gamma^{h'} \sqcup \lambda^h]_{g'} \right\rangle_{h'} \end{cases}$$

and

$$\Phi_g^* \circ \eta_g = \begin{cases} (1 \times 1)(g) \xrightarrow{\Phi_g^* \times \eta_g} TY^{TX}(g) \times TX^X(g) \xrightarrow{\circ} TY^X(g) \\ (*, *) \longmapsto \varphi_{g'}^*,\ \mathfrak{y} \underset{g'}{\longmapsto} (1 \times \Delta)\,;\,(\mathfrak{y} \times 1)\,;\,\varphi^* \overset{?}{=} \varphi \end{cases}$$

where

$$\mathfrak{y}_{g'} \colon \begin{cases} X(g') \times \text{ﻉ}(g)(g') \longrightarrow TX(g') \\ x_{g'},\ \_ \longmapsto \left([0,1]^{g'_L} \ni \lambda \mapsto 1 \cdot |[x_{g'}, !]_{g'}\rangle \right) \end{cases}$$

The morphism $X \times \text{よ}(g) \xrightarrow{1\times\Delta} X \times \text{よ}(g) \times \text{よ}(g) \xrightarrow{\mathfrak{y}\times 1} TX \times \text{よ}(g) \xrightarrow{\varphi^*} TY$ is then given by:

$$\begin{cases} X(g') \times \text{よ}(g)(g') \xrightarrow{(\mathrm{id}\times\Delta)_{g'}} X(g') \times \text{よ}(g)(g')^2 \xrightarrow{(\mathfrak{y}\times\mathrm{id})_{g'}} TX(g') \times \text{よ}(g)(g') \xrightarrow{\varphi^*_{g'}} TY(g') \\ x_{g'},\ g \overset{\iota}{\hookrightarrow} g' \longmapsto x_{g'},\ \iota,\ \iota \quad \longmapsto [0,1]^{g'_L} \ni \lambda \mapsto 1 \cdot |[x_{g'}, !]\rangle\ ,\ \iota \\ \qquad\qquad\qquad\qquad\qquad\qquad \longmapsto \lambda' \mapsto \text{let } \varphi_{g'}(x_{g'}, \iota)(\lambda') \ = \ \left\langle \overrightarrow{q} \mid [y_{h'}, \gamma^{h'}]_{g'} \right\rangle_{h'} \text{ in} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad \langle 1 \mid \overrightarrow{q} \mid [y_{h'}, \underbrace{\gamma^{h'} \sqcup !}_{=\gamma^{h'}}]_{g'} \rangle_{h'} \end{cases}$$

Left unit law $\eta^* = \mathrm{id}$: Let us show that

$$1 \xrightarrow{\eta} TX^X \xrightarrow{(-)^*} TX^{TX} = 1 \xrightarrow{\lambda_{TX}\,\cdot\,\mathrm{id}} TX^{TX}$$

The left-hand side (LHS) is equal to:

$$* \overset{\eta_g}{\longmapsto} \mathfrak{y}_{g'} : \begin{cases} X(g') \times \text{よ}(g)(g') \longrightarrow TX(g') \\ x_{g'},\ \_ \longmapsto \Big([0,1]^{g'_L} \ni \lambda \mapsto 1 \cdot |[x_{g'}, !]\rangle \Big) \end{cases}$$

$$\xmapsto{(-)^* g} \vartheta, g \overset{\iota}{\hookrightarrow} g' \overset{\mathfrak{y}^*_{g'}}{\longmapsto} \lambda' \longmapsto \ \text{let } \vartheta(\lambda') \ = \ \left\langle \overrightarrow{p} \mid [x_h, \lambda^h]_{g'} \right\rangle_h \text{ in}$$

$$\text{for each } h,\ \text{let } \mathfrak{y}_h(x_h,\ g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \ = \ 1 \cdot |[x_h, !]_h\rangle \text{ in}$$

$$\langle \overrightarrow{p} \mid \underbrace{\mathbf{I}}_{\text{identity matrix}} \mid [x_h, \underbrace{! \sqcup \lambda^h}_{=\lambda^h}]_{g'} \rangle_h$$

$$\|$$

$$\underbrace{\vartheta,\ \iota \overset{\mathfrak{y}^*_{g'}}{\longmapsto} \lambda' \longmapsto \vartheta(\lambda')}_{=\,\text{RHS}}$$

Associativity $\Psi^* \circ \Phi^* = (\Psi^* \circ \Phi)^*$: Let us show that

$$
\begin{array}{ccccc}
1 & \xrightarrow{\Psi} & TZ^Y & \xrightarrow{(-)^*} & TZ^{TY} \\
\times & & \times & & \times \xrightarrow{\circ} TY^{TX} \\
1 & \xrightarrow[\Phi]{} & TY^X & \xrightarrow[(-)^*]{} & TY^{TX} \\
 & & \| & & \\
1 & \xrightarrow{\Psi} & TZ^Y & \xrightarrow{(-)^*} & TZ^{TY} \\
\times & & & & \times \xrightarrow{\circ} TY^X \xrightarrow{(-)^*} TY^{TX} \\
1 & & \xrightarrow[\Phi]{} & & TY^X
\end{array}
$$

The LHS

$$(\Psi_g^* \times \Phi_g^*)\,;\circ = (1(g) \times 1(g) \xrightarrow{\Psi_g \times \Phi_g} TZ^Y(g) \times TY^{TX}(g) \xrightarrow{(-)_g^* \times (-)_g^*} TZ^{TY}(g) \times TY^{TX}(g))\,;\circ$$

is equal to:

$$
\begin{aligned}
&*, * \longmapsto \begin{array}{l}\psi \in [Y \times \text{よ}(g), TZ],\\ \varphi \in [X \times \text{よ}(g), TY]\end{array}\\
&\longmapsto \Bigg(\vartheta, g \overset{\iota}{\hookrightarrow} g' \overset{\psi^*_{g'}}{\longmapsto} \lambda' \mapsto \qquad \text{let } \vartheta(\lambda') \;=\; \left\langle \overrightarrow{r} \;\middle|\; [y_h, \theta^h]_{g'} \right\rangle_h \text{ in}\\
&\qquad\qquad\qquad\qquad \text{for each } h,\\
&\qquad\qquad\qquad\qquad\quad \text{let } \psi(y_h,\ g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\theta^h \sqcup \theta') \;=\; \left\langle \overrightarrow{s_h} \;\middle|\; [z_{h'}, \chi^{h'}]_h \right\rangle_{h'} \text{ in}\\
&\qquad\qquad\qquad\qquad \left\langle \overrightarrow{r} \;\middle|\; S \;\middle|\; [z_{h'},\ \chi^{h'} \sqcup \theta^h]_{g'} \right\rangle_{h'} \Bigg),\\
&\qquad \Bigg(\vartheta, g \overset{\iota}{\hookrightarrow} g' \overset{\varphi^*_{g'}}{\longmapsto} \lambda' \mapsto \qquad \text{let } \vartheta(\lambda') \;=\; \left\langle \overrightarrow{p} \;\middle|\; [x_h, \lambda^h]_{g'} \right\rangle_{h'} \text{ in}\\
&\qquad\qquad\qquad\qquad \text{for each } h,\\
&\qquad\qquad\qquad\qquad\quad \text{let } \varphi(x_h,\ g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \;=\; \left\langle \overrightarrow{q_h} \;\middle|\; [y_{h'}, \gamma^{h'}]_h \right\rangle_{h'} \text{ in}\\
&\qquad\qquad\qquad\qquad \left\langle \overrightarrow{p} \;\middle|\; Q \;\middle|\; [y_{h'},\ \gamma^{h'} \sqcup \lambda^h]_{g'} \right\rangle_{h'} \Bigg)\\
&\overset{\circ}{\longmapsto} \underbrace{(1 \times \Delta)\,;\,(\varphi^* \times 1)\,;\,\psi^*}_{=\,\alpha} \overset{?}{=} \underbrace{((1 \times \Delta)\,;\,(\varphi \times 1)\,;\,\psi^*)^*}_{=\,\text{RHS} \,:=\, \beta}
\end{aligned}
$$

where the map

$$\alpha \;:=\; TX \times \text{よ}(g) \xrightarrow{1\times\Delta} TX \times \text{よ}(g)^2 \xrightarrow{\varphi^*\times 1} TY \times \text{よ}(g) \xrightarrow{\psi^*} TZ$$

is given by

$$TX(g') \times \text{ょ}(g)(g') \xrightarrow{(1\times\Delta)_{g'}} TX(g') \times \text{ょ}(g)(g')^2 \xrightarrow{\varphi^*_{g'} \times \mathrm{id}_{g'}} TY(g') \times \text{ょ}(g)(g') \xrightarrow{\psi^*_{g'}} TZ(g')$$

$$\vartheta, \iota \mapsto \vartheta,\ \iota, \iota \mapsto \Big(\lambda' \xmapsto{\varphi^*_{g'}(\vartheta,\iota)} \text{let } \vartheta(\lambda') \ = \ \Big\langle \overrightarrow{p} \ \Big|\ [x_h, \lambda^h]_{g'} \Big\rangle_h \text{ in}$$

$$\text{for each } h,$$

$$\text{let } \varphi_h(x_h,\ g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \ = \ \Big\langle \overrightarrow{q_h} \ \Big|\ [y_{h'}, \gamma^{h'}]_h \Big\rangle_{h'} \text{ in}$$

$$\Big\langle \overrightarrow{p} \ \Big|\ Q \ \Big|\ [y_{h'},\ \gamma^{h'} \sqcup \lambda^h]_{g'} \Big\rangle_{h'} \Big),\ \iota$$

$$\xmapsto{\psi^*_{g'}} \lambda' \mapsto \quad \text{let } \varphi^*_{g'}(\vartheta, \iota)(\lambda') \ = \ \Big\langle \overrightarrow{p} \ \Big|\ Q \ \Big|\ [y_{h'}, \gamma^{h'} \sqcup \lambda^h]_{g'} \Big\rangle_{h'} \text{ in}$$

$$\text{for each } h',$$

$$\text{let } \psi_{h'}(y_{h'},\ g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h')(\gamma^{h'} \sqcup \lambda^h \sqcup \lambda') \ = \ \Big\langle \overrightarrow{s_{h'}} \ \Big|\ [z_{h''}, \chi^{h''}]_{h'} \Big\rangle_{h''} \text{ in}$$

$$\Big\langle \overrightarrow{p} \ \Big|\ Q \ \Big|\ S \ \Big|\ [z_{h''},\ \chi^{h''} \sqcup \gamma^{h'} \sqcup \lambda^h]_{g'} \Big\rangle_{h''}$$

$$\|$$

$$\vartheta, \iota \mapsto \vartheta,\ \iota, \iota \mapsto \lambda' \xmapsto{\varphi^*_{g'}(\vartheta,\iota)} \quad \text{let } \vartheta(\lambda') \ = \ \Big\langle \overrightarrow{p} \ \Big|\ [x_h, \lambda^h]_{g'} \Big\rangle_h \text{ in}$$

$$\text{for each } h,$$

$$\text{let } \varphi_h(x_h,\ g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \ = \ \Big\langle \overrightarrow{q_h} \ \Big|\ [y_{h'}, \gamma^{h'}]_h \Big\rangle_{h'} \text{ in}$$

$$\text{for each } h',$$

$$\text{let } \psi_{h'}(y_{h'},\ g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h')(\gamma^{h'} \sqcup \lambda^h \sqcup \lambda') \ =$$

$$\Big\langle \overrightarrow{s_{h'}} \ \Big|\ [z_{h''}, \chi^{h''}]_{h'} \Big\rangle_{h''} \text{ in}$$

$$\Big\langle \overrightarrow{p} \ \Big|\ Q \ \Big|\ S \ \Big|\ [z_{h''},\ \chi^{h''} \sqcup \gamma^{h'} \sqcup \lambda^h]_{g'} \Big\rangle_{h''}$$

As for the map

$$\beta \ := \ (X \times \text{ょ}(g) \xrightarrow{1\times\Delta} X \times \text{ょ}(g) \times \text{ょ}(g) \xrightarrow{\varphi\times 1} TY \times \text{ょ}(g) \xrightarrow{\psi^*} TZ)^*$$

we have

$$X(g') \times \text{ﻉ}(g)(g') \xrightarrow{\mathrm{id}_{g'} \times \Delta_{g'}} X(g') \times \text{ﻉ}(g)(g')^2 \xrightarrow{\varphi_{g'} \times \mathrm{id}_{g'}} TY(g') \times \text{ﻉ}(g)(g') \xrightarrow{\psi^*_{g'}} TZ(g')$$

$$\begin{aligned}
x_{g'},\ g \xhookrightarrow{\iota} g' &\longmapsto x_{g'},\ \iota, \iota \\
&\longmapsto \varphi_{g'}(x_{g'}, \iota),\ \iota \\
&\xmapsto{\psi^*_{g'}} \lambda' \mapsto \text{let } \varphi_{g'}(x_{g'}, \iota)(\lambda') \ =\ \left\langle \overrightarrow{q} \,\middle|\, [y_{h'}, \gamma^{h'}]_{g'} \right\rangle_{h'} \text{ in} \\
&\qquad \text{for each } h', \\
&\qquad\quad \text{let } \psi_{h'}(y_{h'},\ g \xhookrightarrow{\iota} g' \hookrightarrow h')(\gamma^{h'} \sqcup \lambda') \ =\ \left\langle \overrightarrow{s_{h'}} \,\middle|\, [z_{h''}, \chi^{h''}]_{h'} \right\rangle_{h''} \text{ in} \\
&\qquad \left\langle \overrightarrow{q} \,\middle|\, S \,\middle|\, [z_{h''},\ \chi^{h''} \sqcup \gamma^{h'}]_{g'} \right\rangle_{h''}
\end{aligned}$$

So that β is given by:

$$TX(g') \times \text{ﻉ}(g)(g') \xrightarrow{\zeta^*(g') \coloneqq \left((\mathrm{id}(g') \times \Delta(g'))\,;\,(\varphi(g') \times \mathrm{id}(g'))\,;\,\psi^*(g')\right)^*} TZ(g')$$

$$\begin{aligned}
\vartheta,\ \iota \longmapsto \lambda' \longmapsto &\text{let } \vartheta(\lambda') \ =\ \left\langle \overrightarrow{p} \,\middle|\, [x_h, \lambda^h]_{g'} \right\rangle_h \text{ in} \\
&\quad \text{for each } h, \\
&\qquad \text{let } \zeta_h(x_h,\ g \xhookrightarrow{\iota} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \ = \\
&\qquad\qquad\qquad \left\langle \overrightarrow{q_h} \,\middle|\, S_h \,\middle|\, [z_{h''},\ \chi^{h''} \sqcup \gamma^{h'} \sqcup \lambda^h]_{g'} \right\rangle_{h''} \text{ in} \\
&\quad \left\langle \overrightarrow{p} \,\middle|\, Q \,\middle|\, S \,\middle|\, [z_{h''},\ \chi^{h''} \sqcup \gamma^{h'} \sqcup \lambda^h]_{g'} \right\rangle_{h''} \\
&\quad \Vert \\
\vartheta,\ \iota \longmapsto \lambda' \longmapsto &\text{let } \vartheta(\lambda') \ =\ \left\langle \overrightarrow{p} \,\middle|\, [x_h, \lambda^h]_{g'} \right\rangle_h \text{ in} \\
&\quad \text{for each } h, \\
&\qquad \text{let } \varphi(x_h,\ g \xhookrightarrow{\iota} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \ =\ \left\langle \overrightarrow{q_h} \,\middle|\, [y_{h'}, \gamma^{h'}]_h \right\rangle_{h'} \text{ in} \\
&\qquad\quad \text{for each } h', \\
&\qquad\qquad \text{let } \psi_{h'}(y_{h'},\ g \xhookrightarrow{\iota} g' \hookrightarrow h')(\gamma^{h'} \sqcup \lambda^h \sqcup \lambda') \ =\ \left\langle \overrightarrow{s_{h'}} \,\middle|\, [z_{h''}, \chi^{h''}]_{h'} \right\rangle_{h''} \text{ in} \\
&\qquad \left\langle \overrightarrow{p} \,\middle|\, Q \,\middle|\, S \,\middle|\, [z_{h''},\ \chi^{h''} \sqcup \gamma^{h'} \sqcup \lambda^h]_{g'} \right\rangle_{h''}
\end{aligned}$$

Therefore, $\alpha = \beta$.

2.4 Strong commutativity

Theorem 2.12 (Strong commutativity)**.** *T is a strong commutative monad,* i.e. *in the internal language of the topos: for every $a\colon 1 \to TA, b\colon 1 \to TB$:*

$$a \gg\!= \lambda x.\, b \gg\!= \lambda y.\, \eta(x, y) \ =\ b \gg\!= \lambda y.\, a \gg\!= \lambda x.\, \eta(x, y) \tag{1}$$

Proof. The term $b \gg\!= \lambda y.\, \eta(x, y)$ is given by:

$$\begin{array}{ccc} A & \xrightarrow{\lambda_B.\eta} & T(A \times B)^B \xrightarrow{(-)^*} T(A \times B)^{TB} \\ \times & & \times \xrightarrow{\text{ev}} T(A \times B) \\ 1 & \xrightarrow[b]{} & TB \end{array}$$

So eq. (1) corresponds to the following equality of commutative diagrams:

$$\begin{array}{ccc} 1 & \xrightarrow{\lambda_A.\left(\left((\lambda_B.\eta)^* \times b\right);\text{ev}\right)} & T(A \times B)^A \xrightarrow{(-)^*} T(A \times B)^{TA} \\ \times & & \times \xrightarrow{\text{ev}} T(A \times B) \\ 1 & \xrightarrow[a]{} & TA \end{array}$$

$$\Big\Vert$$

$$\begin{array}{ccc} 1 & \xrightarrow{\lambda_B.\left(\left((\lambda_A.\eta)^* \times a\right);\text{ev}\right)} & T(A \times B)^B \xrightarrow{(-)^*} T(A \times B)^{TB} \\ \times & & \times \xrightarrow{\text{ev}} T(A \times B) \\ 1 & \xrightarrow[b]{} & TB \end{array}$$

Step by step, this unfolds as:

- $A \xrightarrow{\lambda_B.\eta} T(A \times B)^B$:

$$\begin{array}{l} A(g) \xrightarrow{(\lambda_B.\eta)_g} T(A \times B)^B(g) \\ x \longmapsto \mathfrak{y}^x_{g'} : \begin{cases} B(g') \times \not\perp(g, g') \longrightarrow T(A \times B)(g') \\ y,\ g \overset{\iota}{\hookrightarrow} g' \longmapsto \left(\lambda' \mapsto 1 \cdot \left| \left[(A(\iota)(x), y), !\right]_{g'} \right\rangle \right) \end{cases} \end{array}$$

- $A \xrightarrow{\lambda_B.\eta} T(A \times B)^B \xrightarrow{(-)^*} T(A \times B)^{TB}$:

$$
\begin{aligned}
&A(g) \xrightarrow{(\lambda_B.\eta)_g} [B \times \text{ɟ}(g),\, T(A \times B)] \xrightarrow{(-)^*_g} [TB \times \text{ɟ}(g),\, T(A \times B)] \\
&x \mapsto \mathfrak{y}^x \mapsto (\vartheta, g \overset{\iota}{\hookrightarrow} g') \overset{\mathfrak{y}^{x*}_{g'}}{\longmapsto} \lambda' \mapsto \text{let } \vartheta(\lambda') \;=\; \left\langle \overrightarrow{q} \;\middle|\; [y_h,\, \gamma^h]_{g'} \right\rangle_h \text{ in} \\
&\qquad\qquad \text{for each } h, \\
&\qquad\qquad\quad \text{let } \mathfrak{y}^x_h(y_h,\, g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\gamma^h \sqcup \lambda') \;= \\
&\qquad\qquad\qquad 1 \cdot \left| \left[(A(g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(x), y_h),\, ! \right]_h \right\rangle \text{ in} \\
&\qquad\qquad\quad \left\langle \overrightarrow{q} \;\middle|\; \underbrace{\mathbf{I}}_{\text{each line is a one-hot vector}} \;\middle|\; \left[(A(g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(x), y_h), \gamma^h \right]_{g'} \right\rangle_h
\end{aligned}
$$

- $$
\begin{array}{ccccc}
A & \xrightarrow{\lambda_B.\eta} & T(A \times B)^B & \xrightarrow{(-)^*} & T(A \times B)^{TB} \\
\times & & & & \times \\
1 & & \xrightarrow[b]{\qquad\qquad\qquad\qquad} & & TB
\end{array}
\xrightarrow{\quad\text{ev}\quad} T(A \times B) \;:
$$

$$
\begin{aligned}
&A(g) \times 1(g) \xrightarrow{(\lambda_B.\eta)^*_g \times b_g} [TB \times \text{ɟ}(g),\, T(A \times B)] \times TB(g) \xrightarrow{\text{ev}} T(A \times B)(g) \\
&x, * \mapsto \Big(\vartheta, g \overset{\iota}{\hookrightarrow} g' \overset{\mathfrak{y}^{x*}_{g'}}{\longmapsto} \lambda' \mapsto \text{let } \vartheta(\lambda') \;=\; \left\langle \overrightarrow{q} \;\middle|\; [y_h,\, \gamma^h]_{g'} \right\rangle_h \text{ in} \\
&\qquad\qquad\qquad \text{for each } h, \\
&\qquad\qquad\qquad\quad \text{let } \mathfrak{y}^x_h(y_h,\, g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\gamma^h \sqcup \lambda') \;= \\
&\qquad\qquad\qquad\qquad 1 \cdot \left| \left[(A(g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(x), y_h),\, ! \right]_h \right\rangle \text{ in} \\
&\qquad\qquad\qquad \left\langle \overrightarrow{q} \;\middle|\; \left[(A(g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(x), y_h), \gamma^h \right]_{g'} \right\rangle_h \Big), b_g(*) \\
&\qquad \mapsto \lambda \overset{\mathfrak{y}^{x*}_g(b_g(*),\, \mathrm{id}_g)}{\longmapsto} \quad \text{let } b_g(*)(\lambda) \;=\; \left\langle \overrightarrow{q} \;\middle|\; [y_h,\, \gamma^h]_{g'} \right\rangle_h \text{ in} \\
&\qquad\qquad\qquad\quad \text{for each } h,\;\; \text{let } \mathfrak{y}^x_h(y_h,\, g \hookrightarrow h)(\gamma^h \sqcup \lambda) \;= \\
&\qquad\qquad\qquad\qquad\qquad 1 \cdot \left| \left[(A(g \hookrightarrow h)(x), y_h),\, ! \right]_h \right\rangle \text{ in} \\
&\qquad\qquad\qquad\quad \left\langle \overrightarrow{q} \;\middle|\; \left[(A(g \hookrightarrow h)(x), y_h), \gamma^h \right]_g \right\rangle_h
\end{aligned}
$$

- $1 \xrightarrow{\lambda_A.\left(\left((\lambda_B.\eta)^* \times b\right);\mathrm{ev}\right)} T(A \times B)^A$:

$$1(g) \longrightarrow T(A \times B)^A(g)$$

$$* \longmapsto \varphi_{g'} : \begin{cases} A(g') \times \not\perp(g)(g') \longrightarrow T(A \times B(g') \\ x,\, \_ \longmapsto \lambda \overset{\mathfrak{y}^{x*}_g}{\longmapsto} \text{let } b_g(*)(\lambda) \;=\; \left\langle \overrightarrow{q} \;\middle|\; [y_h,\, \gamma^h]_g \right\rangle_h \text{ in} \\ \qquad\qquad\qquad\qquad \text{for each } h,\;\; \text{let } \mathfrak{y}^x_h(y_h,\, g \hookrightarrow h)(\gamma^h \sqcup \lambda) \;= \\ \qquad\qquad\qquad\qquad\qquad\qquad 1 \cdot \left|\left[(A(g \hookrightarrow h)(x), y_h),\, !\right]_h\right\rangle \text{ in} \\ \qquad\qquad\qquad\qquad \left\langle \overrightarrow{q} \;\middle|\; \left[(A(g \hookrightarrow h)(x), y_h), \gamma^h\right]_g \right\rangle_h \end{cases}$$

- $1 \xrightarrow{\lambda_A.\left(\left((\lambda_B.\eta)^* \times b\right);\mathrm{ev}\right)} T(A \times B)^A \xrightarrow{(-)^*} T(A \times B)^{TA}$:

$$1(g) \xrightarrow{\lambda_A.\left(\left((\lambda_B.\eta)^* \times b\right);\mathrm{ev}\right)_g} T(A \times B)^A(g) \xrightarrow{(-)^*_g} T(A \times B)^{TA}(g)$$

$$* \mapsto \varphi \mapsto (\vartheta, g \overset{\iota}{\hookrightarrow} g') \overset{\varphi^*_{g'}}{\longmapsto} \lambda' \;\mapsto\; \text{let } \vartheta(\lambda') \;=\; \left\langle \overrightarrow{p} \;\middle|\; [x_h,\, \lambda^h]_{g'} \right\rangle_h \text{ in}$$

$$\text{for each } h,$$

$$\text{let } \varphi_h(x_h,\, g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \;=$$

$$\left\langle \overrightarrow{q_h} \;\middle|\; \left[(A(h \hookrightarrow h')(x_h), \gamma^{h'}\right]_h \right\rangle_{h'} \text{ in}$$

$$\left\langle \overrightarrow{p} \;\middle|\; Q \;\middle|\; \left[(A(h \hookrightarrow h')(x_h), y_{h'}), \gamma^{h'} \sqcup \lambda^h\right]_{g'} \right\rangle_{h'}$$

- $$\begin{array}{ccccc} 1 & \xrightarrow{\lambda_A.\left(\left((\lambda_B.\eta)^* \times b\right);\mathrm{ev}\right)} & T(A\times B)^A & \xrightarrow{(-)^*} & T(A\times B)^{TA} \\ \times & & & & \times \xrightarrow{\mathrm{ev}} T(A\times B) \\ 1 & \xrightarrow[a]{} & TA & & \end{array} :$$

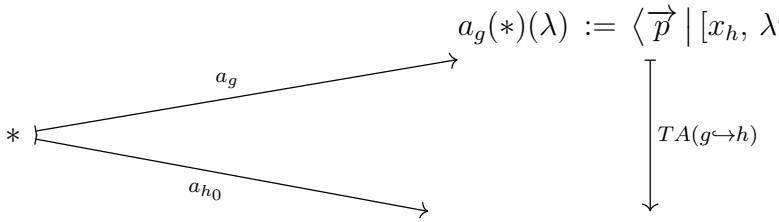
$$(1\times 1)(g) \xrightarrow{\left(\lambda_A.\left(\left((\lambda_B.\eta)^*\times b\right);\mathrm{ev}\right)_g;(-)^*_g\right)\times a_g} T(A\times B)^A(g)\times TA(g) \xrightarrow{\mathrm{ev}_g} T(A\times B)(g)$$

$$\begin{aligned}
, \mapsto \Big(\vartheta, \iota \xmapsto{\varphi^*_{g'}} \lambda' \mapsto\ & \text{let } \vartheta(\lambda') \;=\; \left\langle \overrightarrow{p} \;\middle|\; [x_h,\, \lambda^h]_{g'} \right\rangle_h \text{ in} \\
& \text{for each } h, \\
& \quad \text{let } \varphi_h(x_h,\, g \overset{\iota}{\hookrightarrow} g' \hookrightarrow h)(\lambda^h \sqcup \lambda') \;=\; \\
& \qquad \left\langle \overrightarrow{q_h} \;\middle|\; \left[(A(h\hookrightarrow h')(x_h), \gamma^{h'}\right]_h \right\rangle_{h'} \text{ in} \\
& \quad \left\langle \overrightarrow{p} \;\middle|\; Q \;\middle|\; \left[(A(h\hookrightarrow h')(x_h), y_{h'}), \gamma^{h'} \sqcup \lambda^h\right]_{g'} \right\rangle_{h'} \Big),\ a_g(*) \\
\mapsto \lambda \xmapsto{\varphi^*_g(a_g(*),\,\mathrm{id}_g)}\ & \text{let } a_g(*)(\lambda) \;=\; \left\langle \overrightarrow{p} \;\middle|\; [x_h,\, \lambda^h]_g \right\rangle_h \text{ in} \\
& \quad \text{for each } h, \\
& \qquad \text{let } \varphi_h(x_h,\, g \hookrightarrow h)(\lambda^h \sqcup \lambda) \;=\; \\
& \qquad\quad \left\langle \overrightarrow{q_h} \;\middle|\; \left[(A(h\hookrightarrow h')(x_h), y_{h'}), \gamma^{h'}\right]_h \right\rangle_{h'} \text{ in} \\
& \quad \left\langle \overrightarrow{p} \;\middle|\; Q \;\middle|\; \left[(A(h\hookrightarrow h')(x_h), y_{h'}), \gamma^{h'} \sqcup \lambda^h\right]_g \right\rangle_{h'} \\
\mapsto \lambda \xmapsto{\varphi^*_g(a_g(*),\,\mathrm{id}_g)}\ & \text{let } a_g(*)(\lambda) \;=\; \left\langle \overrightarrow{p} \;\middle|\; [x_h,\, \lambda^h]_g \right\rangle_h \text{ in} \\
& \quad \text{for each } h, \\
& \qquad \text{let } b_h(*)(\lambda^h \sqcup \lambda) \;=\; \left\langle \overrightarrow{q_h} \;\middle|\; [y_{h'},\, \gamma^{h'}]_h \right\rangle_{h'} \text{ in} \\
& \qquad \text{for each } h', \\
& \qquad\quad \text{let } \mathfrak{n}^{x_h}_{h'}(y_{h'}, h\hookrightarrow h')(\gamma^{h'} \sqcup \lambda^h \sqcup \lambda) \;=\; \\
& \qquad\qquad 1 \cdot \left|\left[(A(h\hookrightarrow h')(x_h), y_{h'}), !\right]_{h'}\right\rangle \text{ in} \\
& \quad \left\langle \overrightarrow{p} \;\middle|\; Q \;\middle|\; \left[(A(h\hookrightarrow h')(x_h), y_{h'}), \gamma^{h'} \sqcup \lambda^h\right]_g \right\rangle_{h'}
\end{aligned}$$

Commutativity then boils down to this last map $\varphi^*_g(a_g(*),\, \mathrm{id}_g)$ being equal to:

$$\lambda \xmapsto{\psi_g^*(b_g(*),\,\mathrm{id}_g)} \text{let } b_g(*)(\lambda) \;=\; \left\langle \overrightarrow{q} \;\middle|\; [y_h,\, \gamma^h]_g \right\rangle_h \text{ in}$$
$$\text{for each } h,$$
$$\text{let } a_h(*)(\gamma^h \sqcup \lambda) \;=\; \left\langle \overrightarrow{p_h} \;\middle|\; [x_{h'},\, \lambda^{h'}]_h \right\rangle_{h'} \text{ in}$$
$$\text{for each } h',$$
$$\text{let } \mathfrak{y}_{h'}^{y_h}(x_{h'}, h \hookrightarrow h')(\lambda^{h'} \sqcup \gamma^h \sqcup \lambda) \;=\; 1 \cdot \left| \left[(x_{h'}, B(h \hookrightarrow h')(y_h)), ! \right]_{h'} \right\rangle \text{ in}$$
$$\left\langle \overrightarrow{q} \;\middle|\; P \;\middle|\; \left[(x_{h'}, B(h \hookrightarrow h')(y_h)), \lambda^{h'} \sqcup \gamma^h \right]_g \right\rangle_{h'}$$

And they turn out to be equal indeed: by naturality of $a\colon 1 \to TA$, $b\colon 1 \to TB$, we have, for every h_0 in the coslice category $g/\mathbf{BiGrph}_{emb}$:



and likewise for b. The equality $\langle \overrightarrow{p_{h_0}} \mid [x_{h'}, \lambda^{h'}]_{h_0} \rangle_{h'} = \left\langle \overrightarrow{p} \;\middle|\; [A(h \hookrightarrow h_0 \coprod_g h)(x_h), \lambda^h]_{h_0} \right\rangle_h$ implies that h' and h range over the same index set.

Therefore, if $b_g(*)(\lambda) := \left\langle \overrightarrow{q} \;\middle|\; [y_{h'}, \gamma^{h'}]_g \right\rangle_{h'}$, the map $\varphi_g^*(a_g(*), \mathrm{id}_g)$ can be written as:

$$\lambda \mapsto \text{let } a_g(*)(\lambda) \;=\; \left\langle \overrightarrow{p} \;\middle|\; [x_h, \lambda^h]_g \right\rangle_h \text{ in}$$
$$\text{for each } h,$$
$$\text{let } b_h(*)(\lambda^h \sqcup \lambda) \;=\; \left\langle \overrightarrow{q} \;\middle|\; [B(h' \hookrightarrow h \coprod_g h')(y_{h'}),\, \gamma^{h'}]_h \right\rangle_{h'} \text{ in}$$
$$\text{for each } h',$$
$$\text{let } \mathfrak{y}_{h \coprod_g h'}^{x_h}(B(h' \hookrightarrow h \coprod_g h')(y_{h'}), h \hookrightarrow h \coprod_g h')(\gamma^{h'} \sqcup \lambda^h \sqcup \lambda) \;=$$
$$1 \cdot \left| \left[(A(h \hookrightarrow h \coprod_g h')(x_h), B(h' \hookrightarrow h \coprod_g h')(y_{h'})), ! \right]_{h'} \right\rangle \text{ in}$$
$$\left\langle \overrightarrow{p} \;\middle|\; \mathbf{I} \otimes \langle \overrightarrow{q} | \;\middle|\; \left[(A(h \hookrightarrow h \coprod_g h')(x_h), B(h' \hookrightarrow h \coprod_g h')(y_{h'})), \gamma^{h'} \sqcup \lambda^h \right]_g \right\rangle_{h'}$$

where

$$\mathbf{I} \otimes \langle \overrightarrow{q} | \;:=\; \begin{pmatrix} \langle \overrightarrow{q} | & & \\ & \ddots & \mathbf{0} \\ \mathbf{0} & & \langle \overrightarrow{q} | \end{pmatrix}$$

is the Kronecker product, so that $\langle \overrightarrow{p} \rangle \, (\mathbf{I} \otimes \langle \overrightarrow{q} |)$ is the row vector $\begin{pmatrix} p_{h_1} \cdot \langle \overrightarrow{q} | & \cdots & p_{h_n} \cdot \langle \overrightarrow{q} | \end{pmatrix}$. Similarly, if $a_g(*)(\lambda) := \left\langle \overrightarrow{p} \,\middle|\, [x_h,\, \lambda^h]_g \right\rangle_h$, the map $\psi_g^*(b_g(*),\, \mathrm{id}_g)$ can be written as:

$$\begin{array}{l} \lambda \mapsto \text{let } b_g(*)(\lambda) \;=\; \left\langle \overrightarrow{q} \,\middle|\, [y_{h'},\, \gamma^{h'}]_g \right\rangle_{h'} \text{ in} \\ \qquad \text{for each } h', \\ \qquad\quad \text{let } a_{h'}(*)(\gamma^{h'} \sqcup \lambda) \;=\; \left\langle \overrightarrow{p} \,\middle|\, [A(h \hookrightarrow h' \coprod_g h)(x_h),\, \lambda^h]_{h'} \right\rangle_h \text{ in} \\ \qquad\qquad \text{for each } h, \\ \qquad\qquad\quad \text{let } \mathfrak{y}^{y_{h'}}_{h' \coprod_g h}(A(h \hookrightarrow h' \coprod_g h)(x_h), h' \hookrightarrow h' \coprod_g h)(\lambda^h \sqcup \gamma^{h'} \sqcup \lambda) \;= \\ \qquad\qquad\qquad\qquad\qquad\qquad 1 \cdot \left| \left[(A(h \hookrightarrow h' \coprod_g h)(x_h), B(h' \hookrightarrow h' \coprod_g h)(y_{h'})), ! \right]_h \right\rangle \text{ in} \\ \left\langle \begin{pmatrix} q_{h'_1} \cdot \langle \overrightarrow{p} | & \cdots & q_{h'_m} \cdot \langle \overrightarrow{p} | \end{pmatrix} \,\middle|\, \left[(A(h \hookrightarrow h' \coprod_g h)(x_h), B(h' \hookrightarrow h' \coprod_g h)(y_{h'})), \lambda^h \sqcup \gamma^{h'} \right]_g \right\rangle_h \end{array}$$

But the equivalence class $\left[(A(h \hookrightarrow h' \coprod_g h)(x_h), B(h' \hookrightarrow h' \coprod_g h)(y_{h'})), \lambda^h \sqcup \gamma^{h'} \right]_g$ has probabilistic weight $p_h q_{h'} = q_{h'} p_h$ in both formal convex sums (in the first ket, the index ranges of over the h''s for each h, and in the second ket, it ranges over the h's for each h'), which yields the result. □

2.5 Affineness

Lemma 2.13. *Let X be a constant presheaf on the coslice category $g/\mathbf{BiGrph}_{emb}$,* i.e. *there exists a set S_0 such that*

$$X(g \overset{\iota}{\hookrightarrow} h) = S_0 \xrightarrow{\mathrm{id}} S_0$$

for every $g \overset{\iota}{\hookrightarrow} h \in g/\mathbf{BiGrph}_{emb}$. Then

$$T(X)(g) \cong P_{\mathrm{f}}(S_0)^{[0,1]^{g_L}}$$

Proof. This results from a small calculation:

$$T(X)(g) := \left(P_{\mathrm{f}} \int^{g \hookrightarrow h} X(h) \times [0,1]^{(h-g)_L} \right)^{[0,1]^{g_L}} = \left(P_{\mathrm{f}} \int^{g \hookrightarrow h} S_0 \times [0,1]^{(h-g)_L} \right)^{[0,1]^{g_L}}$$

But as the functor $S_0 \times -$ is left adjoint to $S_0 \Rightarrow -$ in **Set**, it is cocontinuous, so it preserves coends, and:

$$\int^{g \hookrightarrow h} S_0 \times [0,1]^{(h-g)_L} \cong S_0 \times \underbrace{\int^{g \hookrightarrow h} [0,1]^{(h-g)_L}}_{= \operatorname{colim}_{g \hookrightarrow h} [0,1]^{(h-g)_L}}$$

But $\operatorname{colim}_{g \hookrightarrow h} [0,1]^{(h-g)_L} \cong \{*\}$. Indeed, $h = g$ is initial in $g/\mathbf{BiGrph}_{emb}$, so it is terminal in the essential image of $[0,1]^{((-)-g)_L}$ (by precomposition), and the resulting colimit is $[0,1]^{(g-g)_L} \cong [0,1]^{\emptyset} \cong \{*\}$. Therefore, $\int^{g \hookrightarrow h} S_0 \times [0,1]^{(h-g)_L} \cong S_0$ and the result follows. □

Lemma 2.14 (Affineness). *T is affine,* i.e. $T(1) \cong 1$, *where* 1 *is the terminal object of* $[\mathbf{BiGrph}_{emb}, \mathbf{Set}]$.

Proof. By lemma 2.13, for every $g \in \mathbf{BiGrph}_{emb}$: $T(1) \cong P_{\mathrm{f}}(\{*\})^{[0,1]^{g_L}} \cong \{*\}$. □

2.6 Denotation of computation judgements

In our language, the denotational interpretation of values, computations (return and let binding), and matching (elimination of bool's and product types) is standard.

We interpret computation judgements $\Gamma \vdash t : A$ as morphisms $[\![\Gamma]\!] \to T([\![A]\!])$, by induction on the structure of typing derivations. The context Γ is built of bool's, $\mathbb{A}$ and $\mathbb{F}$ and products. Therefore, $[\![\Gamma]\!]$ is isomorphic to $2^k \times \mathbf{BiGrph}_{emb}(\circ, -)^\ell \times \mathbf{BiGrph}_{emb}(\bullet, -)^m$.

Remark 2.15. Note that $\mathbf{BiGrph}_{emb}$ is not cocomplete (morphisms are embeddings), but seen as a subcategory of the category $\mathbf{Grph}$ of graphs: $\mathbf{BiGrph}_{emb}(\circ, g)^\ell \times \mathbf{BiGrph}_{emb}(\bullet, g)^m \cong \mathbf{Grph}(\ell \circ + m\bullet, g)$, and every map in the latter homset factors through an embedding $g_0 \hookrightarrow g$ in $\mathbf{BiGrph}_{emb}$ (epi-mono factorization). However, such a bigraph g_0 cannot be chosen uniformly for all elements in $\mathbf{BiGrph}_{emb}(\circ, g)^\ell \times \mathbf{BiGrph}_{emb}(\bullet, g)^m$ (it depends on each element), so $[\![\Gamma]\!](g)$ cannot be written as $2^k \times \mathbf{BiGrph}_{emb}(g_0, g)$ for some bigraph g_0 in general.

Definition 2.16. For every bigraph g, we denote by R_g (resp. L_g) the set of bigraphs $h \in g/\mathbf{BiGrph}_{emb}$ having one more right (resp. left) node than g, and that are the same otherwise.

$$R_g := \{\, h \in \mathbf{BiGrph}_{emb} \mid h_L = g_L,\ g_R \subseteq h_R \text{ and } |h_R| = |g_R| + 1 \,\}$$
$$L_g := \{\, h \in \mathbf{BiGrph}_{emb} \mid h_R = g_R,\ g_L \subseteq h_L \text{ and } |h_L| = |g_L| + 1 \,\}$$

Remark 2.17. Note that $|R_g| = 2^{|g_L|} < \infty$, $|L_g| = 2^{|g_R|} < \infty$ (all possible edges between the new node and the opposite nodes) since g is finite.

Proposition 2.18. *A computation of type*

- $\mathbb{A}$ *returns the label of an already existing atom or a fresh one with its connections to the already existing functions:*

$$T(\llbracket \mathbb{A} \rrbracket)(g) \;\cong\; P_{\mathrm{f}}(g_R + 2^{g_L})^{[0,1]^{g_L}}$$

- $\mathbb{F}$ *returns the label of an already existing function or create a new function with its connections to already existing atoms and a fixed probabilistic bias:*

$$T(\llbracket \mathbb{F} \rrbracket)(g) \;\cong\; P_{\mathrm{f}}(g_L + 2^{g_R} \times [0,1])^{[0,1]^{g_L}}$$

Proof. Let us show that, for every $g \in \mathbf{BiGrph}_{emb}$:

$$\int^{g \hookrightarrow h} \mathbf{BiGrph}_{emb}(\bullet, h) \times [0,1]^{(h-g)_L} \cong g_R + 2^{g_L}$$
$$\int^{g \hookrightarrow h} \mathbf{BiGrph}_{emb}(\circ, h) \times [0,1]^{(h-g)_L} \cong g_L + 2^{g_R} \times [0,1]$$

Remark 2.19. Note that we cannot straightforwardly use the co-Yoneda lemma as-is, since $\circ$ and $\bullet$ may not be in $g/\mathbf{BiGrph}_{emb}$.

Let $\odot \in \{\circ, \bullet\}$. We have

$$\int^{g \hookrightarrow h} \mathbf{BiGrph}_{emb}(\odot, h) \times [0,1]^{(h-g)_L} \cong \left\{ (g \overset{\iota}{\hookrightarrow} h,\ \odot \overset{\kappa}{\hookrightarrow} h,\ \lambda^h) \;\middle|\; g \overset{\iota}{\hookrightarrow} h \in g/\mathbf{BiGrph}_{emb} \right\} \Big/ \sim$$

There are two cases:

- either $\odot \overset{\kappa}{\hookrightarrow} h = \odot \overset{\iota'}{\hookrightarrow} g \overset{\iota}{\hookrightarrow} h$ factors through $g \overset{\iota}{\hookrightarrow} h$. It then follows that

$$\left[(g \overset{\iota}{\hookrightarrow} h,\ \odot \overset{\kappa}{\hookrightarrow} h,\ \lambda^h)\right] = \left[(g \overset{\mathrm{id}}{\hookrightarrow} g,\ \odot \overset{\iota'}{\hookrightarrow} g,\ !)\right]$$

- or $\odot$ is mapped to a node of $(h - g)$, so that κ factors through a graph $h_0 \in L_g$ (if $\odot = \circ$) or $h_0 \in R_g$ (if $\odot = \bullet$). We write $\odot \overset{\kappa}{\hookrightarrow} h = \odot \overset{\iota'}{\hookrightarrow} h_0 \hookrightarrow h$. In this case, by denoting ι'' the corestriction of ι to h_0, we have:

$$\left[(g \overset{\iota}{\hookrightarrow} h,\ \odot \overset{\kappa}{\hookrightarrow} h,\ \lambda^h)\right] = \left[\left(g \overset{\iota''}{\hookrightarrow} h_0,\ \odot \overset{\iota'}{\hookrightarrow} h_0,\ \begin{cases} p \in [0,1]^{\circ} \cong [0,1] & \text{if } \odot = \circ \\ ! \in [0,1]^{\emptyset} & \text{else if } \odot = \bullet \end{cases}\right)\right]$$

□

2.7 Denotation of language-specific constructs

Denotation of $\Gamma \vdash \mathsf{fresh}() : \mathbb{A}$

The map $[\![\mathsf{fresh}]\!]_g : [\![\Gamma]\!](g) \to T([\![\mathbb{A}]\!])(g)$ randomly chooses connections to each left node according to the state of biases, and makes a fresh right node with those connections.

$$[\![\mathsf{fresh}]\!]_g : \begin{cases} 2^k \times \mathbf{BiGrph}_{emb}(\circ, g)^\ell \times \mathbf{BiGrph}_{emb}(\bullet, g)^m \longrightarrow P_{\mathrm{f}}(g_R + 2^{g_L})^{[0,1]^{g_L}} \\ \_, \_, \_ \mapsto \lambda \mapsto \left\langle \dfrac{1}{Z} \displaystyle\prod_{\mathsf{f} \in g_L} \lambda(\mathsf{f})^{E^h(\mathsf{f}, a_h(\bullet))} (1 - \lambda(\mathsf{f}))^{1 - E^h(\mathsf{f}, a_h(\bullet))} \,\middle|\, [\underbrace{\bullet}_{\cong (h-g)_R} \overset{a_h}{\hookrightarrow} h, !]_g \right\rangle_{h \in R_g} \end{cases}$$

where Z is a normalisation constant.

Remark 2.20. It is enough to consider these h's by garbage collection of the coend.

Denotation of $\Gamma \vdash \lambda_{\text{מ}}\, x.\, u : \mathbb{F}$

As $\lambda_{\text{מ}}$-abstractions are formed based on computation judgements of the form $\Gamma, x : \mathbb{A} \vdash u : \mathsf{bool}$, we first note, by lemma 2.13, that

$$T([\![\mathsf{bool}]\!])g \;\cong\; P_{\mathrm{f}}(2)^{[0,1]^{g_L}} \;\cong\; [0,1]^{[0,1]^{g_L}}$$

Also, we can decompose the extra variable x in the environment $\Gamma,\ x : \mathbb{A}$, the denotation of which is of the form $[\![\Gamma,\ x : \mathbb{A}]\!](g) \;=\; 2^k \times \mathbf{BiGrph}_{emb}(\circ, g)^\ell \times \mathbf{BiGrph}_{emb}(\bullet, g)^m \;\times\; \mathbf{BiGrph}_{emb}(\bullet,\ g)$ for a bigraph $g \in \mathbf{BiGrph}_{emb}$.

Now, the extra part x is a right node, and its valuation will either be a node already in the graph described in the rest of the environment, or a new one with particular edges to the rest of the environment. The argument u can test (if it wants) what kind of node x is, before returning a probability.

As a result, if

$$[\![u]\!]_g : 2^k \times \mathbf{BiGrph}_{emb}(\circ, g)^\ell \times \mathbf{BiGrph}_{emb}(\bullet, g)^m \times \mathbf{BiGrph}_{emb}(\bullet, g) \longrightarrow [0,1]^{[0,1]^{g_L}}$$

the denotation $[\![u]\!]$ gives us the edge probability of the left node that we need to generate, both to the existing right nodes, and to any future right nodes (which needs to be remembered). This can be formalized into a natural transformation $[\![\lambda_{\text{מ}}\, x.\, u]\!] : [\![\Gamma]\!] \to T([\![\mathbb{F}]\!])$.

$$[\![\lambda_{\text{מ}}\, x.u]\!]_g : \begin{cases} 2^k \times \mathbf{BiGrph}_{emb}(\circ, g)^\ell \times \mathbf{BiGrph}_{emb}(\bullet, g)^m \longrightarrow P_{\mathrm{f}}(g_L + 2^{g_R} \times [0,1])^{[0,1]^{g_L}} \\ b^k,\ \begin{matrix} (\circ \overset{\kappa_i}{\hookrightarrow} g)_i, \\ (\bullet \overset{\tau_j}{\hookrightarrow} g)_j \end{matrix} \mapsto \lambda \mapsto \left\langle \dfrac{1}{Z} \displaystyle\prod_{a \in g_R} p_a^{E^h(\mathsf{f}_h(\circ), a)} (1 - p_a)^{1 - E^h(\mathsf{f}_h(\circ), a)} \,\middle|\, [\underbrace{\circ}_{\cong (h-g)_L} \overset{\mathsf{f}_h}{\hookrightarrow} h, \_ \mapsto \tilde{p}]_g \right\rangle_{h \in L_g} \end{cases}$$

where

- Z is a normalization constant.
- for every $a \in g_R$, $\quad p_a \;:=\; [\![u]\!]_g\big(b^k,\, (\circ \overset{\kappa_i}{\hookrightarrow} g)_i,\, (\bullet \overset{\tau_j}{\hookrightarrow} g)_j,\, \bullet \overset{a}{\hookrightarrow} g,\, \lambda\big)$
- $\tilde{p} \;:=\; [\![u]\!]_g\big(b^k,\, (\circ \overset{\kappa_i}{\hookrightarrow} g \overset{\iota_1}{\hookrightarrow} g + \bullet)_i,\, (\bullet \overset{\tau_j}{\hookrightarrow} g \overset{\iota_1}{\hookrightarrow} g + \bullet)_j,\, \bullet \overset{\iota_2}{\hookrightarrow} g + \bullet,\, \lambda\big)$ where $\iota_1,\, \iota_2$ are the coprojections.

3 Operational semantics

We now give an operational semantics to our language.

Henceforth, we fix a countable set of variables $x, y, z, \ldots \in \mathsf{Var}$, and consider the terms up to α-equivalence for the $\lambda_{\text{בּ}}$ operator.

Definition 3.1 (Partial bigraph)**.** A partial bigraph $\mathfrak{g} \;:=\; (\mathfrak{g}_L, \mathfrak{g}_R, E)$ is a bipartite graph where the edge relation $E\colon \mathfrak{g}_L \times \mathfrak{g}_R \to \{\mathsf{true}, \mathsf{false}, \bot\}$ is either true, false or undefined ($\bot$) on each pair of left and right nodes $(\mathfrak{f}, a) \in \mathfrak{g}_L \times \mathfrak{g}_R$.

Remark 3.2. By abuse of notation, syntactic truth values will be conflated with semantic ones.

Notation 3.3. For a partial graph $\mathfrak{g}$, $E(\mathfrak{f}, a) = \beta \in \{\mathsf{true}, \mathsf{false}, \bot\}$ will be written $\mathfrak{f} \overset{\beta}{\longrightarrow} a$.

3.1 Extended expressions

We introduce extended expressions, which are typed computations decorated with memoization reduction contexts $\{\!\{-\}\!\}$. In the following, $\Delta \in (\mathfrak{g}_L \times \mathfrak{g}_R)^{(\mathbb{N})}$ is a finite stack of function label-atom label pairs, indicating that we are currently computing the result of these functions at these atoms for the first time.

$$\frac{\Gamma \vdash^{\mathsf{c}} u : A}{\Gamma \mid \emptyset \vdash^{\mathsf{c}} u : A} \qquad \frac{\Gamma \mid \Delta \vdash^{\mathsf{c}} u : A}{\Gamma \mid (\mathfrak{f}, a), \Delta \vdash^{\mathsf{c}} \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma} : A}\ (\mathfrak{f}, a) \notin \Delta$$

$$\frac{\Gamma \mid \Delta_1 \vdash^{\mathsf{c}} u : A \qquad \Gamma, x : A \mid \Delta_2 \vdash^{\mathsf{c}} t : B}{\Gamma \mid \Delta_1, \Delta_2 \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \leftarrow u\ \mathsf{in}\ t : B}$$

$$\frac{\Gamma \mid \Delta_1 \vdash^{\mathsf{c}} u : A \qquad \Gamma, x : A \mid \Delta_2 \vdash^{\mathsf{c}} t : B}{\Gamma \mid (\mathfrak{f}, a), \Delta_1, \Delta_2 \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \leftarrow \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma}\ \mathsf{in}\ t : B}\ (\mathfrak{f}, a) \notin \Delta_1 \cup \Delta_2$$

$$\frac{\Gamma \mid \Delta_1 \vdash^{\mathsf{c}} u : A \qquad \Gamma, x : A \mid \Delta_2 \vdash^{\mathsf{c}} t : B}{\Gamma \mid (\mathfrak{f}, a), \Delta_1, \Delta_2 \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \leftarrow u\ \mathsf{in}\ \{\!\{t\}\!\}^{\mathfrak{f},a}_{\gamma} : B}\ (\mathfrak{f}, a) \notin \Delta_1 \cup \Delta_2$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \mathsf{bool} \qquad \Gamma \mid \Delta_1 \vdash^{\mathsf{c}} u : A \qquad \Gamma \mid \Delta_2 \vdash^{\mathsf{c}} t : A}{\Gamma \mid \Delta_1, \Delta_2 \vdash^{\mathsf{c}} \mathsf{if}\ v\ \mathsf{then}\ u\ \mathsf{else}\ t : A}$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \mathsf{bool} \qquad \Gamma \mid \Delta_1 \vdash^{\mathsf{c}} u : A \qquad \Gamma \mid \Delta_2 \vdash^{\mathsf{c}} t : A}{\Gamma \mid (\mathfrak{f}, a), \Delta_1, \Delta_2 \vdash^{\mathsf{c}} \mathsf{if}\ v\ \mathsf{then}\ \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma}\ \mathsf{else}\ t : A}\ (\mathfrak{f}, a) \notin \Delta_1 \cup \Delta_2$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : \mathsf{bool} \qquad \Gamma \mid \Delta_1 \vdash^{\mathsf{c}} u : A \qquad \Gamma \mid \Delta_2 \vdash^{\mathsf{c}} t : A}{\Gamma \mid (\mathfrak{f}, a), \Delta_1, \Delta_2 \vdash^{\mathsf{c}} \mathsf{if}\ v\ \mathsf{then}\ u\ \mathsf{else}\ \{\!\{t\}\!\}^{\mathfrak{f},a}_{\gamma} : A}\ (\mathfrak{f}, a) \notin \Delta_1 \cup \Delta_2$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : A \times B \qquad \Gamma, x : A, y : B \mid \Delta \vdash^{\mathsf{c}} t : C}{\Gamma \mid \Delta \vdash^{\mathsf{c}} \mathsf{match}\ v\ \mathsf{as}\ (x, y)\ \mathsf{in}\ t : C}$$

$$\frac{\Gamma \vdash^{\mathsf{v}} v : A \times B \qquad \Gamma, x : A, y : B \mid \Delta \vdash^{\mathsf{c}} t : C}{\Gamma \mid (\mathfrak{f}, a), \Delta \vdash^{\mathsf{c}} \mathsf{match}\ v\ \mathsf{as}\ (x, y)\ \mathsf{in}\ \{\!\{t\}\!\}^{\mathfrak{f},a}_{\gamma} : C}\ (\mathfrak{f}, a) \notin \Delta$$

3.2 Configurations

Definition 3.4. If S is a finite set, $\mathsf{Tree}(S) \cong \sum_{n \geq 1} C_n \, S^n$ (where C_n is the n-th Catalan number) denotes the set of all possible non-empty trees with internal nodes the cartesian product and leaf nodes taken in S.

Example 3.5. *If* $S := \{s_1, s_2\}$, *then* $s_1, (s_2, s_1), (s_1, (s_1, s_2)), \ldots \in \mathsf{Tree}(S)$.

Definition 3.6 (Set-theoretic denotation of contexts for a partial bigraph)**.** Let $\mathfrak{g}$ be a partial bigraph. The set-theoretic denotation $(\!|-|\!)$ is defined as $(\!|\mathsf{bool}|\!) := 2 \cong \{\mathsf{true}, \mathsf{false}\}$, $(\!|\mathbb{F}|\!) := \mathfrak{g}_L$, $(\!|\mathbb{A}|\!) := \mathfrak{g}_R$ and $(\!|-|\!)$ is readily extended to every context Γ.

Moreover, in the following, $\gamma \in (\!|\Gamma|\!) \subseteq \mathsf{Tree}(2 + \mathfrak{g}_L + \mathfrak{g}_R)^{\mathsf{Var}}$ denotes a context value.

Example 3.7.

$$\Gamma = (x : \mathsf{bool}, y : \mathbb{F}, z : ((\mathbb{F} \times 2) \times \mathbb{A}))$$
$$(\!|\Gamma|\!) = \{x \mapsto 2, y \mapsto \mathfrak{g}_L, z \mapsto ((\mathfrak{g}_L \times 2) \times \mathfrak{g}_R)$$
$$\gamma = \{x \mapsto \mathsf{true}, y \mapsto \mathfrak{f}_0, z \mapsto ((\mathfrak{f}_1, \mathsf{true}), a_0)\}$$

Values v and Terminal computations r:

$$v ::= x \mid (v, v) \mid \mathsf{true} \mid \mathsf{false}$$
$$r ::= \mathsf{return}(v) \mid \lambda_{\text{מ}} x.\, u$$

Redexes:

$$\begin{aligned} \rho ::= {} & \mathsf{let\ val}\ x \leftarrow r\ \mathsf{in}\ u \mid \{\!\{\mathsf{return}(v)\}\!\}^{\mathfrak{f},a}_{\gamma} \qquad \text{where } \mathfrak{f} \in \mathfrak{g}_L,\ a \in \mathfrak{g}_R,\ \gamma \in \mathsf{Tree}(2 + \mathfrak{g}_L + \mathfrak{g}_R)^{\mathsf{Var}} \\ & \mid \mathsf{match}\ v\ \mathsf{as}\ (x, y)\ \mathsf{in}\ t \mid \mathsf{if}\ v\ \mathsf{then}\ t\ \mathsf{else}\ u \\ & \mid \mathsf{flip}() \mid \mathsf{fresh}() \mid (v = w) \mid (v @ w) \end{aligned}$$

Reduction contexts:

$$\mathcal{C}[-] ::= [-] \mid \mathsf{let\ val}\ x \leftarrow \mathcal{C}[-]\ \mathsf{in}\ u \mid \{\!\{\mathcal{C}[-]\}\!\}^{\mathfrak{f},a}_{\gamma}$$

Configurations:

$$(\gamma, u, \mathfrak{g}, \lambda)$$

where

- $\gamma \in \mathsf{Tree}(2 + \mathfrak{g}_L + \mathfrak{g}_R)^{\mathsf{Var}}$ is a context value.
- u is an extended expression $\Gamma \mid \Delta \vdash^{\complement} u : A$.
- $\mathfrak{g} = (\mathfrak{g}_L, \mathfrak{g}_R, E)$ is a partial graph.
- $\lambda \colon \mathfrak{g}_L \to \mathsf{Closures}$, where $\mathsf{Closures} := \left\{ (\lambda_{\text{מ}}\, x.\, u, \gamma) \mid \Gamma \vdash^{\complement} \lambda_{\text{מ}}\, x.\, u : \mathbb{F} \text{ and } \gamma \in (\!|\Gamma|\!) \right\}$

3.3 Reduction rules

Let $(\!|-|\!)_\gamma$ be the function evaluating a value in a context value γ and Ty be the one returning the type of such evaluations:

$$
\begin{array}{ll}
(\!|-|\!)_\gamma \colon \mathsf{Values} \to \mathsf{Tree}(2 + \mathfrak{g}_L + \mathfrak{g}_R) & \mathrm{Ty} \colon \mathsf{Tree}(2 + \mathfrak{g}_L + \mathfrak{g}_R) \longrightarrow \mathsf{Tree}(\{\mathsf{bool}, \mathbb{A}, \mathbb{F}\}) \\
(\!|x|\!)_\gamma = \gamma(x) & \mathrm{Ty}(\mathsf{true}) = \mathsf{bool} = \mathrm{Ty}(\mathsf{false}) \\
(\!|\mathsf{true}|\!)_\gamma = \mathsf{true} \qquad (\!|\mathsf{false}|\!)_\gamma = \mathsf{false} & \mathrm{Ty}(a) = \mathbb{A} \quad \forall a \in \mathfrak{g}_R \qquad \mathrm{Ty}(\mathfrak{f}) = \mathbb{F} \quad \forall \mathfrak{f} \in \mathfrak{g}_L \\
(\!|(v, w)|\!)_\gamma = ((\!|v|\!)_\gamma, (\!|w|\!)_\gamma) & \mathrm{Ty}((s, t)) = \mathrm{Ty}(s) \times \mathrm{Ty}(t)
\end{array}
$$

We will also make use of the function Ctxt, which associates to every context value γ its corresponding context:

$$
\mathrm{Ctxt} \colon \begin{cases} \mathsf{Tree}(2 + \mathfrak{g}_L + \mathfrak{g}_R)^{\mathsf{Var}} & \longrightarrow \mathsf{Contexts} \\ \gamma & \longmapsto \big(x : \mathrm{Ty}(\gamma(x))\big)_{x \,\in\, \mathrm{supp}(\gamma)} \end{cases}
$$

where $\mathrm{supp}(\gamma)$ is the support/domain of γ.

Reduction rules:

$$(\gamma, \mathsf{let\ val}\ x\ \leftarrow\ \mathsf{return}(v)\ \mathsf{in}\ u, \mathfrak{g}, \lambda) \longrightarrow (\gamma \sqcup \{x \mapsto (\!|v|\!)_\gamma\}, u, \mathfrak{g}, \lambda)$$

$$(\gamma, \{\!\{\mathsf{return}(v)\}\!\}_{\gamma'}^{(\mathfrak{f},a)}, \mathfrak{g}, \lambda) \longrightarrow (\gamma', \mathsf{return}((\!|v|\!)_\gamma), (\mathfrak{g}_L, \mathfrak{g}_R, E \cup \{\mathfrak{f} \xrightarrow{(\!|v|\!)_\gamma} a\}), \lambda)$$

$$\begin{aligned}(\gamma, \mathsf{let\ val}\ x\ \leftarrow\ \lambda_{\text{ᵐ}} y.\ u\ \mathsf{in}\ t, \mathfrak{g}, \lambda) \longrightarrow \big(&\gamma \sqcup \{x \mapsto \mathfrak{f}\}, t,\\ &(\mathfrak{g}_L \sqcup \{\mathfrak{f}\}, \mathfrak{g}_R, E \sqcup \{f \xrightarrow{\perp} a\}_{a \in \mathfrak{g}_R}),\\ &\lambda \sqcup \{\mathfrak{f} \mapsto (\lambda_{\text{ᵐ}} y.\ u, \gamma)\}\big)\end{aligned}$$

$$(\gamma, (v@w), \mathfrak{g}, \lambda) \longrightarrow \begin{cases} (\gamma, \mathsf{return}(\beta), \mathfrak{g}, \lambda) & \text{if } \beta := E(\gamma(v), \gamma(w)) \neq \perp \\ (\gamma_0 \sqcup \{y \mapsto (\!|w|\!)_\gamma\}, \{\!\{u\}\!\}_\gamma^{\mathfrak{f},a}, \mathfrak{g}, \lambda) & \text{else,} \\ & \text{where } \lambda(\gamma(v)) := (\lambda_{\text{ᵐ}} y.\ u, \gamma_0) \end{cases}$$

$$(\gamma, v = w, \mathfrak{g}, \lambda) \longrightarrow (\gamma, \mathsf{return}(\beta), \mathfrak{g}, \lambda) \text{ where } \beta := (\gamma(v) = \gamma(w))$$

$$\begin{aligned}(\gamma, \mathsf{fresh}(), \mathfrak{g}, \lambda) \longrightarrow \big(&\gamma \sqcup \{x \mapsto a\}, \mathsf{return}(x),\\ &(\mathfrak{g}_L, \mathfrak{g}_R \sqcup \{a\}, \mathfrak{g}_R, E \sqcup \{\mathfrak{f} \xrightarrow{\perp} a\}_{\mathfrak{f} \in \mathfrak{g}_L}), \lambda\big)\end{aligned}$$

$$(\gamma, \mathsf{flip}(), \mathfrak{g}, \lambda) \overset{\text{with proba } \frac{1}{2}}{\longrightarrow} (\gamma, \mathsf{return}(\beta), g, \lambda) \quad \text{where } \beta \in \{\mathsf{true}, \mathsf{false}\}$$

$$(\gamma, \mathsf{match}\ v\ \mathsf{as}\ (x, y)\ \mathsf{in}\ t, \mathfrak{g}, \lambda) \longrightarrow (\gamma \sqcup \{x \mapsto (\!|v|\!)_\gamma, y \mapsto (\!|w|\!)_\gamma\}, t, \mathfrak{g}, \lambda)$$

$$(\gamma, \mathsf{if}\ v\ \mathsf{then}\ t\ \mathsf{else}\ u) \longrightarrow \begin{cases} (\gamma, t, \mathfrak{g}, \lambda) & \text{if } v = \mathsf{true} \\ (\gamma, u, \mathfrak{g}, \lambda) & \text{else, if } v = \mathsf{false} \end{cases}$$

$$\frac{(\gamma, e, \mathfrak{g}, \lambda) \longrightarrow (\gamma', e', \mathfrak{g}', \lambda')}{(\gamma, \mathcal{C}[e], \mathfrak{g}, \lambda) \longrightarrow (\gamma', \mathcal{C}[e'], \mathfrak{g}', \lambda')}$$

Remark 3.8. These reduction rules use lexical binding.

Example 3.9.

(a)

$$\Big(\{x_0 \mapsto a_0\}, \quad \mathsf{let\ val}\ f_1 \leftarrow \lambda_{\text{מ}} x.\,(\mathsf{let\ val}\ b \leftarrow (x = x_0)\ \mathsf{in\ return}(b))\ \mathsf{in}$$
$$\mathsf{let\ val}\ f_2 \leftarrow \lambda_{\text{מ}} y.\, f_1@y\ \mathsf{in}\ f_2@x_0,$$
$$(\emptyset, \{a_0\}, \emptyset),\ \emptyset\Big)$$

$$\longrightarrow^2 \Big(\overbrace{\{x_0 \mapsto a_0,\ f_1 \mapsto \mathfrak{f}_1,\ f_2 \mapsto \mathfrak{f}_2\}}^{:=\gamma_0}, \quad f_2@x_0, \quad (\{\mathfrak{f}_1, \mathfrak{f}_2\},\ \{a_0\}, \{\mathfrak{f}_1 \xrightarrow{\perp} a_0,\ \mathfrak{f}_2 \xrightarrow{\perp} a_0\}),$$

$$\big\{\mathfrak{f}_1 \mapsto (\lambda_{\text{מ}} x.\,(\mathsf{let\ val}\ b \leftarrow (x = x_0)\ \mathsf{in\ return}(b)), \{x_0 \mapsto a_0\}),$$
$$\mathfrak{f}_2 \mapsto (\lambda_{\text{מ}} y.\, f_1@y, \{x_0 \mapsto a_0, f_1 \mapsto \mathfrak{f}_1\})\big\}\Big)$$

$$\longrightarrow \Big(\overbrace{\{x_0 \mapsto a_0,\ f_1 \mapsto \mathfrak{f}_1,\ y \mapsto a_0\}}^{:=\gamma_1}, \quad \{\!\{f_1@y\}\!\}^{\mathfrak{f}_2, a_0}_{\gamma_0},$$
$$(\{\mathfrak{f}_1, \mathfrak{f}_2\},\ \{a_0\}, \{\mathfrak{f}_1 \xrightarrow{\perp} a_0,\ \mathfrak{f}_2 \xrightarrow{\perp} a_0\}),$$
$$\big\{\mathfrak{f}_1 \mapsto (\lambda_{\text{מ}} x.\,(\mathsf{let\ val}\ b \leftarrow (x = x_0)\ \mathsf{in\ return}(b)), \{x_0 \mapsto a_0\}),$$
$$\mathfrak{f}_2 \mapsto (\lambda_{\text{מ}} y.\, f_1@y, \{x_0 \mapsto a_0, f_1 \mapsto \mathfrak{f}_1\})\big\}\Big)$$

$$\longrightarrow \Big(\{x_0 \mapsto a_0,\ x \mapsto a_0\}, \quad \Big\{\!\Big\{ \{\!\{\mathsf{let\ val}\ b \leftarrow (x = x_0)\ \mathsf{in\ return}(b)\}\!\}^{\mathfrak{f}_1, a_0}_{\gamma_1} \Big\}\!\Big\}^{\mathfrak{f}_2, a_0}_{\gamma_0},$$
$$(\{\mathfrak{f}_1, \mathfrak{f}_2\},\ \{a_0\}, \{\mathfrak{f}_1 \xrightarrow{\perp} a_0,\ \mathfrak{f}_2 \xrightarrow{\perp} a_0\}),$$
$$\big\{\mathfrak{f}_1 \mapsto (\lambda_{\text{מ}} x.\ \mathsf{let\ val}\ b \leftarrow (x = x_0)\ \mathsf{in\ return}(b), \{x_0 \mapsto a_0\}),$$
$$\mathfrak{f}_2 \mapsto (\lambda_{\text{מ}} y.\, f_1@y, \{x_0 \mapsto a_0, f_1 \mapsto \mathfrak{f}_1\})\big\}\Big)$$

$$\longrightarrow^2 \Big(\{x_0 \mapsto a_0,\ x \mapsto a_0,\ b \mapsto \mathsf{true}\}, \quad \Big\{\!\Big\{ \{\!\{\mathsf{return}(b)\}\!\}^{\mathfrak{f}_1, a_0}_{\gamma_1} \Big\}\!\Big\}^{\mathfrak{f}_2, a_0}_{\gamma_0},$$
$$(\{\mathfrak{f}_1, \mathfrak{f}_2\},\ \{a_0\}, \{\mathfrak{f}_1 \xrightarrow{\perp} a_0,\ \mathfrak{f}_2 \xrightarrow{\perp} a_0\}),$$
$$\big\{\mathfrak{f}_1 \mapsto (\lambda_{\text{מ}} x.\ \mathsf{let\ val}\ b \leftarrow (x = x_0)\ \mathsf{in\ return}(b), \{x_0 \mapsto a_0\}),$$
$$\mathfrak{f}_2 \mapsto (\lambda_{\text{מ}} y.\, f_1@y, \{x_0 \mapsto a_0, f_1 \mapsto \mathfrak{f}_1\})\big\}\Big)$$

$$\longrightarrow^2 \Big(\overbrace{\{x_0 \mapsto a_0,\ f_1 \mapsto \mathfrak{f}_1,\ f_2 \mapsto \mathfrak{f}_2\}}^{:=\gamma_0}, \quad \mathsf{return}(\mathsf{true}),$$
$$(\{\mathfrak{f}_1, \mathfrak{f}_2\},\ \{a_0\}, \{\mathfrak{f}_1 \xrightarrow{\mathsf{true}} a_0,\ \mathfrak{f}_2 \xrightarrow{\mathsf{true}} a_0\}),$$
$$\big\{\mathfrak{f}_1 \mapsto (\lambda_{\text{מ}} x.\ \mathsf{let\ val}\ b \leftarrow (x = x_0)\ \mathsf{in\ return}(b), \{x_0 \mapsto a_0\}),$$

$\mathsf{f}_2 \mapsto (\lambda_{\text{ת}}\, y.\ f_1@y, \{x_0 \mapsto a_0, f_1 \mapsto \mathsf{f}_1\})\}\Big)$

(b)

$$\Big(\{x_0 \mapsto a_0\},$$
$$\mathsf{let\ val}\ f_1 \ \leftarrow\ \lambda_{\text{ת}}\, x.\ \mathsf{return(false)}\ \mathsf{in}$$
$$\mathsf{let\ val}\ f_2 \ \leftarrow\ \lambda_{\text{ת}}\, y.\ \big(\mathsf{let\ val}\ b \ \leftarrow\ f_1@y\ \mathsf{in\ return(true)}\big)\ \mathsf{in}\ f_2@x_0$$
$$(\emptyset, \{a_0\}, \emptyset),\ \emptyset\Big)$$

$$\longrightarrow^2\Big(\underbrace{\{x_0 \mapsto a_0,\ f_1 \mapsto \mathsf{f}_1,\ f_2 \mapsto \mathsf{f}_2\}}_{:=\gamma_0},\quad f_2@x_0,\quad (\{\mathsf{f}_1, \mathsf{f}_2\},\ \{a_0\}, \{\mathsf{f}_1 \xrightarrow{\perp} a_0,\ \mathsf{f}_2 \xrightarrow{\perp} a_0\}),$$
$$\big\{\mathsf{f}_1 \mapsto (\lambda_{\text{ת}}\, x.\ \mathsf{return(false)}, \{x_0 \mapsto a_0\}),$$
$$\mathsf{f}_2 \mapsto (\lambda_{\text{ת}}\, y.\ \mathsf{let\ val}\ b \ \leftarrow\ f_1@y\ \mathsf{in\ return(true)}, \{x_0 \mapsto a_0, f_1 \mapsto \mathsf{f}_1\})\big\}\Big)$$

$$\longrightarrow\Big(\underbrace{\{x_0 \mapsto a_0, f_1 \mapsto \mathsf{f}_1, y \mapsto a_0\}}_{:=\gamma_1},\ \ \{\!\!\{\mathsf{let\ val}\ b \ \leftarrow\ f_1@y\ \mathsf{in\ return(true)}\}\!\!\}^{\mathsf{f}_2,a_0}_{\gamma_0},$$
$$(\{\mathsf{f}_1, \mathsf{f}_2\},\ \{a_0\}, \{\mathsf{f}_1 \xrightarrow{\perp} a_0,\ \mathsf{f}_2 \xrightarrow{\perp} a_0\}),$$
$$\big\{\mathsf{f}_1 \mapsto (\lambda_{\text{ת}}\, x.\ \mathsf{return(false)}, \{x_0 \mapsto a_0\}),$$
$$\mathsf{f}_2 \mapsto (\lambda_{\text{ת}}\, y.\ \mathsf{let\ val}\ b \ \leftarrow\ f_1@y\ \mathsf{in\ return(true)}, \{x_0 \mapsto a_0, f_1 \mapsto \mathsf{f}_1\})\big\}\Big)$$

$$\longrightarrow\Big(\{x_0 \mapsto a_0, x \mapsto a_0\},\quad \Big\{\!\!\Big\{\mathsf{let\ val}\ b \ \leftarrow\ \{\!\!\{\mathsf{return(false)}\}\!\!\}^{\mathsf{f}_1,a_0}_{\gamma_1}\ \mathsf{in\ return(true)}\Big\}\!\!\Big\}^{\mathsf{f}_2,a_0}_{\gamma_0},$$
$$(\{\mathsf{f}_1, \mathsf{f}_2\},\ \{a_0\}, \{\mathsf{f}_1 \xrightarrow{\perp} a_0,\ \mathsf{f}_2 \xrightarrow{\perp} a_0\}),$$
$$\big\{\mathsf{f}_1 \mapsto (\lambda_{\text{ת}}\, x.\ \mathsf{return(false)}, \{x_0 \mapsto a_0\}),$$
$$\mathsf{f}_2 \mapsto (\lambda_{\text{ת}}\, y.\ \mathsf{let\ val}\ b \ \leftarrow\ f_1@y\ \mathsf{in\ return(true)}, \{x_0 \mapsto a_0, f_1 \mapsto \mathsf{f}_1\})\big\}\Big)$$

$$\longrightarrow\Big(\{x_0 \mapsto a_0, f_1 \mapsto \mathsf{f}_1, y \mapsto a_0\},\quad \{\!\!\{\mathsf{let\ val}\ b \ \leftarrow\ \mathsf{return(false)\ in\ return(true)}\}\!\!\}^{\mathsf{f}_2,a_0}_{\gamma_0},$$
$$(\{\mathsf{f}_1, \mathsf{f}_2\},\ \{a_0\}, \{\mathsf{f}_1 \xrightarrow{\mathsf{false}} a_0,\ \mathsf{f}_2 \xrightarrow{\perp} a_0\}),$$
$$\big\{\mathsf{f}_1 \mapsto (\lambda_{\text{ת}}\, x.\ \mathsf{return(false)}, \{x_0 \mapsto a_0\}),$$
$$\mathsf{f}_2 \mapsto (\lambda_{\text{ת}}\, y.\ \mathsf{let\ val}\ b \ \leftarrow\ f_1@y\ \mathsf{in\ return(true)}, \{x_0 \mapsto a_0, f_1 \mapsto \mathsf{f}_1\})\big\}\Big)$$

$$\longrightarrow\Big(\{x_0 \mapsto a_0, f_1 \mapsto \mathsf{f}_1, y \mapsto a_0, b \mapsto \mathsf{false}\},\quad \{\!\!\{\mathsf{return(true)}\}\!\!\}^{\mathsf{f}_2,a_0}_{\gamma_0},$$

$$
\begin{array}{l}
(\{\mathfrak{f}_1,\mathfrak{f}_2\},\ \{a_0\},\{\mathfrak{f}_1 \xrightarrow{\mathsf{false}} a_0,\ \mathfrak{f}_2 \xrightarrow{\perp} a_0\}),\\
\big\{\mathfrak{f}_1 \mapsto (\lambda_{\text{ם}}\, x.\ \mathsf{return}(\mathsf{false}), \{x_0 \mapsto a_0\}),\\
\mathfrak{f}_2 \mapsto (\lambda_{\text{ם}}\, y.\ \mathsf{let\ val}\ b\ \leftarrow\ f_1@y\ \mathsf{in}\ \mathsf{return}(\mathsf{true}), \{x_0 \mapsto a_0, f_1 \mapsto \mathfrak{f}_1\})\big\}\Big)
\end{array}
$$

$$
\begin{array}{l}
\longrightarrow\Big(\{x_0 \mapsto a_0, f_1 \mapsto \mathfrak{f}_1, f_2 \mapsto \mathfrak{f}_2\},\quad \mathsf{return}(\mathsf{true}),\\
\quad (\{\mathfrak{f}_1,\mathfrak{f}_2\},\ \{a_0\},\{\mathfrak{f}_1 \xrightarrow{\mathsf{false}} a_0,\ \mathfrak{f}_2 \xrightarrow{\mathsf{true}} a_0\}),\\
\quad \big\{\mathfrak{f}_1 \mapsto (\lambda_{\text{ם}}\, x.\ \mathsf{return}(\mathsf{false}), \{x_0 \mapsto a_0\}),\\
\quad \mathfrak{f}_2 \mapsto (\lambda_{\text{ם}}\, y.\ \mathsf{let\ val}\ b\ \leftarrow\ f_1@y\ \mathsf{in}\ \mathsf{return}(\mathsf{true}), \{x_0 \mapsto a_0, f_1 \mapsto \mathfrak{f}_1\})\big\}\Big)
\end{array}
$$

(c) The following example illustrates the effect of lexical binding:

$$
\begin{array}{l}
\Big(\emptyset,\quad \mathsf{let\ val}\ x_1\ \leftarrow\ \mathsf{fresh}()\ \mathsf{in}\\
\qquad\quad \mathsf{let\ val}\ x_2\ \leftarrow\ \mathsf{return}(x_1)\ \mathsf{in}\\
\qquad\quad \mathsf{let\ val}\ f\ \leftarrow\ \lambda_{\text{ם}}\, y.\ (y = x_1)\ \mathsf{in}\\
\qquad\quad \mathsf{let\ val}\ x_1\ \leftarrow\ \mathsf{fresh}()\ \mathsf{in}\ f@x_2\\
\quad (\emptyset,\{a_0\},\emptyset),\ \emptyset\Big)
\end{array}
$$

$$
\begin{array}{l}
\longrightarrow^*\Big(\{x_1' \mapsto a_1', x_1 \mapsto a_1', x_2 \mapsto a_1', y \mapsto a_1'\},\quad \{\!\{\mathsf{return}(\mathsf{true})\}\!\}^{\mathfrak{f},a_1'}_{\gamma_0},\\
\qquad (\{\mathfrak{f}\},\ \{a_1', a_1''\},\{\mathfrak{f} \xrightarrow{\perp} a_1',\ \mathfrak{f} \xrightarrow{\perp} a_1''\}),\\
\qquad \big\{\mathfrak{f} \mapsto (\lambda_{\text{ם}}\, y.\ (y = x_1), \{x_1' \mapsto a_1', x_1 \mapsto a_1', x_2 \mapsto a_1'\})\big\}\Big)\\
\qquad \textit{where}\quad \gamma_0 := \{x_1' \mapsto a_1',\ x_2 \mapsto a_1',\ f \mapsto \mathfrak{f},\ x_1'' \mapsto a_1'',\ x_1 \mapsto a_1''\}
\end{array}
$$

$$
\begin{array}{l}
\longrightarrow\Big(\gamma_0,\quad \mathsf{return}(\mathsf{true}),\quad (\{\mathfrak{f}\},\ \{a_1', a_1''\},\{\mathfrak{f} \xrightarrow{\mathsf{true}} a_1',\ \mathfrak{f} \xrightarrow{\perp} a_1''\}),\\
\qquad \big\{\mathfrak{f} \mapsto (\lambda_{\text{ם}}\, y.\ (y = x_1), \{x_1' \mapsto a_1', x_1 \mapsto a_1', x_2 \mapsto a_1'\})\big\}\Big)
\end{array}
$$

In comparison, the end result would be $\mathsf{return}(\mathsf{true})$ *with dynamic binding.*

(d) The importance of restoring the environment after a function's value is memoized is highlighted by the following program:

$$
\begin{array}{l}
\mathsf{let\ val}\ x_0\ \leftarrow\ \mathsf{fresh}()\ \mathsf{in}\ \mathsf{let\ val}\ x_1\ \leftarrow\ \mathsf{return}(x_0)\ \mathsf{in}\\
\mathsf{let\ val}\ f_1\ \leftarrow\ \lambda_{\text{ם}}\, y_1.\ (y_1 = x_0)\ \mathsf{in}
\end{array}
$$

$$\begin{array}{l}\mathsf{let\ val}\ x_0 \leftarrow \mathsf{fresh}()\ \mathsf{in}\\ \mathsf{let\ val}\ f_2 \leftarrow \lambda_{\text{ם}}\, y_2.\ \big(\mathsf{let\ val}\ z_0 \leftarrow f_1@y_2\ \mathsf{in}\ (x_0 = x_1)\big)\ \mathsf{in}\\ \mathsf{let\ val}\ z_1 \leftarrow f_1@x_1\ \mathsf{in}\ f_2@x_1\end{array}$$

Indeed, whether the let binding z_1 be present or not should not affect the overall result: $\mathsf{return}(\mathsf{false})$. *But if we omit restoring the environment after memoization: the result is* $\mathsf{return}(\mathsf{false})$ *in case $f_1@x_1$ is executed before $f_2@x_1$ (due to $f_1@x_1$ being already memoized inside the body of f_2), whereas it is* $\mathsf{return}(\mathsf{true})$ *otherwise.*

Initial configurations:

$$(\gamma, e, \emptyset, \emptyset)$$

One can associate a typing judgement (called *configuration judgment*) $\mathrm{J}(\gamma, e, \emptyset, \emptyset) := \Gamma \mid \emptyset \vDash^{\text{ç}} e : A$ to such a configuration, which is derivable and such that $\gamma \in (\!|\Gamma|\!)$.

Lemma 3.10 (Configuration Judgement)**.** *If the configuration $(\gamma, e, \mathfrak{g}, \lambda)$ is accessible* – i.e. *there exists an initial configuration $(\gamma_0, e_0, \emptyset, \emptyset)$ and a reduction trace $s = (\gamma_0, e_0, \emptyset, \emptyset) \longrightarrow^* (\gamma, e, \mathfrak{g}, \lambda)$ with probability > 0 – then there exists a corresponding* configuration judgement $\mathrm{J}(\gamma, e, \mathfrak{g}, \lambda) := \Gamma \mid \Delta \vDash^{\text{ç}} e : A$ *(where $\gamma \in (\!|\Gamma|\!)$) such that:*

1. $\mathrm{J}(\gamma, e, \mathfrak{g}, \lambda)$ *is derivable*
2. *There is no duplicate in Δ*
3. *For all $\mathfrak{f} \xrightarrow{\beta} a$ in $\mathfrak{g}$ (where $\beta \in \{\mathsf{true}, \mathsf{false}\}$), there exist $\mathfrak{g}_0, \lambda_0, \gamma', \mathfrak{g}', \lambda'$ and a reduction subtrace of s:*

   $$(\gamma_0 \sqcup \{y \mapsto a\}, \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma'}, \mathfrak{g}_0, \lambda_0) \longrightarrow^* (\gamma', \mathsf{return}(\beta), \mathfrak{g}', \lambda')$$

   with probability > 0, where $\lambda(\gamma(v)) := (\lambda_{\text{ם}}\, y.\ u, \gamma_0)$
4. *If $e = \{\!\{u'\}\!\}^{\mathfrak{f},a}_{\gamma'}$, there exist $\mathfrak{g}_0, \lambda_0$ and a reduction subtrace of s:*

   $$(\gamma_0 \sqcup \{y \mapsto a\}, \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma'}, \mathfrak{g}_0, \lambda_0) \longrightarrow^* (\gamma, e, \mathfrak{g}, \lambda)$$

   with probability > 0, where $\lambda(\mathfrak{f}) := (\lambda_{\text{ם}}\, y.\ u, \gamma_0)$

Proof. By induction on the length of the derivation. The base case is clear.

- If $\mathrm{J}(\gamma, \mathsf{let\ val}\ x \leftarrow \mathsf{return}(v)\ \mathsf{in}\ u, \mathfrak{g}, \lambda) := \Gamma \mid \Delta \vDash^{\text{ç}} \mathsf{let\ val}\ x \leftarrow \mathsf{return}(v)\ \mathsf{in}\ u : A$:

  Let $\mathrm{J}(\gamma \sqcup \{x \mapsto (\!|v|\!)_\gamma\}, u, \mathfrak{g}, \lambda) := \Gamma, x : \mathrm{Ty}((\!|v|\!)_\gamma) \mid \Delta \vDash^{\text{ç}} u : A$ where, by induction

– we can either derive

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \vdash^{\mathsf{v}} v : \mathrm{Ty}(\!|v|\!)_\gamma}}{\Gamma \vdash^{\mathsf{c}} \mathsf{return}(v) : \mathrm{Ty}(\!|v|\!)_\gamma} \qquad \dfrac{\vdots}{\Gamma, x : \mathrm{Ty}(\!|v|\!)_\gamma \mid \Delta_2 \vdash^{\mathsf{c}} u : A}}{\Gamma \mid \underbrace{\emptyset, \Delta_2}_{=\Delta} \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \leftarrow \mathsf{return}(v)\ \mathsf{in}\ u : A}$$

– or

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma \vdash^{\mathsf{v}} v : \mathrm{Ty}(\!|v|\!)_\gamma}}{\Gamma \vdash^{\mathsf{c}} \mathsf{return}(v) : \mathrm{Ty}(\!|v|\!)_\gamma} \qquad \dfrac{\vdots}{\Gamma, x : \mathrm{Ty}(\!|v|\!)_\gamma \mid \Delta_2 \vdash^{\mathsf{c}} t : A}}{\Gamma \mid \underbrace{(\mathfrak{f}, a), \Delta_2}_{=\Delta} \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \leftarrow \mathsf{return}(v)\ \mathsf{in}\ \{\!\{t\}\!\}^{\mathfrak{f},a}_\gamma : A}\ (\mathfrak{f}, a) \notin \Delta_2$$

from which we can deduce

$$\dfrac{\dfrac{\vdots}{\Gamma, x : \mathrm{Ty}(\!|v|\!)_\gamma \mid \Delta_2 \vdash^{\mathsf{c}} t : A}}{\Gamma, x : \mathrm{Ty}(\!|v|\!)_\gamma \mid (\mathfrak{f}, a), \Delta_2 \vdash^{\mathsf{c}} \underbrace{\{\!\{t\}\!\}^{\mathfrak{f},a}_\gamma}_{=u} : A}\ (\mathfrak{f}, a) \notin \Delta_2$$

In both cases, $\mathrm{J}(\gamma \sqcup \{x \mapsto (\!|v|\!)_\gamma\}, u, \mathfrak{g}, \lambda)$ is derivable, and, by induction hypothesis, Δ has no duplicates and the last two conditions are clearly satisfied.

- If $\mathrm{J}(\gamma, \{\!\{\mathsf{return}(v)\}\!\}^{\mathfrak{f},a}_{\gamma'}, \mathfrak{g}, \lambda) := \Gamma \mid \Delta \vdash^{\mathsf{c}} \{\!\{\mathsf{return}(v)\}\!\}^{\mathfrak{f},a}_{\gamma'} : \mathsf{bool}$:

  Let $\mathrm{J}(\gamma', \mathsf{return}(v), (\mathfrak{g}_L, \mathfrak{g}_R, E \cup \{\mathfrak{f} \xrightarrow{v} a\}), \lambda) := \underbrace{\mathrm{Ctxt}(\gamma')}_{:=\Gamma'} \mid \underbrace{\Delta \backslash \{(\mathfrak{f}, a)\}}_{:=\Delta'} \vdash^{\mathsf{c}} \mathsf{return}(v) : \mathsf{bool}$

  where:

$$\dfrac{\dfrac{\dfrac{\overline{\Gamma \vdash^{\mathsf{v}} v : \mathsf{bool}}}{\Gamma \vdash^{\mathsf{c}} \mathsf{return}(v) : \mathsf{bool}}}{\Gamma \mid \Delta' \vdash^{\mathsf{c}} \mathsf{return}(v) : \mathsf{bool}}\ \text{so } \Delta' = \emptyset}{\Gamma \mid \underbrace{(\mathfrak{f}, a), \Delta'}_{=\Delta} \vdash^{\mathsf{c}} \{\!\{\mathsf{return}(v)\}\!\}^{\mathfrak{f},a}_{\gamma'} : \mathsf{bool}}\ (\mathfrak{f}, a) \notin \Delta'$$

  from which it follows, since v is either true, false, or a variable in the context Γ':

$$\dfrac{\dfrac{\dfrac{}{\Gamma' \vdash^{\mathsf{v}} v : \mathsf{bool}}}{\Gamma' \vdash^{\mathsf{c}} \mathsf{return}(v) : \mathsf{bool}}}{\Gamma' \mid \underbrace{\emptyset}_{=\Delta} \vdash^{\mathsf{c}} \mathsf{return}(v) : \mathsf{bool}}$$

Moreover, by induction hypothesis, as $e = \{\!\{\mathsf{return}(v)\}\!\}^{\mathfrak{f},a}_{\gamma'}$, there exist $\mathfrak{g}_0, \lambda_0$ and a reduction subtrace of s:

$$(\gamma_0 \sqcup \{y \mapsto a\}, \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma'}, \mathfrak{g}_0, \lambda_0) \longrightarrow^* (\gamma, e, \mathfrak{g}, \lambda)$$

with probability > 0, where $\lambda(\mathfrak{f}) := (\lambda_{\text{ם}}\, y.\, u, \gamma_0)$

This yields the desired result.

- If $\mathrm{J}(\gamma, \mathsf{let\ val}\ x \leftarrow \lambda_{\text{ם}}\, y.\, u\ \mathsf{in}\ t, \mathfrak{g}, \lambda) := \Gamma \mid \Delta \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \leftarrow \lambda_{\text{ם}}\, y.\, u\ \mathsf{in}\ t : A$:

  Let

  $$\mathrm{J}\left(\gamma \sqcup \{x \mapsto \mathfrak{f}\}, t, (\mathfrak{g}_L \sqcup \{\mathfrak{f}\}, \mathfrak{g}_R, E \sqcup \{f \xrightarrow{\perp} a\}_{a \in \mathfrak{g}_R}), \lambda \sqcup \{\mathfrak{f} \mapsto (\lambda_{\text{ם}}\, y.\, u, \gamma)\}\right) := \Gamma, x : \mathbb{F} \mid \Delta \vdash^{\mathsf{c}} t : A$$

  where we have:

  - either
    $$\dfrac{\begin{array}{c}\vdots\end{array}}{\dfrac{\Gamma \mid \emptyset \vdash^{\mathsf{c}} \lambda_{\text{ם}}\, y.\, u : \mathbb{F} \qquad \Gamma, x : \mathbb{F} \mid \Delta \vdash^{\mathsf{c}} t : A}{\Gamma \mid \emptyset, \Delta \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \leftarrow \lambda_{\text{ם}}\, y.\, u\ \mathsf{in}\ t : A}}$$
  - or
    $$\dfrac{\begin{array}{c}\vdots\end{array} \qquad\qquad \begin{array}{c}\vdots\end{array}}{\dfrac{\Gamma \mid \emptyset \vdash^{\mathsf{c}} \lambda_{\text{ם}}\, y.\, u : \mathbb{F} \qquad \Gamma, x : \mathbb{F} \mid \Delta_2 \vdash^{\mathsf{c}} t' : A}{\Gamma \mid (\mathfrak{f}, a), \emptyset, \Delta_2 \vdash^{\mathsf{c}} \mathsf{let\ val}\ x \leftarrow \lambda_{\text{ם}}\, y.\, u\ \mathsf{in}\ \{\!\{t'\}\!\}^{\mathfrak{f},a}_{\gamma} : A}\ (\mathfrak{f}, a) \notin \Delta_2}$$

    from which we get
    $$\dfrac{\dfrac{\begin{array}{c}\vdots\end{array}}{\Gamma, x : \mathbb{F} \mid \Delta_2 \vdash^{\mathsf{c}} t' : A}}{\Gamma, x : \mathbb{F} \mid \underbrace{(\mathfrak{f}, a), \Delta_2}_{=\Delta} \vdash^{\mathsf{c}} \underbrace{\{\!\{t'\}\!\}^{\mathfrak{f},a}_{\gamma}}_{=t} : A}$$

  This yields the desired result.

- If $\mathrm{J}(\gamma, (v@w), \mathfrak{g}, \lambda) := \Gamma \mid \Delta \vDash^{\mathfrak{C}} (v@w) : \mathsf{bool}$:

  By induction, we have:

  $$\dfrac{\dfrac{\vdots \qquad \vdots}{\Gamma \vDash^{\mathfrak{V}} v : \mathbb{F} \qquad \Gamma \vDash^{\mathfrak{V}} w : \mathbb{A}}}{\Gamma \mid \underbrace{\emptyset}_{=\Delta} \vDash^{\mathfrak{C}} (v@w) : \mathsf{bool}}$$

  - if $\beta := E(\overbrace{\gamma(v)}^{:=\mathfrak{f}}, \overbrace{\gamma(w)}^{:=a}) \in \{\mathsf{true}, \mathsf{false}\}$:
    By induction, there exist $\mathfrak{g}_0, \lambda_0, \gamma', \mathfrak{g}', \lambda'$ and a reduction subtrace of s:

    $$(\gamma_0 \sqcup \{y \mapsto a\}, \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma'}, \mathfrak{g}_0, \lambda_0) \longrightarrow^* (\gamma', \mathsf{return}(\beta), \mathfrak{g}', \lambda')$$

    with probability > 0, where $\lambda(\gamma(v)) := (\lambda_{\text{ᵐ}} y.\ u, \gamma_0)$

    Let $\mathrm{J}(\gamma, \mathsf{return}(\beta), \mathfrak{g}, \lambda) := \Gamma \mid \Delta \vDash^{\mathfrak{C}} \mathsf{return}(\beta) : \mathsf{bool}$

  - else: Let

    $$J(\gamma_0 \sqcup \{y \mapsto (\!|w|\!)_\gamma\}, \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma}, \mathfrak{g}, \lambda) := \mathrm{Ctxt}(\gamma_0), y : \underbrace{\mathbb{A}}_{=\mathrm{Ty}((\!|w|\!)_\gamma)} \mid (\mathfrak{f}, a), \Delta \vDash^{\mathfrak{C}} \{\!\{u\}\!\}^{\mathfrak{f},a}_{\gamma} : \mathsf{bool}$$

    where $\lambda(\gamma(v)) := (\lambda_{\text{ᵐ}} y.\ u, \gamma_0)$. And it can be easily shown that the result follows.

The rest of the proof is omitted. □

4 Soundness

We will now state the soundness property without proof (ongoing work).

Definition 4.1 (Presheaf associated to a partial bigraph)**.** To every partial bigraph $\mathfrak{g} = (\mathfrak{g}_L, \mathfrak{g}_R, E^{\mathfrak{g}})$, we associate a covariant presheaf $P_{\mathfrak{g}} \colon \mathbf{BiGrph}_{emb} \to \mathbf{Set}$ given by:

$$P_{\mathfrak{g}}(g) := \Big\{ (\mathfrak{f}'_1, \ldots, \mathfrak{f}'_{|\mathfrak{g}_L|}, a'_1, \ldots, a'_{|\mathfrak{g}_R|}) \mid \forall i, j. \mathfrak{f}'_i \in g_L,\ a'_j \in g_R \\ \text{and } \forall i, j.\ E^{\mathfrak{g}}(\mathfrak{f}_i, a_j) \neq \bot \Rightarrow E^{g}(\mathfrak{f}'_i, a'_j) = E^{\mathfrak{g}}(\mathfrak{f}_i, a_j) \Big\}$$

where $\mathfrak{g}_L := \{\mathfrak{f}_1, \ldots, \mathfrak{f}_{|\mathfrak{g}_L|}\}$ and $\mathfrak{g}_R := \{a_1, \ldots, a_{|\mathfrak{g}_R|}\}$

The denotational semantics can be extended expressions and configurations.

Let $(\gamma, e, \mathfrak{g}, \lambda)$ be a configuration such that $\mathrm{J}(\gamma, e, \mathfrak{g}, \lambda) = \Gamma \mid \Delta \vDash^{\subseteq} e : A$.

The interpretation of e given the partial bigraph $\mathfrak{g}$ yields a morphism:

$$[\![(e, \mathfrak{g})]\!] \colon 2^k \times P_{\mathfrak{g}} \longrightarrow T([\![A]\!])$$

where k is the number of boolean variables in Γ.

As a result,

$$[\![(e, \mathfrak{g})]\!]_g \colon 2^k \times P_{\mathfrak{g}}(g) \longrightarrow \underbrace{T([\![A]\!])(g)}_{= [0,1]^{g_L} \to P_{\mathrm{f}} \int^{g \hookrightarrow h} A(h) \times [0,1]^{(h-g)_L}}$$

and

$$[\![(e, \mathfrak{g})]\!]_g\big([\![\gamma]\!]^{\mathfrak{g}}_g, [\![\lambda]\!]^{\mathfrak{g}}_g\big) \colon P_{\mathrm{f}} \int^{g \hookrightarrow h} A(h) \times [0,1]^{(h-g)_L}$$

where

- $[\![\gamma]\!]^{\mathfrak{g}}_g \in 2^k \times P_{\mathfrak{g}}(g)$ is comprised of the boolean values specified by γ, and the nodes of g laid out according to $\mathfrak{g}$.
- $[\![\lambda]\!]^{\mathfrak{g}}_g \in [0,1]^{g_L}$ is the state of biases obtained from λ by as follows: every left node $\mathfrak{f}$ of g that is associated to a node of $\mathfrak{g}$ (written $\mathfrak{f}$ as well) via $[\![\gamma]\!]^{\mathfrak{g}}_g$ is mapped to the probability corresponding to the codomain of the (recursively defined) denotation of

$$[\![(u, \mathfrak{g}\backslash\{\mathfrak{f}\} + \bullet)]\!]_{g\backslash\{\mathfrak{f}\}+\bullet}\Big([\![\gamma_0 \sqcup \{y \mapsto \bullet\}]\!]^{\mathfrak{g}\backslash\{\mathfrak{f}\}+\bullet}_{g\backslash\{\mathfrak{f}\}+\bullet}, [\![\lambda\backslash\{\mathfrak{f}\}]\!]^{\mathfrak{g}\backslash\{\mathfrak{f}\}+\bullet}_{g\backslash\{\mathfrak{f}\}+\bullet}\Big)$$

Theorem 4.2 (Soundness)**.** *If*

$$(\gamma, e, \mathfrak{g}, \lambda) \longrightarrow (\gamma', e', \mathfrak{g}', \lambda')$$

then

$$[\![(e, \mathfrak{g})]\!]_g\big([\![\gamma]\!]^{\mathfrak{g}}_g, [\![\lambda]\!]^{\mathfrak{g}}_g\big) = [\![(e', \mathfrak{g}')]\!]_g\big([\![\gamma']\!]^{\mathfrak{g}'}_g, [\![\lambda']\!]^{\mathfrak{g}'}_g\big)$$

naturally in $g \in \mathbf{BiGrph}_{emb}$

References

[Heu+17] Chris Heunen et al. *A Convenient Category for Higher-Order Probability Theory*. Apr. 18, 2017. arXiv: 1701.02547 [cs, math]. URL: http://arxiv.org/abs/1701.02547 (visited on 02/11/2020).

[Lev06] Paul Blain Levy. "Call-by-Push-Value: Decomposing Call-by-Value and Call-by-Name". In: *Higher-Order and Symbolic Computation* 19.4 (Dec. 2006), pages 377–414. ISSN: 1388-3690, 1573-0557. DOI: 10/fwb7vh. URL: http://link.springer.com/10.1007/s10990-006-0480-6 (visited on 11/12/2021).

[Mel14] Paul-André Melliès. "Local States in String Diagrams". In: *Rewriting and Typed Lambda Calculi*. Edited by Gilles Dowek. Redacted by David Hutchison et al. Volume 8560. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pages 334–348. ISBN: 978-3-319-08917-1 978-3-319-08918-8. DOI: 10.1007/978-3-319-08918-8_23. URL: http://link.springer.com/10.1007/978-3-319-08918-8_23 (visited on 05/23/2021).

[Mic68] Donald Michie. ""Memo" Functions and Machine Learning". In: (1968), page 4.

[Pit13] A. M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*. Cambridge Tracts in Theoretical Computer Science 57. Cambridge ; New York: Cambridge University Press, 2013. 276 pages. ISBN: 978-1-107-01778-8.

[PP02] Gordon Plotkin and John Power. "Notions of Computation Determine Monads". In: *Foundations of Software Science and Computation Structures*. Edited by Mogens Nielsen and Uffe Engberg. Redacted by Gerhard Goos, Juris Hartmanis and Jan van Leeuwen. Volume 2303. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pages 342–356. ISBN: 978-3-540-43366-8 978-3-540-45931-6. DOI: 10.1007/3-540-45931-6_24. URL: http://link.springer.com/10.1007/3-540-45931-6_24 (visited on 05/24/2021).

[Roy+08] D. Roy et al. "A Stochastic Programming Perspective on Nonparametric Bayes". In: 2008. URL: https://www.semanticscholar.org/paper/A-stochastic-programming-perspective-on-Bayes-Roy-Mansinghka/946ed04f705a2a28539a8af1f5e9ccdedd7fabc2 (visited on 01/21/2022).

[Sta10] Sam Staton. "Completeness for Algebraic Theories of Local State". In: *Foundations of Software Science and Computational Structures*. Edited by Luke Ong. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pages 48–63. ISBN: 978-3-642-12032-9. DOI: 10.1007/978-3-642-12032-9_5.

[Sta21] Sam Staton. "Some Formal Structures in Probability". 2021. URL: http://www.cs.ox.ac.uk/people/samuel.staton/2021fscd.pdf.

[Sta] Sam Staton. "Name-Passing Process Calculi: Operational Models and Structural Operational Semantics". In: (), page 245.

[Woo+09] Frank Wood et al. “A Stochastic Memoizer for Sequence Data”. In: *Proceedings of the 26th Annual International Conference on Machine Learning - ICML ’09*. The 26th Annual International Conference. Montreal, Quebec, Canada: ACM Press, 2009, pages 1–8. ISBN: 978-1-60558-516-1. DOI: 10/fg8z4q. URL: http://portal.acm.org/citation.cfm?doid=1553374.1553518 (visited on 01/21/2022).