Preorders on Monads and Coalgebraic Simulations

Shin-ya Katsumata and Tetsuya Sato

Research Institute for Mathematical Sciences, Kyoto University, Kyoto, 606-8502, Japan
{sinya,satoutet}@kurims.kyoto-u.ac.jp

Abstract. We study the construction of preorders on **Set**-monads by the semantic $\top\top$-lifting. We show the universal property of this construction, and characterise the class of preorders on a monad as a limit of a **Card**^{op}-chain. We apply these theoretical results to identifying preorders on some concrete monads, including the powerset monad, maybe monad, and their composite monad. We also relate the construction of preorders and coalgebraic formulation of simulations.

1 Introduction

In the coalgebraic treatment of labelled transition systems and process calculi, several coalgebraic formulations of *bisimulations* are proposed [1, 12, 18], and their relationships are well-studied [25]. On the other hand, to express the asymmetry of simulations between coalgebras, we need to generalise the framework of bisimulations. One of the earliest works in this direction is [13], where Hesselink and Thijs introduced a class of relational liftings of **Set**-functors called *relational extensions*, with which simulations can be coalgebraically captured. Hughes and Jacobs took *preordered functors* as a basis for constructing relational extensions of endofunctors. This approach was further developed in the subsequent studies on coalgebraic trace semantics [10] and forward and backward simulations of coalgebras [9]. The key assumption in the last two works is that an *order enrichment* is given to the Kleisli category of a monad.

One natural problem arising in this line of research is how to systematically construct preordered functors. In fact, many coalgebra functors of transition systems contain the functor part of *monads* to describe branching types of transition systems, and they are the focal point when considering relational liftings and preorders on endofunctors. Upon this observation, we address the problem of constructing preorders on monads, and study its relationship to the coalgebraic formulation of simulations.

The main technical vehicle to tackle the problem is *semantic $\top\top$-lifting* [16], which originates from the proof of the strong normalisation of Moggi's computational metalanguage by reducibility candidates [21, 22]. We apply the semantic $\top\top$-lifting to construct preorders on monads, and show that this construction satisfies a universal property. We also characterise the class of preorders on a monad as the limit of a large chain of certain preorders. We then apply these theoretical results to identifying preorders on some concrete monads, including the semiring-valued multiset monad, powerset monad and maybe monad. We finally show that the semantic $\top\top$-lifting satisfies a couple of properties that are relevant to the coalgebraic formulation of simulations.

Preliminaries

Throughout this paper we assume the axiom of choice. We write **Pre** (resp. **Pos**) for the cartesian monoidal category of preorders (resp. posets) and monotone functions between them. For sets I, J, by $I \Rightarrow J$ we mean the set of functions from I to J. Each preorder $\leq$ on a set J extends to the pointwise preorder on a function space $I \Rightarrow J$, which we denote by $\dot{\leq}$. In this paper the metavariable $\mathcal{T}$ (and its variants) is reserved for monads over **Set**. Its components are written by (T, η, μ). For a function $f : I \to TJ$, by $f^{\#}$ we mean the Kleisli lifting of f, that is, the function $\mu_J \circ Tf$. A *preordered functor* [13, 15] consists of an endofunctor $F : \textbf{Set} \to \textbf{Set}$ and an assignment $I \mapsto \sqsubseteq_I$ of a preorder on FI such that for any function $f : I \to J$, Ff is a monotone function from $(FI, \sqsubseteq_I)$ to $(FJ, \sqsubseteq_J)$.

2 Preorders on Monads

Definition 1. *Let I be a set. We call a binary relation $S \subseteq TI \times TI$ substitutive if for each function $f : I \to TI$ and $(x, y) \in S$, $(f^{\#}(x), f^{\#}(y)) \in S$.*

Especially, a preorder $\leq$ on TI is substitutive if and only if for each function $f : I \to TI$, $f^{\#}$ is a monotone function of type $(TI, \leq) \to (TI, \leq)$.

Definition 2. *Let I be a set. We call a preorder $\leq$ on TI congruent if for each set J and functions $f, g : J \to TI$, $f \dot{\leq} g$ implies $f^{\#} \dot{\leq} g^{\#}$.*

Under the correspondence between monads and algebraic theories, TI may be viewed as the set of I-many variable polynomials in the algebraic theory corresponding to $\mathcal{T}$. Then a binary relation $S \subseteq TI \times TI$ is substitutive if for each polynomial pair $(t, u) \in S$ and a simultaneous substitution $[i := v_i]_{i \in I}$ of polynomials, we have $(t[i := v_i]_{i \in I}, u[i := v_i]_{i \in I}) \in S$. The congruence of a preorder $\leq$ on TI means that for each polynomial $v \in TJ$ and two simultaneous substitutions $[j := t_j]_{j \in J}$ and $[j := u_j]_{j \in J}$ such that $t_j \leq u_j$, we have $v[j := t_j]_{j \in J} \leq v[j := u_j]_{j \in J}$.

We introduce the main subject of this paper, *preorders on monads*.

Definition 3. *A preorder $\sqsubseteq$ on $\mathcal{T}$ is an assignment of a preorder $\sqsubseteq_I$ on TI to each set I such that*

1. *each preorder $\sqsubseteq_I$ is congruent, and*
2. *for each function $f : I \to TJ$, $f^{\#}$ is a monotone function from $(TI, \sqsubseteq_I)$ to $(TJ, \sqsubseteq_J)$ (we also call this property* substitutivity*).*

From this definition, $\sqsubseteq_I$ is substitutive for each set I, and $(T, \sqsubseteq)$ is a preordered functor. We write **Pre**$(\mathcal{T})$ for the class of preorders on $\mathcal{T}$. We define a pointwise partial order $\leq$ on **Pre**$(\mathcal{T})$ by: $\sqsubseteq \leq \sqsubseteq'$ if $\sqsubseteq_I \subseteq \sqsubseteq'_I$ holds for each set I. The class **Pre**$(\mathcal{T})$ admits intersections of arbitrary size: for a subcollection $\boldsymbol{\sqsubseteq}$ of **Pre**$(\mathcal{T})$, its intersection is the preorder $\bigcap \boldsymbol{\sqsubseteq}$ on $\mathcal{T}$ defined by: $a (\bigcap \boldsymbol{\sqsubseteq})_I b$ if $a \sqsubseteq_I b$ holds for each preorder $\sqsubseteq \in \boldsymbol{\sqsubseteq}$.

Example 1. We write $\mathcal{T}_p$ for the powerset monad. For each set I, T_pI has a natural preorder given by the set inclusion. This is a preorder on T_p.

Example 2. We write $\mathcal{T}_l$ for the monad whose functor part is given by $T_l I = I + \{*\}$; this is known as the *maybe monad* in Haskell. We assign to each set I the flat partial order on $T_l I$ that makes $\iota_2(*)$ the least element. This is a preorder on T_l.

Example 3. We write $\mathcal{T}_m$ for the free monoid monad. For each set I, we define a preorder $\sqsubseteq_I$ on $T_m I$ by: $x \sqsubseteq_I y$ if the length of x is equal or shorter than y. This is *not* a preorder on T_m because it is not substitutive.

Suppose that the Kleisli category $\mathbf{Set}_{\mathcal{T}}$ of a monad $\mathcal{T}$ is $\mathbf{Pre}$-enriched, and moreover the enrichment is *pointwise*, that is, $(\forall x \in \mathbf{Set}_{\mathcal{T}}(1, I) . f^\# \circ x \sqsubseteq_{1,J} g^\# \circ x)$ implies $f \sqsubseteq_{I,J} g$ for all $f, g \in \mathbf{Set}_{\mathcal{T}}(I, J)$. Then the assignment $I \mapsto \sqsubseteq_{1,I}$ gives a preorder on $\mathcal{T}$ under the identification $\mathbf{Set}_{\mathcal{T}}(1, I) \simeq TI$. Conversely, given a preorder $\sqsubseteq$ on $\mathcal{T}$, the assignment of the preorder $\dot{\sqsubseteq}_J$ to $\mathbf{Set}_{\mathcal{T}}(I, J)$ gives a pointwise $\mathbf{Pre}$-enrichment. This correspondence between pointwise $\mathbf{Pre}$-enrichments on $\mathbf{Set}_{\mathcal{T}}$ and preorders on $\mathcal{T}$ is bijective.

3 Relational Liftings and Preorders on Monads

After reviewing a coalgebraic formulation of (bi)simulations in the category $\mathbf{BRel}$ of binary relations and relation-respecting functions, we introduce a relational lifting of monads, called *preorder* $\top\top$-*lifting*, and show that it gives rise to preorders on monads.

3.1 The Category BRel of Binary Relations

We define the category $\mathbf{BRel}$ (which is the same as Rel in [15]) by the following data. An object in $\mathbf{BRel}$ is a triple (X, I_1, I_2) such that $X \subseteq I_1 \times I_2$. A morphism from (X, I_1, I_2) to (Y, J_1, J_2) is a pair (f_1, f_2) of functions $f_1 : I_1 \to J_1$ and $f_2 : I_2 \to J_2$ such that for each $(i_1, i_2) \in X$, $(f_1(i_1), f_2(i_2)) \in Y$. We use bold letters $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ to range over objects in $\mathbf{BRel}$, and refer to each component of $\mathbf{X} \in \mathbf{BRel}$ by $(\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2)$. We write $i_{\mathbf{X}} : \mathbf{X}_0 \to \mathbf{X}_1 \times \mathbf{X}_2$ for the inclusion function. We say that $\mathbf{X} \in \mathbf{BRel}$ is *above* $(I_1, I_2) \in \mathbf{Set}^2$ if $\mathbf{X}_1 = I_1$ and $\mathbf{X}_2 = I_2$. Objects above the same $\mathbf{Set}^2$-object are ordered by the inclusion of their relation part. We denote this order by $\subseteq$. For each object $\mathbf{X}, \mathbf{Y}$ in $\mathbf{BRel}$ and morphism $(f_1, f_2) : (\mathbf{X}_1, \mathbf{X}_2) \to (\mathbf{Y}_1, \mathbf{Y}_2)$ in $\mathbf{Set}^2$, we abbreviate $(f_1, f_2) \in \mathbf{BRel}(\mathbf{X}, \mathbf{Y})$ to $(f_1, f_2) : \mathbf{X} \nrightarrow \mathbf{Y}$. We call a pair $(\mathbf{X}, \mathbf{Y})$ of objects in $\mathbf{BRel}$ *composable* if $\mathbf{X}_2 = \mathbf{Y}_1$. Their composition $\mathbf{X} * \mathbf{Y}$ is given by the relational composition of $\mathbf{X}_0$ and $\mathbf{Y}_0$:

$$\mathbf{X} * \mathbf{Y} = (\{(x_1, y_2) \mid \exists z \in \mathbf{X}_2 . (x_1, z) \in \mathbf{X}_0, (z, y_2) \in \mathbf{Y}_0\}, \mathbf{X}_1, \mathbf{Y}_2).$$

A preorder $\leq$ on a set I determines a $\mathbf{BRel}$-object $(\leq, I, I)$, which we also denote by $\leq$. We write Eq_I for the $\mathbf{BRel}$-object of the identity relation on I.

The category $\mathbf{BRel}$ arises as the vertex of the pullback of the subobject fibration $p : \mathbf{Sub}(\mathbf{Set}) \to \mathbf{Set}$ (see [14, Chapter 0]) along the product functor $D : \mathbf{Set}^2 \to \mathbf{Set}$:

$$
\begin{array}{ccc}
\mathbf{BRel} & \longrightarrow & \mathbf{Sub}(\mathbf{Set}) \\
\pi \downarrow & & \downarrow p \\
\mathbf{Set}^2 & \xrightarrow{\ D\ } & \mathbf{Set}
\end{array}
\qquad
\text{where } \pi \text{ is }
\begin{cases}
\pi(\mathbf{X}) = (\mathbf{X}_1, \mathbf{X}_2), \\
\pi(f_1, f_2) = (f_1, f_2)
\end{cases}
$$

The leg $\pi : \mathbf{BRel} \to \mathbf{Set}^2$ of the pullback is a partial order fibration [14]. For an object $\mathbf{X}$ in $\mathbf{BRel}$ and a morphism $(f_1, f_2) : (I_1, I_2) \to (\mathbf{X}_1, \mathbf{X}_2)$ in $\mathbf{Set}^2$, we define the *inverse image object* $(f_1, f_2)^* \mathbf{X}$ by

$$(f_1, f_2)^* \mathbf{X} = (\{(x_1, x_2) \mid (f_1(x_1), f_2(x_2)) \in X\}, I_1, I_2).$$

The category $\mathbf{BRel}$ has a bi-cartesian closed structure that is strictly preserved by π. The object part of this structure is given as follows:

$$\dot{\textstyle\prod}_{i \in I} \mathbf{X}_i = (\{(x, y) \mid \forall i \in I . (\pi_i(x), \pi_i(y)) \in (\mathbf{X}_i)_0\}, {\textstyle\prod}_{i \in I}(\mathbf{X}_i)_1, {\textstyle\prod}_{i \in I}(\mathbf{X}_i)_2)$$
$$\dot{\textstyle\coprod}_{i \in I} \mathbf{X}_i = ({\textstyle\bigcup}_{i \in I}\{(\iota_i(x), \iota_i(y)) \mid (x, y) \in (\mathbf{X}_i)_0\}, {\textstyle\coprod}_{i \in I}(\mathbf{X}_i)_1, {\textstyle\coprod}_{i \in I}(\mathbf{X}_i)_2)$$
$$\mathbf{X} \dot{\Rightarrow} \mathbf{Y} = (\{(f, g) \mid \forall (x, y) \in \mathbf{X}_0 . (f(x), g(y)) \in \mathbf{Y}_0\}, \mathbf{X}_1 \Rightarrow \mathbf{Y}_1, \mathbf{X}_2 \Rightarrow \mathbf{Y}_2).$$

This structure captures the essence of *logical relations* for product, coproduct and arrow types interpreted in type hierarchies [23]. We note that the equality functor Eq : $\mathbf{Set} \to \mathbf{BRel}$ also preserves the bi-CC structure (*identity extension*).

3.2 Relational Liftings and Coalgebraic Simulations

Definition 4. *A* relational lifting *of an endofunctor $F : \mathbf{Set} \to \mathbf{Set}$ is an assignment $\dot{F} : |\mathbf{BRel}| \to |\mathbf{BRel}|$ such that for each morphism $(f, g) : \mathbf{X} \dot{\to} \mathbf{Y}$, we have $(Ff, Fg) : \dot{F}\mathbf{X} \dot{\to} \dot{F}\mathbf{Y}$. We say that $\dot{F}$ is*

- reflexive *if* $\mathrm{Eq}_{FI} \subseteq \dot{F}\mathrm{Eq}_I$,
- lax compositional *if* $\dot{F}\mathbf{X} * \dot{F}\mathbf{Y} \subseteq \dot{F}(\mathbf{X} * \mathbf{Y})$,
- compositional *if* $\dot{F}\mathbf{X} * \dot{F}\mathbf{Y} = \dot{F}(\mathbf{X} * \mathbf{Y})$, *and*
- a relational extension *[13] if it is reflexive and compositional.*

A relational lifting bijectively corresponds to an endofunctor $\dot{F} : \mathbf{BRel} \to \mathbf{BRel}$ such that $\pi \circ \dot{F} = F^2 \circ \pi$. We later see that the lax compositionality guarantees the *composability* of simulations between coalgebras.

Example 4. The bi-cartesian closed structure on $\mathbf{BRel}$ gives canonical relational extensions of functors consisting of Id, C_A (the constant functor for a set A), $+$ and $\times$. For instance, the canonical lifting of $FX = C_A + X \times X$ is $\dot{F}\mathbf{X} = \mathrm{Eq}_A \dot{+} \mathbf{X} \dot{\times} \mathbf{X}$.

Example 5. The following relational lifting $\overline{F}$ is known to capture the concept of bisimulation between F-coalgebras in many cases (see e.g. [12]):

$$\overline{F}\mathbf{X} = (\mathrm{Im}, F\mathbf{X}_1, F\mathbf{X}_2),$$

where Im is the image of $\langle F\pi_1, F\pi_2 \rangle \circ Fi_\mathbf{X} : F\mathbf{X}_0 \to F\mathbf{X}_1 \times F\mathbf{X}_2$. It is always reflexive, and also compositional if and only if F preserves weak pullbacks [3].

Example 6. In [13, Section 4.1] Hesselink and Thijs give the following construction of a relational lifting $F^{+(\sqsubseteq)}(\mathbf{X})$ from a preordered functor $(F, \sqsubseteq)$:

$$F^{+(\sqsubseteq)}(\mathbf{X}) = \sqsubseteq_{\mathbf{X}_1} * \overline{F}\mathbf{X} * \sqsubseteq_{\mathbf{X}_2}.$$

They show that every relational extension $\dot{F}$ of a $\mathbf{Set}$-functor F gives rise to a preordered functor $(F, \dot{F}(\mathrm{Eq}_-))$, and $\dot{F}$ can be recovered as $\dot{F} = F^{+(\dot{F}(\mathrm{Eq}_-))}$. In [20], it is shown that the preordered functor $(F, \sqsubseteq)$ is *stable* (Definition 10, [20]) if and only if $F^{+(\sqsubseteq)}$ is a relational extension such that $(F^{+(\sqsubseteq)}, F^2)$ is an endomorphism over π.

A natural generalisation of the coalgebraic formulation of (bi)simulations in [12, 15] is to make it parametrised by relational liftings of coalgebra functors.

Definition 5. *Let $\dot{F}$ be a relational lifting of an endofunctor F : **Set** → **Set**. An $\dot{F}$-simulation from an F-coalgebra (I_1, f_1) to another F-coalgebra (I_2, f_2) is an object **X** ∈ **BRel** above (I_1, I_2) such that $(f_1, f_2) : \mathbf{X} \dashrightarrow \dot{F}\mathbf{X}$.*

Example 7. Hermida and Jacobs formulated bisimulations between F-coalgebras as $\overline{F}$-simulations [12]. Later, Hughes and Jacobs employed $F^{+(\sqsubseteq)}$-simulations to capture the concept of simulations between F-coalgebras [15].

Here are some properties of $\dot{F}$-simulations. I) $\dot{F}$-simulations are closed under the union of arbitrary family. II) If $\dot{F}$ is reflexive, $\dot{F}$-simulations are $\overline{F}$-simulations. III) If $\dot{F}$ is lax compositional, $\dot{F}$-simulations are closed under the relational composition $*$.

We extend the concept of relational liftings of endofunctors to monads.

Definition 6. *A relational lifting of $\mathcal{T}$ is an assignment $\dot{T}$: $|\mathbf{BRel}|$ → $|\mathbf{BRel}|$ such that*

- *For each object **X** in **BRel**, we have $(\eta_{\mathbf{X}_1}, \eta_{\mathbf{X}_2}) : \mathbf{X} \dashrightarrow \dot{T}\mathbf{X}$, and*
- *for each morphism $(f_1, f_2) : \mathbf{X} \dashrightarrow \dot{T}\mathbf{Y}$, we have $(f_1^\#, f_2^\#) : \dot{T}\mathbf{X} \dashrightarrow \dot{T}\mathbf{Y}$.*

A relational lifting of $\mathcal{T}$ bijectively corresponds to a monad $\dot{\mathcal{T}} = (\dot{T}, \dot{\eta}, \dot{\mu})$ over **BRel** such that

$$\pi(\dot{T}\mathbf{X}) = (T\mathbf{X}_1, T\mathbf{X}_2), \quad \pi(\dot{T}(f_1, f_2)) = (Tf_1, Tf_2), \quad \dot{\eta}_\mathbf{X} = (\eta_{\mathbf{X}_1}, \eta_{\mathbf{X}_2}), \quad \dot{\mu}_\mathbf{X} = (\mu_{\mathbf{X}_1}, \mu_{\mathbf{X}_2}).$$

We note that every relational lifting $\dot{T}$ of $\mathcal{T}$ is a strong monad over **BRel**, and its strength $\dot{\theta}$ satisfies $\pi(\dot{\theta}_{\mathbf{X},\mathbf{Y}}) = (\theta_{\mathbf{X}_1,\mathbf{Y}_1}, \theta_{\mathbf{X}_2,\mathbf{Y}_2})$, where θ is the canonical strength of $\mathcal{T}$.

The relational lifting in Example 5 extends to monads:

Proposition 1. *For each monad $\mathcal{T}$, $\overline{T}$ is a relational lifting of $\mathcal{T}$.*

Larrecq, Lasota and Nowak further generalised this fact using subscones and mono factorisation systems [8]. Hesselink and Thijs's construction in Example 6 also yields relational liftings of monads, when preorders on monads are supplied:

Proposition 2. *For each monad $\mathcal{T}$ and preorder $\sqsubseteq$ on $\mathcal{T}$, $T^{+(\sqsubseteq)}$ is a lifting of $\mathcal{T}$.*

3.3   Preorder ⊤⊤-Lifting

Inspired from [22, 21, 24], in [16] the first author introduced *semantic ⊤⊤-lifting*, a method to lift strong monads on the base category $\mathbb{B}$ of a certain partial order fibration $p : \mathbb{E} \to \mathbb{B}$ to its total category $\mathbb{E}$. This method takes a pair (R, S) such that $pS = TR$ as a parameter of the lifting, and by varying this parameter we can derive various liftings of $\mathcal{T}$. In this paper, we apply the semantic ⊤⊤-lifting to the strong monad $\mathcal{T}^2$ over **Set**^{2} and the fibration π : **BRel** → **Set**^{2}, and we supply congruent (and substitutive) preorders to the semantic ⊤⊤-lifting as parameters.

Definition 7. *A preorder parameter for $\mathcal{T}$ is a pair $(R, \leq)$ of a set R and a congruent preorder $\leq$ on TR.*

The following is a special case of the semantic $\top\top$-lifting [16, Definition 3.2], where a preorder parameter is supplied.

Definition 8. *Let* $(R, \leq)$ *be a preorder parameter for* $\mathcal{T}$. *We write* σ_I *for the function* $\lambda x k \,.\, k^{\#}(x) : TI \to (I \Rightarrow TR) \Rightarrow TR.$[1] *We define the assignment* $T^{\top\top(R,\leq)} : |\mathbf{BRel}| \to |\mathbf{BRel}|$ *by:*

$$T^{\top\top(R,\leq)}\mathbf{X} = (\sigma_{\mathbf{X}_1}, \sigma_{\mathbf{X}_2})^*((\mathbf{X}\dot{\Rightarrow} \leq)\dot{\Rightarrow} \leq). \tag{1}$$

Below we call $T^{\top\top(R,\leq)}$ *preorder* $\top\top$-*lifting* to distinguish it from the general semantic $\top\top$-lifting. When the preorder parameter is obvious from context, we simply write $T^{\top\top}$ instead of $T^{\top\top(R,\leq)}$. An equivalent definition of $T^{\top\top}\mathbf{X}$ using an auxiliary object $\mathbf{X}^{\top}$ is:

$$\mathbf{X}^{\top} = \mathbf{X} \dot{\Rightarrow} \leq = (\{(f_1, f_2) \mid \forall(x_1, x_2) \in \mathbf{X}_0 \,.\, f_1(x_1) \leq f_2(x_2)\}, \mathbf{X}_1 \Rightarrow TR, \mathbf{X}_2 \Rightarrow TR),$$
$$T^{\top\top}\mathbf{X} = ((\{(x_1, x_2) \mid \forall(f_1, f_2) \in (\mathbf{X}^{\top})_0 \,.\, f_1^{\#}(x_1) \leq f_2^{\#}(x_2)\}, T\mathbf{X}_1, T\mathbf{X}_2).$$

Theorem 1 ([16]). *The preorder* $\top\top$-*lifting* $T^{\top\top}$ *is a relational lifting of* $\mathcal{T}$.

Example 8 (Example 3.6, [16]). We regard $T_p 1 = \{\emptyset, 1\}$ as the congruent preorder $\emptyset \leq 1$. The preorder $\top\top$-lifting of $\mathcal{T}_p$ with this preorder parameter is

$$T_p^{\top\top}\mathbf{X} = (\{(P_1, P_2) \mid \forall x_1 \in P_1 \,.\, \exists x_2 \in P_2 \,.\, (x_1, x_2) \in \mathbf{X}_0\}, T_p\mathbf{X}_1, T_p\mathbf{X}_2).$$

Every preorder $\top\top$-lifting of a monad $\mathcal{T}$ yields a preorder on $\mathcal{T}$.

Theorem 2. *Let* $(R, \leq)$ *be a preorder parameter for* $\mathcal{T}$.

1. *For each set* I, *we have* $T^{\top\top}\mathrm{Eq}_I = (\{(x, y) \mid \forall f : I \to TR \,.\, f^{\#}(x) \leq f^{\#}(y)\}, TI, TI)$.
2. *The assignment* $I \mapsto T^{\top\top}\mathrm{Eq}_I$ *is a preorder on* $\mathcal{T}$ *(which we denote by* $[\leq]^R$).

Proof. We note that $(T^{\top\top}\mathrm{Eq}_I)_0 = \{(x, y) \mid \forall f, g : I \to TR \,.\, f \dot{\leq} g \implies f^{\#}(x) \leq g^{\#}(y)\}$.

1. ($\supseteq$) Immediate. ($\subseteq$) Let $x, y \in TI$ and assume $\forall h : I \to TR \,.\, h^{\#}(x) \leq h^{\#}(y)$. For functions $f, g : I \to TR$ such that $f \dot{\leq} g$, we have $f^{\#}(x) \leq g^{\#}(x)$ as $\leq$ is congruent, and $g^{\#}(x) \leq g^{\#}(y)$ from the assumption. Therefore $f^{\#}(x) \leq g^{\#}(y)$ holds by the transitivity of $\leq$.
2. (Transitivity) Let $(x, y), (y, z) \in T^{\top\top}\mathrm{Eq}_I$. From 1, for any function $f : I \to TR$, we have $f^{\#}(x) \leq f^{\#}(y)$ and $f^{\#}(y) \leq f^{\#}(z)$, hence $f^{\#}(x) \leq f^{\#}(z)$. (Reflexivity) Reflexivity is immediate from the congruence of $\leq$. (Congruence) The Kleisli lifting of $(f, g) : \mathrm{Eq}_I \dot{\to} T^{\top\top}\mathrm{Eq}_J$ satisfies $(f^{\#}, g^{\#}) : T^{\top\top}\mathrm{Eq}_I \dot{\to} \dot{T}\mathrm{Eq}_J$. From the reflexivity of $T^{\top\top}\mathrm{Eq}_I$, we have $(f^{\#}, g^{\#}) : \mathrm{Eq}_{TI} \subseteq T^{\top\top}\mathrm{Eq}_I \dot{\to} T^{\top\top}\mathrm{Eq}_I$. (Substitutivity) Let $f : I \to TJ$ be a function and $x, y \in TI$ such that $(x, y) \in T^{\top\top}\mathrm{Eq}_I$. For each function $g : J \to TR$, we have
$$g^{\#}(f^{\#}(x)) = (g^{\#} \circ f)^{\#}(x) \leq (g^{\#} \circ f)^{\#}(y) = g^{\#}(f^{\#}(y)),$$
implying $(f^{\#}(x), f^{\#}(x)) \in T^{\top\top}\mathrm{Eq}_J$.

Below we write $\mathbf{CSPre}(\mathcal{T}, I)$ for the set of congruent and substitutive preorders on TI, ordered by inclusion. The mapping $(-)_I : \sqsubseteq \mapsto \sqsubseteq_I$ is a monotone function of type $\mathbf{Pre}(\mathcal{T}) \to \mathbf{CSPre}(\mathcal{T}, I)$. We characterise the assignment $\leq \mapsto [\leq]^R$ as the right adjoint $[-]^I : \mathbf{CSPre}(\mathcal{T}, I) \to \mathbf{Pre}(\mathcal{T})$ to $(-)_I$.

---

[1] This is called the *unit* of the continuation monad transformer [4].

Theorem 3. *For each set I, we have the following adjunction $(-)_I \dashv [-]^I$ such that $[-]^I_I = \mathrm{id}$.*

$$(\mathbf{CSPre}(\mathcal{T}, I), \subseteq) \underset{(-)_I}{\overset{[-]^I}{\rightleftarrows_{\top}}} (\mathbf{Pre}(\mathcal{T}), \leq). \tag{2}$$

Proof. Monotonicity of $[-]^I$ is easy. We show $\sqsubseteq \ \leq \ [\sqsubseteq_I]^I$. Let J be a set and suppose $x \sqsubseteq_J y$. Then from the substitutivity of $\sqsubseteq$, for each function $f : J \to TI$, we have $f^{\#}(x) \sqsubseteq_I f^{\#}(y)$, that is, $x\ [\sqsubseteq_I]^I_J\ y$. Next, we show $[\leq]^I_I = \ \leq$. We first calculate $[\leq]^I_I$:

$$[\leq]^I_I = \{(x, y) \mid \forall f : I \to TI \ . \ f^{\#}(x) \leq f^{\#}(y)\}$$

Then $\leq \ \subseteq \ [\leq]^I_I$ is equivalent to the substitutivity of $\leq$, which is already assumed. To show $[\leq]^I_I \subseteq \ \leq$, use the unit $\eta_I : I \to TI$ of $\mathcal{T}$. $\quad\square$

Example 9. 1. We define a congruent preorder $\leq$ on $T_m 2 = 2^*$ by: $x \leq y$ if x is a subsequence of y. Then we have $x\ [\leq]^2_I\ y$ if and only if x is a subsequence of y.
 2. For $x \in T_m I$ and $i \in I$, by $o(x, i)$ we mean the number of occurrences of i in x. For each congruent preorder $\leq$ on $T_m 1 \simeq \mathbf{N}$, we have $x\ [\leq]^1_I\ y$ if and only if $\forall i \in I \ . \ o(x, i) \leq o(y, i)$.

4 Characterising Pre($\mathcal{T}$) as the Limit of a Large Chain

Using the family of adjunctions (2), for sets I, J we define the monotone function $\varphi_{I,J} : \mathbf{CSPre}(\mathcal{T}, I) \to \mathbf{CSPre}(\mathcal{T}, J)$ by $\varphi_{I,J}(\leq) = [\leq]^J_I$. Theorem 3 asserts $\varphi_{I,I} = \mathrm{id}$.

Lemma 1. *For each $\sqsubseteq \ \in \mathbf{Pre}(\mathcal{T})$ and sets I, J such that $\mathrm{card}(I) \leq \mathrm{card}(J)$, we have $\sqsubseteq_I = [\sqsubseteq_J]^J_I$.*

Proof. From $\sqsubseteq \ \leq \ [\sqsubseteq_J]^J$, we have $\sqsubseteq_I \ \subseteq \ [\sqsubseteq_J]^J_I$. We show the converse. We take an injection $i : I \rightarrowtail J$ and a surjection $s : J \twoheadrightarrow I$ such that $s \circ i = \mathrm{id}$. Suppose $x[\sqsubseteq_J]^J_I y$. Then for the function $\eta \circ i : I \to TJ$, the following holds:

$$Ti(x) = (\eta \circ i)^{\#}(x) \sqsubseteq_J (\eta \circ i)^{\#}(y) = Ti(y).$$

From the substitutivity of $\sqsubseteq$, we obtain $x \sqsubseteq_I y$, because

$$x = Ts \circ Ti(x) = (\eta \circ s)^{\#}(Ti(x)) \sqsubseteq_I (\eta \circ s)^{\#}(Ti(y)) = Ts \circ Ti(y) = y.$$

Lemma 2. *For sets I, J, K such that $\mathrm{card}(I) \leq \mathrm{card}(J)$, we have $\varphi_{J,I} \circ \varphi_{K,J} = \varphi_{K,I}$.*

Proof. We have $\varphi_{K,I}(\leq) = [\leq]^K_I \overset{*}{=} [[\leq]^K_J]^J_I = \varphi_{J,I} \circ \varphi_{K,J}(\leq)$; here, $\overset{*}{=}$ is by Lemma 1. $\quad\square$

This implies that when $\mathrm{card}(I) \leq \mathrm{card}(J)$, we have $\varphi_{J,I} \circ \varphi_{I,J} = \mathrm{id}$, hence $\varphi_{I,J}$ is a split monomorphism in **Pos**.

Lemma 3. *For each $\sqsubseteq \ \in \mathbf{Pre}(\mathcal{T})$ and sets I, J such that $\mathrm{card}(I) \leq \mathrm{card}(J)$, we have $[\sqsubseteq_I]^I \geq [\sqsubseteq_J]^J$.*

Proof. We have $[\sqsubseteq_J]^J \leq [[\sqsubseteq_J]_I^J]^I = [\sqsubseteq_I]^I$; the last step is by Lemma 1.

Thus each $\sqsubseteq \,\in \mathbf{Pre}(\mathcal{T})$ determines a descending chain of preorders on $\mathcal{T}$ indexed by cardinals: $[\sqsubseteq_0]^0 \geq [\sqsubseteq_1]^1 \geq \cdots$, and $\sqsubseteq$ is a lower bound by Theorem 3. In fact, $\sqsubseteq$ is the *greatest* lower bound:

Theorem 4. *For each* $\sqsubseteq \,\in \mathbf{Pre}(\mathcal{T})$, *we have* $\sqsubseteq \,= \bigcap_{\alpha \in \mathbf{Card}} [\sqsubseteq_\alpha]^\alpha$.

Proof. It is sufficient to show $\bigcap_{\alpha \in \mathbf{Card}} [\sqsubseteq_\alpha]^\alpha \leq \,\sqsubseteq$. Let I be a set, $x, y \in TI$ and suppose that $x \, [\sqsubseteq_\alpha]_I^\alpha \, y$ holds for any cardinal α; so this especially holds at $\mathrm{card}(I)$. Taking a bijection $h : I \to \mathrm{card}(I)$, we obtain $Th(x) \sqsubseteq_{\mathrm{card}(I)} Th(y)$. As $\sqsubseteq$ is substitutive, we have $x = Th^{-1} \circ Th(x) \sqsubseteq_I Th^{-1} \circ Th(y) = y$.

Let us write **Card** for the linear order of cardinals (recall that we assume the axiom of choice). To clarify the relationship between $\mathbf{Pre}(\mathcal{T})$ and $\mathbf{CSPre}(\mathcal{T}, -)$, we extend the assignment $\alpha \in \mathbf{Card} \mapsto \mathbf{CSPre}(\mathcal{T}, \alpha)$ to a functor $\mathbf{CSPre}(\mathcal{T}, -) : \mathbf{Card}^{\mathrm{op}} \to \mathbf{Pre}$; the morphism part is given by φ. We thus obtain a large chain:

$$\mathbf{CSPre}(\mathcal{T}, 0) \xleftarrow{\varphi_{1,0}} \mathbf{CSPre}(\mathcal{T}, 1) \xleftarrow{\varphi_{2,1}} \cdots \longleftarrow \mathbf{CSPre}(\mathcal{T}, \aleph_0) \xleftarrow{\varphi_{\aleph_0, \aleph_1}} \cdots$$

We characterise $\mathbf{Pre}(\mathcal{T})$ as a limit of this large chain.

Theorem 5. *The family* $(-)_\alpha : \mathbf{Pre}(\mathcal{T}) \to \mathbf{CSPre}(\mathcal{T}, \alpha)$ *is a limiting cone.*

Proof. We first show that $(-)_\alpha : \mathbf{Pre}(\mathcal{T}) \to \mathbf{CSPre}(\mathcal{T}, \alpha)$ is a cone over $\mathbf{CSPre}(\mathcal{T}, -)$. Let $\sqsubseteq \,\in \mathbf{Pre}(\mathcal{T})$ and α, β be cardinals such that $\alpha \leq \beta$. Then $\varphi_{\beta, \alpha}(\sqsubseteq_\beta) = [\sqsubseteq_\beta]_\alpha^\beta = \,\sqsubseteq_\alpha$ by Lemma 1.

Next, let V be a class and $p : V \to \mathbf{CSPre}(\mathcal{T}, -)$ be a cone. We construct the unique mediating mapping $m : V \to \mathbf{Pre}(\mathcal{T})$ such that $(-)_\alpha \circ m = p_\alpha$. For this, we first prove the following lemma:

Lemma 4. *For each class* V, *cone* $p : V \to \mathbf{CSPre}(\mathcal{T}, -)$ *and cardinals* α, β *such that* $\alpha \leq \beta$, *we have* $[p_\alpha(v)]^\alpha \geq [p_\beta(v)]^\beta$.

Proof. As p is a cone, for any cardinal $\alpha \leq \beta$, we have $\varphi_{\beta, \alpha}(p_\beta(v)) = [p_\beta(v)]_\alpha^\beta = p_\alpha(v)$. Then $[p_\alpha(v)]^\alpha = [[p_\beta(v)]_\alpha^\beta]^\alpha \geq [p_\beta(v)]^\beta$; the last step is by Lemma 3.

Therefore every $v \in V$ determines a decreasing sequence of preorders on $\mathcal{T}$: $[p_0(v)]^0 \geq [p_1(v)]^1 \geq \cdots$. We then define a mapping $m : V \to \mathbf{Pre}(\mathcal{T})$ by

$$m(v) = \bigcap_{\alpha \in \mathbf{Card}} [p_\alpha(v)]^\alpha.$$

This mapping satisfies $m(v)_\alpha = p_\alpha(v)$ because

$$m(v)_\alpha = \bigcap_{\beta \in \mathbf{Card}} [p_\beta(v)]_\alpha^\beta = \bigcap_{\beta \in \mathbf{Card}, \alpha \leq \beta} [p_\beta(v)]_\alpha^\beta = \bigcap_{\beta \in \mathbf{Card}, \alpha \leq \beta} p_\alpha(v) = p_\alpha(v).$$

When another mapping $m' : V \to \mathbf{Pre}(\mathcal{T})$ satisfies $m'(v)_\alpha = p_\alpha(v)$, then $m'(v) = m(v)$ because

$$m'(v) = \bigcap_{\alpha \in \mathbf{Card}} [m'(v)_\alpha]^\alpha = \bigcap_{\alpha \in \mathbf{Card}} [p_\alpha(v)]^\alpha = m(v).$$

Corollary 1. *We have an isomorphism* $\mathbf{CSPre}(\mathcal{T}, \alpha) \simeq \mathbf{Pre}(\mathcal{T})$ *if* $\varphi_{\beta,\alpha}$ *is an isomorphism for each cardinal* $\beta \geq \alpha$.

Finding such a cardinal α is not obvious and depends on $\mathcal{T}$. Below we present a convenient condition for finding such α; see Example 11 for a concrete case.

Definition 9. *We say that a cardinal* α *is* large enough for preorder axioms *on* $\mathcal{T}$ *if for each cardinal* $\beta \geq \alpha$ *and* $x, y \in T\beta$, *there exists functions* $g : \beta \to T\alpha$ *and* $f : \alpha \to T\beta$ *(depending on* x, y*) such that* $f^{\#} \circ g^{\#}(x) = x$ *and* $f^{\#} \circ g^{\#}(y) = y$.

Theorem 6. *If* α *is large enough for preorder axioms on* $\mathcal{T}$, *then* $\mathbf{CSPre}(\mathcal{T}, \alpha) \simeq \mathbf{Pre}(\mathcal{T})$.

Proof. We show that $\varphi_{\alpha,\beta}$ is surjective as a function for any cardinal $\beta \geq \alpha$. When this is shown, $\varphi_{\alpha,\beta}$ becomes the inverse of $\varphi_{\beta,\alpha}$ in **Pos** because $\varphi_{\alpha,\beta}$ is a split monomorphism.

Let β be a cardinal such that $\beta \geq \alpha$, and suppose that it is witnessed by an injection $w : \alpha \rightarrowtail \beta$. For each congruent and substitutive preorder $\leq \in \mathbf{CSPre}(T, \beta)$, we define a binary relation $\leq' \subseteq T\alpha \times T\alpha$ by

$$a \leq' b \iff \text{there exists an injection } m : \alpha \rightarrowtail \beta \text{ such that } Tm(a) \leq Tm(b).$$

Lemma 5. $\leq' \in \mathbf{CSPre}(\mathcal{T}, \alpha)$.

We omit the proof of this lemma. We next show that $\leq$ is the image of $\leq'$ by $\varphi_{\alpha,\beta}$.

Lemma 6. $\varphi_{\alpha,\beta}(\leq') = \leq$.

Proof. Let $x, y \in T\beta$ such that $x \leq y$. For each function $f : \beta \to T\alpha$, we obtain

$$Tw \circ f^{\#}(x) = (Tw \circ f)^{\#}(x) \leq (Tw \circ f)^{\#}(y) = Tw \circ f^{\#}(y)$$

from the substitutivity of $\leq$, thus $f^{\#}(x) \leq' f^{\#}(y)$. Therefore we obtain $x \, [\leq']^{\alpha}_{\beta} \, y$.

Conversely, suppose $x \, [\leq']^{\alpha}_{\beta} \, y$. From the assumption, we have $g : \beta \to T\alpha$ and $f : \alpha \to T\beta$ such that $f^{\#} \circ g^{\#}(x) = x$ and $f^{\#} \circ g^{\#}(y) = y$. We thus have $g^{\#}(x) \leq' g^{\#}(y)$, hence there is an injection $m : \alpha \rightarrowtail \beta$ such that $Tm \circ g^{\#}(x) \leq Tm \circ g^{\#}(y)$. Now take a surjection $s : \beta \twoheadrightarrow \alpha$ such that $s \circ m = \mathrm{id}_\alpha$. Then we have a function $f \circ s : \beta \to T\beta$, and as the preorder $\leq$ is substitutive, we have

$$x = (f \circ s)^{\#} \circ Tm \circ g^{\#}(x) \leq (f \circ s)^{\#} \circ Tm \circ g^{\#}(y) = y.$$

Theorem 7. *The rank of a monad* $\mathcal{T}$, *if it exists, is large enough for preorder axioms on* $\mathcal{T}$.

Proof. We write α for the rank of $\mathcal{T}$. Let β be a cardinal such that $\beta \geq \alpha$ and $x_1, x_2 \in T\beta$. There exists a cardinal $0 < \gamma < \alpha$ (witnessed by an injection $i' : \gamma \rightarrowtail \alpha$), $m_1, m_2 \in T\gamma$ and an injection $i : \gamma \rightarrowtail \beta$ such that $T(i)(m_i) = x_i$ $(i = 1, 2)$. We then take surjections $s : \beta \twoheadrightarrow \gamma$ and $s' : \alpha \twoheadrightarrow \gamma$ that are left inverses to i and i', respectively. Then $f = \eta \circ i' \circ s : \beta \to T\alpha$ and $g = \eta \circ i \circ s' : \alpha \to T\beta$ satisfy $g^{\#} \circ f^{\#}(x_i) = x_i$ because

$$g^{\#} \circ f^{\#}(x_i) = Ti \circ Ts' \circ Ti' \circ Ts \circ Ti(m_i) = Ti(m_i) = x_i \quad (i = 1, 2).$$

5 Enumerating and Identifying Preorders on Monads

The understanding of the categorical status of $\mathbf{Pre}(\mathcal{T})$ allows us to identify its contents in several ways. Below we illustrate some methods with concrete monads.

5.1 Showing the Adjunction (2) being an Isomorphism

Let M be a semiring. We write $\mathcal{T}_c^M$ for the M-valued finite multiset monad, whose functor part is given by $T_c^M I = \{f : I \to M \mid \mathrm{supp}(f) \text{ is finite}\}$; here, $\mathrm{supp}(f) = \{i \in I \mid f(i) \neq 0\}$. Below we show that the adjunction (2) becomes an isomorphism for $I = 1$. The following is the key lemma, which states that each preorder on $\mathcal{T}_c^M$ is pointwise:

Lemma 7. *Each preorder $\sqsubseteq$ on $\mathcal{T}_c^M$ satisfies: $d \sqsubseteq_I d' \iff \forall i \in I . d(i) \sqsubseteq_1 d'(i)$.*

This implies $[\sqsubseteq_1]^1 \leq \sqsubseteq$. Therefore from Theorem 3, we obtain:

Theorem 8. *We have $\mathbf{CSPre}(\mathcal{T}_c^M, 1) \simeq \mathbf{Pre}(\mathcal{T}_c^M)$.*

By letting M be the two-point boolean algebra and removing the finiteness restriction, $\mathcal{T}_c^M$ becomes the powerset monad $\mathcal{T}_p$. A similar argument then identifies $\mathbf{Pre}(\mathcal{T}_p)$:

Theorem 9. *We have $4 \simeq \mathbf{CSPre}(\mathcal{T}_p, 1) \simeq \mathbf{Pre}(\mathcal{T}_p)$. The preorders on $\mathcal{T}_p$ are: I) the discrete order, II) the inclusion order, III) the opposite of II and IV) the trivial preorder (that is, $\sqsubseteq_I = T_p I \times T_p I$).*

5.2 Collecting Preorders of the Form $[\leq]^R$

From Theorem 4, every preorder $\sqsubseteq$ on $\mathcal{T}$ is the intersection of preorders of the form $[\leq]^R$. Therefore if the collection $\{[\leq]^R \mid R \in \mathbf{Set}, \leq \in \mathbf{CSPre}(\mathcal{T}, R)\}$ is closed under intersections of arbitrary size, then it is equal to $\mathbf{Pre}(\mathcal{T})$. Below we identify $\mathbf{Pre}(\mathcal{T}_l)$ using this fact. We note that Levy identified $\mathbf{Pre}(\mathcal{T}_l)$ using a different method called *boolean precongruences* [19]; see Section 7.

Example 10. Let $(R, \leq)$ be a preorder parameter for $\mathcal{T}_l$. Then $[\leq]^R$ is either I) the discrete order, II) the flat order with $\iota_2(*)$ being the least element, III) the opposite of II, or IV) the trivial order. For proving this statement, we consider the combinations of two subcases: A) whether $\iota_2(*)$ is the least element in $(R, \leq)$ or not, and B) whether $\iota_2(*)$ is the greatest element in $(R, \leq)$ or not. From this, we conclude that I—IV are the only preorders on $\mathcal{T}_l$.

5.3 Computing $\mathbf{CSPre}(\mathcal{T}, \alpha)$ with a Large Enough α for Preorder Axioms

In the previous method, we have managed to find a good case analysis of preorder parameters. However, when the monad $\mathcal{T}$ becomes more complex, we immediately have no idea what kind of case analysis on preorder parameters is sufficient for classifying all the preorders on the monad. The second method presented in this section circumvents this problem by exploiting Theorem 6. We find a finite cardinal α that is large enough for preorder axioms on $\mathcal{T}$, then compute $\mathbf{CSPre}(\mathcal{T}, \alpha)$. Below we examine the

case where this computation is feasible. First, we assume that $T\alpha$ is finite. We introduce the following preorder $\lhd$ on $T\alpha \times T\alpha$:

$$(x_1, y_1) \lhd (x_2, y_2) \iff \exists f : \alpha \to T\alpha . (f^\# x_1, f^\# y_1) = (x_2, y_2)$$

and the following *congruent closure operator* C:

$$C(B) = \{(f^\#(w), g^\#(w)) \mid X \in \mathbf{Set}, w \in \mathcal{T}X, (f, g) : \mathrm{Eq}_X \dot{\to} (B, T\alpha, T\alpha)\}.$$

For a finite set D, a subset $A \subseteq D$ and a monotone increasing function f over $T_p D$, the following function $\mathtt{lfp}$ computes the least fixpoint of f including A:

```
lfp(A, f){
   if(A = f(A)) { return A; } else { return lfp(f(A), f); }
}
```

If f is computable then $\mathtt{lfp}$ terminates in finite steps.

We construct the following algorithm $\mathtt{Naive}$ to compute $\mathbf{CSPre}(\mathcal{T}, \alpha)$:

```
CTU(A) { return lfp( A, C ∘ T ∘ U ); }
  f1(L) { return L ∪ { CTU(B ∪ {(x,y)}) | B ∈ L, (x,y) ∈ Tα × Tα \ B }; }
Naive() { return lfp( {Eq_Tα}, f1 ); }
```

where, $\mathtt{U}$ is the upward closure operator on $(T\alpha \times T\alpha, \lhd)$ and $\mathtt{T}$ is the transitive closure operator; they are both computable. The function $\mathtt{CTU}$ thus computes the congruent transitive upward closure of a given binary relation over $T\alpha$. When $\mathtt{C}$ is computable, the above algorithm is also computable.

Proposition 3. $\mathtt{Naive}() = \mathbf{CSPre}(\mathcal{T}, \alpha)$.

We explain how the algorithm $\mathtt{Naive}$ runs with the following example.

Example 11. First, the cardinal 3 is large enough for preorder axioms on the nonempty powerset monad $\mathcal{T}_{p^+}$, because for each pair $(x, y) \in T_{p^+} X \times T_{p^+} X$, the following two functions $f : X \to T_{p^+} 3$ and $g : 3 \to T_{p^+} X$ satisfy $g^\# \circ f^\# x = x$ and $g^\# \circ f^\# y = y$:

$$f(a) = \begin{cases} \{0\} & a \in x \backslash y \\ \{1\} & a \in y \backslash x \\ \{2\} & \text{otherwise} \end{cases}, \qquad g(b) = \begin{cases} y & b = 1 \\ x \cap y & b = 2 \text{ and } x \cap y \neq \emptyset \\ x & \text{otherwise} \end{cases}$$

Since $T_{p^+} 3$ is finite and the multiplication of $\mathcal{T}_{p^+}$ is the set union operation, R is congruent if and only if R satisfies $(x_1, y_1), (x_2, y_2) \in R \implies (x_1 \cup x_2, y_1 \cup y_2) \in R$. Therefore, the following algorithm computes C:

```
C(A){ return lfp( A, f2 ); }  where  f2(B){ return B ∪ {x ∪ y | x, y ∈ B}; }
```

We have $\mathbf{CSPre}(\mathcal{T}, \alpha) \simeq 4$. The orders on $\mathcal{T}_{p^+}$ remains the same as the one for $\mathcal{T}_p$.

Type of preorders	The definition of $x \sqsubseteq_I y$
Trivial preorder	true
Equivalence relations	$x = y$, $\quad (x = y) \vee (\bot \in x \wedge \bot \in y)$,
	$x \backslash \{\bot\} = y \backslash \{\bot\}$
Partial orders	$x \subseteq y$, $\quad x = y \vee x = y \backslash \{\bot\}$,
	$x = y \vee (x \subseteq y \wedge \bot \in x)$, $\quad x = y \vee (x \subseteq y \wedge \bot \in y)$,
	$(x = y) \vee (x \backslash \{\bot\} \subseteq y \backslash \{\bot\} \wedge \bot \in x)$
Proper preorders	$x = y \vee \bot \in x$, $\quad x \subseteq y \vee \bot \in y$,
	$(x \subseteq y) \vee (x \backslash \{\bot\} \subseteq y \backslash \{\bot\} \wedge \bot \in x)$

Table 1. All Preorders on $\mathcal{T}_{pl}$ (we omit opposite ones)

We rewrite the naive algorithm to an efficient one. The basic idea to improve the efficiency is to work on the poset $(T\alpha \times T\alpha/\sim, [\lhd])$ rather than the preorder $(T\alpha \times T\alpha, \lhd)$, where $\sim$ is the equivalence relation $\lhd \cap \rhd$ and $[\lhd]$ is the extension of $\lhd$ to the partial order on $\sim$-equivalence classes.

Since $T\alpha$ is finite, the set of all $\sim$-equivalence classes and the order $[\lhd]$ between them are computable. We then rewrite the naive algorithms CTU and Naive to,

$$\text{CTU}(A) \ \{ \ \texttt{return lfp}(\ A, \ \texttt{C}' \circ \texttt{T}' \circ \texttt{U}' \ ); \ \}$$

$$\texttt{f3}(L) \ \{ \ \texttt{return } L \cup \{ \ \text{CTU}(B \cup \{d\}) \mid B \in L, d \in (T\alpha \times T\alpha/\sim) \setminus B\}; \ \}$$

$$\texttt{Modified}() \ \{ \ \texttt{return lfp}(\{\{[(x,y)] \mid (x,y) \in \text{Eq}_{T\alpha}\}\}, \ \texttt{f3} \ ); \ \}$$

respectively. Here, U' is the upward closure operator on $(T\alpha \times T\alpha/\sim, [\lhd])$, $\texttt{C}'(B) = \{[(x,y)] \mid (x,y) \in \texttt{C}(\bigcup B)\}$, and $\texttt{T}'(B) = \{[(x,y)] \mid (x,y) \in \texttt{T}(\bigcup B)\}$. Since an upward closed subset B of $(T\alpha \times T\alpha, \lhd)$ is the union $\bigcup B'$ of an upward closed subset B' of $(T\alpha \times T\alpha/\sim, [\lhd])$, we have $\{\bigcup B \mid B \in \texttt{Modified}()\} = \mathbf{CSPre}(\mathcal{T}, \alpha)$.
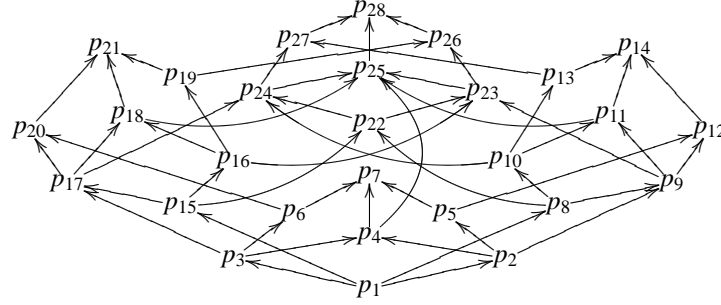
Algorithm Modified is faster than Naive because the upward closure operator U' and the set comprehension in f3 works on the smaller poset $(T\alpha \times T\alpha/\sim, [\lhd])$ than $(T\alpha \times T\alpha, \lhd)$. Function f1 also has a redundant computation: it computes $\text{CTU}(B \cup \{(x,y)\})$ for each pair $(x,y) \in T\alpha \times T\alpha \setminus B$, but the results of this computation are the same when $\sim$-equivalent pairs are supplied. The function f3 avoids such duplicated computation by working on $\sim$-equivalence classes.

We demonstrate an execution of Modified. Below, we write $\mathcal{T}_{pl}$ for the composite monad $\mathcal{T}_p \circ \mathcal{T}_l$ using the canonical distributive law between $\mathcal{T}_p$ and $\mathcal{T}_l$.

Example 12. The cardinal $2 = \{a, b\}$ is large enough for preorder axioms on $\mathcal{T}_{pl}$. First we calculate all $\sim$-equivalence classes and the partial order $[\lhd]$. We have $\mathcal{T}_{pl}2 \times \mathcal{T}_{pl}2/\sim = \{p_1, p_2, \cdots, p_{28}\}$ where,

$p_1 = [(\{a\}, \{b\})]$ $p_8 = [(\{a, \bot\}, \{b\})]$ $p_{15} = [(\{a\}, \{b, \bot\})]$ $p_{22} = [(\{a, \bot\}, \{b, \bot\})]$

$p_2 = [(\{a, b\}, \{b\})]$ $p_9 = [(\{a, b, \bot\}, \{b\})]$ $p_{16} = [(\{a, b\}, \{b, \bot\})]$ $p_{23} = [(\{a, b, \bot\}, \{b, \bot\})]$

$p_3 = [(\{a\}, \{a, b\})]$ $p_{10} = [(\{a, \bot\}, \{a, b\})]$ $p_{17} = [(\{a\}, \{a, b, \bot\})]$ $p_{24} = [(\{a, \bot\}, \{a, b, \bot\})]$

$p_4 = [(\{a\}, \{a\})]$ $p_{11} = [(\{a, \bot\}, \{a\})]$ $p_{18} = [(\{a\}, \{a, \bot\})]$ $p_{25} = [(\{a, \bot\}, \{a, \bot\})]$

$p_5 = [(\{a\}, \emptyset)]$ $p_{12} = [(\{a, \bot\}, \emptyset)]$ $p_{19} = [(\{a\}, \{\bot\})]$ $p_{26} = [(\{a, \bot\}, \{\bot\})]$

$p_6 = [(\emptyset, \{a\})]$ $p_{13} = [(\{\bot\}, \{a\})]$ $p_{20} = [(\emptyset, \{a, \bot\})]$ $p_{27} = [(\{\bot\}, \{a, \bot\})]$

$p_7 = [(\emptyset, \emptyset)]$ $p_{14} = [(\{\bot\}, \emptyset)]$ $p_{21} = [(\emptyset, \{\bot\})]$ $p_{28} = [(\{\bot\}, \{\bot\})]$

We draw the following Hasse diagram of the poset $(\mathcal{T}_{pl}2 \times \mathcal{T}_{pl}2/\sim, [\lhd])$.

Next, we demonstrate the execution of `Modified()`. It computes the least fixpoint of `f3` containing $\{\{p_4, p_7, p_{25}, p_{28}\}\}$. We now see the first loop of `lfp` in the execution of `Modified()` in detail. The function `f3` picks up an equivalence class other than $\{p_4, p_7, p_{25}, p_{28}\}$, say p_6, then pass $\{p_4, p_7, p_{25}, p_{28}, p_6\}$ to `CTU`. The function `CTU` processes its argument by the closure operators `U'`, `T'`, `C'` repeatedly until it gets stationary. The following is the first pass of this process:

- $\mathtt{U'}(\{p_4, p_7, p_{25}, p_{28}, p_6\}) = \{p_4, p_6, p_7, p_{20}, p_{21}, p_{25}, p_{28}\}$
- $\mathtt{T'}(\{p_4, p_6, p_7, p_{20}, p_{21}, p_{25}, p_{28}\}) = \{p_4, p_6, p_7, p_{20}, p_{21}, p_{25}, p_{28}\}$
- $\mathtt{C'}(\{p_4, p_6, p_7, p_{20}, p_{21}, p_{25}, p_{28}\}) = \{p_3, p_4, p_6, p_7, p_{17}, p_{18}, p_{20}, p_{21}, p_{24}, p_{25}, p_{27}, p_{28}\}$

The result of the last calculation by `C'`, which we call H below, is already closed under `U'`, `T'` and `C'`. Therefore, $\mathtt{CTU}(\{p_4, p_7, p_{25}, p_{28}, p_6\}) = H$. The function `f3` similarly calculates $\mathtt{CTU}(\{p_4, p_7, p_{25}, p_{28}, p\})$ for each equivalence class p other than $p_4, p_7, p_{25}, p_{28}, p_6$, and returns the union of the results of the calculations of `CTU`s and `f3`'s argument L. This finishes the first call of `f3`. The function `lfp` in `Modified` repeats calling `f3` until we obtain the least fixpoint of `f3`. The algorithm `Modified()` yields 20 sets of equivalence classes, hence $\mathbf{CSPre}(\mathcal{T}_{pl}, 2) \simeq 20$ (see also Section 7 for Levy's result).

After this computation, we manually extract the definitions of preorders on $\mathcal{T}_{pl}$ from each set of equivalence classes. The 20 preorders are listed in Table 5.3. For this extraction, we first identify the meaning of the binary relation $\bigcup B$ over $T_{pl}2$ for each set $B \in \mathtt{Modified}()$ of equivalence classes, then manually characterise $[\bigcup B]_I^2$ for each set I. For instance, $\bigcup H = \subseteq_2$, and from this we obtain $[\subseteq_2]_I^2 = \subseteq_I$.

Another method to enumerate congruent substitutive preorders on $T\alpha$ is to reduce the problem to finding the valuations ρ that satisfy the following boolean formula:

$$\bigwedge_{(Q_1, Q_2) \in V} \left(\bigwedge_{p \in Q_1} P_p \implies \bigwedge_{p \in Q_2} P_q \right) \tag{3}$$

Here, P_p is the propositional variable assigned to each $p \in T\alpha \times T\alpha/\sim$, and V is the set of the following pairs:

- $(\{p\}, \{q\})$ for all $p, q \in T\alpha \times T\alpha/\sim$ such that $p \lhd q$
- $(\emptyset, [(x, x)])$ for all $x \in T\alpha$
- $(\{[(x, y)], [(y, z)]\}, \{[(x, z)]\})$ for all $x, y, z \in T\alpha$

– $(Q, \mathsf{C}'(Q))$ for all $Q \subseteq T\alpha \times T\alpha/\sim$ such that $\mathsf{C}'(Q) \neq Q$.

The set V encodes the conditions of congruent substitutive preorder. If $T\alpha$ is finite and C is computable, the boolean formula (3) is finite and can be generated by an algorithm.

The satisfying assignments of the boolean formula (3) bijectively correspond to preorders in $\mathbf{CSPre}(\mathcal{T}, \alpha)$. The number of $\mathbf{CSPre}(\mathcal{T}, \alpha)$ is the solution of the problem of counting the number of satisfying assignments of the formula, and this problem is known as #SAT problem [5].

6 Some Properties on Preorder $\top\top$-Lifting

We show that preorder $\top\top$-liftings satisfy a couple of properties that are relevant to the coalgebraic simulations discussed in Section 3.2. The first property relates oplax coalgebra morphisms and simulations. We restrict our attention to TF-coalgebras, where T is the functor part of a monad and F consists of $\mathrm{Id}, C_A, +, \times$ only. Below for each function $g : I \to J$, we define $\mathbf{Gr}(g)$ to be the $\mathbf{BRel}$-object $(\{(i, g(i)) \mid i \in I\}, I, J)$ of the graph of g. We note $\dot{F}(\mathbf{Gr}g) = \mathbf{Gr}(Fg)$.

Theorem 10. *Let $(R, \leq)$ be a preorder parameter, and (I_i, f_i) be TF-coalgebras ($i = 1, 2$). For each function $g : I_1 \to I_2$, $\mathbf{Gr}(g)$ is a $T^{\top\top}\dot{F}$-simulation from (I_1, f_1) to (I_2, f_2) if and only if g is an oplax morphism of coalgebras with respect to $[\leq]^R$, that is, $TFg \circ f_1[\dot{\leq}]^R_{FI_2} f_2 \circ g$.*

In general, preorder $\top\top$-liftings may not be lax compositional. We here present a condition to guarantee the lax compositionality.

Theorem 11. *Let $(R, \leq)$ be a congruent preorder such that $\leq$ satisfies the following condition for all subsets $X, Y \subseteq TR$:*

$$(\forall x \in X, y \in Y . \ x \leq y) \implies \exists z \in TR. \forall x \in X, y \in Y . \ x \leq z \wedge z \leq y. \qquad (4)$$

Then $T^{\top\top}$ is lax compositional.

For instance, (4) is satisfied when the preorder parameter $(R, \leq)$ is a complete lattice.

7 Conclusion and Related Work

We showed that preorder $\top\top$-liftings construct preorders on monads, and this construction enjoys a universal property. We gave a characterisation of the collection $\mathbf{Pre}(\mathcal{T})$ of preorders on $\mathcal{T}$ as the limit of the large diagram $\mathbf{CSPre}(\mathcal{T}, -) : \mathbf{Card}^{op} \to \mathbf{Set}$. We then applied these theoretical results to identifying preorders on some concrete monads. We also showed the properties of the preorder $\top\top$-lifting that are relevant to the coalgebraic formulation of simulations.

Besides [13, 11, 15], we briefly mention some recent works on (bi)simulations and relational liftings. Cîrstea studies modular constructions of relational extensions and modal logics characterising simulations using the categorical structures on $\mathbf{BRel}$ [7]. Klin studies the least fibred lifting of $\mathbf{Set}$-functors across the fibration $\mathbf{ERel} \to \mathbf{Set}$,

where **ERel** is the category of equivalence relations [17]. His lifting works for mono-preserving functors, and when they preserve weak pullbacks, his lifting coincides with the one in Example 5. Balan and Kurz give liftings and extensions of finitary **Set**-functors to endofunctors over **Pre** and **Pos** [2]. Their method uses the fact that every finitary **Set**-functor T is presented as $\mathrm{Lan}_I(T \circ I)$, where $I : \textbf{Finord} \to \textbf{Set}$ is the inclusion functor. Bilkova et al. derive a natural definition of relations between preorders using Sierpinski-space enriched categories, and give relational liftings of endofunctors over **Pre** in this context [6]. Levy extends the characterisation of bisimilarity by final coalgebras to similarity [20].

The novelty of our approach is that we exploit the structure of *monad* to relationally lift functors. The principle of the semantic $\top\top$-lifting seems fundamentally different from the lifting methods employed in the above works. One distinguishing feature of the semantic $\top\top$-lifting is its flexibility. By changing the preorder parameter, we can uniformly derive various relational liftings and preorders on monads. The source of this flexibility lies at *continuation monads*, which are a special case of enriched *right* Kan extensions.

Levy introduces the concept called *deterministic / nondeterministic boolean pre-congruences* (*DBP* and *NDBP* for short) in [19]. They are defined in our language by:

$$\mathrm{DBP}_E = \textbf{CSPre}(\mathcal{T}_e^E, 2), \quad \mathrm{NDBP}_E = \textbf{CSPre}(\mathcal{T}_{p^+} \circ \mathcal{T}_e^E, 2);$$

here, $\mathcal{T}_e^E$ is the *error monad*, whose functor part is given by $T_e^E I = I + E$. He shows $\textbf{CSPre}(\mathcal{T}, 2) \simeq \textbf{Pre}(\mathcal{T})$ for $\mathcal{T} = \mathcal{T}_e^E$ and $\mathcal{T} = \mathcal{T}_{p^+} \circ \mathcal{T}_e^E$, and enumerates the following boolean precongruences together with their definitions:

$$\mathrm{DBP}_0 \simeq 2, \quad \mathrm{DBP}_1 \simeq 4, \quad \mathrm{DBP}_2 \simeq 13, \quad \mathrm{NDBP}_0 \simeq 4, \quad \mathrm{NDBP}_1 \simeq 20.$$

He also gives modal logics that have Hennesy-Milner property with respect to the concept of simulations derived from boolean precongruences. His results are derived by the method that is specialised to these monads.

References

1. P. Aczel and N. Mendler. A final coalgebra theorem. In D. Pitt, D. Rydeheard, P. Dybjer, A. Pitts, and A. Poigné, editors, *Category Theory and Computer Science*, volume 389 of *LNCS*, pages 357–365. Springer, 1989.
2. A. Balan and A. Kurz. Finitary functors: From set to preord and poset. In Andrea Corradini, Bartek Klin, and Corina Cîrstea, editors, *CALCO*, volume 6859 of *LNCS*, pages 85–99. Springer, 2011.
3. M. Barr. Relational algebras. In S. MacLane, H. Applegate, M. Barr, B. Day, E. Dubuc, Phreilambud, A. Pultr, R. Street, M. Tierney, and S. Swierczkowski, editors, *Reports of the Midwest Category Seminar IV*, volume 137 of *LNM*, pages 39–55. Springer Berlin Heidelberg, 1970.

4. N. Benton, J. Hughes, and E. Moggi. Monads and effects. In *Proc. APPSEM '00*, volume 2395 of *LNCS*, pages 42–122. Springer, 2002.

5. A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2009.

6. M. Bílková, A. Kurz, D. Petrisan, and J. Velebil. Relation liftings on preorders and posets. *CoRR*, abs/1210.1433, 2012.

7. C. Cîrstea. A modular approach to defining and characterising notions of simulation. *Information and Computation*, 204(4):469 – 502, 2006.

8. J. G.-Larrecq, S. Lasota, and D. Nowak. Logical relations for monadic types. In *Proc. CSL '02*, volume 2471 of *LNCS*, pages 553–568. Springer, 2002.

9. I. Hasuo. Generic forward and backward simulations. In C. Baier and H. Hermanns, editors, *CONCUR*, volume 4137 of *LNCS*, pages 406–420. Springer, 2006.

10. I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace theory. *Electr. Notes Theor. Comput. Sci.*, 164(1):47–65, 2006.

11. C. Hermida and B. Jacobs. An algebraic view of structural induction. In *Proc. CSL '94*, volume 933 of *LNCS*, pages 412–426. Springer-Verlag, 1995.

12. C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. Comput.*, 145(2):107–152, 1998.

13. W. H. Hesselink and A. Thijs. Fixpoint semantics and simulation. *Theor. Comput. Sci.*, 238(1-2):275–311, 2000.

14. B. Jacobs. *Categorical Logic and Type Theory*. Elsevier, 1999.

15. B. Jacobs and J. Hughes. Simulations in coalgebra. *Electr. Notes Theor. Comput. Sci.*, 82(1):128–149, 2003.

16. S. Katsumata. A semantic formulation of ⊤⊤-lifting and logical predicates for computational metalanguage. In *Proc. CSL '05*, volume 3634 of *LNCS*, pages 87–102. Springer, 2005.

17. B. Klin. The least fibred lifting and the expressivity of coalgebraic modal logic. In *Procs. CALCO*, volume 3629 of *LNCS*, pages 247–262, 2005.

18. A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwig-Maximilians-Universität, Munchen, 2000.

19. P. Levy. Boolean precongruences. Manuscript, 2009.

20. P. Levy. Similarity quotients as final coalgebras. In Martin Hofmann, editor, *FOSSACS*, volume 6604 of *LNCS*, pages 27–41. Springer, 2011.

21. S. Lindley. *Normalisation by Evaluation in the Compilation of Typed Functional Programming Languages*. PhD thesis, University of Edinburgh, 2004.

22. S. Lindley and I. Stark. Reducibility and tt-lifting for computation types. In Pawel Urzyczyn, editor, *TLCA*, volume 3461 of *LNCS*, pages 262–277. Springer, 2005.

23. J. Mitchell. *Foundations for Programming Languages*. MIT Press, 1996.

24. A. Pitts. Parametric polymorphism and operational equivalence. *Mathematical Structures in Computer Science*, 10(3):321–359, 2000.

25. S. Staton. Relating coalgebraic notions of bisimulation. *Logical Methods in Computer Science*, 7(1), 2011.