

Coinductive Reasoning for Parametrized Functors and Monads

Ugo Dal Lago 

University of Bologna, Italy

Zeinab Galal 

RIMS, Kyoto University, Japan

Abstract

Lax extensions (also called relators or relation liftings) are a categorical notion to reason about functors acting on functions and relations in a compatible way. They play a central role to develop sound proof principles for behavioral equivalence of state-based systems and are also important for establishing contextual equivalence for effectful programs.

In this paper, we develop the theory of lax extensions for parametrized functors and monads and consider notions of behavioral preorders, equivalence relations or metrics which can now be modulated by additional parameters. From an operational viewpoint, we replace standard contextual equivalence where we quantify over all possible contexts by a refined notion of equivalence where the user can regulate the allowed contexts via chosen parameters.

2012 ACM Subject Classification Theory of computation → Categorical semantics; Theory of computation → Operational semantics

Keywords and phrases categorical semantics, parametrized monads, effects, relations, lax extensions, behavioral equivalence, coalgebra, global state

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2026.29

Funding The first author was partly supported by the ANR Project HOPR (ANR-24-CE48-5521-01) and the second author was supported by the Research Institute for Mathematical Sciences (RIMS) and by the JSPS KAKENHI Grant 26K21167.

Introduction

Relations play a central role in modeling various notions of behavioral preorders and equivalences for state-based systems and programming languages. In particular, coinductive techniques on relations provide a formal framework to reason about circular or infinite objects and they are an important tool in concurrency theory and in operational semantics for establishing behavioral equivalence of processes or programs [38]. Simulation and bisimulation are among the main examples of relations defined coinductively and they have a categorical counterpart in the world of coalgebras. Coalgebras are morphisms $S \rightarrow F(S)$ where S represents the state space and F is a functor encoding the branching type of the system [37, 23]. By varying the functor, we can capture a wide range of state-based systems: streams, labelled transition systems, automata, Markov chains, etc.

A fundamental component of the study of behavioral preorders and equivalences is to extend the behavior functor of the system from sets to relations using the notion of lax extension. This notion also plays an important role in the operational semantics of higher-order functional languages with monadic effects where applicative bisimulation is generalized to the effectful setting using lax extensions of monads from sets to relations [13].

Lax extensions are also used in modal logic, effectful intersection type systems, monoidal topology and in the theory of double categories [30, 15, 22, 27]. They were initially considered by Barr for the ultrafilter monad on sets and were since then generalized to quantale-enriched relations [8, 36, 46, 40, 39, 10] to measure the *behavioral distance* between two states for non-deterministic and probabilistic systems [44, 7, 26].



© Ugo Dal Lago and Zeinab Galal;
licensed under Creative Commons License CC-BY 4.0

37th International Conference on Concurrency Theory (CONCUR 2026).

Editors: Ana Sokolova and Patrick Totzke; Article No. 29; pp. 29:1–29:24

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The main purpose of this paper is to extend the theory of lax extensions for endofunctors and monads to *parametrized* functors and monads. A first generalization is to replace endofunctors $F : \mathbb{C} \rightarrow \mathbb{C}$ with functors $F : \mathbb{P} \times \mathbb{C} \rightarrow \mathbb{C}$ where $\mathbb{P}$ is viewed as a parameter category. This simple generalization allows us to vary the parameter and consider notions of behavioral preorders or equivalences which depend on relations on the parameter.

In the case of monads, the underlying endofunctor of an ordinary monad $T : \mathbb{C} \rightarrow \mathbb{C}$ is replaced by a functor $T : \mathbb{P}^{\text{op}} \times \mathbb{P} \times \mathbb{C} \rightarrow \mathbb{C}$ [5]. A parametrized monad has therefore two additional parameters providing information about the starting and ending state of a computation. It is important to note that the input parameter is contravariant and the output parameter is covariant, following the intuition of preconditions and postconditions for program logics. Parametrized monads were introduced by Atkey [5] and they offer a convenient setting to model sequential effectful computations, they also have a categorical characterization as the notion of monads arising from parametrized adjunctions [28, 12] (also called adjunction with a parameter or multivariable adjunction) offering the natural generalization of the standard monad-adjunction correspondence. One of the motivating examples is the global state monad: in the ordinary case, the set of global stores is fixed and remains constant which is not flexible enough to keep track of which memory regions a computation has access to and can modify. In the parametrized case, the change in the state type over time is captured by additional parameters corresponding to the memory region *before* and *after* the execution of a computation.

In this paper, we develop the theory of lax extensions for parametrized monads to have a framework that allows us to modulate the notion of program equivalence or metric depending on the power we want to give to contexts. For example, if programs can read and write on some memory locations, standard contextual equivalence means that contexts have access to the same memory locations as programs. We are interested in having well-behaved notions of program equivalence where contexts only have access to a *restricted* set of memory locations. In this setting, our framework allows to weaken the notion of equivalence by explicitly controlling the power we give to contexts depending on the relations added to parameters.

Plan of the paper

- After a background section on parametrized monads (Section 1), we introduce the notion of parametrized lax extensions for parametrized functors and monads in Section 2.
- In Section 3, we generalize the Barr canonical extension to the parametrized setting and show how additional nuances need to be addressed in the presence of mixed variance variables. In particular, there are several options for the Barr extension, and we study their properties.
- In Section 4, we study notions of bisimulation and behavioral equivalence in the parametrized setting where the behavior of two states is compared relative to some chosen relations on the parameters.
- Lastly, we instantiate our framework in Section 5 to coalgebras for parametrized functors in the first order setting (Mealy machines) and to parametrized monadic higher order languages where we introduce a parametrized version of contextual equivalence.

1 Background on parametrized monads

We begin by recalling the notion of parametrized (strong) monad and provide some examples. While the term “parametrized” has been used in various contexts, we would like to emphasize that the notion of parametrized monad we consider in this paper is the one by Atkey [5] and is distinct from the other generalizations of monads considered in the literature such as indexed monads [14], parametric effect monads [25], Dijkstra monads [29], graded monads [42], etc.

► **Definition 1** (Definitions 5.2.7 and 5.2.9 in [3]). *For categories $\mathbb{C}$ and $\mathbb{D}$, a parametrized monad consists of a functor $T : \mathbb{C}^{\text{op}} \times \mathbb{C} \times \mathbb{D} \rightarrow \mathbb{D}$ together with*

- *a family of morphisms $\eta_{S,X} : X \rightarrow T(S, S, X)$ indexed by objects S in $\mathbb{C}$ and X in $\mathbb{D}$ natural in X and dinatural in S*
 - *a family of morphisms $\mu_{S_1, S_2, S_3, X} : T(S_1, S_2, T(S_2, S_3, X)) \rightarrow T(S_1, S_3, X)$ indexed by objects S_1, S_2, S_3 in $\mathbb{C}$ and X in $\mathbb{D}$ natural in X, S_1, S_3 and dinatural in S_2*
- satisfying some compatibility axioms. If $(\mathbb{D}, \otimes, 1)$ is monoidal, a parametrized strong monad is additionally equipped with*
- *a family of morphisms $\text{str}_{S_1, S_2, X, Y} : X \otimes T(S_1, S_2, Y) \rightarrow T(S_1, S_2, X \otimes Y)$ in $\mathbb{D}$ indexed by objects S_1, S_2 in $\mathbb{C}$ and X, Y in $\mathbb{D}$, natural in all variables, and satisfying some compatibility axioms.*

► **Example 2.**

1. The parametrized state monad [5, Section 2.3.2]: for a fixed set of states S , the global state monad maps a set X to the set $(X \times S)^S$ while the parametrized version allows for the set of input states and output states to vary, its underlying functor $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ has the following action on objects: $(S_1, S_2, A) \mapsto (A \times S_2)^{S_1}$.
2. The composable continuation monad [3, Example 5.2.6]: for a fixed response object R in a cartesian closed category, the ordinary continuation monad maps an object X to $(X \Rightarrow R) \Rightarrow R$. In the parametrized case, we obtain the notion of composable continuations where the response type may vary: the composable continuation monad $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ is given by $(R_1, R_2, X) \mapsto (X \Rightarrow R_1) \Rightarrow R_2$.
3. The parametrized I/O monad [5, Section 2.3.4]: for a fixed set of inputs I and observable outputs O , the interactive input-output (I/O) monad is the functor mapping a set X to the underlying set of the initial algebra $\mu Z.(X + Z^I + O \times Z)$. It induces three possible types of computations: either compute a value, wait for an input and then resume or produce an output and then resume [31, Example 1.1]. The parametrized I/O-monad $\mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ maps a triple (I, O, X) to the underlying set of $\mu Z.(X + Z^I + O \times Z)$.
4. The parametrized action monad [3, Example 5.2.5]: any monoid $(M, e : I \rightarrow M, m : M \otimes M \rightarrow M)$ internal to a monoidal category $\mathbb{C}$ induces a monad (called the writer, output or action monad) on $\mathbb{C}$ given by $X \mapsto M \otimes X$. This construction has been generalized by Atkey in the case of parametrized monads where any category $\mathbb{P}$ induces a parametrized monad $\mathbb{P}^{\text{op}} \times \mathbb{P} \times \mathbf{Set} \rightarrow \mathbf{Set}$ whose action on objects is $(A, B, X) \mapsto \mathbb{P}(A, B) \times X$.

2 Parametrized lax extensions

2.1 Relations and dualities

The simplest notion of relation is a binary relation between sets $R \subseteq X \times Y$ which we denote $R : X \multimap Y$. The identity relation $\text{id}_X : X \multimap X$ is the diagonal $\{(x, x) \mid x \in X\}$ and the composite of two relations $R : X \multimap Y$ and $S : Y \multimap Z$ is the relation $S \circ R : X \multimap Z$ given by $\{(x, z) \mid \exists y \in Y (x, y) \in R \text{ and } (y, z) \in S\}$. We will also use the sequential notation $R; S$ for the composite $S \circ R$ and we denote by **Rel** the category of sets and binary relations.

The underlying functor of a parametrized monad $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ has both contravariant and covariant inputs so it is important to understand the dualities it induces on relations to motivate the definition of parametrized extension. They are better understood in light of the 2-categorical structure of **Rel**: for two relations $R, S : X \multimap Y$, the inclusion $R \subseteq S$ viewed as subsets of $X \times Y$ induces an ordering on hom-sets $(\mathbf{Rel}(X, Y), \leq)$. Composition of relations is monotone with respect to this ordering which implies that **Rel** is a posetal 2-category.

For a general 2-category $\mathcal{C}$, we have the “op” operator which reverses the direction of 1-morphisms and we also have the “co” operator which reverses the direction of 2-morphisms. For example, the transpose operator that maps a relation $R : X \multimap Y$ to the relation $R^t = \{(y, x) \mid (x, y) \in R\} : Y \multimap X$ can be seen as a 2-functor $(-)^t : \mathbf{Rel}^{\text{op}} \rightarrow \mathbf{Rel}$.

The exponentiation operation in $\mathbf{Set}$ is a functor $\mathbf{Set}^{\text{op}} \times \mathbf{Set} \rightarrow \mathbf{Set}$ mapping a pair (X, Y) to Y^X , the set of all functions from X to Y . It induces an exponentiation¹ 2-functor $\mathbf{Rel}^{\text{co}} \times \mathbf{Rel} \rightarrow \mathbf{Rel}$ that maps a pair of relations $(R : X_1 \multimap X_2, S : Y_1 \multimap Y_2)$ to $S^R := \{(\varphi, \psi) \in Y_1^{X_1} \times Y_2^{X_2} \mid \forall (x_1, x_2) \in R, (\varphi(x_1), \psi(x_2)) \in S\}$, which reverses the inclusion of relations in the first argument, *i.e.* $R \leq R'$ implies $S^{R'} \leq S^R$. This operation will guide the formulation of the theory of parametrized lax extensions, and the exponentiation of relations will provide the canonical extension for the state monad.

Another important fact for the theory of lax extensions is the inclusion of functions into relations: any function $f : X \rightarrow Y$ induces two relations $f_* : X \multimap Y$ and $f^* : Y \multimap X$ given by $f_* := \{(x, f(x)) \mid x \in X\}$ and $f^* := \{(f(x), x) \mid x \in X\}$. Moreover, f_* is left adjoint to f^* in (the posetal 2-category) $\mathbf{Rel}$.

2.2 Parametrized (lax) extensions

In the non-parametrized setting, for an endofunctor $F : \mathbf{Set} \rightarrow \mathbf{Set}$, a *strict extension* (or simply an extension) of F to $\mathbf{Rel}$ consists of an endofunctor $\bar{F} : \mathbf{Rel} \rightarrow \mathbf{Rel}$ that agrees with F on objects ($F(A) = \bar{F}(A)$) and that is compatible with the inclusion of functions into relations, *i.e.* for a function $f : A \rightarrow B$, $(F(f))_* = \bar{F}(f_*)$. An extension $\bar{F}$ is said to be *monotone* if it preserves the inclusion of relations: if $R \leq R'$, then $\bar{F}(R) \leq \bar{F}(R')$. Equivalently, $\bar{F}$ is a 2-functor $\mathbf{Rel} \rightarrow \mathbf{Rel}$.

The general idea is that, in the strict case, a monotone extension of a functor $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ will be a 2-functor $\bar{T} : \mathbf{Rel}^{\text{co}} \times \mathbf{Rel} \times \mathbf{Rel} \rightarrow \mathbf{Rel}$ and not $\bar{T} : \mathbf{Rel}^{\text{op}} \times \mathbf{Rel} \times \mathbf{Rel} \rightarrow \mathbf{Rel}$.

► **Definition 3.** For a functor $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$, a 1-functor $\bar{T} : \mathbf{Rel} \times \mathbf{Rel} \times \mathbf{Rel} \rightarrow \mathbf{Rel}$ extends T strictly if the following holds:

1. for all $f : A \rightarrow B, g : C \rightarrow D$ and $h : X \rightarrow Y$, $\bar{T}(f^*, g_*, h_*) = (T(f, g, h))_*$
2. we say that $\bar{T}$ is monotone if for all $M' \leq M : A \multimap B$, $N \leq N' : C \multimap D$ and $R \leq R' : X \multimap Y$, $\bar{T}(M, N, R) \leq \bar{T}(M', N', R')$.

Strict extensions are in practice too restrictive and various weakenings have been considered in the non-parametrized setting. In this paper, we focus on the notion of *lax extension* considered by Barr and later generalized to the quantale-enriched setting [8, 46, 40, 39, 9, 10, 16].

► **Definition 4.** For a functor $F : \mathbf{Set} \rightarrow \mathbf{Set}$, a lax extension of F to $\mathbf{Rel}$ consists of an indexed family of functions $\bar{T} : \mathbf{Rel}(X, Y) \rightarrow \mathbf{Rel}(TX, TY)$ satisfying the following axioms:

1. $\bar{T}$ is monotone: if $R \leq R' : X \multimap Y$, then $\bar{T}(R) \leq \bar{T}(R')$;
2. for composable relations $R : X \multimap Y, S : Y \multimap Z$, $\bar{T}(S) \circ \bar{T}(R) \leq \bar{T}(S \circ R)$;
3. for all functions $f : X \rightarrow Y$; $(T(f))_* \leq \bar{T}(f_*)$ and $(T(f))^* \leq \bar{T}(f^*)$.

If $\bar{T}$ preserves identities strictly, *i.e.* $\text{id}_{TX} = \bar{T}(\text{id}_X)$, we say that it is a normal lax extension (also called flat). If conditions 2. and 3. are equalities, then we recover exactly the notion of monotone strict extension.

¹ While $\mathbf{Rel}$ is not a cartesian closed category, we still use the terminology of exponentiation because $\mathbf{Rel}$ is cartesian closed as a double category [32].

Lax extensions are not unique, there can be more than one lax extension for the same functor:

► **Example 5.** In the case of the power set functor $\mathcal{P} : \mathbf{Set} \rightarrow \mathbf{Set}$, for a relation $R : X \multimap Y$, the following mappings

$$\begin{aligned}\mathcal{L}_1(R) &:= \{(A, B) \in \mathcal{P}(X) \times \mathcal{P}(Y) \mid \forall x \in A, \exists y \in B, (x, y) \in R\} \\ \mathcal{L}_2(R) &:= \{(A, B) \in \mathcal{P}(X) \times \mathcal{P}(Y) \mid \forall y \in B, \exists x \in A, (x, y) \in R\} \\ \mathcal{L}_3(R) &:= \mathcal{L}_1(R) \cap \mathcal{L}_2(R)\end{aligned}$$

are all lax extensions. The last one corresponds to the canonical Barr extension in Section 3.1.

In the parametrized case, we are also interested in the lax version where the strict equalities in Definition 3 are relaxed to inequalities.

► **Definition 6.** For a functor $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$, a parametrized lax extension consists of a family of functions: $\bar{T} : \mathbf{Rel}(A, B) \times \mathbf{Rel}(C, D) \times \mathbf{Rel}(X, Y) \rightarrow \mathbf{Rel}(T(A, C, X), T(B, D, Y))$ indexed by A, B, C, D, X, Y that satisfies the following axioms:

1. it is antitone in the first argument and monotone in the last two arguments: for all $M' \leq M : A \multimap B$, $N \leq N' : C \multimap D$ and $R \leq R' : X \multimap Y$, $\bar{T}(M, N, R) \leq \bar{T}(M', N', R')$
2. lax composition: for all $M : A_1 \multimap A_2$, $M' : A_2 \multimap A_3$, $N : B_1 \multimap B_2$, $N' : B_2 \multimap B_3$, $R : X_1 \multimap X_2$ and $R' : X_2 \multimap X_3$, $\bar{T}(M', N', R') \circ \bar{T}(M, N, R) \leq \bar{T}(M' \circ M, N' \circ N, R' \circ R)$
3. for all $f : A \rightarrow B$, $g : C \rightarrow D$ and $h : X \rightarrow Y$, $(T(f, g, h))_* \leq \bar{T}(f^*, g_*, h_*)$ and $(T(f, g, h))^* \leq \bar{T}(f_*, g^*, h^*)$.

If $\bar{T}$ preserves identities strictly ($\text{id}_{T(A, B, X)} = \bar{T}(\text{id}_A, \text{id}_B, \text{id}_X)$), we say that it is a normal parametrized lax extension. If conditions 2. and 3. are equalities, then we obtain a monotone strict extension. If $\bar{T}$ commutes with the transpose operation as below, it is called symmetric:

4. for all $M : A \multimap B$, $N : C \multimap D$ and $R : X \multimap Y$, $(\bar{T}(M, N, R))^t = \bar{T}(M^t, N^t, R^t)$

► **Example 7.** The underlying functor of the parametrized state monad (Example 2.1) has a parametrized lax extension $\bar{T}$ which we describe explicitly: for relations $M : A \multimap B$, $N : C \multimap D$ and $R : X \multimap Y$, $\bar{T}(M, N, R) := (N \times R)^M = \{(\varphi, \psi) \mid \forall (a, b) \in M, (\varphi(a), \psi(b)) \in N \times R\} : (C \times X)^A \multimap (D \times Y)^B$.

3 The Parametrized Barr extensions

3.1 Extending functors from Set to Rel

The span representation of relations

The Barr extension (also called canonical relation lifting) is based on the universal representation of relations as pairs of functions which we recall here. A relation $R : X \multimap Y$ induces a pair of projection functions $\pi_1^R : \mathbf{gr}(R) \rightarrow X$ and $\pi_2^R : \mathbf{gr}(R) \rightarrow Y$ where $\mathbf{gr}(R) := \{(x, y) \in X \times Y \mid (x, y) \in R\} \leq X \times Y$ is called the *graph* or *tabulator* of R . These two functions completely describe the relation in the following sense:

1. tabulator universal property: for every other span of functions $(h : A \rightarrow X, k : A \rightarrow Y)$ satisfying $h^*; k_* \leq R$, there exists a unique function $u : A \rightarrow \mathbf{gr}(R)$ such that $\pi_1^R \circ u = h$ and $\pi_2^R \circ u = k$.
2. tabulators are strong: we can recover the original relation $R = (\pi_1^R)^*; (\pi_2^R)_*$.

The cospan representation of relations

There is a dual cospan representation of relations with the notion of *collage* or *cotabulator*. It is used to define the Barr extension of quantale-enriched relations, we will only define it for ordinary relations on sets and make use of its universal property in Section 4. For a relation $R : X \multimap Y$, if we take the pushout of the span $(\pi_1^R : \mathbf{gr}(R) \rightarrow X, \pi_2^R : \mathbf{gr}(R) \rightarrow Y)$, we obtain a pair of functions $\text{in}_1^R : X \rightarrow \mathbf{cl}(R)$ and $\text{in}_2^R : Y \rightarrow \mathbf{cl}(R)$ where $\mathbf{cl}(R) := X + Y / \sim$ is the quotient of the disjoint union $X + Y$ by the equivalence relation $\sim$ generated by $(2, y) \sim (1, x)$ for $(x, y) \in R$. It verifies the following:

1. cotabulator universal property: for every other cospan of functions $(h : X \rightarrow A, k : Y \rightarrow A)$ satisfying $R \leq h_*; k^*$, there exists a unique function $u : \mathbf{cl}(R) \rightarrow A$ such that $u \circ \text{in}_1^R = h$ and $u \circ \text{in}_2^R = k$.

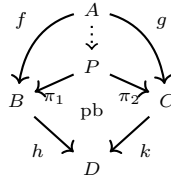
Rel does not have strong cotabulators, we only have $R \leq (\text{in}_1^R)_*; (\text{in}_2^R)^*$ in general. Relations for which the cotabulator is strong are called *difunctional* and they subsume equivalence relations [35, 21, 17].

The Barr extension

The representation of a relation by a universal span of functions is at the heart of the Barr extension construction. The Barr extension $\bar{T} : \mathbf{Rel}(X, Y) \rightarrow \mathbf{Rel}(TX, TY)$ is the operator mapping a relation $R : X \multimap Y$ to the relation $\bar{T}(R) := (T\pi_1^R)^*; (T\pi_2^R)_* : TX \multimap TY$.

Without any additional assumptions on T , the operator $\bar{T}$ is monotone (axiom (1) of Definition 4) and satisfies axiom (3) strictly. However, it is not in general a lax extension since it does not always satisfy the lax composition axiom (2), it preserves composition in the reverse op-lax direction instead. To obtain a lax-extension, we need to assume additional properties on T based on the notion of *weak pullback*:

► **Definition 8.** A commutative square $h \circ f = k \circ g$ in **Set** as below is a *weak pullback* if the universal map from A to the pullback of h and k is split epi.



This notion (also called Beck-Chevalley property) plays a central role in the theory of lax extensions as weak pullback squares can be equivalently characterized by the equality $g_* \circ f^* = k^* \circ h_*$ in **Rel** allowing to commute the left $(-)_*$ and right adjoints $(-)^*$.

► **Theorem 9** ([8, 43, 11]). For a functor $T : \mathbf{Set} \rightarrow \mathbf{Set}$, T preserves weak-pullbacks if and only if the Barr extension is a monotone strict extension and it is the unique one.

Before considering the extension of functors $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$, we first explain how to extend functors $\mathbf{Set}^{\text{op}} \rightarrow \mathbf{Set}$ and $\mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ to isolate the different issues that each generalization entails.

3.2 Extending contravariant functors

For a functor $T : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{Set}$, we define its Barr extension as the operator mapping a relation $R : X \multimap Y$ to the relation $\bar{T}(R) := (T\pi_1^R)^*; (T\pi_2^R)_* : TX \multimap TY$. The operator $\bar{T}$

is antitone with respect to the inclusion of relations, is normal and verifies $\overline{T}(f^*) = (Tf)_*$ and $\overline{T}(f_*) = (Tf)^*$ for a function f . The important point is that it always preserves composition of relations laxly without requiring additional conditions on T . To obtain a strict extension, it suffices for T to map weak pullback squares to weak pullback squares in the following sense:

$$\begin{array}{ccc}
 & A & \\
 p_1 \nearrow & & \searrow f \\
 P & \text{wpb} & C \\
 p_2 \searrow & & \nearrow g \\
 & B &
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 & TA & \\
 T(f) \nearrow & & \searrow T(p_1) \\
 TC & \text{wpb} & TP \\
 T(g) \searrow & & \nearrow T(p_2) \\
 & TB &
 \end{array}$$

Note however that this condition is quite strong and that the typical examples built from the functor $(-) \Rightarrow A : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{Set}$ do not satisfy this property as the exponentiation functor is only lax in $\mathbf{Rel}$.

3.3 Extending multivariable functors

We consider here functors of the form $T : \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ with two arguments but this can be generalized to functors $\mathbf{Set}^n \rightarrow \mathbf{Set}$. We define the *parametrized Barr extension* of T as the operator $\overline{T}$ mapping a pair of relations $(N : C \multimap D, R : X \multimap Y)$ to $\overline{T}(N, R) := (T(\pi_1^N, \pi_1^R))^* ; T(\pi_2^N, \pi_2^R)_* : T(C, X) \multimap T(D, Y)$. It verifies the expected properties which we do not prove as they are a special case of Theorem 10 where we have an additional contravariant variable. Instead, we want to emphasize the fact that in the multivariable case, we have other possibilities for the choice of extensions we can define:

$$\begin{aligned}
 \mathcal{L}_1(N, R) &:= (T(\pi_1^N, \text{id}))^* ; (T(\pi_2^N, \text{id}))_* ; (T(\text{id}, \pi_1^R))^* ; (T(\text{id}, \pi_2^R))_* \\
 \mathcal{L}_2(N, R) &:= (T(\text{id}, \pi_1^R))^* ; (T(\text{id}, \pi_2^R))_* ; (T(\pi_1^N, \text{id}))^* ; (T(\pi_2^N, \text{id}))_*
 \end{aligned}$$

We have that $\overline{T}(N, R) \leq \mathcal{L}_1(N, R)$ and $\overline{T}(N, R) \leq \mathcal{L}_2(N, R)$. While $\overline{T}$ preserves composition opaxly in both variables, $\mathcal{L}_1$ and $\mathcal{L}_2$ preserve composition opaxly in each variable separately: $\mathcal{L}_i(N' \circ N, \text{id}) \leq \mathcal{L}_i(N', \text{id}) \circ \mathcal{L}_i(N, \text{id})$ and $\mathcal{L}_i(\text{id}, R' \circ R) \leq \mathcal{L}_i(\text{id}, R') \circ \mathcal{L}_i(\text{id}, R)$.

Recall that for an oplax 2-functor, we have both $T(f, g) \leq T(f, \text{id}) \circ T(\text{id}, g)$ and $T(f, g) \leq T(\text{id}, g) \circ T(f, \text{id})$ but $T(f, \text{id}) \circ T(\text{id}, g)$ and $T(\text{id}, g) \circ T(f, \text{id})$ are not in general comparable. For multivariable oplax 2-functors, the equality $T(f, \text{id}) \circ T(\text{id}, g) = T(\text{id}, g) \circ T(f, \text{id})$ (or coherent isomorphism if we are not in the posetal case) corresponds to the intermediate notion of cubical or quasi-functors [20, 24]) between strict (or pseudo) 2-functors and oplax ones. If for every $f : A \rightarrow B$, $g : C \rightarrow D$, the following interchange square is a weak-pullback

$$\begin{array}{ccccc}
 & & T(A, C) & & \\
 T(f, \text{id}) \swarrow & & & \searrow T(\text{id}, g) & \\
 & T(B, C) & \text{wpb} & T(A, D) & \\
 & \searrow & & \swarrow & \\
 T(\text{id}, g) & & T(B, D) & & T(f, \text{id})
 \end{array}$$

then $\overline{T} = \mathcal{L}_1 = \mathcal{L}_2$ which intuitively means that we can commute the order in which we do the Barr extension. This issue will be relevant in the mixed variance case where we have to consider the covariant and contravariant variables separately.

3.4 Combining both

For a functor $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$, we obtain a lattice of *parametrized Barr extensions*: for a triple of relations $(M : A \multimap B, N : C \multimap D, R : X \multimap Y)$, we define

$$\begin{aligned}\mathcal{L}_1(M, N, R) &:= (T(\text{id}, \pi_1, \pi_1))^*; (T(\pi_1, \text{id}, \text{id}))_*; (T(\pi_2, \text{id}, \text{id}))^*; (T(\text{id}, \pi_2, \pi_2))_* \\ \mathcal{L}_2(M, N, R) &:= (T(\pi_1, \text{id}, \text{id}))_*; (T(\pi_2, \pi_1, \pi_1))^*; (T(\text{id}, \pi_2, \pi_2))_* \\ \mathcal{L}_3(M, N, R) &:= (T(\text{id}, \pi_1, \pi_1))^*; (T(\pi_1, \pi_2, \pi_2))_*; (T(\pi_2, \text{id}, \text{id}))^* \\ \mathcal{L}_4(M, N, R) &:= (T(\pi_1, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1, \pi_1))^*; (T(\text{id}, \pi_2, \pi_2))_*; ((T(\pi_2, \text{id}, \text{id}))^*)\end{aligned}$$

These operators are related by the following inclusions: $\mathcal{L}_1 \leq \mathcal{L}_2 \leq \mathcal{L}_4$ and $\mathcal{L}_1 \leq \mathcal{L}_3 \leq \mathcal{L}_4$.

► **Theorem 10.** *For a functor T as above and $i \in \{1, 2, 3, 4\}$, the Barr extension $\mathcal{L}_i$ verifies:*

1. *it is normal: for all A, B, X , $\text{id}_{T(A, B, X)} = \mathcal{L}_i(\text{id}_A, \text{id}_B, \text{id}_X)$*
2. *it is antitone in the first argument and monotone in the last two arguments: for all $M' \leq M : A \multimap B$, $N \leq N' : C \multimap D$ and $R \leq R' : X \multimap Y$, $\mathcal{L}_i(M, N, R) \leq \mathcal{L}_i(M', N', R')$*
3. *it preserves composition laxly in the first argument: for all $M : A_1 \multimap A_2$, $M' : A_2 \multimap A_3$, $\mathcal{L}_i(M', \text{id}, \text{id}) \circ \mathcal{L}_i(M, \text{id}, \text{id}) \leq \mathcal{L}_i(M' \circ M, \text{id}, \text{id})$*
4. *it preserves composition op-laxly in the last two arguments: for all $N : B_1 \multimap B_2$, $N' : B_2 \multimap B_3$, $R : X_1 \multimap X_2$ and $R' : X_2 \multimap X_3$, $\mathcal{L}_i(\text{id}, N' \circ N, R' \circ R) \leq \mathcal{L}_i(\text{id}, N', R') \circ \mathcal{L}_i(\text{id}, N, R)$*
5. *for all $f : A \rightarrow B$, $g : C \rightarrow D$ and $h : X \rightarrow Y$, $(T(f, g, h))_* = \mathcal{L}_i(f^*, g^*, h^*)$ and $(T(f, g, h))^* = \mathcal{L}_i(f_*, g_*, h_*)$*
6. *it is symmetric: for all $M : A \multimap B$, $N : C \multimap D$ and $R : X \multimap Y$, $(\mathcal{L}_i(M, N, R))^t = \mathcal{L}_i(M^t, N^t, R^t)$*
7. *for any lax extension $\mathcal{L}$ of T and relations M, N, R , we have $\mathcal{L}_i(M, N, R) \leq \mathcal{L}(M, N, R)$.*

To obtain that $\mathcal{L}_i$ preserves composition laxly in the last two arguments, we need the functor T to preserve weak pullbacks in the last two arguments.

► **Lemma 11.** *For $i \in \{1, 2, 3, 4\}$, if for every weak pullback squares as on the left below,*

$$\begin{array}{ccc} \begin{array}{ccccc} & A & & & \\ f \swarrow & & \searrow g & & \\ B & & & C & \\ & \text{wpb} & & & \\ h \searrow & & \swarrow k & & \\ & D & & & \end{array} & \begin{array}{ccccc} & W & & & \\ i \swarrow & & \searrow j & & \\ X & & & Y & \\ & \text{wpb} & & & \\ l \searrow & & \swarrow m & & \\ & Z & & & \end{array} & \Rightarrow & \begin{array}{ccc} T(\text{id}, f, i) & \xrightarrow{T(U, A, W)} & T(\text{id}, g, j) \\ & \searrow & \swarrow \\ T(U, B, X) & \text{wpb} & T(U, C, Y) \\ & \searrow & \swarrow \\ T(\text{id}, h, l) & \xrightarrow{T(U, D, Z)} & T(\text{id}, k, m) \end{array} \end{array}$$

their image by T as on the right is a weak pullback, then for all $N : B_1 \multimap B_2$, $N' : B_2 \multimap B_3$, $R : X_1 \multimap X_2$ and $R' : X_2 \multimap X_3$, $\mathcal{L}_i(\text{id}, N', R') \circ \mathcal{L}_i(\text{id}, N, R) \leq \mathcal{L}_i(\text{id}, N' \circ N, R' \circ R)$.

However, this is not sufficient to obtain a parametrized lax extension as in Definition 6. The important point is the question of interchange of variables we explained in the previous section when dealing with two covariant variables. Here, we consider the interaction between the contravariant variable and the covariant ones. We do not obtain $\mathcal{L}_i(M', N', R') \circ \mathcal{L}_i(M, N, R) \leq \mathcal{L}_i(M' \circ M, N' \circ N, R' \circ R)$ from the inclusions $\mathcal{L}_i(\text{id}, N', R') \circ \mathcal{L}_i(\text{id}, N, R) \leq \mathcal{L}_i(\text{id}, N' \circ N, R' \circ R)$ and $\mathcal{L}_i(M', \text{id}, \text{id}) \circ \mathcal{L}_i(M, \text{id}, \text{id}) \leq \mathcal{L}_i(M' \circ M, \text{id}, \text{id})$. In order to obtain a parametrized lax extension, it suffices to have the following:

► **Proposition 12.** *We say that T satisfies the strong interchange property if for every $f : A \rightarrow B$, $g : C \rightarrow D$ and $h : X \rightarrow Y$, the following square is a weak pullback*

$$\begin{array}{ccccc}
& T(B, C, X) & & & \\
T(f, \text{id}, \text{id}) \swarrow & & \text{wpb} & & \searrow T(\text{id}, g, h) \\
T(A, C, X) & & & & T(B, D, Y) \\
T(\text{id}, g, h) \searrow & & & & \swarrow T(f, \text{id}, \text{id}) \\
& T(A, D, Y) & & &
\end{array}$$

If T preserves weak pullbacks in its last two arguments as in Lemma 11 and satisfies the strong interchange property, then all the operators $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ and $\mathcal{L}_4$ become equal and are parametrized lax extensions.

Intuitively, the strong interchange property means that the contravariant variable is independent from the covariant variables. For example, if $T(A, C, X) = F(A) \times G(C, X)$ for some functors $F : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{Set}$ and $G : \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$, then T satisfies this property. An example which does not satisfy this property is the underlying functor of the state monad $T(A, C, X) = (C \times X)^A$ for which the operators $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ and $\mathcal{L}_4$ are all distinct, and only $\mathcal{L}_4$ is a parametrized lax extension. The functor T satisfies a weaker form of interchange:

► **Proposition 13.** *We say that T satisfies the weak interchange property if for every spans $(g : P \rightarrow C, h : P \rightarrow D)$ and $(u : Q \rightarrow X, v : Q \rightarrow Y)$ and split epi $f : A \rightarrow B$, the canonical inclusion $(T(\text{id}, g, u))^*; (T(f, \text{id}, \text{id}))_*; (T(f, \text{id}, \text{id}))^*; (T(\text{id}, h, v))_* \leq (T(f, \text{id}, \text{id}))_*; (T(\text{id}, g, u))^*; (T(\text{id}, h, v))_*; (T(f, \text{id}, \text{id}))^*$ is an equality. If T preserves weak pullbacks in its last two arguments as in Lemma 11 and satisfies the weak interchange property, then $\mathcal{L}_4$ is a parametrized lax extension.*

We spelled out the results for functors of type $\mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$ because they correspond to the underlying functor of parametrized monads, but the constructions presented above extend to arbitrary mixed variance functor $\mathbf{Set}^{v_1} \times \dots \times \mathbf{Set}^{v_n} \rightarrow \mathbf{Set}$ where $\mathbf{Set}^{v_i}$ is either $\mathbf{Set}$ or $\mathbf{Set}^{\text{op}}$. For example, for a functor $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$, we define $\mathcal{L}_4(M, N, O, P, Q)$ as

$$(T(\pi_1, \text{id}, \pi_1, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1, \text{id}, \pi_1, \pi_1))^*; (T(\text{id}, \pi_2, \text{id}, \pi_2, \pi_2))^*; (T(\pi_2, \text{id}, \pi_2, \text{id}, \text{id}))^*$$

and we obtain analogous results.

3.5 Extending the monad structure

Recall that for an ordinary natural transformation $\alpha : F \Rightarrow G : \mathbf{Set} \rightarrow \mathbf{Set}$ with lax extensions $\overline{F}$ and $\overline{G}$ respectively, its canonical extension $\overline{\alpha} : \overline{F} \Rightarrow \overline{G}$ has components $(\overline{\alpha})_X := (\alpha_X)_* : F(X) \rightarrow G(X)$. If $\overline{F}$ and $\overline{G}$ are the Barr extensions of F and G , then $\overline{\alpha}$ is in general an oplax transformation: *i.e.* for a relation $R : X \rightarrowtail Y$, $\overline{F}(R); (\alpha_Y)_* \leq (\alpha_X)_*; \overline{G}(R)$. To obtain (strict) naturality, we need α to be a weakly cartesian transformation, *i.e.* that all its naturality squares are weak pullbacks.

► **Definition 14.** *A lax extension of a strong monad $(T, \mu, \eta, \text{str})$ on $\mathbf{Set}$ to $\mathbf{Rel}$ consists a lax extension $\overline{T}$ of T , as in Definition 4, satisfying the following axioms:*

1. *for all $R : X \rightarrowtail Y$, $(\eta_Y)_* \circ R \leq \overline{T}(R) \circ (\eta_X)_*$;*
 2. *for all $R : X \rightarrowtail Y$, $(\mu_Y)_* \circ \overline{T}(\overline{T}(R)) \leq \overline{T}(R) \circ (\mu_X)_*$.*
 3. *for all $R : A \rightarrowtail B$ and $S : X \rightarrowtail Y$, $(\text{str}_{B,Y})_* \circ (R \times \overline{T}(S)) \leq \overline{T}(R \times S) \circ \text{str}_{A,X}$.*
- If $\overline{T}$ is a strict extension for T and conditions 1. – 3. are equalities, we say that $(\overline{T}, \mu_*, \eta_*, \text{str}_*)$ is a strict extension for $(T, \mu, \eta, \text{str})$.*

Notable examples of monads that do not have a strict extension are the powerset monad, the probability distribution monad, the ultrafilter monad and the global state monad. In the parametrized case, the unit and multiplication are transformations which are both natural in some variables and dinatural in others. We therefore need to understand what it means to extend a dinatural transformation. Recall that for ordinary 1-categorical functors $F, G : \mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{D}$, a dinatural transformation $F \Rightarrow G$ is a family of morphisms $\theta_X : F(X, X) \rightarrow G(X, X)$ in $\mathcal{D}$ such that for all $f : X \rightarrow Y$ in $\mathcal{C}$, the hexagon axiom holds: $G(\text{id}, f) \circ \theta_X \circ F(f, \text{id}) = G(f, \text{id}) \circ \theta_Y \circ F(\text{id}, f)$. In the general 2-categorical setting, the notion of 2-dinaturality takes into account the various possible dualities and we have transformations between functors of type $\mathcal{C}^{\text{coop}} \times \mathcal{C}^{\text{op}} \times \mathcal{C}^{\text{co}} \times \mathcal{C} \rightarrow \mathcal{D}$ which satisfy coherences with 2-morphisms [41]. In our case, we are only working with posetal 2-categories so the situation is simplified. For 2-functors $F, G : \mathbf{Rel}^{\text{co}} \times \mathbf{Rel} \rightarrow \mathbf{Rel}$, a dinatural transformation $F \Rightarrow G$ is simply a family of relations $\theta_X : F(X, X) \dashrightarrow G(X, X)$ such that for all $R : X \dashrightarrow Y$, $\theta_X; G(R, R) = F(R, R); \theta_Y$. It is oplax if $F(R, R); \theta_Y \leq \theta_X; G(R, R)$ and lax if we have the reverse inclusion.

► **Definition 15.** For a parametrized monad (T, η, μ) , a parametrized lax extension consists of a lax extensions $\bar{T}$ as in Definition 6, such that the relations $(\eta_{A,X})_*$ and $(\mu_{A,C,E,X})_*$ verify:

1. for all $M : A \dashrightarrow B$ and $R : X \dashrightarrow Y$, $(\eta_{B,Y})_* \circ R \leq \bar{T}(M, M, R) \circ (\eta_{A,X})_*$
2. for all $M : A \dashrightarrow B$, $N : C \dashrightarrow D$, $O : E \dashrightarrow F$ and $R : X \dashrightarrow Y$, $(\mu_{B,D,F,Y})_* \circ \bar{T}(M, N, \bar{T}(N, O, R)) \leq \bar{T}(M, O, R) \circ (\mu_{A,C,E,X})_*$.

If the parametrized monad is equipped with a strength str , then a strong lax extension for T is a lax extension as above satisfying the additional axiom:

3. for all $M : A \dashrightarrow B$, $N : C \dashrightarrow D$, $R : X \dashrightarrow Y$ and $S : Z \dashrightarrow W$, $(\text{str}_{B,D,Y,W})_* \circ (R \times \bar{T}(M, N, S)) \leq \bar{T}(M, N, R \times S) \circ (\text{str}_{A,C,X,Z})_*$.

A strict parametrized extension consists of a strict extension $\bar{T}$ as in Definition 3 for which the inequalities 1. – 3. above are equalities.

► **Proposition 16.** For a strong parametrized monad $(T, \eta, \mu, \text{str})$, if $\mathcal{L}_4$ is a parametrized lax extension for T , then $(\mathcal{L}_4, \eta_*, \mu_*, \text{str}_*)$ is a parametrized lax extension for $(T, \eta, \mu, \text{str})$ as in Definition 15.

Note that if any of $\mathcal{L}_1, \mathcal{L}_2$ or $\mathcal{L}_3$ are lax extensions, they must be equal to $\mathcal{L}_4$ by Theorem 10.7.

► **Example 17.** The parametrized state monad has underlying functor T as in Example 2.1. For sets A, X , the unit has components $\eta_{A,X} : X \rightarrow (A \times X)^A$ given by the function $x \mapsto \lambda a. (a, x)$. For sets A, B, C and X , the component of the multiplication $\mu_{A,B,C,X} : T(A, B, T(B, C, X)) \rightarrow T(A, C, X)$ maps $\varphi \in (B \times (C \times X)^B)^A$ to $\lambda a. (\pi_2(\varphi(a)), (\pi_1(\varphi(a))))$. It is a strong monad: for sets A, B, X and Y , the component of the strength $\text{str}_{A,B,X,Y} : X \times (B \times Y)^A \rightarrow (B \times X \times Y)^A$ maps $(x, \psi) \in X \times (B \times Y)^A$ to $\lambda a. (\pi_1(\psi(a)), x, \pi_2(\psi(a)))$. We obtain a parametrized lax extension $(\mathcal{L}_4, \eta_*, \mu_*, \text{str}_*)$, it corresponds to the monad structure induced from the parametrized adjunction for the (double) cartesian closed structure of $\mathbf{Rel}$.

4 Parametrized behavioral equivalence

Many coinductive preorders and equivalences to reason on systems can be described coalgebraically. In this paper, we will focus on the three following notions: Aczel-Mendler bisimulation, $\bar{F}$ -bisimulation (for some lax extension $\bar{F}$) and behavioral equivalence. An important problem is how to develop sound and/or complete proof principles to establish

behavioral equivalence of states, which is often difficult to show directly in practice. This problem has been well-studied for set-functors [1], and it has also been addressed in the quantale-enriched setting [46, 16].

4.1 Non-parametrized bisimulation and behavioral equivalence

For an endofunctor $F : \mathbf{Set} \rightarrow \mathbf{Set}$ and coalgebras $(X, \alpha : X \rightarrow F(X))$, $(Y, \beta : Y \rightarrow F(Y))$, a relation $R : X \multimap Y$ is a behavioral equivalence if there exists a coalgebra $(Z, \zeta : Z \rightarrow F(Z))$ and coalgebra morphisms $f : (X, \alpha) \rightarrow (Z, \zeta)$, $g : (Y, \beta) \rightarrow (Z, \zeta)$ as on the left below such that the diagram on the right is a weak pullback:

$$\begin{array}{ccc}
 X & \xrightarrow{f} & Z \\
 \alpha \downarrow & & \downarrow \zeta \\
 F(X) & \xrightarrow{F(f)} & F(Z) \\
 & \nearrow F(g) & \\
 & F(Y) &
 \end{array}
 \quad
 \begin{array}{ccc}
 & \mathbf{gr}(R) & \\
 \pi_1^R \swarrow & & \searrow \pi_2^R \\
 X & \xrightarrow{\text{wpb}} & Y \\
 f \searrow & & \swarrow g \\
 & Z &
 \end{array}$$

Two states $x \in X$ and $y \in Y$ are *behaviorally equivalent* if there exists a relation R as above such that $(x, y) \in R$. An important characterization of behavioral equivalence is that if F has a final coalgebra $\text{out}_F : \nu F \rightarrow F(\nu F)$, then two states x and y are behaviorally equivalent if and only if they have the same *final semantics* i.e. they are mapped to equal states in the final coalgebra. We typically restrict to the case where $\alpha = \beta$ and R is an actual equivalence relation, not just a difunctional relation as above.

Two states $x \in X$ and $y \in Y$ are *Aczel-Mendler bisimilar* (AM-bisimilar) if there exists a relation $R : X \multimap Y$ and a coalgebra $\gamma : \mathbf{gr}(R) \rightarrow F(\mathbf{gr}(R))$ such that $(x, y) \in R$ and the projection maps $\pi_1 : \mathbf{gr}(R) \rightarrow X$ and $\pi_2 : \mathbf{gr}(R) \rightarrow Y$ are coalgebra morphisms.

► **Definition 18.** For coalgebras (X, α) and (Y, β) as above, a relation $R : X \multimap Y$ is *sound* for behavioral equivalence if for any $(x, y) \in R$, the states x and y are behaviorally equivalent. It is *complete* if the reverse inclusion holds and it is *fully abstract* if both inclusions hold.

► **Theorem 19** ([37]). *AM-bisimilarity is sound for behavioral equivalence; if F preserves weak-pullbacks, then it is also fully abstract.*

We are mainly interested in this paper in the notion of equivalence induced by a lax extension $\bar{F}$ of F , two states $x \in X$ and $y \in Y$ are $\bar{F}$ -bisimilar if there exists a relation $R : X \multimap Y$ such that $(x, y) \in R$ and R is included in $\alpha_*; \bar{F}(R); \beta^*$. Bisimulations induced by lax extensions subsume AM-bisimulations.

► **Proposition 20.** *If R is an AM-bisimulation, then it is a $\bar{F}$ -bisimulation for any lax extension $\bar{F}$.*

► **Theorem 21** ([30, 18]). *For a lax extension $\bar{F}$, $\bar{F}$ -bisimilarity is complete for behavioral equivalence and if $\bar{F}$ is normal, then $\bar{F}$ -bisimilarity is fully abstract.*

4.2 Parametrized bisimulation from lax extensions

We fix a parametrized lax-extension $\bar{T}$ of a functor $T : \mathbf{Set}^{\text{op}} \times \mathbf{Set} \times \mathbf{Set} \rightarrow \mathbf{Set}$. We first need to determine what is the notion of coalgebra in this setting. In this paper, we do not follow the approach of [4] where a parametrized coalgebra would be a functor $c : \mathbf{Set}^{\text{op}} \rightarrow \mathbf{Set}$ equipped with a family of functions $\alpha_{A,C} : c(C) \rightarrow T(A, C, c(C))$ natural in A and dinatural in C . This definition requires the state space to depend on the output parameter, which

is too restrictive for some of the examples we consider. Moreover, we want to be able to consider a single coalgebra $\alpha : X \rightarrow T(A, C, X)$ at a time, together with relations between parameters chosen by the user, without requiring those relations to be defined for all sets. In our setting, we fix coalgebras $\alpha : X \rightarrow T(A, C, X)$ and $\beta : Y \rightarrow T(B, D, Y)$ and relations $M : A \rightarrowtail B$ and $N : C \rightarrowtail D$ on the parameters. We want to understand what it means for a relation $R : X \rightarrowtail Y$ between states to be a bisimulation with respect to (M, N) . The most straightforward generalization to our setting is the lax-extension notion of bisimulation:

► **Definition 22.** *A relation $R : X \rightarrowtail Y$ is a $\bar{T}$ -bisimulation w.r.t (M, N) if it satisfies $R \leq \alpha_*; \bar{T}(M, N, R); \beta^*$.*

Since $\bar{T}$ is monotone in its third argument, the mapping $R \mapsto \alpha_*; \bar{T}(M, N, R); \beta^*$ has a greatest fixpoint which we call $\bar{T}$ -bisimilarity w.r.t (M, N) . Note that using the formalism of [33], $\bar{T}(M, N, -)$ is an example of *relational connector* between the functors $T(A, C, -)$ and $T(B, D, -)$ and the notion of $\bar{T}$ -bisimilarity w.r.t (M, N) coincides with the notion of bisimilarity induced by this relational connector.

Assume now that $\beta = \alpha$ (so that $M : A \rightarrowtail A$ and $N : C \rightarrowtail C$), in the non-parametrized case, the axioms of lax extension ensure that $\bar{T}$ -bisimulations are closed under composition and identity which implies that $\bar{T}$ -bisimilarity is a preorder. In our case, we need additional conditions on M and N :

► **Lemma 23.** *If $M \leq \text{id}_A$ and $\text{id}_C \leq N$, then id_X is a $\bar{T}$ -bisimulation w.r.t (M, N) . If $M \leq M \circ M$, $N \circ N \leq N$ and R is a $\bar{T}$ -bisimulation w.r.t (M, N) , then so is $R \circ R$.*

In practice, the conditions on M are too restrictive and we typically want M to be a preorder. While $\bar{T}$ -bisimilarity will not be a preorder for arbitrary coalgebras α , it will hold for the specific cases we are interested in.

To obtain that $\bar{T}$ -bisimilarity is an equivalence relation, we need it to be symmetric and it suffices to require that $\bar{T}$ commutes with the operation of taking the transpose of relations:

► **Lemma 24.** *Assume that $M^t = M$, $N^t = N$ and $\bar{T}$ is symmetric (axiom 4. in Definition 6). If R is a $\bar{T}$ -bisimulation w.r.t (M, N) , then so is R^t .*

4.3 Parametrized AM-bisimulation

A relation $R : X \rightarrowtail Y$ is an *AM-bisimulation w.r.t (M, N)* if there exists a coalgebra $\gamma : \mathbf{gr}(R) \rightarrow T(\mathbf{gr}(M), \mathbf{gr}(N), \mathbf{gr}(R))$ such that the two pentagons below commute:

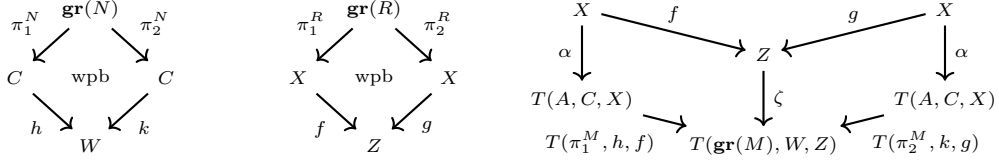
$$\begin{array}{ccccc}
 X & \xleftarrow{\pi_1^R} & \mathbf{gr}(R) & \xrightarrow{\pi_2^R} & Y \\
 \alpha \downarrow & & \downarrow \gamma & & \downarrow \beta \\
 T(A, C, X) & & T(\mathbf{gr}(M), \mathbf{gr}(N), \mathbf{gr}(R)) & & T(B, D, Y) \\
 \searrow T(\pi_1^M, \text{id}, \text{id}) & & \swarrow T(\text{id}, \pi_1^N, \pi_1^R) \quad \searrow T(\text{id}, \pi_2^N, \pi_2^R) & & \swarrow T(\pi_2^M, \text{id}, \text{id}) \\
 & T(\mathbf{gr}(M), C, X) & & & T(\mathbf{gr}(M), D, Y)
 \end{array}$$

This notion can be defined without any restrictions on the parameter relations M and N , and from the axioms of parametrized lax extension, we obtain soundness similarly to Proposition 20:

► **Lemma 25.** *If R is an AM-bisimulation w.r.t (M, N) , then it is a $\bar{T}$ -bisimulation w.r.t (M, N) .*

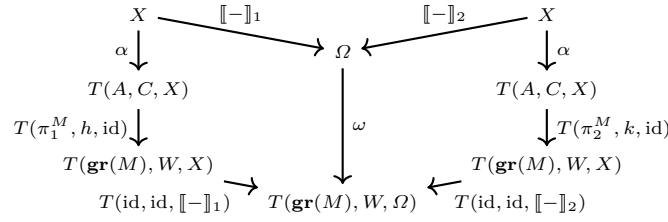
4.4 Parametrized behavioral equivalence

We consider the case where $\alpha = \beta$ and $M : A \multimap A$ and $N : C \multimap C$ are endorelations. For the notion of (M, N) -behavioral equivalence, we also require the relation N to be difunctional: there exists $h : C \rightarrow W$ and $k : C \rightarrow W$ such that the left square below is a weak-pullback



In this setting, two states $x, y \in X$ are (M, N) -behaviorally equivalent if there exist a relation $R : X \multimap X$ and functions $f : X \rightarrow Z$, $g : X \rightarrow Z$ such that the middle square above is a weak-pullback, $(x, y) \in R$, and there exists a coalgebra $\zeta : Z \rightarrow T(\text{gr}(M), W, Z)$ such that the two squares on the right commute.

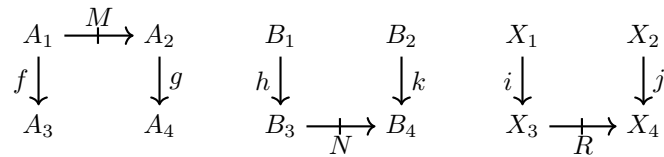
► **Lemma 26.** *For α and (M, N) as above, if the functor $T(\text{gr}(M), W, -) : \mathbf{Set} \rightarrow \mathbf{Set}$ has a final coalgebra $\omega : \Omega \rightarrow T(\text{gr}(M), W, \Omega)$, then two states $(x, y) \in X \times X$ are behaviorally equivalent if and only if $\llbracket x \rrbracket_1 = \llbracket y \rrbracket_2$ where $\llbracket - \rrbracket_1$ and $\llbracket - \rrbracket_2$ are the two unique coalgebra morphisms as below:*



► **Proposition 27.** *If T preserves weak-pullbacks in the last two variables, then (M, N) -behavioral equivalence is sound for AM-bisimilarity w.r.t (M, N) .*

Before stating the full-abstraction result analogous to Theorem 21, we give a useful lemma which is similar to Proposition III.1.4.3 [22] for the non-parametrized setting:

► **Lemma 28.** *For relations M, N, R and functions f, g, h, k, i, j as below*



if $\bar{T}$ is a parametrized lax extension, then

$$\bar{T}(f^*; M; g_*, h_*, N; k^*, i_*, R; j^*) = (T(f, h, i))_*; \bar{T}(M, N, R); (T(g, k, j))^*$$

► **Theorem 29.** *$\bar{T}$ -bisimilarity w.r.t (M, N) is complete for (M, N) -behavioral equivalence and if $\bar{T}$ is normal, then it is fully abstract.*

5 Case studies

5.1 Mealy Machines

We first consider a simple example of deterministic finite state machines: for fixed sets I and O of inputs and outputs respectively, a *Mealy machine* is a triple $M = (X \in \mathbf{Set}, \text{next} : X \times I \rightarrow X, \text{out} : X \times I \rightarrow O)$, or equivalently a coalgebra for the functor $F : X \mapsto (X \times O)^I$ (e.g. [34]) which has a strict extension $\bar{F}$ to $\mathbf{Rel}$. For two machines M_1 and M_2 , a relation $R : X_1 \rightarrow X_2$ between their state spaces is an $\bar{F}$ -bisimulation if for every $(x_1, x_2) \in R$, and $i \in I$, $(\text{next}_1(x_1, i), \text{next}_2(x_2, i)) \in R$ and $\text{out}_1(x_1, i) = \text{out}_2(x_2, i)$.

In the parametrized case, we allow the sets of inputs and outputs to vary and consider the parametrized functor $T : (I, O, X) \mapsto (O \times X)^I$ which verifies the conditions of Proposition 13 and therefore has a parametrized lax extension $\bar{T}$. We can now consider two machines M_1, M_2 on different sets of inputs I_1, I_2 and outputs O_1, O_2 . For relations $M : I_1 \rightarrow I_2$ and $N : O_1 \rightarrow O_2$ specifying how inputs and outputs can be related, the induced notion of $\bar{T}$ -bisimulation w.r.t (M, N) relaxes the notion of bisimulation above by requiring only that on two related inputs, the two corresponding outputs should be related: for all $(x_1, x_2) \in R$, $(i_1, i_2) \in M$, $(\text{next}_1(x_1, i_1), \text{next}_2(x_2, i_2)) \in R$ and $(\text{out}_1(x_1, i_1), \text{out}_2(x_2, i_2)) \in N$.

5.2 Parametrized state contextual equivalence

We finally instantiate our framework to a higher-order calculus with parametrized monadic effects and sketch how it supports notions of parametrized contextual equivalence that cannot be directly addressed using the ordinary notion of lax extension (Definition 4).

We fix a (possibly partial) monoid $(\mathcal{L}, \varepsilon, \cdot, \leq)$. Our motivating example is the monoid of memory regions for a fixed set of locations Loc , a memory region is simply a subset $L \subseteq \text{Loc}$. The monoid multiplication $L_1 \cdot L_2$ is defined when the two subsets are disjoint and it is given by their union. We consider a monadic type system with state and computation types, programs are terms in a standard λ -calculus given in fine-grain call-by-value style, which we give below.

State types $L_1, L_2 ::= \varepsilon \mid L_1 \cdot L_2$

Computation types $A_1, A_2 ::= \mathbf{bool} \mid \mathbf{1} \mid (A_1, L_1) \rightarrow (A_2, L_2)$

$$\begin{array}{c}
\frac{}{\Gamma, x : A \vdash_v x : A} \quad \frac{}{\Gamma \vdash_v \mathbf{t} : \mathbf{bool}} \quad \frac{}{\Gamma \vdash_v \mathbf{f} : \mathbf{bool}} \quad \frac{}{\Gamma \vdash_v \star : \mathbf{1}} \\
\\
\frac{\Gamma, x : A_1; L_1 \vdash_c M : A_2; L_2}{\Gamma \vdash_v \lambda(x : A_1; L_1).M : (A_1, L_1) \rightarrow (A_2, L_2)} \quad \frac{\Gamma \vdash_v V : A}{\Gamma; L \vdash_c \mathbf{return}_L V : A; L} \\
\\
\frac{\Gamma \vdash_v V : (A_1, L_1) \rightarrow (A_2, L_2) \quad \Gamma \vdash_v W : A_1}{\Gamma; L_1 \vdash_c VW : A_2; L_2} \quad \frac{\Gamma; L_1 \vdash_c M : A; L_2 \quad \Gamma, x : A; L_2 \vdash_c N : B; L_3}{\Gamma; L_1 \vdash_c \mathbf{let } x = M \mathbf{ in } N : B; L_3} \\
\\
\frac{\Gamma \vdash_v V_i : A_1 \text{ op } (A_1, L_1)^n \rightarrow (A_2, L_2)}{\Gamma; L_1 \vdash_c \text{op}(V_1, \dots, V_n) : A_2; L_2} \quad \frac{\Gamma; L_1 \vdash_c M : A; L_2}{\Gamma; L_1 \cdot L_3 \vdash_c M : A; L_2 \cdot L_3}
\end{array}$$

The metavariable op stands for an operation accessing or modifying the underlying store. Examples of operations are $\mathbf{set}_\ell^L : (\mathbf{bool}, L) \rightarrow (\mathbf{1}, L)$ and $\mathbf{get}_\ell^L : (\mathbf{1}, L) \rightarrow (\mathbf{bool}, L)$ where ℓ is a location in the region L and $\mathbf{new}_\ell^L : (\mathbf{bool}, L) \rightarrow (\mathbf{1}, L \cdot \{\ell\})$ for a new location $\ell \in \text{Loc} \setminus L$. For simplicity, we restrict locations to carry only boolean values.

We denote by $\Lambda(L_1, L_2, A)$ the set of terms M such that $\emptyset; L_1 \vdash_c M : A; L_2$ and by $\mathcal{V}(A)$ is the set of values V such that $\emptyset \vdash_v V : A$. The small step operational semantics is an

indexed family of partial functions $\{\Lambda(L_1, L_2, A) \rightarrow T(L_1, L_2, \Lambda(L_1, L_2, A))\}_{L_1, L_2, A}$ where T is the parameterized state monad defined as $T(L_1, L_2, X) = (\llbracket L_2 \rrbracket \times X)^{\llbracket L_1 \rrbracket}$ and $\llbracket L \rrbracket$ is the set of stores (functions mapping locations in L to boolean values). Examples of small step reductions are:

$$(\text{return}_L(\lambda(x : A; L_1).M)V, s) \rightarrow (M[V/x], s) \text{ and } (\text{new}_\ell^L \mathbf{t}, s) \rightarrow (\text{return} \star, [s, \ell \mapsto \mathbf{t}])$$

where $[s, \ell \mapsto \mathbf{t}] : L \cup \{\ell\} \rightarrow \{\mathbf{t}, \mathbf{f}\}$ maps $\ell' \in L$ to $s(\ell')$ and ℓ to $\mathbf{t}$. The induced big step operational semantics is an indexed family of functions $\{\Downarrow : \Lambda(L_1, L_2, A) \rightarrow T(L_1, L_2, \mathcal{V}(A))\}_{A, L_1, L_2}$.

We are interested in comparing terms, a term relation consists of a pair of families of relations $(R^\mathcal{V} : \mathcal{V} \rightarrow \mathcal{V}, R^A : A \rightarrow A)$ indexed following the aforementioned pattern. Standard contextual equivalence is the largest equivalence that is adequate with respect to the big step operational semantics and compatible with respect to all term constructors. The theory we have developed allows us to refine this notion by modulating the power of contexts depending on which memory region they have access to. This is precisely what happens in the so-called adversarial rules often used in security proofs (e.g. [2]), where the role of the context is played by an adversary that has access only to some (and not all) of the memory locations available to the honest parties.

Consider a memory region $L = L_1 \uplus L_2$ and a term $\emptyset; L_1 \vdash_c M : A; L_1$ which has access only to the region L_1 . Assume that we make changes to the memory outside of L_1 , for example, define $N := \text{let } x = \text{set}_{\ell_2}^L \mathbf{t} \text{ in } M$ where $\ell_2 \in L_2$, then a context which has access to the whole region L will be able to distinguish M from N (even if the variable x does not occur in M). Indeed, for M and N to be contextually equivalent, the following must hold

$$\forall s, r \in \llbracket L \rrbracket, (C[M](s) \Downarrow (\mathbf{t}, r) \Leftrightarrow C[N](s) \Downarrow (\mathbf{t}, r))$$

for every context $C[\cdot]$ such that $\emptyset; L \vdash_c C[M], C[N] : \mathbf{bool}; L$. We can easily construct a context in which the equivalence above fails, for instance by using $\text{get}_{\ell_2}^L$.

Instead, M and N are equivalent for weaker contexts that can access only the region L_1 . We define an equivalence relation E on stores $\llbracket L \rrbracket = \mathbf{Set}(L, \{\mathbf{t}, \mathbf{f}\})$ as $E := \{(s, s') \mid \forall \ell \in L_1, s(\ell) = s'(\ell)\}$, where two stores are equivalent when they are equal in the restricted region L_1 . We say that two terms M, N are L_1 -equivalent if for all $(s_1, s'_1) \in E$:

$$\begin{aligned} (\exists s_2 \in \llbracket L \rrbracket, C[M](s_1) \Downarrow (\mathbf{t}, s_2)) &\Rightarrow (\exists s'_2 \in \llbracket L \rrbracket, C[N](s'_1) \Downarrow (\mathbf{t}, s'_2) \wedge (s_2, s'_2) \in E) \\ (\exists s'_2 \in \llbracket L \rrbracket, C[N](s'_1) \Downarrow (\mathbf{t}, s'_2)) &\Rightarrow (\exists s_2 \in \llbracket L \rrbracket, C[M](s_1) \Downarrow (\mathbf{t}, s_2) \wedge (s_2, s'_2) \in E) \end{aligned}$$

for every context $C[\cdot]$ containing only constructors with locations in L_1 and such that $\emptyset; L \vdash_c C[M], C[N] : \mathbf{bool}; L$. Coming back to our example, M and N will be equivalent for this notion of refined equivalence. This equivalence can further be used to capture a finer form of commutativity for effects acting on disjoint sets of memory locations which we cannot show using ordinary contextual equivalence since the state monad is a typical example of a non-commutative monad where the effect of writing on the memory is not commutative. We can also recover standard contextual equivalence by replacing E with the equality relation and allowing contexts $C[\cdot]$ to be built with constructor accessing and modifying all locations.

This refined notion of contextual equivalence can be coinductively characterized as the largest term equivalence relation $R = (R^\mathcal{V}, R^A)$ satisfying the following conditions:

1. R is (E, E) -adequate with respect to the big step operational semantics, which corresponds to a $\bar{T}$ -bisimulation w.r.t (E, E) : $R^A(L, L, A) \leq (\Downarrow)_*; \bar{T}(E, E, R^\mathcal{V}(A)); (\Downarrow)^*$ where $\bar{T}$ is the lax-extension of Example 7.
2. R is L_1 -compatible: when defining the compatible closure of R , we restrict the standard definition to term constructors with locations in L_1 .

So far, we used the same relation E for both parameters to restrict the scope of testing contexts but our framework also allows the user to specify different predicates on the memory region before and after the execution of a computation.

Conclusion and future work

In this paper, we have extended the theory of lax extensions to parametrized functors and monads on sets, and showed that it allows to refine the usual notions of behavioral equivalence by studying conditions on the parameters to derive sound and complete proof principles establishing behavioral equivalence. The parametrized setting is particularly well-suited for monadic effects where the parameters correspond to a notion of pre and post condition updated by the execution of a program such as global state and I/O (input/output) effects.

In the ordinary non-parametrized setting, there is a close connection between predicate liftings and lax extensions [6, 30, 45, 19, 17] which we aim to develop in the parametrized case. In particular, the adequate notion of coalgebraic logics derived from liftings and lax extensions in the parametrized case remains to be investigated.

References

- 1 Peter Aczel and Nax Mendler. A final coalgebra theorem. In *Category Theory and Computer Science: Manchester, UK, September 5–8, 1989 Proceedings*, pages 357–365. Springer, 2005.
- 2 Alejandro Aguirre, Gilles Barthe, Marco Gaboardi, Deepak Garg, Shin-ya Katsumata, and Tetsuya Sato. Higher-order probabilistic adversarial computations: categorical semantics and program logics. *Proc. ACM Program. Lang.*, 5(ICFP):1–30, 2021. doi:10.1145/3473598.
- 3 Robert Atkey. *Substructural Simple Type Theories for Separation and In-place Update*. PhD thesis, University of Edinburgh, College of Science and Engineering., 2006.
- 4 Robert Atkey. Algebras for parameterised monads. In *International Conference on Algebra and Coalgebra in Computer Science*, pages 3–17. Springer, 2009. doi:10.1007/978-3-642-03741-2_2.
- 5 Robert Atkey. Parameterised notions of computation. *Journal of functional programming*, 19(3-4):335–376, 2009. doi:10.1017/S095679680900728X.
- 6 Paolo Baldan, Filippo Bonchi, Henning Kerstan, and Barbara König. Behavioral metrics via functor lifting. *arXiv preprint arXiv:1410.3385*, 2014. arXiv:1410.3385.
- 7 Paolo Baldan, Filippo Bonchi, Henning Kerstan, and Barbara König. Coalgebraic behavioral metrics. *Logical Methods in Computer Science*, 14, 2018. doi:10.23638/LMCS-14(3:20)2018.
- 8 Michael Barr. Relational algebras. In *Reports of the Midwest Category Seminar IV*, pages 39–55. Springer, 1969.
- 9 Marta Bílková, Alexander Kurz, Daniela Petrişan, and Jiří Velebil. Relation liftings on preorders and posets. In *Algebra and Coalgebra in Computer Science: 4th International Conference, CALCO 2011, Winchester, UK, August 30–September 2, 2011. Proceedings 4*, pages 115–129. Springer, 2011. doi:10.1007/978-3-642-22944-2_9.
- 10 Marta Bilkova, Alexander Kurz, Daniela Petrisan, and Jiri Velebil. Relation lifting, with an application to the many-valued cover modality. *Logical Methods in Computer Science*, 9, 2013.
- 11 Aurelio Carboni, G Max Kelly, and Richard J Wood. A 2-categorical approach to change of base and geometric morphisms i. *Cahiers de topologie et géométrie différentielle catégoriques*, 32(1):47–95, 1991.
- 12 Eugenia Cheng, Nick Gurski, and Emily Riehl. Cyclic multicategories, multivariable adjunctions and mates. *Journal of K-Theory*, 13(2):337–396, 2014. doi:10.1017/is013012007jkt250.
- 13 Ugo Dal Lago and Francesco Gavazzo. A relational theory of effects and coefficients. *Proceedings of the ACM on Programming Languages*, 6(POPL):1–28, January 2022. doi:10.1145/3498692.
- 14 Soichiro Fujii. A 2-categorical study of graded and indexed monads. *arXiv preprint arXiv:1904.08083*, 2019. arXiv:1904.08083.

- 15 Francesco Gavazzo, Riccardo Treglia, and Gabriele Vanoni. Monadic intersection types, relationally. In *European Symposium on Programming*, pages 22–51. Springer, 2024. doi:10.1007/978-3-031-57262-3_2.
- 16 Sergey Goncharov, Dirk Hofmann, Pedro Nora, Lutz Schröder, and Paul Wild. Quantalic behavioural distances.
- 17 Sergey Goncharov, Dirk Hofmann, Pedro Nora, Lutz Schröder, and Paul Wild. Identity-Preserving Lax Extensions and Where to Find Them. In *42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025)*, volume 327, pages 40:1–40:20, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2025.40.
- 18 Sergey Goncharov, Dirk Hofmann, Pedro Nora, Lutz Schröder, and Paul Wild. Relators and notions of simulation revisited. In *2025 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 776–789. IEEE, 2025. doi:10.1109/LICS65433.2025.00064.
- 19 Sergey Goncharov, Dirk Hofmann, Pedro Nora, Lutz Schröder, and Paul Wild. Kantorovich Functors and Characteristic Logics for Behavioural Distances. In Orna Kupferman and Pawel Sobocinski, editors, *Foundations of Software Science and Computation Structures*, volume 13992, pages 46–67. Springer Nature Switzerland, Cham, 2023. Series Title: Lecture Notes in Computer Science. doi:10.1007/978-3-031-30829-1_3.
- 20 John W Gray. *Formal category theory: adjointness for 2-categories*, volume 391. Springer, 2006.
- 21 H Peter Gumm and Mehdi Zarrad. Coalgebraic simulations and congruences. In *International Workshop on Coalgebraic Methods in Computer Science*, pages 118–134. Springer, 2014. doi:10.1007/978-3-662-44124-4_7.
- 22 Dirk Hofmann, Gavin J Seal, and Walter Tholen. *Monoidal Topology: A Categorical Approach to Order, Metric, and Topology*, volume 153. Cambridge University Press, 2014.
- 23 Bart Jacobs. *Introduction to coalgebra*, volume 59. Cambridge University Press, 2017.
- 24 Niles Johnson and Donald Yau. *2-dimensional categories*. Oxford University Press (UK), 2021.
- 25 Shin-ya Katsumata. Parametric effect monads and semantics of effect systems. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 633–645, 2014. doi:10.1145/2535838.2535846.
- 26 Barbara König and Christina Mika-Michalski. (Metric) Bisimulation Games and Real-Valued Modal Logics for Coalgebras. In *29th International Conference on Concurrency Theory (CONCUR 2018)*, volume 118 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.CONCUR.2018.37.
- 27 Michael Lambert. Characterizing double categories of relations. *Applied Category Theory*, 2021.
- 28 Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 1998.
- 29 Kenji Maillard, Danel Ahman, Robert Atkey, Guido Martínez, Cătălin Hrițcu, Exequiel Rivas, and Éric Tanter. Dijkstra monads for all. *Proceedings of the ACM on Programming Languages*, 3(ICFP):1–29, 2019. doi:10.1145/3341708.
- 30 Johannes Marti and Yde Venema. Lax extensions of coalgebra functors and their logic. *Journal of Computer and System Sciences*, 81(5):880–900, 2015. doi:10.1016/J.JCSS.2014.12.006.
- 31 Eugenio Moggi. Notions of computation and monads. *Information and computation*, 93(1):55–92, 1991. doi:10.1016/0890-5401(91)90052-4.
- 32 Susan Niefield. Cartesian closed double categories. *Theory and Applications of Categories*, 40(3):63–79, 2024.
- 33 Pedro Nora, Jurriaan Rot, Lutz Schröder, and Paul Wild. Relational connectors and heterogeneous simulations. In *International Conference on Foundations of Software Science and Computation Structures*, pages 111–132. Springer, 2025. doi:10.1007/978-3-031-90897-2_6.

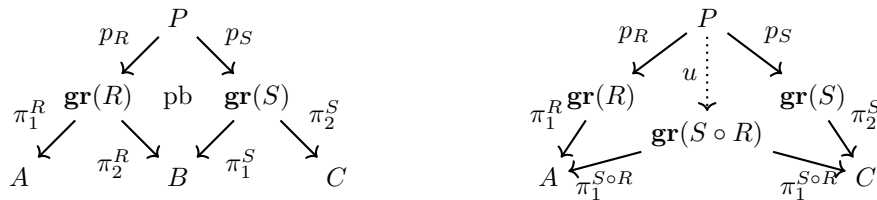
- 34 Dirk Pattinson. An introduction to the theory of coalgebras. *Course notes for NASSLLI*, 2003.
- 35 Jacques Riguet. Relations binaires, fermetures, correspondances de galois. *Bulletin de la société mathématique de France*, 76:114–155, 1948.
- 36 Jan JMM Rutten. Relators and metric bisimulations. *Electronic Notes in Theoretical Computer Science*, 11:252–258, 1998. doi:10.1016/S1571-0661(04)00063-5.
- 37 Jan JMM Rutten. Universal coalgebra: a theory of systems. *Theoretical computer science*, 249(1):3–80, 2000. doi:10.1016/S0304-3975(00)00056-6.
- 38 Davide Sangiorgi. *Introduction to bisimulation and coinduction*. Cambridge University Press, 2011.
- 39 Christoph Schubert and Gavin J Seal. Extensions in the theory of lax algebras. *Theory and Applications of Categories [electronic only]*, 21:118–151, 2008.
- 40 Gavin J Seal. Canonical and op-canonical lax algebras. *Theory Appl. Categ*, 14(10):221–243, 2005.
- 41 Mike Shulman. 2-dinatural transformation. URL: <https://ncatlab.org/nlab/show/2-dinatural%20transformation>.
- 42 AL Smirnov. Graded monads and rings of polynomials. *Journal of Mathematical Sciences*, 151(3):3032–3051, 2008.
- 43 Věra Trnková. Relational automata in a category and their languages. In *International Conference on Fundamentals of Computation Theory*, pages 340–355. Springer, 1977.
- 44 Franck Van Breugel and James Worrell. A behavioural pseudometric for probabilistic transition systems. *Theoretical Computer Science*, 331(1):115–142, 2005. doi:10.1016/J.TCS.2004.09.035.
- 45 Paul Wild and Lutz Schröder. Characteristic logics for behavioural hemimetrics via fuzzy lax extensions. *Logical Methods in Computer Science*, 18, 2022. doi:10.46298/LMCS-18(2:19)2022.
- 46 James Worrell. *On coalgebras and final semantics*. PhD thesis, University of Oxford, 2000.

A Proofs for Section 3

A.1 Extending parametrized functors from Set to Rel

► **Lemma 30.** *If a function $f : A \rightarrow B$ is split epi, then $f^*; f_* = \text{id}_B : B \rightarrowtail B$ and if a function $f : A \rightarrow B$ is split mono, then $f_*; f^* = \text{id}_A : A \rightarrowtail A$.*

► **Lemma 31.** *For relations $R : A \rightarrowtail B$ and $S : B \rightarrowtail C$, if we take the pullback as on the left below (or equivalently the composition of the two tabulation spans), then the unique morphism $u : P \rightarrow \text{gr}(S \circ R)$ as on the right is split epi.*



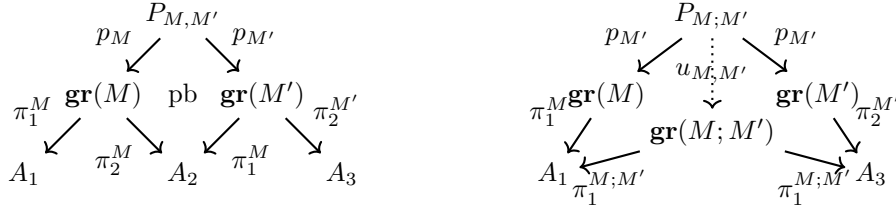
Proof of Theorem 10. We will only prove the theorem for $\mathcal{L}_1$, the proofs for $i = 2, 3, 4$ follow the same pattern.

1. To obtain that $\mathcal{L}_1$ is normal, we use the fact that the tabulation of the identity relation is isomorphic to the identity span.

2. If $M' \leq M : A \multimap B$, $N \leq N' : C \multimap D$ and $R \leq R' : X \multimap Y$, then there exist functions $\alpha : \mathbf{gr}(M') \rightarrow \mathbf{gr}(M)$, $\beta : \mathbf{gr}(N) \rightarrow \mathbf{gr}(N')$ and $\gamma : \mathbf{gr}(R) \rightarrow \mathbf{gr}(R)'$ such that $\pi_1^M \circ \alpha = \pi_1^{M'}$, $\pi_2^M \circ \alpha = \pi_2^{M'}$, $\pi_1^{N'} \circ \beta = \pi_1^N$, $\pi_2^{N'} \circ \beta = \pi_2^N$, $\pi_1^{R'} \circ \gamma = \pi_1^R$ and $\pi_2^{R'} \circ \gamma = \pi_2^R$. We obtain:

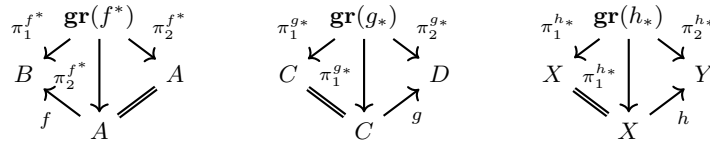
$$\begin{aligned}
\mathcal{L}_1(M, N, R) &= (T(\text{id}, \pi_1^N, \pi_1^R))^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\pi_2^M, \text{id}, \text{id}))_* ; (T(\text{id}, \pi_2^N, \pi_2^R))^* \\
&\leq (T(\text{id}, \pi_1^N, \pi_1^R))^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\alpha, \text{id}, \text{id}))_* ; \\
&\quad (T(\alpha, \text{id}, \text{id}))_* ; (T(\pi_2^M, \text{id}, \text{id}))_* ; (T(\text{id}, \pi_2^N, \pi_2^R))^* \\
&= (T(\text{id}, \pi_1^N, \pi_1^R))^* ; (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M'}, \text{id}, \text{id}))_* ; (T(\text{id}, \pi_2^N, \pi_2^R))^* \\
&= (T(\text{id}, \pi_1^{N'}, \pi_1^{R'}))^* ; (T(\text{id}, \beta, \gamma))^* ; (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; \\
&\quad (T(\pi_2^{M'}, \text{id}, \text{id}))_* ; (T(\text{id}, \beta, \gamma))^* ; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^* \\
&\leq (T(\text{id}, \pi_1^{N'}, \pi_1^{R'}))^* ; (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; (T(\text{id}, \beta, \gamma))^* ; \\
&\quad (T(\text{id}, \beta, \gamma))^* ; (T(\pi_2^{M'}, \text{id}, \text{id}))_* ; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^* \\
&\leq (T(\text{id}, \pi_1^{N'}, \pi_1^{R'}))^* ; (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M'}, \text{id}, \text{id}))_* ; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^* \\
&= \mathcal{L}_1(M', N', R')
\end{aligned}$$

3. for $M : A_1 \multimap A_2$, $M' : A_2 \multimap A_3$, we have:



$$\begin{aligned}
&\mathcal{L}_1(M, \text{id}_C, \text{id}_X) ; \mathcal{L}_1(M', \text{id}_C, \text{id}_X) \\
&= (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M'}, \text{id}, \text{id}))_* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\pi_2^M, \text{id}, \text{id}))_* \\
&\leq (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; (T(p_M, \text{id}, \text{id}))_* ; (T(p_{M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M'}, \text{id}, \text{id}))_* \\
&= (T(\pi_1^{M;M'}, \text{id}, \text{id}))_* ; (T(u_{M,M'}, \text{id}, \text{id}))_* ; (T(u_{M,M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M;M'}, \text{id}, \text{id}))_* \\
&= (T(\pi_1^{M;M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M;M'}, \text{id}, \text{id}))_* = \mathcal{L}_1(M; M', \text{id}_C, \text{id}_X)
\end{aligned}$$

4. the proof for $\mathcal{L}_1(\text{id}, N' \circ N, R' \circ R) \leq \mathcal{L}_1(\text{id}, N', R') \circ \mathcal{L}_1(\text{id}, N, R)$ is similar.
 5. for $f : A \rightarrow B$, $g : C \rightarrow D$ and $h : X \rightarrow Y$, we have



where π_2^{f*} , π_1^{g*} and π_1^{h*} are split epi. We obtain:

$$\begin{aligned}
\mathcal{L}_1(f^*, g_*, h_*) &= (T(\text{id}, \pi_1^{f*}, \pi_1^{h*}))_* ; (T(\pi_1^{f*}, \text{id}, \text{id}))_* ; (T(\pi_2^{f*}, \text{id}, \text{id}))_* ; (T(\text{id}, \pi_2^{g*}, \pi_2^{h*}))_* \\
&= (T(\text{id}, \pi_1^{f*}, \pi_1^{h*}))_* ; (T(f, \text{id}, \text{id}))_* ; (T(\pi_2^{f*}, \text{id}, \text{id}))_* ; \\
&\quad (T(\pi_2^{f*}, \text{id}, \text{id}))_* ; (T(\text{id}, \pi_1^{g*}, \pi_1^{h*}))_* ; T(\text{id}, g, h))_* \\
&= (T(\text{id}, \pi_1^{f*}, \pi_1^{h*}))_* ; (T(f, \text{id}, \text{id}))_* ; (T(\text{id}, \pi_1^{g*}, \pi_1^{h*}))_* ; (T(\text{id}, g, h))_* \\
&= (T(\text{id}, \pi_1^{f*}, \pi_1^{h*}))_* ; (T(\text{id}, \pi_1^{g*}, \pi_1^{h*}))_* ; (T(f, \text{id}, \text{id}))_* ; (T(\text{id}, g, h))_* \\
&= (T(f, g, h))_*
\end{aligned}$$

The proof for $(T(f, g, h))^* = \mathcal{L}_1(f^*, g^*, h^*)$ is similar.

6. for $M : A \multimap B$, $N : C \multimap D$ and $R : X \multimap Y$, we use that the graph of the transpose switches source and target in spans:

$$\begin{array}{ccccc}
 \pi_1^{M^t} & \text{gr}(M^t) & \pi_2^{M^t} & \pi_1^{N^t} & \text{gr}(N^t) & \pi_2^{N^t} & \pi_1^{R^t} & \text{gr}(R^t) & \pi_2^{R^t} \\
 & \swarrow & \downarrow & \swarrow & \downarrow & \swarrow & \swarrow & \downarrow & \swarrow \\
 B & \cong & A & D & \cong & C & Y & \cong & X \\
 \pi_2^M & \swarrow & \pi_1^M & \pi_2^N & \swarrow & \pi_1^N & \pi_2^R & \swarrow & \pi_1^R \\
 & \text{gr}(M) & & \text{gr}(N) & & \text{gr}(R) & & &
 \end{array}$$

We obtain:

$$\begin{aligned}
 \mathcal{L}_1(M^t, N^t, R^t) &= (T(\text{id}, \pi_2^N, \pi_2^R))^* ; (T(\pi_2^M, \text{id}, \text{id}))_* ; (T(\pi_1^M, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_1^N, \pi_1^R))^* \\
 &= ((T(\text{id}, \pi_1^N, \pi_1^R))^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\pi_2^M, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_2^N, \pi_2^R))^*)^t \\
 &= (\mathcal{L}_1(M, N, R))^t
 \end{aligned}$$

7. let $\mathcal{L}$ be an arbitrary lax extension for T , for relations $M : A \multimap B$, $N : C \multimap D$ and $R : X \multimap Y$, we have:

$$\begin{aligned}
 \mathcal{L}_1(M, N, R) &= (T(\text{id}, \pi_1^N, \pi_1^R))^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\pi_2^M, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_2^N, \pi_2^R))^* \\
 &\leq \mathcal{L}(\text{id}, (\pi_1^N)^*, (\pi_1^R)^*) ; \mathcal{L}((\pi_1^M)^* \text{id}, \text{id}) ; \mathcal{L}((\pi_2^M)^* \text{id}, \text{id}) ; \mathcal{L}(\text{id}, (\pi_2^N)^*, (\pi_2^R)^*) \\
 &\leq \mathcal{L}((\pi_1^M)^* ; (\pi_2^M)^*, (\pi_1^N)^* ; (\pi_2^N)^* ; (\pi_1^R)^* ; (\pi_2^R)^*) = \mathcal{L}(M, N, R)
 \end{aligned}$$

Proof of Proposition 12. It is straightforward to obtain $\mathcal{L}_1 = \mathcal{L}_2 = \mathcal{L}_3 = \mathcal{L}_4$ from the strong interchange property. We show that $\mathcal{L}_1$ satisfies axiom (2) in Definition 6:

For relations $M : A_1 \multimap A_2$, $M' : A_2 \multimap A_3$, $N : B_1 \multimap B_2$, $N' : B_2 \multimap B_3$, $R : X_1 \multimap X_2$ and $R' : X_2 \multimap X_3$, we have:

$$\begin{aligned}
 &\mathcal{L}_1(M, N, R) ; \mathcal{L}_1(M', N', R') \\
 &= (T(\text{id}, \pi_1^N, \pi_1^R))^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\pi_2^M, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_2^N, \pi_2^R))^* \\
 &\quad ; (T(\text{id}, \pi_1^{N'}, \pi_1^{R'}))^* ; (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M'}, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^* \\
 &= (T(\text{id}, \pi_1^N, \pi_1^R))^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\pi_2^M, \text{id}, \text{id}))^* ; (T(\text{id}, p_N, p_R))^* \\
 &\quad ; (T(\text{id}, p_{N'}, p_{R'}))^* ; (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M'}, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^* \\
 &= T(\text{id}, \pi_1^N, \pi_1^R)^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\text{id}, p_N, p_R))^* ; (T(\pi_2^M, \text{id}, \text{id}))^* \\
 &\quad ; (T(\pi_1^{M'}, \text{id}, \text{id}))_* ; (T(\text{id}, p_{N'}, p_{R'}))^* ; (T(\pi_2^{M'}, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^* \\
 &\leq T(\text{id}, \pi_1^N, \pi_1^R)^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(\text{id}, p_N, p_R))^* ; (T(p_M, \text{id}, \text{id}))_* \\
 &\quad ; (T(p_{M'}, \text{id}, \text{id}))_* ; (T(\text{id}, p_{N'}, p_{R'}))^* ; (T(\pi_2^{M'}, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^* \\
 &= T(\text{id}, \pi_1^N, \pi_1^R)^* ; (T(\text{id}, p_N, p_R))^* ; (T(\pi_1^M, \text{id}, \text{id}))_* ; (T(p_M, \text{id}, \text{id}))_* \\
 &\quad ; (T(p_{M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M'}, \text{id}, \text{id}))^* ; (T(\text{id}, p_{N'}, p_{R'}))^* ; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^* \\
 &= (T(\text{id}, \pi_1^{N;N'}, \pi_1^{R;R'}))^* ; (T(\pi_1^{M;M'}, \text{id}, \text{id}))_* ; (T(\pi_2^{M;M'}, \text{id}, \text{id}))^* ; (T(\text{id}, \pi_2^{N;N'}, \pi_2^{R;R'}))^*
 \end{aligned}$$

We use that T preserves pullbacks in the last two variables for the second equality and that the interchange squares are weak pullbacks for the penultimate equality. $\blacktriangleleft$

Proof of Proposition 13. We show that $\mathcal{L}_4$ satisfies axiom (2) in Definition 6:

For relations $M : A_1 \multimap A_2$, $M' : A_2 \multimap A_3$, $N : B_1 \multimap B_2$, $N' : B_2 \multimap B_3$, $R : X_1 \multimap X_2$ and $R' : X_2 \multimap X_3$, we have:

$$\begin{aligned}
& \mathcal{L}_4(M, N, R); \mathcal{L}_4(M', N', R') \\
&= (T(\pi_1^M, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1^N, \pi_1^R))^*; (T(\text{id}, \pi_2^N, \pi_2^R))^*; (T(\pi_2^M, \text{id}, \text{id}))^* \\
&\quad ; (T(\pi_1^{M'}, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1^{N'}, \pi_1^{R'}))^*; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^*; (T(\pi_2^{M'}, \text{id}, \text{id}))^* \\
&\leq (T(\pi_1^M, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1^N, \pi_1^R))^*; (T(\text{id}, \pi_2^N, \pi_2^R))^*; (T(p_M, \text{id}, \text{id}))_* \\
&\quad ; (T(p_{M'}, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1^{N'}, \pi_1^{R'}))^*; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^*; (T(\pi_2^{M'}, \text{id}, \text{id}))^* \\
&= (T(\pi_1^M, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1^N, \pi_1^R))^*; (T(p_M, \text{id}, \text{id}))_*; (T(\text{id}, \pi_2^N, \pi_2^R))^*; \\
&\quad ; (T(\text{id}, \pi_1^{N'}, \pi_1^{R'}))^*; (T(p_{M'}, \text{id}, \text{id}))_*; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^*; (T(\pi_2^{M'}, \text{id}, \text{id}))^* \\
&\leq (T(\pi_1^M, \text{id}, \text{id}))_*; (T(p_M, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1^N, \pi_1^R))^*; (T(\text{id}, \pi_2^N, \pi_2^R))^*; \\
&\quad ; (T(\text{id}, \pi_1^{N'}, \pi_1^{R'}))^*; (T(\text{id}, \pi_2^{N'}, \pi_2^{R'}))^*; (T(p_{M'}, \text{id}, \text{id}))_*; (T(\pi_2^{M'}, \text{id}, \text{id}))^* \\
&= (T(\pi_1^{M;M'}, \text{id}, \text{id}))_*; (T(u_{M,M'}, \text{id}, \text{id}))_*; T(\text{id}, \pi_1^{N;N'}, \pi_1^{R;R'})_* \\
&\quad ; T(\text{id}, \pi_2^{N;N'}, \pi_2^{R;R'})_* (T(u_{M,M'}, \text{id}, \text{id}))_*; (T(\pi_2^{M;M'}, \text{id}, \text{id}))^* \\
&= (T(\pi_1^{M;M'}, \text{id}, \text{id}))_*; T(\text{id}, \pi_1^{N;N'}, \pi_1^{R;R'})_*; (T(u_{M,M'}, \text{id}, \text{id}))_* \\
&\quad ; (T(u_{M,M'}, \text{id}, \text{id}))_*; T(\text{id}, \pi_2^{N;N'}, \pi_2^{R;R'})_*; (T(\pi_2^{M;M'}, \text{id}, \text{id}))^* \\
&= (T(\pi_1^{M;M'}, \text{id}, \text{id}))_*; T(\text{id}, \pi_1^{N;N'}, \pi_1^{R;R'})_*; T(\text{id}, \pi_2^{N;N'}, \pi_2^{R;R'})_*; (T(\pi_2^{M;M'}, \text{id}, \text{id}))^* \\
&= \mathcal{L}_4(M; M', N; N', R; R')
\end{aligned}$$

We use that T preserves pullbacks in the last two variables for the third equality and the weak interchange property for the fourth equality. $\blacktriangleleft$

B Proofs of Section 4

Proof of Lemma 25. Assume that we have an AM-bisimulation w.r.t (M, N) as in Section 4.3, then we obtain that it is a $\bar{T}$ -bisimulation w.r.t (M, N) from the axioms of parametrized lax extension:

$$\begin{aligned}
& (\pi_1^R)^*; (\pi_2^R)^* \\
& \leq \alpha_*; (T(\pi_1^M, \text{id}, \text{id}))_*; (T(\text{id}, \pi_1^N, \pi_1^R))^*; (T(\text{id}, \pi_2^N, \pi_2^R))^*; (T(\pi_2^M, \text{id}, \text{id}))^*; \beta^* \\
& \leq \alpha_*; \bar{T}((\pi_1^M)^*, \text{id}, \text{id}); \bar{T}(\text{id}, (\pi_1^N)^*, (\pi_1^R)^*); \bar{T}(\text{id}, (\pi_2^N)^*, (\pi_2^R)^*); \bar{T}((\pi_2^M)^*, \text{id}, \text{id}); \alpha^* \\
& \leq \alpha_*; \bar{T}(M, N, R); \beta^*
\end{aligned}$$

$\blacktriangleleft$

Proof of Lemma 26.

- Assume that (x, y) are behaviorally equivalent. Let $\llbracket - \rrbracket_3 : Z \rightarrow \Omega$ be the unique coalgebra morphism from γ to ω . By uniqueness, we obtain $\llbracket - \rrbracket_1 = \llbracket - \rrbracket_3 \circ f$ and $\llbracket - \rrbracket_2 = \llbracket - \rrbracket_3 \circ g$. Since $(x, y) \in \mathbf{gr}(R)$ and $f(x) = g(y)$, we obtain $\llbracket x \rrbracket_1 = \llbracket y \rrbracket_2$.
- For the other direction, assume that $\llbracket x \rrbracket_1 = \llbracket y \rrbracket_2$. Let (P, p_1, p_2) be the following pullback:

$$\begin{array}{ccc}
& P & \\
p_1 \swarrow & & \searrow p_2 \\
X & \text{pb} & X \\
f \searrow & & \swarrow g \\
& Z &
\end{array}$$

and define $R := f_*; g^*$. From the universal property of tabulators, we obtain that the span (P, p_1, p_2) is isomorphic to $(\mathbf{gr}(R), \pi_1^R, \pi_2^R)$ which finishes the proof. $\blacktriangleleft$

29:22 Coinductive Reasoning for Parametrized Functors and Monads

Proof of Proposition 27. Assume that $N : C \rightrightarrows C$ is difunctional *i.e.* that there exists $h : C \rightarrow W$ and $k : C \rightarrow W$ such that the left square below is a weak-pullback

$$\begin{array}{ccc} \pi_1^N & \mathbf{gr}(N) & \pi_2^N \\ \swarrow & & \searrow \\ C & \text{wpb} & C \\ \searrow & & \swarrow \\ h & & k \\ & W & \end{array} \quad \begin{array}{ccc} \pi_1^R & \mathbf{gr}(R) & \pi_2^R \\ \swarrow & & \searrow \\ X & \text{wpb} & X \\ \searrow & & \swarrow \\ f & & g \\ & Z & \end{array}$$

By definition, a relation $R : X \rightrightarrows X$ is a *behavioral equivalence w.r.t* (M, N) if there exists $f : X \rightarrow Z$ and $g : X \rightarrow Z$ and a coalgebra $\zeta : Z \rightarrow T(\mathbf{gr}(M), W, Z)$ such that the right square above is a weak-pullback and the following two squares below commute:

$$\begin{array}{ccccc} X & & f & & Z & & g & & X \\ \alpha \downarrow & & & & & & & & \downarrow \alpha \\ T(A, C, X) & & & & & & & & T(A, C, X) \\ & \searrow & & & \downarrow \zeta & & \swarrow & & \\ & T(\pi_1^M, h, f) & & & T(\mathbf{gr}(M), W, Z) & & & & T(\pi_2^M, k, g) \end{array}$$

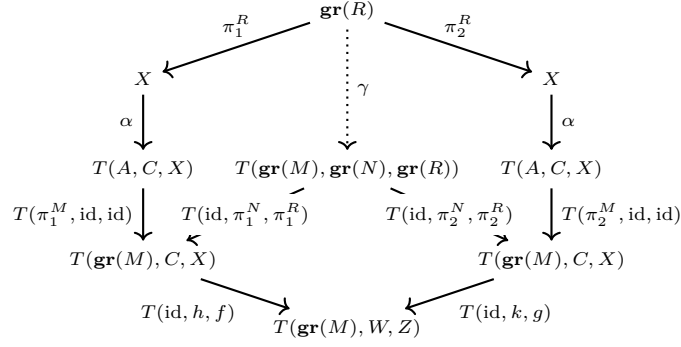
We want to show that R is an AM-bisimulation w.r.t (M, N) . Since T preserves weak pullbacks in the last two variables, the square below is a weak pullback.

$$\begin{array}{ccc} & T(\mathbf{gr}(M), \mathbf{gr}(N), \mathbf{gr}(R)) & \\ T(\text{id}, \pi_1^N, \pi_1^R) \swarrow & & \searrow T(\text{id}, \pi_2^N, \pi_2^R) \\ & T(\mathbf{gr}(M), C, X) \text{ wpb } T(\mathbf{gr}(M), C, X) & \\ T(\text{id}, h, f) \swarrow & & \searrow T(\text{id}, k, g) \\ & T(\mathbf{gr}(M), W, Z) & \end{array}$$

Note that we see here why both N and R are assumed to be difunctional. By the universal property of weak-pullbacks, since the diagram below commutes,

$$\begin{array}{ccccc} & & \mathbf{gr}(R) & & \\ \pi_1^R \swarrow & & & & \searrow \pi_2^R \\ X & & & & X \\ \alpha \downarrow & & f & & g \\ T(A, C, X) & & & & Z \\ \downarrow T(\pi_1^R, \text{id}, \text{id}) & & & & \downarrow T(\pi_2^M, \text{id}, \text{id}) \\ T(\mathbf{gr}(M), C, X) & & & & T(\mathbf{gr}(M), C, X) \\ \downarrow T(\text{id}, h, f) & & \downarrow \zeta & & \downarrow T(\text{id}, k, g) \\ & T(\mathbf{gr}(M), W, Z) & \end{array}$$

there exists a morphism $\gamma : \mathbf{gr}(R) \rightarrow T(\mathbf{gr}(M), \mathbf{gr}(N), \mathbf{gr}(R))$ such that:



which finishes the proof. ◀

Proof of Lemma 28. We show both inclusions:

$$\begin{aligned} (T(f, h, i))_*; \overline{T}(M, N, R); (T(g, k, j))^* &\leq \overline{T}(f^*, h_*, i_*); \overline{T}(M, N, R); \overline{T}(g_*, k^*, j^*) \\ &\leq \overline{T}(f^*; M; g_*, h_*; N; k^*, i_*; R; j^*) \end{aligned}$$

$$\begin{aligned} &\overline{T}(f^*; M; g_*, h_*; N; k^*, i_*; R; j^*) \\ &\leq (T(f, h, i))_*; (T(f, h, i))^*; \overline{T}(f^*; M; g_*, h_*; N; k^*, i_*; R; j^*); (T(g, k, j))_*; (T(g, k, j))^* \\ &\leq (T(f, h, i))_*; \overline{T}(f_*, h^*, i^*); \overline{T}(f^*; M; g_*, h_*; N; k^*, i_*; R; j^*); \overline{T}(g^*, k_*, j_*); (T(g, k, j))^* \\ &\leq (T(f, h, i))_*; \overline{T}(f_*; f^*; M; g_*; g^*, h_*; h^*; N; k_*; k^*, i^* i_*; R; j^*; j_*); (T(g, k, j))^* \\ &\leq (T(f, h, i))_*; \overline{T}(M, N, R); (T(g, k, j))^* \end{aligned} \quad \blacktriangleleft$$

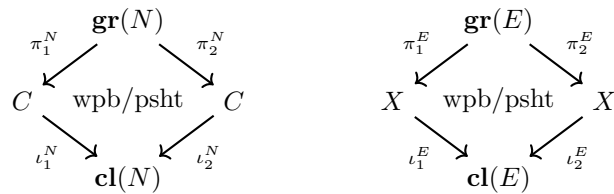
Proof of Theorem 29.

- Assume that R is an (M, N) -behavioral equivalence as in Section 4.4, then we obtain that it is a $\overline{T}$ -bisimulation w.r.t (M, N) :

$$\begin{aligned} f_*; g^* &\leq \alpha_*; (T(\pi_1, h, f))_*; (T(\pi_2, k, g))^*; \alpha^* \\ &\leq \alpha_*; \overline{T}((\pi_1)^*, h_*, f_*); \overline{T}((\pi_2)_*, k^*, g^*); \alpha^* \\ &= \alpha_*; \overline{T}(M, N, R); \alpha^* \end{aligned}$$

- For the other direction, we want to show that $\overline{T}$ -bisimilarity w.r.t (M, N) is an (M, N) -behavioral equivalence.

The operator $\mathbf{Rel}(X, X) \rightarrow \mathbf{Rel}(X, X)$ mapping $R : X \multimap X$ to $\alpha_*; \overline{T}(M, N, R); \alpha^*$ is monotone in R so it has a greatest fixpoint which we denote $E : X \multimap X$. If $M \leq \text{id}_A$, $M \leq M \circ M$, $M \leq M^t$, $\text{id}_C \leq N$, $N \circ N \leq N$, $N^t \leq N$ and $\overline{T}$ verifies axiom (6) in Definition 6, then E is an equivalence relation (in particular it is difunctional). Since N and E are difunctional, taking the collage of N and E , the two squares below are both pushout and weak pullback squares.



29:24 Coinductive Reasoning for Parametrized Functors and Monads

In particular, we have $N = (\iota_1^N)_*; (\iota_2^N)^*$ and $E = (\iota_1^E)_*; (\iota_2^E)^*$. By Lemma 28,

$$\bar{T}(M, N, R) = (T(\pi_1^M, \iota_1^N, \iota_1^R))_*; \bar{T}(\text{id}_A, \text{id}_C, \text{id}_X); (T(\pi_2^M, \iota_2^N, \iota_2^R))^*$$

so if $\bar{T}$ is normal, we have $\bar{T}(M, N, R) = (T(\pi_1^M, \iota_1^N, \iota_1^R))_*; (T(\pi_2^M, \iota_2^N, \iota_2^R))^*$. By the universal property of the collage, we obtain that there is a unique function $\mathbf{cl}(E) \rightarrow T(\mathbf{gr}(M), \mathbf{cl}(N), \mathbf{cl}(E))$ such that:

$$\begin{array}{ccc} \begin{array}{ccc} X & \xrightarrow{E} & X \\ \alpha \downarrow & | \wedge & \downarrow \alpha \\ T(A, C, X) & \xrightarrow{\bar{T}(M, N, R)} & T(A, C, X) \\ \swarrow T(\pi_1^M, \iota_1^N, \iota_1^R) & | \wedge & \searrow T(\pi_2^M, \iota_2^N, \iota_2^R) \\ & T(\mathbf{gr}(M), \mathbf{cl}(N), \mathbf{cl}(E)) & \end{array} & = & \begin{array}{ccc} X & \xrightarrow{E} & X \\ \swarrow \iota_1^E & | \wedge & \searrow \iota_2^E \\ & \mathbf{cl}(E) & \\ \vdots \zeta & & \\ & T(\mathbf{gr}(M), \mathbf{cl}(N), \mathbf{cl}(E)) & \end{array} \end{array}$$

In particular, we obtain that $\zeta \circ \iota_1^E = T(\pi_1^M, \iota_1^N, \iota_1^R) \circ \alpha$ and $\zeta \circ \iota_2^E = T(\pi_2^M, \iota_2^N, \iota_2^R) \circ \alpha$ as desired. $\blacktriangleleft$