Logical Relations for Monadic Types*

Jean Goubault-Larrecq[1], Slawomir Lasota[1,2], and David Nowak[1]

[1] LSV, CNRS & ENS Cachan, France
[2] Institute of Informatics, Warsaw University, Poland
{goubault,lasota,nowak}@lsv.ens-cachan.fr

Abstract. Logical relations and their generalizations are a fundamental tool in proving properties of lambda-calculi, e.g., yielding sound principles for observational equivalence. We propose a natural notion of logical relations able to deal with the monadic types of Moggi's computational lambda-calculus. The treatment is categorical, and is based on notions of subsconing and distributivity laws for monads. Our approach has a number of interesting applications, including cases for lambda-calculi with non-determinism (where being in logical relation means being bisimilar), dynamic name creation, and probabilistic systems.
Keywords: logical relations, monads, semantics, typed lambda-calculus.

1 Introduction

Motivation and context. Logical relations and their generalizations [13] are a fundamental tool in proving properties of lambda-calculi, e.g., characterizing lambda-definability [19, 9, 2, 4], proving equational completeness [13, 24], studying parametric polymorphism [21, 12, 11] notably. On the other hand, Moggi's computational lambda-calculus [16] has proved useful to define various notions of computations on top of the lambda-calculus: side-effects, input-output, continuations, non-determinism [26], probabilistic computation [20] in particular.

What should then be a natural notion of logical relation for Moggi's computational lambda-calculus? Although there is no unique answer to this question, we propose one that is satisfying in practice. We shall demonstrate the relevance of our approach by illustrating our construction on monads for non-determinism, dynamic name creation, and probabilistic computation.

Moggi's insight is based on categorical semantics: while categorical models of the λ-calculus are cartesian closed categories (CCCs), the computational lambda-calculus requires CCCs with a strong monad $(\boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{t})$. The *monadic types* of the computational lambda-calculus are given by the syntax:

$$\tau ::= b | \tau \to \tau | \tau \times \tau | \boldsymbol{T}(\tau)$$

* The first author acknowledges partial support by the RNTL project EVA. The first and third authors acknowledge partial support by the ACI jeunes chercheurs "Sécurité informatique, protocoles cryptographiques et détection d'intrusions". The second author acknowledges partial support by the post-doc fellowship of the Foundation for Polish Science and by the Polish KBN grant 7 T11C 002 21.

where b ranges over a set B of so-called *base types*, and $\boldsymbol{T}(\tau)$ is meant to denote the type of *computations* of type τ. Compared to the lambda-calculus, Moggi's calculus has an additional **val** operation, of type $\tau \to \boldsymbol{T}(\tau)$, and an additional **let** $x = u$ **in** v construct, of type $\boldsymbol{T}(\tau')$ provided u has type $\boldsymbol{T}(\tau)$ and v has type $\boldsymbol{T}(\tau')$ under the assumption $x : \tau$. Every computational lambda-term has a unique interpretation as a morphism in a CCC with a strong monad. In fact the category ***Comp*** whose objects are types and whose morphisms are terms up to $\beta\eta$-conversion is the free CCC-with-a-strong-monad over the set B.

Accordingly, our study will rest on categorical principles. While there is a flurry of generalizations of logical relations (Kripke logical relations [13], lax logical relations [18], pre-logical relations [7], etc.), we use *subscones* [14] as a unifying framework for defining logical relations. Recall that subscones over ***Set*** allow us to define logical relations, and subscones over the presheaf category $\boldsymbol{Set}^{\mathcal{I}}$ lead to $\mathcal{I}$-indexed Kripke logical relations [14]. The important property of logical relations is the so-called Basic Lemma [13]: meanings of a lambda-term in different models w.r.t. related environments are related. This is immediate for subscones, and stems from the fact that ***Comp*** is the free CCC-with-a-strong-monad on B (a trivial adaptation of Proposition 5.2 in [14]). In particular, that any two closed terms that are in logical relation are observationally equivalent is immediate. Our whole endeavor then reduces to finding appropriate *liftings* of monads on categories $\boldsymbol{C}$ to the subscone category $\mathrm{Subscone}^{\mathbb{C}}_{\boldsymbol{C}}$ (see Section 3).

Outline. We define liftings of monads to *scones* in Section 2; this is simpler than for subscones, and of independent interest. This requires *distributivity laws*, slightly extending [25]. We then lift monads to *sub*scones in Section 3. The important case where the target category $\boldsymbol{C}$ is a product of two categories is investigated in Section 4: this is where *binary* logical relations arise, allowing us to compare two models. We terminate our lifting construction by lifting monad strengths in Section 5. It remains to test the relevance of our construction (Section 6): the logical relations thus defined characterize bisimulations when $\boldsymbol{T}$ is the non-determinism monad (as suggested in [11]), a generalization of Larsen and Skou's [10] probabilistic bisimulations when $\boldsymbol{T}$ is a measure monad [8], and a notion close to Pitts and Stark's logical relations for observational equivalence of programs that create names dynamically [17, 23]. We conclude in Section 7.

Preliminaries. Fix two categories $\boldsymbol{C}$ and $\mathbb{C}$ and a functor $|\_| : \boldsymbol{C} \to \mathbb{C}$. Consider the comma category $(\mathbb{C} \downarrow |\_|)$, whose objects are tuples $\langle S, f, A\rangle$, with $f : S \to |A|$ in $\mathbb{C}$ and whose morphisms are pairs $\langle g, h\rangle : \langle S, f, A\rangle \to \langle S', f', A'\rangle$, $g : S \to S'$ in $\mathbb{C}$ and $h : A \to A'$ in $\boldsymbol{C}$, such that the diagram on the right commutes in $\mathbb{C}$:

$$\begin{array}{ccc} S & \xrightarrow{f} & |A| \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle |h|} \\ S' & \xrightarrow[f']{} & |A'| \end{array} \qquad (1)$$

This category is the *scone of* $\boldsymbol{C}$ *over* $\mathbb{C}$, $\mathrm{Scone}^{\mathbb{C}}_{\boldsymbol{C}}$. The second projection functor $U : (\mathbb{C} \downarrow |\_|) \to \boldsymbol{C}$ maps $\langle S, f, A\rangle$ to A and a morphism $\langle g, h\rangle$ to h.

In the sequel we shall be especially interested in the case where $\mathbb{C} = \boldsymbol{Set}$, and $|\_| = \boldsymbol{C}(1, \_)$ is the *global section* functor, where 1 is terminal in $\boldsymbol{C}$. Another interesting situation arises when $\boldsymbol{C} = \mathbb{C}\times\mathbb{C}$ and $|(A, B)| = A\times B$, assuming that $\mathbb{C}$

has finite products. Objects of the scone then represent binary relations between objects in $\boldsymbol{C}$. In this case, given two functors $|\_|_1 : \boldsymbol{C}_1 \to \mathbb{C}$ and $|\_|_2 : \boldsymbol{C}_2 \to \mathbb{C}$, we may define $|\_| : \boldsymbol{C} \to \mathbb{C}$, for $\boldsymbol{C} = \boldsymbol{C}_1 \times \boldsymbol{C}_2$, by $|\langle A_1, A_2\rangle| = |A_1|_1 \times |A_2|_2$.

Further assume we are given a monad $(\boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\mu})$ on $\boldsymbol{C}$. When $\boldsymbol{C} = \boldsymbol{C}_1 \times \boldsymbol{C}_2$, the monad $\boldsymbol{T}$ on $\boldsymbol{C}$ will be usually defined pointwise, by two monads $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ on $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, respectively: $\boldsymbol{T}(\langle A_1, A_2\rangle) = \langle \boldsymbol{T}_1(A_1), \boldsymbol{T}_2(A_2)\rangle$.

Related work. We have already said that there is no unique notion of monad lifting. One of the simplest is the lifting, proposed in [3], $\widehat{T}$ of $\boldsymbol{T}$, which maps the object $\langle S, f, A\rangle$ of the scone to $\langle S, f; |\boldsymbol{\eta}_A|, \boldsymbol{T}(A)\rangle$. Turi [25] considers lifting monads to the category of coalgebras of a given endofunctor. This is a special case of our framework, when $\boldsymbol{C} = \mathbb{C}$ (and $\boldsymbol{T} = T$) and moreover only objects of the form $S \xrightarrow{f} |S|$ are taken into consideration, and only morphisms of the form $\langle g, g\rangle$. This defines the category of $|\_|$-coalgebras as a proper subcategory of scones. Turi uses a simpler version of the distributivity law: distributivity of a monad over an endofunctor; our law involves two monads and a functor between distinct categories. Neither Pitts nor Turi deal with subscones.

In the same way that we lift a monad to relations, Rutten [22] defines an extension of an endofunctor in ***Set*** to a category of relations. The latter has relations as *morphisms* between sets. An endofunctor extends to relations iff it preserves weak pullbacks, and if so, the extension is unique. The approach taken by Rutten is different from ours, where relations are *objects* rather than morphisms. Hence, Rutten imposes a different functoriality condition: the action of a lifted endofunctor on a composition of two relations must coincide with a composition of actions of the lifted endofunctor on these two relations. This amounts to closedness under composition of relations yielded by the lifted endofunctor.

An approach related to ours is [5], where a *comonad* lifting is defined. This relies on pullbacks, whereas we use mono factorization systems. Nonetheless, the *commutators* of [5] *are* dual to our distributivity laws.

2 Lifting of a monad to a scone

By *lifting of a monad* $(\boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\mu})$ to the scone of $\boldsymbol{C}$ over $\mathbb{C}$ we mean a monad $(\widetilde{T}, \widetilde{\eta}, \widetilde{\mu})$ on $\mathrm{Scone}^{\mathbb{C}}_{\boldsymbol{C}}$ such that the diagram on the right commutes. That is, $U \circ \widetilde{T} = \boldsymbol{T} \circ U$ and moreover

$$\begin{tikzcd} \mathrm{Scone}^{\mathbb{C}}_{\boldsymbol{C}} \arrow[r, "\widetilde{T}"] \arrow[d, "U"'] & \mathrm{Scone}^{\mathbb{C}}_{\boldsymbol{C}} \arrow[d, "U"] \\ \boldsymbol{C} \arrow[r, "\boldsymbol{T}"'] & \boldsymbol{C} \end{tikzcd} \tag{2}$$

$$U\widetilde{\eta} = \boldsymbol{\eta}_U \quad \text{and} \quad U\widetilde{\mu} = \boldsymbol{\mu}_U. \tag{3}$$

By $U\widetilde{\eta}$ and $\boldsymbol{\eta}_U$ we mean the two possible compositions of a natural transformation with U, similarly $U\widetilde{\mu}$ and $\boldsymbol{\mu}_U$. The equations (3) amount to the requirement that the two diagrams on the right commute, for all objects X in $\mathrm{Scone}^{\mathbb{C}}_{\boldsymbol{C}}$:

$$\begin{tikzcd} & \boldsymbol{T}UX \arrow[d, equal, "(2)"] \\ UX \arrow[ur, "\boldsymbol{\eta}_{UX}"] \arrow[r, "U\widetilde{\eta}_X"'] & U\widetilde{T}X \end{tikzcd} \qquad \begin{tikzcd} & \boldsymbol{T}^2UX \arrow[dl, "\boldsymbol{\mu}_{UX}"'] \arrow[d, equal, "(2)"] \\ \boldsymbol{T}UX \arrow[d, equal, "(2)"] & \boldsymbol{T}U\widetilde{T}X \arrow[d, equal, "(2)"] \\ U\widetilde{T}X & U\widetilde{T}^2X \arrow[l, "U\widetilde{\mu}_X"] \end{tikzcd}$$

In other words, the functor U together with the identity natural transformation is a morphism of monads from $\widetilde{T}$ to $\boldsymbol{T}$. Note that the equations (3) determine the $\boldsymbol{C}$-components of $\widetilde{\eta}$ and $\widetilde{\mu}$ unambiguously. Moreover, diagram (2) determines the $\boldsymbol{C}$-component of the action of $\widetilde{T}$ on objects and morphisms, i.e. $\langle S, f, A\rangle$ is necessarily mapped to $\langle \widetilde{S}, \widetilde{f}, \boldsymbol{T}A\rangle$, for some $\widetilde{S}$, $\widetilde{f}$ and a morphism $\langle g, h\rangle$ is necessarily mapped to $\langle \widetilde{g}, \boldsymbol{T}h\rangle$, for some $\widetilde{g}$.

To be able to give an appropriate lifting we assume another monad (T, η, μ) on $\mathbb{C}$ such that $\boldsymbol{T}$ and T are related by a *distributivity law*, i.e. a natural transformation $\sigma : T|\_| \Rightarrow |\boldsymbol{T}|$ making the two diagrams on the right commute, for each object A in $\boldsymbol{C}$:

$$\begin{tikzcd}
 & T|A| \arrow[d, "\sigma_A"] \\
|A| \arrow[ur, "\eta_{|A|}"] \arrow[r, "|\boldsymbol{\eta}_A|"'] & |\boldsymbol{T}A|
\end{tikzcd}
\qquad
\begin{tikzcd}
 & T^2|A| \arrow[dl, "\mu_{|A|}"'] \arrow[d, "T\sigma_A"] \\
T|A| \arrow[d, "\sigma_A"] & T|\boldsymbol{T}A| \arrow[d, "\sigma_{\boldsymbol{T}A}"] \\
|\boldsymbol{T}A| & |\boldsymbol{T}^2A| \arrow[l, "|\boldsymbol{\mu}_A|"]
\end{tikzcd} \tag{4}$$

Having σ, we define $\widetilde{T}$ on objects by $\langle S, f, A\rangle \mapsto \langle TS, \sigma_X \circ Tf, \boldsymbol{T}A\rangle$ exploiting that if $S \xrightarrow{f} |A|$ then $TS \xrightarrow{Tf} T|A| \xrightarrow{\sigma_A} |\boldsymbol{T}A|$ is an arrow. On morphisms, note that

$$\begin{tikzcd}
TS \arrow[r, "Tf"] \arrow[d, "Tg"'] & T|A| \arrow[r, "\sigma_A"] & |\boldsymbol{T}A| \arrow[d, "|\boldsymbol{T}h|"] \\
TS' \arrow[r, "Tf'"'] & T|A'| \arrow[r, "\sigma_{A'}"'] & |\boldsymbol{T}A'|
\end{tikzcd}$$

commutes since

$$\begin{tikzcd}
S \arrow[r, "f"] \arrow[d, "g"'] & |A| \arrow[d, "|h|"] \\
S' \arrow[r, "f'"'] & |A'|
\end{tikzcd}$$

commutes and σ is natural. So we define $\widetilde{T}$ by $\langle g, h\rangle \mapsto \langle Tg, \boldsymbol{T}h\rangle$. Moreover, we put $\widetilde{\eta}_{\langle S,f,A\rangle} = \langle \eta_S, \boldsymbol{\eta}_A\rangle$ and $\widetilde{\mu}_{\langle S,f,A\rangle} = \langle \mu_S, \boldsymbol{\mu}_A\rangle$.

Checking that this defines a monad is straightforward. First, to check that unit and multiplication are well defined it is sufficient to merge the commuting diagrams (4) and complete them with naturality squares for η and μ as shown on the right.
Unit $\widetilde{\eta}$ and multiplication $\widetilde{\mu}$ are natural since they are defined pointwise and $\boldsymbol{\eta}$, $\boldsymbol{\mu}$, η and μ are. Verifying monad laws is immediate, by the same argument.

$$\begin{tikzcd}
S \arrow[r, "\eta_S"] \arrow[dddd, "f"'] & TS \arrow[ddd, "Tf"] & T^2S \arrow[l, "\mu_S"'] \arrow[d, "T^2f"] \\
 & & T^2|A| \arrow[dl, "\mu_{|A|}"'] \arrow[d, "T\sigma_A"] \\
 & & \\
 & T|A| \arrow[d, "\sigma_A"] & T|\boldsymbol{T}A| \arrow[d, "\sigma_{\boldsymbol{T}A}"] \\
|A| \arrow[ur, "\eta_{|A|}"] \arrow[r, "|\boldsymbol{\eta}_A|"'] & |\boldsymbol{T}A| & |\boldsymbol{T}^2A| \arrow[l, "|\boldsymbol{\mu}_A|"]
\end{tikzcd}$$

3 Lifting of a monad to a subscone

The full subcategory of $\mathrm{Scone}^{\mathbb{C}}_{\boldsymbol{C}}$ consisting of all objects $\langle S, f, A\rangle$ with f a mono, written $S \overset{f}{\hookrightarrow} |A|$, we call *the subscone of* $\boldsymbol{C}$ *over* $\mathbb{C}$ and denote by $\mathrm{Subscone}^{\mathbb{C}}_{\boldsymbol{C}}$. Objects When $\mathbb{C} = \boldsymbol{Set}$ and $|A|$ is given by $\boldsymbol{C}(1, A)$, each object $S \hookrightarrow |A|$ in the subscone represents a subset of global elements of A. In the binary case, i.e. when $\boldsymbol{C} = \boldsymbol{C}_1 \times \boldsymbol{C}_2$ and $|\langle A_1, A_2\rangle| = \boldsymbol{C}_1(1_1, A_1) \times \boldsymbol{C}_2(1_2, A_2)$, $S \hookrightarrow |\langle A_1, A_2\rangle|$

corresponds to a binary relation on global elements of A_1 and A_2—when A_1 and A_2 are the respective denotations of type τ in two given models, this will be the logical relation at type τ.

For technical reasons, we require that $\mathbb{C}$ has a *mono factorization system.* This is essentially an epi-mono factorization [1], except we relax part of the definition: we keep the mono part but do not require the epis in the sequel.

Formally, a mono factorization system is given by two distinguished subclasses of morphisms in $\mathbb{C}$, the so-called *pseudoepis* $\twoheadrightarrow$ and the so-called *relevant monos* $\multimap$. The latter must be monos, while the former are not required to be epis. Both classes must contain all isomorphisms and be closed under composition with isomorphisms.

Each morphism f in $\mathbb{C}$ must factor as $f = m \circ e$ for some pseudoepi e and some relevant mono m; and each commuting square (5) has a diagonal making both triangles commute as in (6). Note that the diagonal is necessarily unique and that whenever the lower-right triangle commutes, the upper-left triangle does too. Furthermore, the latter property guarantees that the factorization $f = m \circ e$ is unique up to iso.

$$\begin{tikzcd} \cdot \arrow[r, two heads] \arrow[d] & \cdot \arrow[d] \\ \cdot \arrow[r, tail] & \cdot \end{tikzcd} \tag{5}$$

$$\begin{tikzcd} \cdot \arrow[r, two heads] \arrow[d] & \cdot \arrow[d] \arrow[dl, dotted] \\ \cdot \arrow[r, tail] & \cdot \end{tikzcd} \tag{6}$$

Additionally, we assume that functor T preserves pseudoepis, i.e. T maps a pseudoepi to a pseudoepi. This will be needed in diagram (11) below.

Note the following simple and important fact:

Fact 1 *The first component g of a morphism $\langle g, h\rangle$ in a subscone is uniquely determined by the second component h.*

This is because the bottom arrow in (1) is now mono.

Let us define a lifting of the monad to the subscone by analogy with (2) and (3) for the scone. In the binary case mentioned at the beginning of this section, this corresponds to a lifting of a monad to the category of binary relations (as objects) and relation preserving functions (as morphisms).

The lifting $\widetilde{T}$ on objects is given by the mono part of the mono factorization of the lifting of the previous section: $\langle S, f, A\rangle$ is taken to $\langle \widetilde{S}, m, \boldsymbol{T}A\rangle$ given by the diagram on the right:

$$\begin{tikzcd} TS \arrow[r, "Tf"] \arrow[d, "e"'] & T|A| \arrow[d, "\sigma_A"] \\ \widetilde{S} \arrow[r, tail, "m"'] & |\boldsymbol{T}A| \end{tikzcd} \tag{7}$$

Clearly $\widetilde{T}$ is defined only up to iso. Formally, the construction would be unambiguous if we worked with subobjects of $|\boldsymbol{T}A|$, which are determined uniquely.

Given a morphism $\langle g, h\rangle$, the diagram on the right commutes. Then the action of $\widetilde{T}$ on $\langle g, h\rangle$ will be obtained from the unique diagonal guaranteed by (6). We construct diagram (9) below from two copies of (7).

$$\begin{tikzcd} S \arrow[r, hook, "f"] \arrow[dr, "g"'] & {|A|} \arrow[dr, "|h|"] & \\ & S' \arrow[r, tail, "f'"'] & {|A'|} \end{tikzcd} \tag{8}$$

All four given faces of the cube commute. Both front and back faces commute by definition of $\widetilde{T}$ on objects: they are copies of diagram (7). The right-hand face is a naturality square of σ; the top face is by application of T to diagram (8), hence commutes by definition of morphisms in the subscone.

$$\begin{tikzcd}
TS \arrow[rr, "Tf"] \arrow[dr, "Tg"] \arrow[dd, "e"'] & & T|A| \arrow[dr, "\boldsymbol{T}|h|"] \arrow[dd, "\sigma_A" near start] & \\
& TS' \arrow[rr, "Tf'" near start] \arrow[dd, "e'" near start] & & T|A'| \arrow[dd, "\sigma_{A'}"] \\
\widetilde{S} \arrow[rr, tail, "m" near start] & & |\boldsymbol{T}A| \arrow[dr, "|\boldsymbol{T}h|"] & \\
& \widetilde{S'} \arrow[rr, tail, "m'"'] & & |\boldsymbol{T}A'|
\end{tikzcd} \tag{9}$$

Now, an instance of diagram (5) can be found in (9) by two walks from TS to $|\boldsymbol{T}A'|$: one starts with the pseudoepi $TS \xrightarrow{e} \widetilde{S}$, the other ends with the relevant mono $\widetilde{S'} \xrightarrow{m'} |\boldsymbol{T}A'|$. Since all faces commute, there is an arrow $\widetilde{S} \dashrightarrow \widetilde{S'}$ as in diagram (6), making the two newly created faces of the cube commute. This arrow is unique by Fact 1. Now $\widetilde{T}\langle g, h\rangle$ is given by the bottom face. Functoriality follows immediately from uniqueness of the diagonal arrow in (6).

The ($\mathbb{C}$-component of the) unit $\widetilde{\eta}_{\langle S,f,A\rangle}$ is defined by post-composing η_S with the pseudoepi part of the mono factorization in (7). This is well defined since everything in sight in the diagram on the right commutes. Indeed, the right triangle is one of the distributivity law diagrams,

$$\begin{tikzcd}
S \arrow[rr, hook, "f"] \arrow[d, "\eta_S"'] & & |A| \arrow[dl, "\eta_{|A|}"'] \arrow[dd, "|\boldsymbol{\eta}_A|"] \\
TS \arrow[r, "Tf"] \arrow[d, "e"'] & T|A| \arrow[dr, "\sigma_A"] & \\
\widetilde{S} \arrow[rr, tail, "m"'] & & |\boldsymbol{T}A|
\end{tikzcd} \tag{10}$$

the upper square is the naturality of η while the lower one is a copy of (7). The ($\mathbb{C}$-component of the) multiplication $\widetilde{\mu}_{\langle S,f,A\rangle}$ will be induced by a diagram similar to (9) (below).

Again, all the faces not having the required dotted arrow as edge commute. The front face and the lower half of the back face are instances of (7), defining $\widetilde{T}\langle S, f, A\rangle$ and $\widetilde{T}^2\langle S, f, A\rangle$, respectively. The upper half of the back face is by application of T to the front face. The right-hand face is the other distributivity law diagram, which we had not used yet, while the upper one is a naturality square for μ.

$$\begin{tikzcd}
T^2S \arrow[rr, "T^2f"] \arrow[d, "Te"'] \arrow[dr, "\mu_S"] & & T^2|A| \arrow[d, "T\sigma_A"'] \arrow[dr, "\mu_{|A|}"] & \\
T\widetilde{S} \arrow[rr, "Tm" near end] \arrow[dd, "\widetilde{e}"'] \arrow[ddr, dashed] & TS \arrow[rr, "Tf" near start] \arrow[dd, "e" near end] & T|\boldsymbol{T}A| \arrow[dd, "\sigma_{\boldsymbol{T}A}" near start] & T|A| \arrow[dd, "\sigma_A"] \\
& & & \\
\widetilde{\widetilde{S}} \arrow[rr, tail, "\widetilde{m}" near end] \arrow[r, dotted, "!"'] & \widetilde{S} \arrow[rr, tail, "m"'] & |\boldsymbol{T}^2A| \arrow[r, "|\boldsymbol{\mu}_A|"] & |\boldsymbol{T}A|
\end{tikzcd} \tag{11}$$

Note that Te is a pseudoepi, since T preserves pseudoepis. Composition $\widetilde{e} \circ Te$ is not necessarily a pseudoepi, hence we will need a diagonal (6) twice. First, similarly as in diagram (9) we find an instance of diagram (5) by two walks from T^2S to $|\boldsymbol{T}A|$, one starting with Te and the other ending with m. Hence, the unique dashed arrow exists and makes the two triangles commute. One of them, involving the pseudoepi Te, is the upper part of the left-hand side. The other one, namely that involving the relevant mono m, allows us to apply (5) again, since the following two walks from $T\widetilde{S}$ to $|\boldsymbol{T}A|$ commute: one starting with the pseudoepi $\widetilde{e}$ and the other consisting of the dashed arrow followed by m. Hence, the unique dotted arrow exists and makes the bottom face as well as the triangle in the left-hand face commute. The multiplication $\widetilde{\mu}_{\langle S,f,A\rangle}$ is then defined by the bottom face of the cube.

Verification of the monad laws is a formality due to the following:

Fact 2 *Given two parallel arrows in* $\mathrm{Subscone}_{\mathbb{C}}^{\boldsymbol{C}}$, *say* $\langle g_1, h_1\rangle$ *and* $\langle g_2, h_2\rangle$, *they are equal whenever the second components* h_1 *and* h_2 *are.*

The proof is immediate by Fact 1. Using this fact, and knowing that second components of $\widetilde{\eta}$ and $\widetilde{\mu}$ satisfy the monad laws (as they are unit and multiplication of $\boldsymbol{T}$, respectively), we deduce immediately that $\widetilde{\eta}$ and $\widetilde{\mu}$ satisfy the monad laws too. Similarly one proves naturality of $\widetilde{\eta}$ and $\widetilde{\mu}$.

It is useful to summarize the ingredients we have used here. To lift a monad $(\boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\mu})$ on $\boldsymbol{C}$ to $\mathrm{Subscone}_{\boldsymbol{C}}^{\mathbb{C}}$, we need:

> **(i)** a category $\mathbb{C}$ and a functor $|\_| : \boldsymbol{C} \to \mathbb{C}$,
> **(ii)** a monad (T, η, μ) on $\mathbb{C}$, related to $(\boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\mu})$ by a distributivity law σ,
> **(iii)** a mono factorization system on $\mathbb{C}$, **(iv)** T preserves pseudoepis.

Recall that to lift the CCC structure of $\boldsymbol{C}$ to the subscone, we additionally require $\mathbb{C}$ to be a CCC with pullbacks, and $|\_|$ to preserve finite products [14]. Description of the construction can be found eg. in [5], Section 5.4.

4 Lifting of a monad to relations

Recall that we would like to lift monads to categories of binary relations as objects. Hence, assume in this section that $\boldsymbol{C}$ is a product category, $\boldsymbol{C} = \boldsymbol{C}_1 \times \boldsymbol{C}_2$ and that both $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are equipped with monads $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$, and functors $|\_|_1 : \boldsymbol{C}_1 \to \mathbb{C}$ and $|\_|_2 : \boldsymbol{C}_2 \to \mathbb{C}$. A monad $\boldsymbol{T}$ on $\boldsymbol{C}$ can be defined pairwise: $\boldsymbol{T}\langle A_1, A_2\rangle = \langle \boldsymbol{T}_1 A_1, \boldsymbol{T}_2 A_2\rangle$ and similarly we define $|\_| : \boldsymbol{C} \to \mathbb{C}$ by $|\langle A_1, A_2\rangle| = |A_1|_1 \times |A_2|_2$—to this aim we assume binary products in $\mathbb{C}$.

In the same vein distributivity laws for $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ induce a distributivity law for $\boldsymbol{C}$. Assumed two distributivity laws, $\sigma^1 : T|\_|_1 \Rightarrow |\boldsymbol{T}_1|_1$ and $\sigma^2 : T|\_|_2 \Rightarrow |\boldsymbol{T}_2|_2$, we can define $\sigma_{\langle A_1, A_2\rangle} : T(|A_1|_1 \times |A_2|_2) \to |\boldsymbol{T}_1 A_1|_1 \times |\boldsymbol{T}_2 A_2|_2$ by

$$\sigma_{\langle A_1, A_2\rangle} = \langle T\pi_1; \sigma^1_{A_1}, T\pi_2; \sigma^2_{A_2}.\rangle \tag{12}$$

where π_1 and π_2 denote the projections from $|A_1|_1 \times |A_2|_2$.

The situation gets much simpler when $\mathbb{C} = \boldsymbol{Set}$, $|\_|_1 = \boldsymbol{C}_1(1_1, \_)$ and $|\_|_2 = \boldsymbol{C}_2(1_2, \_)$. We assume that $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ have terminal objects, 1_1 and 1_2 respectively. Each object $S \hookrightarrow |\langle A_1, A_2\rangle|$ in the subscone defines a binary relation (again noted S) on global elements of A_1 and A_2. Obviously $\boldsymbol{Set}$ satisfies all requirements from previous sections, with surjections as pseudoepis and injections as relevant monos.

Given two CCCs $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ with respective strong monads $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$, the fact that ***Comp*** is the free CCC with strong monad on the set B of base types means that there are two representations of CCCs-with-strong-monads, $[\![\_]\!]_1$ and $[\![\_]\!]_2$, from ***Comp*** to $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ respectively: they are the natural meaning functions for monadic types and computational λ-terms.

Our construction of a lifting together with standard constructions on subscones [14] yield another representation of CCCs-with-strong-monads $[\![\_]\!]$ from ***Comp*** to $\mathrm{Subscone}^{\boldsymbol{Set}}_{\boldsymbol{C}_1\times\boldsymbol{C}_2}$. That $[\![\_]\!]$ is a lifting means that $U\circ[\![\_]\!] = \langle [\![\_]\!]_1, [\![\_]\!]_2\rangle$, i.e., the diagram on the right commutes. When $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are concrete categories, this means that

$$
\begin{tikzcd}
 & \mathrm{Subscone}^{\boldsymbol{Set}}_{\boldsymbol{C}_1\times\boldsymbol{C}_2} \arrow[d, "U"] \\
\boldsymbol{Comp} \arrow[ur, "{[\![\_]\!]}"] \arrow[r, "{\langle [\![\_]\!]_1, [\![\_]\!]_2\rangle}"'] & \boldsymbol{C}_1\times\boldsymbol{C}_2
\end{tikzcd}
$$

$$\forall a_1 \in [\![\Gamma]\!]_1, a_2 \in [\![\Gamma]\!]_2 . (a_1,a_2)\in[\![\Gamma]\!] \Rightarrow ([\![t]\!]_1(a_1), [\![t]\!]_2(a_2)) \in [\![\tau]\!] \tag{13}$$

for all terms t of type τ in the context $\Gamma = x_1 : \tau_1, \ldots, x_n : \tau_n$; representations of Γ are taken to be products of the representations of $\tau_1, \ldots, \tau_n$; $[\![\tau]\!]$ is a relation between $[\![\tau]\!]_1$ and $[\![\tau]\!]_2$, defined by induction on types τ (the case where τ is a base type is arbitrary):

$$
\begin{aligned}
(f_1,f_2)\in[\![\tau\to\tau']\!] &\iff \forall (a_1,a_2)\in[\![\tau]\!] . (f_1(a_1), f_2(a_2)) \in [\![\tau']\!] \\
((a_1,a_1'),(a_2,a_2'))\in[\![\tau\times\tau']\!] &\iff (a_1,a_2)\in[\![\tau]\!] \wedge (a_1',a_2')\in[\![\tau']\!] \\
(B_1,B_2)\in[\![\boldsymbol{T}\tau]\!] &\iff (B_1,B_2)\in\widetilde{T}[\![\tau]\!]
\end{aligned}
$$

These equations (except possibly the last one) are the standard definition of a *logical relation*. (13) is the already cited *Basic Lemma.*

Further simplification is gained when $\boldsymbol{C}_1 = \boldsymbol{C}_2 = \boldsymbol{Set}$, the three monads $\boldsymbol{T}_1$, $\boldsymbol{T}_2$ and T are identical and both $|\_|_1$ and $|\_|_2$ are identity functors. The distributivity law reduces to distributivity of the monad T over binary product, and (12) rewrites to $\sigma_{(A_1,A_2)} = \langle T\pi_1, T\pi_2\rangle : T(A_1\times A_2) \to TA_1\times TA_2$, where by T we denote a given single monad on ***Set***. This is a particularly interesting special case, so we study it in more detail.

Every binary relation $S \subseteq A_1\times A_2$ has a representation $S \xhookrightarrow{\langle {\pi^S}_1, {\pi^S}_2\rangle} A_1\times A_2$ where the arrow is the inclusion induced by two projections ${\pi^S}_1 : S\to A_1$ and ${\pi^S}_2 : S \to A_2$. In fact, the full subcategory of subscone consisting exclusively of inclusions instead of all injections is equivalent to the whole subscone, so without loss of generality we consider only inclusions in the rest of this section.

Recall the action of a lifted monad $\widetilde{T}$ on a relation $S \xhookrightarrow{\langle {\pi^S}_1, {\pi^S}_2\rangle} A_1\times A_2$:

$$
\begin{tikzcd}
TS \arrow[r, "{T\langle {\pi^S}_1, {\pi^S}_2\rangle}"] \arrow[d, two heads] & T(A_1\times A_2) \arrow[d, "\sigma_{(A_1,A_2)}"] \\
\widetilde{S} \arrow[r, hook] & TA_1\times TA_2
\end{tikzcd}
$$

The functor $\widetilde{T}$ maps a relation S to the relation between sets TA_1 and TA_2 defined as the direct image of the function $\langle T{\pi^S}_1, T{\pi^S}_2\rangle : TS \to TA_1\times TA_2$, since $\langle T{\pi^S}_1, T{\pi^S}_2\rangle = \langle T\pi_1, T\pi_2\rangle\circ T\langle {\pi^S}_1, {\pi^S}_2\rangle$.

5 Lifting of a strong monad to a scone and a subscone

In this section we assume that both categories $\boldsymbol{C}$ and $\mathbb{C}$ have finite products given explicitly. By the same symbol 1 we denote the terminal object both in $\boldsymbol{C}$ and $\mathbb{C}$. Moreover we assume that natural isomorphisms are given:

$$\begin{array}{rll} \boldsymbol{r}_A : 1\times A \to A & \boldsymbol{\alpha}_{A,B,C} : (A\times B)\times C \to A\times(B\times C) & \text{in } \boldsymbol{C} \\ r_S : 1\times S \to S & \alpha_{S,R,Q} : (S\times R)\times Q \to S\times(R\times Q) & \text{in } \mathbb{C} \end{array} \tag{14}$$

and that $|\_|$ preserves strictly finite products as well as $\boldsymbol{r}$ and $\boldsymbol{\alpha}$, i.e., $|A\times B| = |A|\times|B|$, $|\boldsymbol{r}_A| = r_{|A|}$, $|\boldsymbol{\alpha}_{A,B,C}| = \alpha_{|A|,|B|,|C|}$. We will also need an assumption that products preserve pseudoepis, i.e., if f and g are pseudoepis then $f\times g$ is pseudoepi too. Furthermore we assume that two monads $\boldsymbol{T}$ and T are *strong*, i.e., a *strength* natural transformation $\boldsymbol{t}_{A,B} : A\times\boldsymbol{T}B \to \boldsymbol{T}(A\times B)$ is given such that the diagrams in Definition 3.2 in [16] commute and analogously a strength $t_{S,R} : S\times TR \to T(S\times R)$ for T is assumed.

Our definition of monad lifting $\boldsymbol{T}$ to scones (or subscones) in diagram (2) and equations (3) will be extended to strong monads below. First observe that we do have finite products and natural isomorphisms $\widetilde{r}$ and $\widetilde{\alpha}$ in scones and subscones, if we put $\widetilde{r}_{\langle S,f,A\rangle} = \langle r_S, \boldsymbol{r}_A\rangle$ and $\widetilde{\alpha}_{\langle S,f,A\rangle,\langle R,g,B\rangle,\langle Q,h,C\rangle} = \langle \alpha_{S,R,Q}, \boldsymbol{\alpha}_{A,B,C}\rangle$.

Lemma 1. *Both* $\mathrm{Scone}_{\boldsymbol{C}}^{\mathbb{C}}$ *and* $\mathrm{Subscone}_{\boldsymbol{C}}^{\mathbb{C}}$ *have finite products, given explicitly. Functor U preserves strictly finite products as well as $\widetilde{r}$ and $\widetilde{\alpha}$.* □

By lifting of $\boldsymbol{T}$ to $\mathrm{Scone}_{\boldsymbol{C}}^{\mathbb{C}}$ we now mean a strong monad, i.e. a monad $(\widetilde{T}, \widetilde{\eta}, \widetilde{\mu})$ together with a strength $\widetilde{t}_{X,Y} : X\times\widetilde{T}Y \to \widetilde{T}(X\times Y)$, such that diagram (2) commutes, equations (3) hold and $U\widetilde{t}_{X,Y} = \boldsymbol{t}_{UX,UY}$, i.e., U preserves strength. To be able to give an appropriate lifting, we extend distributivity law (4) by one more condition, relating strengths $\boldsymbol{t}_{A,B}$ and $t_{|A|,|B|}$:

$$\begin{tikzcd} {|A|\times T|B|} \arrow[r, "\mathrm{id}_{|A|}\times\sigma_B"] \arrow[d, "t_{|A|,|B|}"] & {|A|\times|\boldsymbol{T}B| = |A\times\boldsymbol{T}B|} \arrow[d, "|\boldsymbol{t}_{A,B}|"] \\ {T(|A|\times|B|) = T|A\times B|} \arrow[r, "\sigma_{A\times B}"] & {|\boldsymbol{T}(A\times B)|} \end{tikzcd} \tag{15}$$

Having lifted $\boldsymbol{T}$ to scones and subscones in previous sections, we only need to give a lifting of the strength $\boldsymbol{t}$. For scones this is straightforward—define $\widetilde{t}$ pointwise by $\widetilde{t}_{\langle S,f,A\rangle,\langle R,g,B\rangle} := \langle t_{S,R}, \boldsymbol{t}_{A,B}\rangle$.

Verifying that this is well-defined amounts to pasting together a naturality square for t and a diagram (15):

$$\begin{tikzcd} S\times TR \arrow[r, "f\times Tg"] \arrow[d, "t_{S,R}"] & {|A|\times T|B|} \arrow[r, "\mathrm{id}_{|A|}\times\sigma_B"] \arrow[d, "t_{|A|,|B|}"'] & {|A\times\boldsymbol{T}B|} \arrow[d, "|\boldsymbol{t}_{A,B}|"'] \\ T(S\times R) \arrow[r, "T(f\times g)"] & {T(|A|\times|B|)} \arrow[r, "\sigma_{A\times B}"] & {|\boldsymbol{T}(A\times B)|} \end{tikzcd}$$

Note that the upper side of this diagram is precisely $\langle S,f,A\rangle\times\widetilde{T}\langle R,g,B\rangle$ in the scone while the lower side is $\widetilde{T}(\langle S,f,A\rangle\times\langle R,g,B\rangle)$. Checking naturality of $\widetilde{t}$ and strength laws is immediate since $\widetilde{t}$, $\widetilde{\alpha}$, $\widetilde{r}$, $\widetilde{\eta}$ and $\widetilde{\mu}$ are all defined pointwise.

Now we move to subscones. Call $\widetilde{T}$ the lifted monad defined in (7) and (9).

As in previous sections, $\widetilde{t}$ in subscones will differ from the case of scones only in its $\mathbb{C}$-component, and this component will be induced as a unique diagonal guaranteed by diagram (6) in the diagram on the right:

$$
\begin{tikzcd}
S\times TR \arrow[rr, "f\times Tg"] \arrow[dr, "t_{S,R}"'] \arrow[dd, "\mathrm{id}_S\times e"'] & & {|A|\times T|B|} \arrow[dr, "t_{|A|,|B|}"] \arrow[dd, two heads, "\mathrm{id}_{|\_|A}\times\sigma_B"] & \\
 & T(S\times R) \arrow[rr, "T(f\times g)"] \arrow[ddd] & & T(|A|\times|B|) \arrow[d, equal] \\
S\times\widetilde{R} \arrow[rr, hook, "f\times m"] \arrow[ddr, dotted] & & {|A|\times|\boldsymbol{T}B|} \arrow[d, equal] & T|A\times B| \arrow[dd, "\sigma_{A\times B}"] \\
 & & {|A\times\boldsymbol{T}B|} \arrow[dr, "|\boldsymbol{t}_{A,B}|"] & \\
 & \cdot \arrow[rr, tail, "n"] & & {|\boldsymbol{T}(A\times B)|}
\end{tikzcd}
\tag{16}
$$

As ingredients of this diagram we have used instances of diagram (7) for $\widetilde{T}\langle R, g, B\rangle$ and $\widetilde{T}(\langle S, f, A\rangle\times\langle R, g, B\rangle)$:

$$
\begin{tikzcd}
TR \arrow[r, "Tg"] \arrow[d, "e"'] & T|B| \arrow[d, "\sigma_B"] \\
\widetilde{R} \arrow[r, tail, "m"] & {|\boldsymbol{T}B|}
\end{tikzcd}
\qquad
\begin{tikzcd}
T(S\times R) \arrow[r, "T(f\times g)"] \arrow[d] & T(|A|\times|B|) \arrow[r, equal] & T|A\times B| \arrow[d, "\sigma_A"] \\
\cdot \arrow[rr, tail, "n"] & & {|\boldsymbol{T}(A\times B)|}
\end{tikzcd}
$$

The right diagram is the front face of (16), while the product of f with the left diagram generates the back face. The upper face of (16) is a naturality square for t and the right-hand face is exactly diagram (15). Note that $\mathrm{id}_{|\_|A}\times\sigma_B$ is marked as pseudoepi due to our assumption that products preserve pseudoepis. Since the bottom (slightly deformed) face commutes, $\widetilde{t}$ is well-defined. And again, checking naturality of $\widetilde{t}$ and strength laws is immediate by Fact 2.

Here is the final set of ingredients for lifting a strong monad $(\boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{r}, \boldsymbol{\alpha}, \boldsymbol{t})$ on category $\boldsymbol{C}$ with explicitly given finite products, to $\mathrm{Subscone}_{\boldsymbol{C}}^{\mathbb{C}}$:

(i) a category $\mathbb{C}$ with explicitly given finite products and natural isos r and α,
(ii) a functor $|\_| : \boldsymbol{C} \to \mathbb{C}$, preserving finite products, $\boldsymbol{r}$ and $\boldsymbol{\alpha}$, strictly,
(iii) a strong monad $(T, \eta, \mu, r, \alpha, t)$ on $\mathbb{C}$, related to $(\boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{r}, \boldsymbol{\alpha}, \boldsymbol{t})$ by a distributivity law σ defined in (4) and (15),
(iv) a mono factorization system on $\mathbb{C}$,
(v) pseudoepis are preserved by T as well as by finite products.

5.1 Building distributivity laws from adjunctions

It is often the case that we have a (strong) monad on $\boldsymbol{C}$, and wish to build another one on $\mathbb{C}$ related to the latter by a distributivity law. The following results are then of some help. For lack of space, proofs are omitted and can be found in the full version [6].

Proposition 1. *Let $\boldsymbol{C}$ and $\mathbb{C}$ be two categories with explicitly given finite products and with natural isomorphisms $\boldsymbol{r}$, $\boldsymbol{\alpha}$ in $\boldsymbol{C}$ and r, α in $\mathbb{C}$ as in (14). Let $|\_| : \boldsymbol{C} \to \mathbb{C}$ be a functor with a left adjoint $\boldsymbol{D} : \mathbb{C} \to \boldsymbol{C}$. Assume that both functors strictly preserve finite products and the natural isomorphisms: $|\boldsymbol{r}_A| = r_{|A|}$,*

$|\boldsymbol{\alpha}_{A,B,C}| = \alpha_{|A|,|B|,|C|}$, $\boldsymbol{D}(r_E) = \boldsymbol{r}_{\boldsymbol{D}E}$ *and* $\boldsymbol{D}(\alpha_{E,F,G}) = \boldsymbol{\alpha}_{\boldsymbol{D}E,\boldsymbol{D}F,\boldsymbol{D}G}$. *Furthermore assume that the adjunction preserves products:* $\dot\eta_{E\times F} = \dot\eta_E \times \dot\eta_F$ *(and hence* $\dot\epsilon_{A\times B} = \dot\epsilon_A \times \dot\epsilon_B$*), where* $\dot\eta_E : E \to |\boldsymbol{D}(E)|$ *is the unit of the adjunction and* $\dot\epsilon_A : \boldsymbol{D}|A| \to A$ *is the counit. Let* $(\boldsymbol{T},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{r},\boldsymbol{\alpha},\boldsymbol{t})$ *be a strong monad on* $\boldsymbol{C}$.

Define $T = |\_| \circ \boldsymbol{T} \circ \boldsymbol{D}$, $\eta_E = |\boldsymbol{\eta}_{\boldsymbol{D}(E)}| \circ \dot\eta_E$, $\mu_E = |\boldsymbol{\mu}_{\boldsymbol{D}(E)} \circ \boldsymbol{T}\dot\epsilon_{\boldsymbol{TD}(E)}|$, *and* $t_{E,F} = |\boldsymbol{t}_{\boldsymbol{D}(E),\boldsymbol{D}(F)}| \circ (\dot\eta_E \times \mathrm{id}_{TF})$. *Finally, let* $\sigma_A = |\boldsymbol{T}\dot\epsilon_A| : T|A| \to |\boldsymbol{T}A|$. *Then* (T,η,μ,r,α,t) *is a strong monad on* $\mathbb{C}$ *and* σ *is a distributivity law of strong monads from* $T|\_|$ *to* $|\boldsymbol{T}\_|$.

Proposition 1 is surprisingly powerful: in each of the the examples given in the following section, a distributivity law that induces the relevant lifting can be obtained by Proposition 1.

Any category with explicitly given finite products offers a standard choice for $\boldsymbol{r}$ and $\boldsymbol{\alpha}$: let $\boldsymbol{r}_A : 1 \times A \to A$ be the second projection π_2, and $\boldsymbol{\alpha} : (A \times B) \times C \to A \times (B \times C)$ be $\langle \pi_1 \circ \pi_1, \langle \pi_2 \circ \pi_1, \pi_2 \rangle\rangle$. Call these isomorphisms *standard*.

Corollary 1. *Let* $(\boldsymbol{T},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{r},\boldsymbol{\alpha},\boldsymbol{t})$ *be a strong monad on a category* $\boldsymbol{C}$ *with explicit finite products,* $|\_| : \boldsymbol{C} \to \mathbb{C}$*, with left adjoint* $\boldsymbol{D} : \mathbb{C} \to \boldsymbol{C}$*, where* $\boldsymbol{r}$ *and* $\boldsymbol{\alpha}$ *are standard, and* $\mathbb{C}$ *has explicit finite products. Assume that* $|\_|$ *and* $\boldsymbol{D}$ *preserve finite products strictly, and* $|\boldsymbol{D}|$ *is the identity functor on* $\mathbb{C}$ *and* $\dot\eta_E = \mathrm{id}_E$.

Let T, η, μ, σ *be as in Proposition 1 , and* $t_{E,F} = |\boldsymbol{t}_{\boldsymbol{D}(E),\boldsymbol{D}(F)}|$.

Then (T,η,μ,r,α,t) *is a strong monad on* $\mathbb{C}$*, where* r *and* α *are standard, and* σ *is a distributivity law of strong monads from* $T|\_|$ *to* $|\boldsymbol{T}\_|$.

It's only for simplicity that we have assumed strict preservation of products in this section – in fact all the development can be done when products are preserved only up to iso, see [6].

6 Examples

As in Section 4, suppose $\mathbf{C}_1 = \mathbf{C}_2 = \mathbb{C} = \boldsymbol{Set}$, and $|\_|_1$ and $|\_|_2 = |\_|$ are identities. Below we summarize the action of $\widetilde{T}$ on a relation $S \multimap A_1 \times A_2$, for different computational monads T of Moggi [16]. This is parameterized by a binary relation R_{St} on states in the state monad $(A \times \mathrm{St})^{\mathrm{St}}$, and by a binary relation $R_{\mathcal{R}}$ in the continuation monad $\mathcal{R}^{\mathcal{R}^A}$.

Monad T	relation $\widetilde{S} \subseteq TA_1 \times TA_2$
$TA = A_\perp = A \cup \{\perp\}$	$\widetilde{S} = S \cup \{\perp, \perp\}$
$TA = (A \times \mathrm{St})^{\mathrm{St}}$	$(f,g) \in \widetilde{S} \iff \forall s_1, s_2 \in \mathrm{St}.$ $(s_1,s_2) \in \mathrm{St} \Rightarrow (\pi_1(fs_1), \pi_1(gs_2)) \in S \;\wedge\; (\pi_2(fs_1), \pi_2(gs_2)) \in R_{\mathrm{St}}$
$TA = \mathbb{P}_{\mathrm{fin}}(A)$	$(B_1, B_2) \in \widetilde{S} \iff$ $\forall b_1 \in B_1. \exists b_2 \in B_2. (b_1, b_2) \in S \wedge$ $\forall b_2 \in B_2. \exists b_1 \in B_1. (b_1, b_2) \in S$
$TA = \mathcal{R}^{\mathcal{R}^A}$	$(\alpha_1, \alpha_2) \in \widetilde{S} \iff ($ $\forall f_1, f_2. (\forall a_1, a_2. (a_1, a_2) \in S \Rightarrow (f_1(a_1), f_2(a_2)) \in R_{\mathcal{R}}) \Rightarrow$ $(\alpha_1(f_1), \alpha_2(f_2)) \in R_{\mathcal{R}})$

Our construction in the case of the finite powerset monad $\mathbb{P}_{\mathrm{fin}}()$ in fact expands to: $(B_1, B_2) \in \widetilde{S}$ iff $B_1 = \{x | (x, y) \in R\}$ and $B_2 = \{y | (x, y) \in R\}$ for some $R \subseteq S$. (Recall that $\widetilde{T}$ maps relations S to the direct image $\widetilde{S}$ of $\langle T\pi_1, T\pi_2\rangle : TS \to TA_1 \times TA_2$, see the end of Section 4.) This is equivalent to the condition given above, which is the more usual way of defining bisimulations.

Indeed, if $B_1 = \{x | (x, y) \in R\}$ and $B_2 = \{y | (x, y) \in R\}$ for some $R \subseteq S$ then for every $b_1 \in B_1$ by construction there is some $b_2 \in B_2$ such that $(b_1, b_2) \in R$, therefore $(b_1, b_2) \in S$ since $R \subseteq S$, and symmetrically for every $b_2 \in B_2$ there is some $b_1 \in B_1$ such that $(b_1, b_2) \in S$: B_1 and B_2 are bisimilar.

Conversely, if B_1 and B_2 are bisimilar (in the sense just given), then let R be the restriction of S to $B_1 \times B_2$. For every $b_1 \in B_1$, by bisimilarity there is some $b_2 \in B_2$ such that $(b_1, b_2) \in S$, so $(b_1, b_2) \in R$, therefore $b_1 \in \{x | (x, y) \in R\}$; so $B_1 \subseteq \{x | (x, y) \in R\}$. The reverse inclusion is obvious, so $B_1 = \{x | (x, y) \in R\}$. The other equality $B_2 = \{y | (x, y) \in R\}$ is by symmetry.

That logical relations on powersets define bisimulations was conjectured in [11] and, for pre-logical relations, in [7].

6.1 Labelled transition systems and bisimulations

The case $TA = \mathbb{P}_{\mathrm{fin}}(A)$ defines labelled transition systems as elements of $(TA)^{A \times L}$, with labels in L and states in A, as functions mapping states a and labels ℓ to the set of states a' such that $a \xrightarrow{\ell} a'$. Our monad lifting $\widetilde{S}$ in this case is parameterized by a binary relation on R_L on labels and is defined by:

$$(f_1, f_2) \in \widetilde{S} \iff (\forall a_1, a_2, \ell_1, \ell_2 \cdot (a_1, a_2) \in S \wedge (\ell_1, \ell_2) \in R_L \Rightarrow \left\{ \begin{array}{l} \forall b_1 \in f_1(a_1, \ell_1).\exists b_2 \in f_2(a_2, \ell_2).(b_1, b_2) \in S \wedge \\ \forall b_2 \in f_2(a_2, \ell_2).\exists b_1 \in f_1(a_1, \ell_1).(b_1, b_2) \in S \end{array} \right)$$

In case R_L is the equality relation, the relation $\widetilde{S}$ relates f_1 and f_2 iff S is a *strong bisimulation* between the labelled transition systems f_1 and f_2.

6.2 Logical relations for dynamic name creation

Consider Moggi's model of dynamic name creation [15]. Let $\mathcal{I}$ be the category of finite sets and injective functions, and $\boldsymbol{Set}^{\mathcal{I}}$ be the category of functors from $\mathcal{I}$ to $\boldsymbol{Set}$ and natural transformations (the category of *covariant presheaves* over $\mathcal{I}$). For short, write $\boldsymbol{T}As$ for $\boldsymbol{T}(A)(s)$ and similarly for other notations. Let $+$ denote coproduct in $\mathcal{I}$.

We define the strong monad $\boldsymbol{T}$ on $\boldsymbol{Set}^{\mathcal{I}}$ as follows. $\boldsymbol{T}A = \mathrm{colim}_{s'} A(\_ + s') : \mathcal{I} \to \boldsymbol{Set}$. On objects, this is given by $\boldsymbol{T}As = \mathrm{colim}_{s'} A(s + s')$, i.e., $\boldsymbol{T}As$ is the set of all equivalence classes of pairs (s', a) with $s' \in \mathcal{I}$ and $a \in A(s + s')$ modulo the smallest equivalence relation $\equiv$ such that $(s', a) \equiv (s'', A(\mathrm{id}_s + j)a)$ for every morphism $s' \xrightarrow{j} s''$ in $\mathcal{I}$ (intuitively, given a set of *names* s, elements of $\boldsymbol{T}As$ are formal expressions $(\nu s')a$ where all names in s' are bound and every name free in a is in $s + s'$—modulo the fact that $(\nu s', s'')a \equiv (\nu s')a$ for any additional set of new names s'' not free in a). On morphisms $s_1 \xrightarrow{i} s_2$, $\boldsymbol{T}Ai$ maps the equivalence class of (s', a) to the equivalence class of $(s', A(i + \mathrm{id}_{s'})a)$.

It is important to note how $\equiv$ works. The category $\mathcal{I}$ has pushouts: in particular, if $s_0 \xrightarrow{i_1} s_1$ and $s_0 \xrightarrow{i_2} s_2$ are two morphisms in $\mathcal{I}$, then there is a finite set $s_1 +_{s_0} s_2$ and two morphisms $s_1 \xrightarrow{j_1} s_1 +_{s_0} s_2$, $s_2 \xrightarrow{j_2} s_1 +_{s_0} s_2$ such that $i_1; j_1 = i_2; j_2$—take $s_1 +_{s_0} s_2$ to be the disjoint sum $s_1 + s_2$ modulo the equivalence relation relating $i_1(a_0) = i_2(a_0)$ for every $a_0 \in s_0$. It follows that for every $a_1 \in A(s + s_1)$, $a_2 \in A(s + s_2)$, $(s_1, a_1) \equiv (s_2, a_2)$ if and only if there is a finite set s_{12} and two arrows $s_1 \xrightarrow{j_1} s_{12}$ and $s_2 \xrightarrow{j_2} s_{12}$ such that $A(\mathrm{id}_{s_1} + j_1)a_1 = A(\mathrm{id}_{s_2} + j_2)a_2$.

We take $\boldsymbol{C}_1 = \boldsymbol{C}_2 = \mathbb{C} = \boldsymbol{Set}^{\mathcal{I}}$, hence objects in the subscone give raise to the $\mathcal{I}$-indexed *Kripke logical relations*. Furthermore, $|\_|_1 = |\_|_2 = |\_|$ is the identity functor and T is just $\boldsymbol{T}$. Category $\boldsymbol{Set}^{\mathcal{I}}$ has a mono factorization consisting of pointwise surjections and pointwise injections.

As in Section 4, the distributivity law $\sigma_{\langle A_1, A_2 \rangle} s : T(A_1 \times A_2)s \to \boldsymbol{T}A_1 s \times \boldsymbol{T}A_2 s$ is equal to $\langle T\pi_1, T\pi_2 \rangle s$. $\widetilde{S}s \hookrightarrow \boldsymbol{T}A_1 s \times \boldsymbol{T}A_2 s$ is thus given by

$$(s_1, a_1)\ \widetilde{S}s\ (s_2, a_2) \iff \begin{array}{l} \exists s_0 \in \mathcal{I} \cdot \exists i_1 : s_1 \to s_0 \in \mathcal{I} \cdot \exists i_2 : s_2 \to s_0 \in \mathcal{I} \cdot \\ (A_1(\mathrm{id}_s + i_1)a_1)\ S(s + s_0)\ (A_2(\mathrm{id}_s + i_2)a_2) \end{array} \tag{17}$$

where $a_1 \in A_1(s + s_1)$ and $a_2 \in A_2(s + s_2)$.

This is similar to the logical relations of [17, 23]. While (17) is roughly similar to the notion of logical relation of [17], this paper does not rest on Moggi's computational λ-calculus. On the other hand [23] does rest on the computational λ-calculus but does not define a suitable notion of logical relation.

6.3 Monads of Measures and Probabilities

Let us consider a natural CCC equipped with a notion of measure: ***Ipo***, the category of *inductive partial orders* or *ipos* [8]. The objects (ipos) are partial orders such that every directed subset has a least upper bound; the morphisms are continuous functions. This is cartesian-closed, and has pullbacks.

For every ipo A, Jones observed that the set $\boldsymbol{T}A$ of all continuous evaluations, to be defined next, was again an ipo. An *evaluation* $\boldsymbol{\nu}$ maps Scott opens to reals in $[0, +\infty]$ so that $\boldsymbol{\nu}(O \cup O') = \boldsymbol{\nu}(O) + \boldsymbol{\nu}(O') - \boldsymbol{\nu}(O \cap O')$ (for all opens O, O'), $\boldsymbol{\nu}(\emptyset) = 0$ and $\boldsymbol{\nu}$ is monotonic, i.e., $O \subseteq O'$ implies $\boldsymbol{\nu}(O) \leq \boldsymbol{\nu}(O')$. A *continuous evaluation* in addition maps unions of directed sets of opens to the sup of their evaluations. This extends to a strong monad $(\boldsymbol{T}, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{t})$: see Jones' thesis [8].

Take $\boldsymbol{C} = \boldsymbol{Ipo}$, $\mathbb{C} = \boldsymbol{Set}$, $|\_| : \boldsymbol{Ipo} \to \boldsymbol{Set}$ be the underlying set functor, with left adjoint the *discretization* functor $\boldsymbol{D}$: $\boldsymbol{D}(E)$ is E with equality as ordering. By Corollary 1 we get a pair of strong monads related by a distributivity law.

To lift the monads to the subscone, check that T preserves pseudoepis, i.e., that whenever $f : E \to F$ is surjective, then for every continuous evaluation ξ on F (opens are all subsets), there is a continuous evaluation ν of E such that $\xi(Y) = Tf(\nu)(Y)$, i.e., $\xi(Y) = \nu(f^{-1}(Y))$ for every $Y \subseteq F$. Using the axiom of choice, let γ map every $Z \in \mathbb{P}(E) \setminus \{\emptyset\}$ to some element of Z. Then defining $\nu(X)$ as $\xi(\{y \in Y | \gamma(f^{-1}(y)) \in X\})$ for every $X \subseteq E$ fits the bill.

Turning to binary relations is a matter of taking $\boldsymbol{C} = \boldsymbol{Ipo} \times \boldsymbol{Ipo}$, $\mathbb{C} = \boldsymbol{Set} \times \boldsymbol{Set}$, as in Section 4, and letting the distributivity law be given by (12), where $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are both the continuous evaluation monad on $\boldsymbol{Ipo}$, $|\_|_1$ and $|\_|_2$ are both the forgetful functor from $\boldsymbol{Ipo}$ to $\boldsymbol{Set}$, and σ^1 and σ^2 are both given by Corollary 1.

Let us spell this out: for any relation $S \subseteq |A_1|_1 \times |A_2|_2$, the lifted relation $\widetilde{S}$ between continuous evaluations $\boldsymbol{\nu}_1 \in |\boldsymbol{T}_1 A_1|_1$ and $\boldsymbol{\nu}_2 \in |\boldsymbol{T}_2 A_2|_2$ is given by:

$$(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2) \in \widetilde{S} \iff \left(\exists \nu \in TS. \begin{cases} \forall O_1 \subseteq A_1 \text{ open}. \boldsymbol{\nu}_1(O_1) = \nu((O_1 \times A_2) \cap S) \wedge \\ \forall O_2 \subseteq A_2 \text{ open}. \boldsymbol{\nu}_2(O_2) = \nu((A_1 \times O_2) \cap S) \end{cases} \right)$$

Interestingly, and analogously with Section 6.1, we may define a *probabilistic* labelled transition system as an element of $(\boldsymbol{T}A)^{A \times L}$. Then two such transition systems f_1 and f_2 are in relation if and only if:

$$\begin{aligned} &\forall a_1 \in |A_1|_1, a_2 \in |A_2|_2\,, \ell_1, \ell_2 \in L.(a_1, a_2) \in S \wedge (\ell_1, \ell_2) \in R_L \Rightarrow \\ &\qquad \left(\exists \nu \in TS\,.\begin{cases} \forall O_1 \subseteq A_1 \text{ open}. f_1(a_1, \ell)(O_1) = \nu((O_1 \times A_2) \cap S) \wedge \\ \forall O_2 \subseteq A_2 \text{ open}. f_2(a_2, \ell)(O_2) = \nu((A_1 \times O_2) \cap S) \end{cases} \right) \end{aligned} \tag{18}$$

We invite the reader to check that this definition is equivalent to Larsen and Skou's notion of *probabilistic bisimulation* [10] in the case where A_1 and A_2 are *finite* and *discrete*: then relations S as described by our subscone construction have probabilistic bisimulations as reflexive symmetric transitive closures, and probabilistic bisimulations are equivalence relations S obeying (18).

7 Conclusion

The main contribution of this paper is a natural extension of logical relations able to deal with monadic types. We illustrate its naturality and its practical value by demonstrating that various notions of bisimulations and a non-trivial notion of logical relation for dynamic name creation are instances of our construction. Besides, our construction provides a natural integration between notions of simulations between transition systems (possibly probabilistic), higher-order computation (the import of the λ-calculus), and limited forms of side-effects (e.g., dynamic names), yielding streamlined criteria for observational equivalence of those combined systems.

References

1. J. Adamek, H. Herrlich, and G. Strecker. *Abstract and Concrete Categories*. Wiley, New York, 1990.
2. M. Alimohamed. A characterization of lambda definability in categorical models of implicit polymorphism. *Theoretical Computer Science*, 146:5–23, 1995.
3. R. Crole and A. Pitts. New foundations for fixpoint computations: Fix-hyperdoctrines and the fix-logic. *Information and Computation*, 98:171–210, 1992.

4. M. Fiore and A. Simpson. Lambda definability with sums via Grothendieck logical relations. In *TLCA'99*, pages 147–161. Springer Verlag LNCS 1581, 1999.
5. J. Goubault-Larrecq and E. Goubault. On the geometry of intuitionistic S4 proofs. Research Report LSV-01-8, LSV, CNRS & ENS Cachan, 2001. To appear in Homology, Homotopy and Applications.
6. J. Goubault-Larrecq, S. Lasota, and D. Nowak. Logical relations for monadic types. Research Report, LSV, CNRS & ENS Cachan, 2002.
7. F. Honsell and D. Sannella. Pre-logical relations. In *CSL'99*, pages 546–561. Springer Verlag LNCS 1683, 1999.
8. C. Jones. *Probabilistic Non-Determinism*. PhD thesis, University of Edinburgh, 1990. Technical Report ECS-LFCS-90-105.
9. A. Jung and J. Tiuryn. A new characterization of lambda definability. In *TLCA'93*, pages 245–257. Springer Verlag LNCS 664, 1993.
10. K. G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.
11. R. Lazić and D. Nowak. A unifying approach to data-independence. In *CONCUR'2000*, pages 581–595. Springer Verlag LNCS 1877, 2000.
12. Q. Ma and J. C. Reynolds. Types, abstraction, and parametric polymorphism, part 2. In *MFPS'91*, pages 1–40. Springer-Verlag LNCS 598, 1992.
13. J. C. Mitchell. *Foundations for Programming Languages*. MIT Press, 1996.
14. J. C. Mitchell and A. Scedrov. Notes on sconing and relators. In *CSL'92*, pages 352–378. Springer Verlag LNCS 702, 1993.
15. E. Moggi. An abstract view of programming languages. Technical Report ECS-LFCS-90-113, LFCS, Department of Computer Science, University of Edinburgh, 1990.
16. E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.
17. A. Pitts and I. Stark. Observable properties of higher order functions that dynamically create local names, or: What's *new*? In *MFCS'93*, pages 122–141. Springer-Verlag LNCS 711, 1993.
18. G. Plotkin, J. Power, D. Sannella, and R. Tennent. Lax logical relations. In *ICALP'2000*, pages 85–102. Springer Verlag LNCS 1853, 2000.
19. G. D. Plotkin. Lambda-definability in the full type hierarchy. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, pages 363–373. Academic Press, 1980.
20. N. Ramsey and A. Pfeffer. Stochastic lambda calculus and monads of probability distributions. In *POPL'02*, pages 154–165, 2002.
21. J. C. Reynolds. Types, abstraction and parametric polymorphism. In *IFIP'83*, pages 513–523. North-Holland, 1983.
22. J. Rutten. Relators and metric bisimulations. In *CMCS'98*, volume 11 of *Electronic Notes in Theoretical Computer Science*, pages 1–7. Elsevier Science, 1998.
23. I. Stark. Names, equations, relations: Practical ways to reason about *new*. *Fundamenta Informaticae*, 33(4):369–396, April 1998.
24. R. Statman. Logical relations and the typed λ-calculus. *Information and Control*, 65(2–3):85–97, 1985.
25. D. Turi. *Functorial Operational Semantics and its Denotational Dual*. PhD thesis, Free University, Amsterdam, June 1996.
26. P. Wadler. Comprehending monads. *Mathematical Structures in Computer Science*, 2:461–493, 1992.