A CATEGORICAL SETTING FOR TRANSITION SYSTEMS

A thesis submitted to The University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

2023

Daniel T. Mroz

School of Engineering,
Department of Computer Science

Contents

Word Count: 33174

Abstract

A categorical setting for transition systems

Daniel T. Mroz

A thesis submitted to The University of Manchester for the degree of Doctor of Philosophy, 2023

Transition systems of various "transition types" are used to model various computational phenomena in many aspects of theoretical computer science. A convenient categorical approach is to model transition systems as coalgebras for a monad T. Possible transition types include non-determinism, probability, divergence, or weights in a semiring. There are several interesting notions of morphisms of transition systems—transition preserving functions, functional bisimulations, and relation-based like simulations and bisimulations.

This thesis takes the analogy of "simulations as relational morphisms" seriously, and expresses simulations as Kleisli morphisms for a powerset-like monad on the category of non-deterministic transition systems (analogous to the relationship between **Rel** and **Set**). This monad is shown to be an instance of a lax distributive law $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$. In a more general setting, we seek to exhibit a correspondence theorem between lax distributive laws $ST \to TS$ and monads on categories of T-transition systems. The simple category of T-coalgebras turns out to be too small, so we introduce a notion of T-actions, which simultaneously generalise T-coalgebras and monoid actions. We frame many well known constructions of transition systems (including the cartesian closed structure, and a notion of graph homotopy) in terms of T-actions.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

I would like to thank my supervisor Andrea Schalk for her unceasing support over the past three years. Thank-you for always being there when I needed something, and for having an answer to every question. I'm incredibly grateful for the detailed feedback on my roughest drafts, and for all the good advice. I don't think I can imagine a better supervisor.

Thanks to all my other colleagues in the Formal Methods group and to my co-supervisor Renate Schmidt. A special thank-you to Joe Razavi, Toby Osborne, Francisco Lobo, and Bence Babrian, for the enlightening seminars, and for listening to my own half-formed ideas and always having great feedback and suggestions.

Thank-you to all the colleagues I worked with, and my fellow teaching assistants, and to all the students I enjoyed teaching.

I'm grateful to all the friends I made in Manchester. I would have gone crazy without you. A fond thank-you to the Grove Bar gang, to all our merry nights, and to watching the sun go up—Vicky, Maryam, Franz, Ani, Abdullah, Vlad, David, Alessio, Jacob, and Joe. A particular thank-you to Kai Xu, for feeding me when I was hungry, and for pouring me a drink when I was thirsty. For the coffees in the morning, ales in the afternoon sun, and lavish roast dinners, and always being up for a good time.

A massive thank-you to William Matthews. Nothing will beat the old Oxford days, but we had some moments. This is to all the pints and curries and double triples, the scoops and gimmicks, all the lanes we won and the games we lost, to the memories that we forgot we had, and the best days that are yet to come.

A warm thank-you to the woman I admire most, Lisa Malki. I could not have done this without you by my side, or without your constant kindness, and your words of affirmation. Thank you for everything.

And finally, the biggest thank-you to my sister Sofia, and my parents Radoslav and Ľubica. I would not have made it this far without you, and your support in

every way. For being there for me, and always believing in me, ďakujem.

AMDG

Chapter 1

Introduction

The goal of this thesis is to formulate a unified, categorical setting for diverse types of transition systems. There are two over-arching themes of work. The first is defining a notion of "homotopy" for edge-preserving functions on transition systems. This generalises the existing theory of $\times$-homotopy for undirected graphs, extending it to the case of labelled, directed systems, and then later to systems valued over an appropriate semiring (for example, distance weighted graphs, or generalised NFAs). The second theme is understanding simulations (an important class of morphisms between transition systems) as being the Kleisli morphisms of a certain monad of simulation Sim.

Context

Categories of transition systems

Transition systems and similar structures appear in many diverse areas of Computer Science. Some variations include *finite automata* [49], *labelled transition systems* [44, 9], *probabilistic systems* [34], *weighted automata* [10, 43], and even directed and undirected graphs [17].

Category theory, a field of mathematics pioneered by Mac Lane and Eilenberg, has been used by theoretical computer scientists to provide an abstract and generic foundation for their objects of study. By recognising that a deterministic finite automaton and a probabilistic model of a biological system are, in some sense "the same sort of thing", the results of one field of study can be *translated* to the other, instead of having to be rediscovered all over again.

A particular benefit of category theory is that it studies not just *objects* (transition systems, in our case), but *morphisms as well*—that is, ways of *moving from* one transition system to another. There are many notions of a morphism of transition systems. These include transition-preserving functions, but also things like *bounded morphisms* [9], functional bisimulations [29], and even relational notions like simulations and bisimulations [49].

Monads of "transition type"

The approach of this thesis is to model the "computational effects" of transition systems as *monads*, following the approach of Moggi [41]. Intuitively, a monad consists of a sort of computation effect, as well as a way of wrapping a value in the effect, and a way of composing one effectful computation with another.

For example, the monad of non-deterministic choice is the *powerset monad* $\mathcal{P}$. The action of $\mathcal{P}$ on a set A is to return the set of all subsets of A, $\mathcal{P}(A) = \{X : X \subseteq A\}$. What this means in terms of transition systems, is that we can model a non-deterministic transition system as a set of states A, and a transition function $\alpha : A \to \mathcal{P}A$. The transition function provides, for every state $a \in A$, a set of *successor* states $\alpha(a) \subseteq A$. It is also necessary to specify what $\mathcal{P}$ does to functions. If $f : A \to B$ is a function on a set of states, then we can lift it to a function that acts on sets of succesors in the following way. The *direct image* function $\mathcal{P}f : \mathcal{P}A \to \mathcal{P}B$ is defined by $\mathcal{P}f(U) = \{f(u) : u \in U\}$.

The "wrapping" component of the powerset monad is denoted $\eta^{\mathcal{P}} : A \to \mathcal{P}A$. It encodes the trivial transition system where every state can transition only to itself: $\eta^{\mathcal{P}}(a) = \{a\}$. Composing effects is modelled by a function $\mu^{\mathcal{P}} : \mathcal{P}\mathcal{P}A \to \mathcal{P}A$, where $\mu^{\mathcal{P}}F = \bigcup F$. This lets us compute "multi-step transitions functions". Imagine that a is a state of A, then $\alpha(a) \in \mathcal{P}A$ is the set of all succesors of a. In order to find all the states that are *2* transitions away from a, we can find the succesors of all the successors of a, that is, compute $\mathcal{P}(\alpha)(\alpha(a)) \in \mathcal{P}\mathcal{P}A$ to find a set of subsets of A. The function $\mu^{\mathcal{P}}$ lets us collect all of these successors of successors into a single subset of A, $\mu^{\mathcal{P}}(\mathcal{P}(\alpha)(\alpha(a)))$.

Formally, the monad $(\mathcal{P}, \eta^{\mathcal{P}}, \mu^{\mathcal{P}})$ consists of an endofunctor $\mathcal{P}$ on **Set**, and natural transformations $\eta^{\mathcal{P}} : \mathrm{Id} \to \mathcal{P}$ and $\mu^{\mathcal{P}} : \mathcal{P}\mathcal{P} \to \mathcal{P}$.

Some other monads that encode important types of transition effects that we will be considering are the monad $\mathcal{D}D$ of *probability distributions*, and the monads $\mathcal{P}_S$ of *semiring valued subsets*. We take the coalgebraic approach of [45],

and model transition systems for a monad T as the combination of a set of states A, and a transition function $\alpha : A \to TA$. The morphisms, however, cannot be morphisms of T-coalgebras. This condition is too strict, and encodes the functional bisimulations [29]. Rather we will take the *lax cohomomorphisms* [15] (hence we work in a **Pos**-enriched setting).

Functions and simulations

The powerset monad has another role to play. Apart from modelling systems with non-deterministic choice, it is also used to express simulations in terms of edge-preserving functions. This is via the *Kleisli construction*. Functions of the type $f : A \to \mathcal{P}B$ associate to every element a of A a *set* of elements $f(a)$ of B. If we are thinking of $\mathcal{P}$ as encoding *non-determinism*, then we can think of f as being a non-deterministic function $A \to B$, that given an input in A may return one of several possible outputs in B, or perhaps none at all.

That is, the function $f : A \to \mathcal{P}B$ encodes a *relation* $R \subseteq A \times B$. Formally, R is the relation $\{(a, b) : b \in f(a)\}$. And this construction can be reversed—we may think of any relation $R \subseteq A \times B$ as inducing a function $A \to \mathcal{P}B$. Moreover, composition of relations can be defined in terms of composing the non-determinism of $\mathcal{P}$. Essentially, we can construct the category of sets and relations (with relational composition) using nothing more than composition of functions and the tools provided by the $\mathcal{P}$ monad on **Set**.

Similarly, the simulations will be Kleisli morphisms of a monad on the category of transition systems with edge-preserving functions as morphisms. This monad will be induced by a lax distributive law of type $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$, representing the fact that $\mathcal{P}$ is both the monad of transition, and the *relational* monad of simulations.

Distributive laws

In the most general case of a monad (T, η^T, μ^T) on a category $\mathcal{C}$, and a mere functor $S : \mathcal{C} \to \mathcal{C}$, a (functorial) distributive law of type $ST \to TS$ consists of a natural transformation $\lambda : ST \to TS$ that satisfies two conditions, one involving η^T, and another involving μ^T. There is a correspondence (first stated by Beck [7]) between such functorial distributive laws λ, and functors $\underline{S}$ on the Kleisli category $\mathcal{C}_T$ that *extend* S.

This correspondence can be strengthened. When S is not just a functor but

a monad (S, η^S, μ^S), we may formulate two additional η^S and μ^S conditions for distributive laws. This leads to a correspondence between monad distributive laws λ, and *monads* $(\underline{S}, \eta^{\underline{S}}, \mu^{\underline{S}})$ on $\mathcal{C}_T$.

When we seek to generalise this correspondence to the lax setting, things become more complicated. Additional coherence conditions (whiskering conditions) are needed to guarantee a bijective correspondence. And the extended functor $\underline{S}$ (even when S has the structure of a monad) will no longer be a monad on $\mathcal{C}_T$. Instead, the η^S and μ^S conditions mean that η^S and μ^S are *lax cohomomorphisms*.

Contributions

The first set of contributions of this thesis are to do with generalising results from the category of undirected graphs **Gph** to the directed, labelled setting of the category of non-deterministic transition systems labelled in Σ, denoted $\mathbf{TS}_\Sigma$. In particular, we show that there is a cartesian closed structure. While [39] identifies a *weak exponential* object, we show that the true exponential $\mathcal{A} \Rightarrow \mathcal{B}$ must contain all the **Set**-functions $\mathcal{A} \to \mathcal{B}$ and not merely the morphisms. Transitions $f \xrightarrow{\sigma} g$ in the exponential object have a *homotopic* interpretation that generalises the $\times$-homotopy theory of undirected graphs [17].

The next set of results investigates the nature of the category of transition systems with *simulations* as morphisms (denoted $\mathbf{TSS}_\Sigma$). Our working motto is that the relation between $\mathbf{TSS}_\Sigma$ and $\mathbf{TS}_\Sigma$ should be like that of **Rel** and **Set**. And indeed, we present the result of [39] that the category $\mathbf{TSS}_\Sigma$ is isomorphic to the Kleisli category of the Sim monad on $\mathbf{TS}_\Sigma$, and generalise this result to novel monads RevSim and DSim that encode *reverse* and *double* simulations.

The goal of the remainder of the thesis is to generalise these two sets of results to transition systems of different types $T : \mathcal{C} \to \mathcal{C}$. It turns out that the Sim monad is constructed from a *lax functorial extension* of $\mathcal{P}$ to **Rel**. In particular, the monad components *do not* arise as instances of a *lax monad extension*—they remain the components of $\mathcal{P}$ in **Set**. We switch our focus to better understanding lax extensions.

It is known that strict extensions of a monad S to a monad $\underline{S}$ on $\mathcal{C}_T$ correspond to strict monad distributive laws $ST \to TS$. In the lax setting (where the Kleisli category of T is equipped with a **Pos**-enrichment), we demonstrate, in detail, correspondence between lax monad distributive laws and lax extensions of a monad

S to a lax functor $\underline{S}$ on $\mathcal{C}_T$. This generalises a result of Tholen [53], which was limited to the case of $T = \mathcal{P}_Q$, for Q a quantale. Importantly, $\underline{S}$ does not become a monad on the Kleisli category, nor do we expect it to.

The question now becomes: "on which category *is* $\underline{S}$ a monad?" The answer must be something like "a category of T-coalgebras", because that is what happens with the Sim monad on $\mathbf{TS}$. Unfortunately, the category of T-coalgebras is not large enough. A monad on this category does not provide enough information to recover a lax monad extension $\underline{S}$ on $\mathcal{C}_T$.

In order to end up with a robust correspondence theorem, between lax monads extensions $\underline{S}$ on one hand, and monads on a "T-transition system category" on the other, we need a horizontal categorification. Rather than mere T-coalgebras, we show that the appropriate notion is that of a T-action, which is essentially a (possibly lax) functor $\Delta : \mathcal{D} \to \mathcal{C}_T$, for some $\mathbf{Pos}$-enriched category $\mathcal{D}$. We construct a "hom-like" functor $T\text{-}\mathbf{Act}(-) : \mathbf{PosCat}^{\mathrm{op}} \to \mathbf{Cat}$ that generalises the category of T-coalgebras.

We prove a Yoneda-style result. Natural transformations $\mathbb{S} : T\text{-}\mathbf{Act}(-) \to T\text{-}\mathbf{Act}(-)$ are in correspondence with lax functorial distributive laws $ST \to TS$. Furthermore, we show that lax monad distributive laws correspond to monads on this functor (in a 2-categorical sense). The Sim monad is an instance of this correspondence.

Finally, we turn our attention to another specific case: transition systems that are *valued* in a semiring S (that is, for the monad $T = \mathcal{P}_S$), and in particular, the case when S is not just a semiring but a distributive monoidal lattice. We present pairs of lax laws $\mathcal{P}^f\mathcal{P}_S \to \mathcal{P}_S\mathcal{P}^f$ that generalise the laws $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$. Our final line of investigation is into the categories $\mathcal{P}_S\text{-}\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$. We show that when the lattice S admits all joins, the category of $\mathcal{P}_S$ transition systems is cartesian closed, and when S is residuated, the category $\mathcal{P}_S\text{-}\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$ is a *residuated category*. Both of these constructions provide novel notions of S-valued homotopies for S-valued transition systems.

Related work

The exponential object in $\mathbf{TS}_\Sigma$ is essentially a generalisation of the exponential object in $\mathbf{Gph}$. The corresponding notion of homotopy has been identified by Dochtermann [17] as $\times$-homotopy for undirected graphs . The results concerning

"spider moves" in Section 3.3 are a generalisation of results found in [13].

The important result that simulations are Kleisli morphisms of a monad on $\mathbf{TS}_\Sigma$ originates with Malacaria [39]. This seems to have gone mostly unnoticed in further literature.

A very important strand of work is the theory of relators. First introduced by Thijs in [52], relators (for a functor T on **Set**) provide a lifting of relations $A \nrightarrow B$ to relations $TA \nrightarrow TB$. Many notions of simulation (including cosimulations and reverse simulations) can be expressed via $\mathcal{P}$-relators. Formally, T-relators correspond to lax distributive laws $T\mathcal{P} \to \mathcal{P}T$ (although they are not usually described this way). The approach described in this thesis (arriving at simulations through Kleisli categories) uses the same machinery as relators, but in a very different way.

The theory of lax distributive laws has been used to characterise lax extensions of a monad T on **Set** to the category Q-**Rel**, for Q a quantale. Tholen presents a satisfying correspondence theorem in [53]. One of the lax laws described in Chapter 7 is identified in [33]. Lax distributive laws have seen significant application in the field of *monoidal topology*. The connections between such work and this thesis are still unclear.

Outline

This thesis begins by recollecting some important mathematical notions that will be used in the later chapters. Chapter 3 contains a detailed accounting of the "naive" category of transition systems $\mathbf{TS}_\Sigma$, with concrete descriptions of the cartesian closed structure.

In Chapter 4 we detail the *coalgebraic* view of transition systems as coalgebras $\alpha : A \to TA$ for a transition monad T, and explore some specific examples (the monad $\mathcal{P}$ is the transition type for **TS**). We include the correspondence theorem for distributive laws and extensions, and a generalisation to the lax case.

Chapter 5 examines in detail the specific case of lax laws $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$, including the Sim construction. We show that there are two lax laws $\ell^+, \ell^- : \mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$, both of which can be constructed from the unique lifting of $\mathcal{P}$ to **Rel** [23]. We compare the Kleisli approach to simulations with the *relators* of Thijs [52].

In Chapter 6 we present a converse result. The full definition of the T-**Act**$(-)$ functor is built up in several steps. We show that monads on this functor

correspond to lax distributive laws $ST \to TS$.

Finally, Chapter 7 contains some results for *semiring* valued transition systems. We generalise the laws of Chapter 5 to obtain pairs of lax laws $\mathcal{P}^f\mathcal{P}_S \to \mathcal{P}_S\mathcal{P}^f$, when S is a distributive monoidal lattice. Furthermore, we see that when S admits all joins the category $\mathcal{P}_S$-$\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$ is cartesian closed. When S is a *residuated lattice*, we can construct residuated $\mathcal{P}_S$-systems.

Chapter 2

Preliminaries

2.1 Category theory

We shall require some basic definitions of category theory. A good reference is [38]. The reader shall need to be familiar with basic constructions like products, coproducts, and exponentials, as well as notions like functors, natural transformations, and adjunctions. We will also dip into 2-category theory, and some notions of enriched categories [30].

Definition 2.1. The category **Set** has as objects the collection of all sets, and as morphisms $f : A \to B$, functions from A to B. The full subcategory **FinSet** consists of just the finite sets.

The category **Rel** has the same objects as **Set**. A morphism $R : A \nrightarrow B$ is a relation from A to B, that is, a subset of the product $R \subseteq A \times B$.

The category **Pos** has as objects all partially ordred sets $(A, \leq)$. A morphism $(A, \leq) \to (B, \leq)$ is an order-preserving function $f : A \to B$.

The category **Cat** has small categories as objects. The morphisms $F : \mathcal{C} \to \mathcal{D}$ are functors.

We recall the definition of a monad. Let $\mathcal{C}$ be a category.

Definition 2.2 (Monad)**.** A *monad on* $\mathcal{C}$ consists of an endofunctor $T : \mathcal{C} \to \mathcal{C}$, and natural transformations $\eta^T : \mathrm{Id}_{\mathcal{C}} \to T$ and $\mu^T : T^2 \to T$ such that diagrams below (depicted concretely) commute for every object A of $\mathcal{C}$.

This definition can be generalised to an arbitrary 2-category $\mathfrak{K}$. Accordingly, let C be an object of $\mathfrak{K}$.

$$
\begin{tikzcd}
T^3A \arrow[r, "T\mu^T_A"] \arrow[d, "\mu^T_{TA}"'] & T^2A \arrow[d, "\mu^T_A"] \\
T^2A \arrow[r, "\mu^T_A"] & TA
\end{tikzcd}
\qquad
\begin{tikzcd}
T^2A \arrow[dr, "\mu^T_A"'] & TA \arrow[l, "T\eta^T_A"'] \arrow[d, equal] \\
& TA
\end{tikzcd}
\qquad
\begin{tikzcd}
TA \arrow[r, "\eta^T_{TA}"] \arrow[d, equal] & T^2A \arrow[dl, "\mu^T_A"] \\
TA &
\end{tikzcd}
$$

(μ^T) $\qquad$ $(\eta^T\text{-right})$ $\qquad$ $(\eta^T\text{-left})$

Figure 2.1: The concrete monad laws

Definition 2.3. A *monad on* C consists of a 1-cell $t : C \to C$ and a pair of 2-cells, $\eta^t : \mathrm{id}_C \to t$ and $\mu^t : t^2 \to t$, such that the diagrams below commute.

$$
\begin{tikzcd}
t^3 \arrow[r, "t\mu^t"] \arrow[d, "\mu^t t"'] & t^2 \arrow[d, "\mu^t"] \\
t^2 \arrow[r, "\mu^t"] & t
\end{tikzcd}
\qquad
\begin{tikzcd}
t^2 \arrow[dr, "\mu^t"'] & t \arrow[l, "t\eta^t"'] \arrow[d, equal] \\
& t
\end{tikzcd}
\qquad
\begin{tikzcd}
t \arrow[r, "\eta^t t"] \arrow[d, equal] & t^2 \arrow[dl, "\mu^t"] \\
t &
\end{tikzcd}
$$

(μ^t) $\qquad$ $(\eta^t\text{-right})$ $\qquad$ $(\eta^t\text{-left})$

Figure 2.2: The monad laws in a 2-category

Accordingly, a monad in the sense of Definition 2.2 is a monad in the 2-category **Cat**. Most of the monads we shall encounter in this thesis will be concrete monads in **Cat**. However, Definition 2.3 is presented here because in Chapter 6 we define monads of simulation in a 2-category of lax functors.

Example 2.4. The identity functor on any category is a monad $(\mathrm{Id}, 1, 1)$.

Example 2.5. The powerset monad on **Set** is denoted by $(\mathcal{P}, \eta^{\mathcal{P}}, \mu^{\mathcal{P}})$. The action of $\mathcal{P}$ is to send a set A to the collection of all subsets of A. For a function $f : A \to B$, we have

$$\mathcal{P}(f)(U) = \{f(u) : u \in U\}.$$

The components of the monad are given by

$$\eta^{\mathcal{P}}(a) = \{a\}$$
$$\mu^{\mathcal{P}}(F) = \bigcup F.$$

Example 2.6. The $(\mathrm{Maybe}, \eta^{\mathrm{Maybe}}, \mu^{\mathrm{Maybe}})$ monad is defined on **Set** in the following way. The functor Maybe sends a set A to the set $A \cup \{\bot\}$, where $\bot \notin A$

represents "undefined". For a function $f : A \to B$, we define Maybe f by

$$\begin{aligned} \text{Maybe}\, f(a) &= f(a) \text{ for } a \in A \\ \text{Maybe}\, f(\bot) &= \bot. \end{aligned}$$

The components are defined by the rules

$$\begin{aligned} \eta^{\text{Maybe}}(a) &= a \\ \mu^{\text{Maybe}}(a) &= a \\ \mu^{\text{Maybe}}(\bot) &= \bot. \end{aligned}$$

Example 2.7. Let $f : A \to S$ be a function. The *support* of f is the set

$$\text{supp}(f) = \{a \in A : f(a) \neq 0\}.$$

The (finite) *distribution monad* on **Set** is denoted $(\mathcal{D}, \eta^{\mathcal{D}}, \mu^{\mathcal{D}})$. For any set A, we take $\mathcal{D}A$ to be the set

$$\mathcal{D}A = \left\{ U : A \to [0,1] \mid \sum_{a \in A} U(a) = 1 \text{ and } \text{supp}(U) \text{ is finite.} \right\}.$$

The elements U are probability distributions on A. For any element $a \in A$, the value $U(a) \in [0,1]$ indicates how likely it is to occur.

The action of $\mathcal{D}$ on a function $f : A \to B$ is to map a distribution on A to a distribution on B in the following way.

$$\mathcal{D}f(U)(b) = \sum_{a \in f^{-1}(b)} U(a).$$

The components of the monad are given by

$$\begin{aligned} \eta^{\mathcal{D}}(a)(a') &= \begin{cases} 1 & a = a' \\ 0 & a \neq a' \end{cases} \\ \mu^{\mathcal{D}}(F)(a) &= \sum_{U \in F} F(U) \cdot U(a). \end{aligned}$$

The notation $U \in F$ in the sum defining $\mu^{\mathcal{D}}$ is shorthand for $U \in \text{supp}\, F$. As the support is finite, this sum is well-defined.

Definition 2.8. Let T be a monad on $\mathcal{C}$. The *Kleisli category* $\mathcal{C}_T$ has

- the same collection of objects as $\mathcal{C}$
- for every morphism $f : A \to TB$ in $\mathcal{C}$, there is a morphism $f : A \to B$ in $\mathcal{C}_T$.

To avoid confusion, we will write type signatures in Kleisli categories with $\nrightarrow$, so $f : A \nrightarrow B$ indicates that f is a morphism $A \to TB$ in $\mathcal{C}$.

Composition of morphisms in $\mathcal{C}_T$ is indicated with $\bullet$, and defined in the following way. If $f : A \nrightarrow B$ and $g : B \nrightarrow C$,

$$(g \bullet f) = \mu^T_C \circ Tg \circ f : A \to TC = A \nrightarrow C \tag{2.1}$$

The identity morphism is given by $\eta^T_A : A \nrightarrow A$. We will always denote the identity in $\mathcal{C}_T$ with η^T, never with id_A, which will be reserved for the identity morphism $A \to A$ in $\mathcal{C}$.

Associativity of composition and the unital property of the identity follow directly from the monad laws. There is an important adjunction between $\mathcal{C}$ and the Kleisli category $\mathcal{C}_T$.

Definition 2.9. There is a *free functor* $F_T : \mathcal{C} \to \mathcal{C}_T$ with

- $F_T A = A$, and
- for $f : A \to B$, $F_T f = \eta^T_B \circ f : A \nrightarrow B$.

There is also a *forgetful functor* $U_T : \mathcal{C}_T \to \mathcal{C}$ with

- $U_T A = TA$, and
- for $f : A \nrightarrow B$, $U_T f = \mu^T B \circ Tf : TA \to TB$.

There is an adjunction $F_T \dashv U_T$, with unit $\eta : \mathrm{Id} \to U_T F_T$ and counit $\varepsilon : F_T U_T \nrightarrow \mathrm{Id}$ defined by

- $\eta_A = \eta^T_A : A \to TA$, and
- $\varepsilon_A = \mathrm{id}_{TA} : TA \nrightarrow A$.

Note that **Rel** is isomorphic to the Kleisli category of the powerset monad $\mathcal{P}$ on **Set**. Hence we will write type signatures of relations in the Kleisli style, as $A \nrightarrow B$.

Sometimes (this is often the case for **Rel**, and also for categories of *generalised relations*) we will prefer to write composition diagrammatically. This will usually be denoted with $\fatsemi$, hence

$$R \fatsemi S = S \bullet R.$$

The category **Rel** also has the structure of a *dagger category* (in the sense of [1]). For every relation $R : A \nrightarrow B$ we have the corresponding *opposite relation* $R^\dagger : B \nrightarrow A$ given by

$$(b, a) \in R^\dagger \iff (a, b) \in R$$

Lemma 2.10. *Let T be a monad on a category $\mathcal{C}$. A morphism $f : A \nrightarrow B$ is monic in the Kleisli category $\mathcal{C}_T$ if and only if the morphism $\mu^T_B \circ Tf : TA \to TB$ is monic in $\mathcal{C}$.*

Proof. Note that for a morphism $g : X \to TA$ (which may be considered a Kleisli morphism $X \nrightarrow A$) the following equality holds:

$$f \bullet g = \mu^T_B \circ Tf \circ g$$

Therefore, if g and h are parallel morphisms $X \to TA$, the equality

$$f \bullet g = f \bullet h,$$

can be rewritten as

$$\mu \circ Tf \circ g = \mu \circ Tf \circ h.$$

The result follows. □

Corollary 2.11 (Folklore)**.** *The monics in **Rel** are precisely the relations $R : A \nrightarrow B$ with corresponding direct image map $\mathcal{P}(A) \to \mathcal{P}(B)$ injective.*

Definition 2.12. Let T be an endofunctor on $\mathcal{C}$. A *T-coalgebra* consists of an object A of $\mathcal{C}$, and a morphism $\alpha : A \to TA$.

Let $\alpha : A \to TA, \beta : B \to TB$ be T-coalgebras. A morphism of T-coalgebras $\alpha \to \beta$ consists of a morphism $f : A \to B$ such that the diagram below commutes.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\alpha"'] & B \arrow[d, "\beta"] \\ TA \arrow[r, "Tf"] & TB \end{tikzcd}$$

The category of T-coalgebras is denoted $\mathbf{Coalg}(T)$.

2.2 Algebra

We shall recall some basic notions of algebra.

We will write $\Sigma^\star$ for the free monoid on a set Σ. When M is a monoid, we will denote the associated *contravariant* category by $\underline{M}$. This is the category with a single object $\star$, and for every $m \in M$ a morphism $m : \star \to \star$. Composition is given by reversing multiplication in M

$$m \circ n = nm.$$

The identity morphism is the one corresponding to the unit 1 of M.

Note that the covariant version (where $m \circ n = mn$) is perhaps more common in the literature. But for our purposes, contravarience will simplify the notation of Chapter 6.

Definition 2.13. A *semiring* $(S, 0, 1, +, *)$ consists of a set S, with two (distinct) distinguished elements 0 and 1, and two binary operations $+$ and $*$. These satisfy the following axioms.

- $(S, 0, +)$ is a commutative monoid.
- $(S, 1, *)$ is a monoid.
- Multiplication distributes over addition:

$$a * (b + c) = a * b + a * c$$
$$(a + b) * c = a * c + b * c.$$

- Multiplication by zero is annihilating:

$$0 * a = a * 0 = 0.$$

A semiring is called *commutative* if the multiplication is commutative.

A semiring is *idempotent* if the addition is idempotent, that is, if we have $a + a = a$ for all a.

A semiring is called *complete* if we can form infinite sums. That is, if for every family $\{a_i\}_I$ we can take the sum $\sum_I a_i$ such that

$$b * \left(\sum_I a_i\right) = \sum_I (b * a_i)$$
$$\left(\sum_I a_i\right) * b = \sum_I (a_i * b)$$

Example 2.14. Some examples of semirings are:

1. The boolean semiring $(\mathbb{B}, 0, 1, \vee, \wedge)$. It is complete by taking $\sum = \exists$.

2. Every ring is a semiring.

3. The natural numbers with addition and multiplication $(\mathbb{N}, 0, 1, +, *)$.

4. The tropical *min-plus* semiring on the extended natural numbers, $(\mathbb{N} \cup \{\infty\}, \infty, 0, \min, +)$ (with ∞ being absorbing with respect to addition and multiplication) is a complete semiring (because every non empty set of natural numbers has a least element). This is also denoted $\mathcal{N}$.

5. The probabilistic semiring $(\mathbb{R}^{\geq 0}, 0, 1, +, \cdot)$.

6. The semiring of regular languages over an alphabet Σ (or equivalently, regular expressions up to language equivalence), which is denoted $(\mathrm{Regex}\,\Sigma, \emptyset, \varepsilon, \cup, \cdot)$.

7. For any set B, there is a semiring $(\mathcal{P}(B), \emptyset, B, \cup, \cap)$.

When $(S, 0, 1, +, *)$ is a semiring, we may define a notion of an S-valued subset. Let A be a set. Intuitively, an S-valued subset of A is something like a probability

distribution on A. When we have a "real" subset $U \subseteq A$, we can, for any element $a \in A$, ask the question "is a an element of U?" and expect the answer to be true or false. An S-valued subset on the other hand, answers the question "*to what extent* is a and element of U?" with a value in S. Formally, we say that an S-valued subset of A is simply a function $U : A \to S$. The value $U(a) \in S$ expresses how strongly a is a member of U.

Note that when we take $S = \mathbb{B}$ to be the boolean semiring, a $\mathbb{B}$-valued subset is precisely a subset. As the notion of a semiring valued subset generalises that of a subset, so too can we define a monad of S-valued subsets that will generalise the powerset monad $\mathcal{P}$. We will model such S-valued subsets of A as functions $A \to S$. For our purposes, we will consider only those functions that take a non-zero value on only finitely many elements of A.

Definition 2.15. Let $(S, 0, 1, +, *)$ be a semiring.

The *generalised S-valued powerset monad* on **Set** is denoted $(\mathcal{P}_S, \eta^S, \mu^S)$. The functor $\mathcal{P}_S$ sends a set A to the set of all functions $A \to S$ with finite support. For a function $f : A \to B$, the direct image is defined as

$$\begin{aligned} \mathcal{P}_S(f) &: \mathcal{P}_S(A) \to \mathcal{P}_S(B) \\ \mathcal{P}_S(f)(U)(b) &= \sum_{a \in f^{-1}(b)} U(a). \end{aligned}$$

The monad components are given by:

$$\begin{aligned} &\eta_A : A \to \mathcal{P}_S A \\ &\eta_A(a)(a') = \delta_{a,a'} \\ &\mu_A : \mathcal{P}_S(\mathcal{P}_S A) \to \mathcal{P}_S A \\ &\mu_A(F)(a) = \sum_{U : A \to S} F(U) \cdot U(a). \end{aligned}$$

Note that the sum in the definition of μ is well defined, as even though it ranges over infinitely functions $A \to S$, it has only finitely many non-zero terms.

When S is a complete semiring, we may define the *full generalised powerset monad* on **Set**. This sends a set A to the set of all functions $A \to S$. Completeness is required, as the sums in the definition of the direct image and the μ^S may be infinite when A is infinite.

Remark 2.16. Note that the distribution monad $\mathcal{D}$ of Example 2.7 is *not* the

semiring monad associated to the semiring $(\mathbb{R}^{\geq 0}, 0, 1, +, \cdot)$. The elements of $\mathcal{D}A$ are not merely the functions $U : A \to [0,1]$ with finite support—we require the condition that $\sum_{a \in A} U(a) = 1$.

The Kleisli category of the generalised powerset monad is the category of *(finite) generalised relations*, which is denoted $\mathbf{FinRel}_S$. This is due to the isomorphism

$$\mathbf{Set}(A, (B \Rightarrow S))\mathbf{Set}(\cong A \times B, S).$$

Hence a Kleisli morphism $R : A \nrightarrow \mathcal{P}_S(B)$ may be thought of as an S-valued subset of $A \times B$. The value $R(a,b) \in S$ indicates how strongly the pair (a,b) is in R, or, how strongly a is related to b. The Kleisli composition of two generalised relations $R : A \nrightarrow B$ and $S : B \nrightarrow C$ is given by the formula

$$S \bullet R = \sum_{b \in B} R(a,b) \cdot S(b,c). \tag{2.2}$$

Naturally, when $S = \mathbb{B}$ we see that the Kleisli morphisms $A \nrightarrow B$ are precisely the relations with finite image (that is, the relations $R \subseteq A \times B$ with $\{b : (a,b) \in R\}$ finite for any $a \in A$). The Kleisli composition above agrees with composition of relations.

The structure of a semiring is all that is required to define the monads of generalised subset. Nonetheless, many relevant examples have significantly more structure. For example, that of an associated ordering, or additional operations. A particularly interesting class of semirings are those that come from lattices.

Definition 2.17. A lattice comprises a partially ordered set $(S, \leq)$ where every pair of elements x, y has both a unique *greatest lower bound*, denoted $x \wedge y$, and a unique *least upper bound* $x \vee y$. These are also called the *meet* and the *join* of x and y.

A lattice is called *bounded* if, furthermore, there are distinguished *least* and *greatest* elements (respectively denoted $\perp$ and $\top$).

A lattice is *distributive* if either of the following equations holds for all x, y, z:

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$
$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z).$$

A lattice is *complete* if every subset X of S has a greatest lower bound, and a least upper bound.

Example 2.18. The typical example of a lattice is a powerset lattice $(\mathcal{P}(B), \subseteq)$, with join and meet given by $\cup$ and $\cap$. This lattice is bounded by $\emptyset$ and B, and distributive, and complete.

Example 2.19. Every bounded, distributive lattice is a semiring $(S, \bot, \top, \vee, \wedge)$.

The semiring $\mathbb{B}$ is simply the powerset lattice of the singleton set $\{\star\}$.

Example 2.20. The semiring of regular languages forms a lattice. It is a sublattice of $(\mathcal{P}(\Sigma^\star), \subseteq)$. The join is the union, which is the semiring addition. The meet is intersection. Regular languages are closed under union and intersection [49].

Example 2.21. The min-plus semiring can be represented as a lattice. It is isomorphic to a sublattice of $(\mathcal{P}(\mathbb{N}), \subseteq)$. We describe the embedding of the min-plus semiring into this lattice.

The natural number n is represented by the set $\{n, n+1, \cdots\}$. The element ∞ is encoded by the empty set. Taking the minimum of two elements corresponds to the union (with unit $\infty = \emptyset$). The meet of the lattice corresponds to the maximum operation. Addition comes from element-wise addition of subsets.

The cases of Examples 2.20 and 2.21 are interesting, as they show semirings that are based on lattices, with the semiring addition being given by the lattice join. However, the multiplication is not the meet of the lattice, but a different operation. One axiomisation of this notion of a "lattice with a multiplication" is a quantale. Quantales, and in particular, the monads of quantales (or equivalently, quantale valued subsets and relations) have found significant use in the literature to express varying "truth values" [53, 56].

Definition 2.22 ([53])**.** A *quantale* is a complete lattice Q with a multiplication $*$, and a distinguished element $1 \in Q$. The multiplication must distribute over the join:

$$x * \left(\bigvee y_i\right) = \bigvee (x * y_i)$$
$$\left(\bigvee y_i\right) * x = \bigvee (y_i * x)$$

The condition of completeness, however, is quite strong. The semiring of regular languages is not a quantale, as it is not complete. If it were, then arbitrary

unions of regular languages would always be regular. But every language, including the non-regular languages, can be expressed as the (potentially infinite) union of singleton sets (which are all regular). A more appropriate notion is that of a monoidal lattice, which does not require completeness.

Definition 2.23. A monoidal lattice is a lattice $(S, \leq)$ with a distinguished element 1 and a binary operation $*$ such that

- $(S, 1, *)$ is a monoid.
- Multiplication distributes over joins:

$$a * (b \vee c) = (a * b) \vee (a * c)$$
$$(b \vee c) * a = (b * a) \vee (c * a)$$

- Multiplication is order-preserving. If $a \leq b$ then

$$c * a \leq c * b \text{ and}$$
$$a * c \leq b * c.$$

An additional property will allow us to partially undo the multiplication. A *residuated lattice* is a monoidal lattice that further satisfies the conditions below.

- For every c, a there is a greatest b with $a * b \leq c$. In this case we write $b = a \backslash c$. This is called the *right residual* of c by a.
- For every c, b there is a greatest a with $a * b \leq c$, denoted $a = c / b$. This is the *left residual* of c by b.

Example 2.24. The boolean semiring is a residuated lattice. The multiplication is the semiring multiplication ($\wedge$) and residuals are formed by material implication:

$$a \backslash c = a \Rightarrow c$$
$$c / b = b \Rightarrow c$$

Example 2.25. The min-plus semiring is a residuated lattice. The multiplication is the semiring multiplication ($+$), and residuals are formed by truncated

subtraction:

$$a \backslash c = c / a = \begin{cases} c - a & \text{if } a \leq c \\ 0 & \text{otherwise} \end{cases}$$

Example 2.26. The semiring of regular languages is a residuated lattice. The multiplication is the semiring multiplication (concatenation), and residuals are formed in the following way:

$$C / A = \{w : zw \in C \text{ for all } z \in A\}$$
$$B \backslash C = \{w : wz \in C \text{ for all } z \in A\}$$

Some justification for the regularity of these languages is required. We sketch a technique for construction an appropriate automaton that recognises the residual languages (see [49] for details). Let (D, R, s, F) be a DFA for the regular language C. For a word $w = w_1 w_2 \cdots 2_n \in \Sigma^\star$ we will let $R_w(x)$ denote the unique state y with

$$x \xrightarrow{w_1} x_1 \xrightarrow{w_2} x_2 \to \cdots \xrightarrow{w_n} y.$$

- The DFA for $B \backslash C$ is essentially that of C, but with a different set of accepting states. The set of accepting states is given by

  $$F' = \{x \in D : \text{ for all } w \in B, \ \ R_w(x) \in F\}.$$

  We see that a word u is accepted by (D, R, s, F') if it ends up in a state where inputting any word w that is in B ends up being accepted by the DFA for C.

- The case of C / A is a little more complicated. First we define the set of states

  $$S = \{R_w(s) : w \in A\}.$$

  This is essentially the set of states that a word in A could end up in. Now we construct a DFA that consists of $|S|$ copies of the DFA for C (the product construction [49]). The start state is defined by choosing a different state in

S for each copy. Running a word w in this product automata corresponds to running w in the DFA for C several times, starting at each state in S, and accepting only if w always ends up in an accepting state, no matter which state in S we start from. This corresponds to uw being in C for any word $u \in A$, which is exactly the property we need.

Chapter 3

The simple category of transition systems

In this chapter, we shall define a convenient category of transition systems $\mathbf{TS}_\Sigma$. The morphisms will be *transition preserving functions*. We will review the elementary properties of this category. In particular, we show that the it is cartesian closed—for any two transition systems $\mathcal{A}, \mathcal{B}$, we can form the *exponential* system $\mathcal{A} \Rightarrow \mathcal{B}$ consisting of *all functions* $\mathcal{A} \to \mathcal{B}$ (not merely the morphisms).

This construction is known in the case of undirected graphs [24], but does not appear to have been succesfully generalised to transition systems (labelled, directed graphs) yet. Unfortunately, while an appropriately formulated category of graphs is a topos [55], the category $\mathbf{TS}_\Sigma$ is not, essentially because we do not require reflexivity.

Following in this vein, we adapt some definitions of *graph homotopy* theory [40, 17, 27] to the case of transition systems. We also investigate the interesting subcategories that consist of transition systems with *reflexive, symmetric, or transitive* transition relations, and define dual pairs of reflexivisation and symmetrisation functors that are adjoint to the free inclusion functors.

In the second half of this chapter we focus our attention on *simulations* of transition systems, which are essentially "transition-preserving relations", and define a category $\mathbf{TSS}_\Sigma$ of transition systems with simulations as morphisms. The overarching theme is that the relationship between $\mathbf{TS}_\Sigma$ and $\mathbf{TSS}_\Sigma$ mirrors that of **Set** and **Rel**. Indeed, we see that in $\mathbf{TSS}_\Sigma$ the products and coproducts are given by the disjoint union (as in **Rel**). The cartesian product remains a symmetric monoidal product, but there is no monoidal closed structure.

The culmination of this line of investigation is the result of Malacaria [39] that the category of simulations $\mathbf{TSS}_\Sigma$ arises as the Kleisli category of a *simulation monad* Sim on $\mathbf{TS}_\Sigma$, mirroring the fact that **Rel** is the Kleisli category of $\mathcal{P}$ on **Set**. A novel contribution of this thesis is the identification of several variations of the Sim monad that result in categories of transition systems with *reverse simulations*, and other related notions.

We shall begin with the following primitive formulation of a transition system. This is a relatively standard presentation, and can be found in, for example, [21]. Let Σ be a finite set of labels. A transition system (over Σ) comprises

- a set of states A, and
- a transition relation $R \subseteq A \times \Sigma \times A$.

When the transition relation R is clear from context, we may indicate that $(a, \sigma, a') \in R$ by writing

$$a \xrightarrow{\sigma} a'.$$

In writing, we can say that *a transitions to a'* (*by* or *along* the letter σ).

When Σ is a singleton set we may write $\Sigma = *$, and we call such transition systems *unlabelled*.

Let (A, R) and (B, S) be two transition systems. There are several interestion ways in which a function of carrier sets may interact with the transition structure.

Definition 3.1. Let $f : A \to B$ be a function of underlying sets.

1. We say that f is *transition preserving* if $(a, \sigma, a') \in R$ implies $(f(a), \sigma, f(a')) \in S$.
2. We will say that f *reflects transitions* if the converse condition holds. That is, if whenever $(f(a), \sigma, f(a')) \in S$ then we have $(a, \sigma, a') \in R$.
3. If both conditions hold then we will call f *strictly preserving*.

These conditions are depicted in Fig. 3.1.

All of these conditions are found variously in the literature. It is common to refer to a function that satisfies Condition 1 as simply a *morphism* ([39]) or a *homomorphism* ([9]). A modal logician would call a function that satisfies

$$\begin{tikzcd} a \arrow[d, "\sigma"'] & f(a) \arrow[d, dashed, "\sigma"] \\ a' & f(a') \end{tikzcd} \quad \Longrightarrow \qquad\qquad \begin{tikzcd} a \arrow[d, dashed, "\sigma"'] & f(a) \arrow[d, "\sigma"] \\ a' & f(a') \end{tikzcd} \quad \Longleftarrow$$

Figure 3.1: The transition preserving and reflecting conditions

$$\begin{tikzcd} a \arrow[r, dotted] \arrow[d, "\sigma"] & b \arrow[d, "\sigma"] \\ a_1 \arrow[r, dotted] & b' \\ a_2 \arrow[ur, dotted] & \end{tikzcd} \qquad\qquad \begin{tikzcd} a \arrow[r, dotted] & b \arrow[d, "\sigma"] \\ & b' \end{tikzcd}$$

Figure 3.2: Bounded and strictly preserving functions

conditions 1 and 2 a *strong homomorphism* ([9]), or a *functional bisimulation* ([29]).

Being strictly preserving is a very strong condition. There is another way we can strengthen the notion of a transition preserving function.

Definition 3.2. A function $f : A \to B$ is a *bounded morphism* if it is transition preserving and satisfies the condition below:

$$\text{If } (f(a), \sigma, b) \in S \text{ there exists an } a' \text{ with } f(a') = b \text{ and } (a, \sigma, a') \in R \tag{Bounded morphism}$$

The difference between a strictly preserving function and a bounded morphism is subtle. A function is strictly preserving if every transition between images can be pulled back to a transition between all preimages. On the other hand, a function is bounded if every transition originating from an image can be pulled back to a transition into a single preimage.

The following example will provide intuition. The diagram in Fig. 3.2 depicts two functions of transition systems. Both are transition preserving. The function on the left is bounded, because the transition $b \to b'$ can be pulled back to $a \to a_1$. It is not reflective, because this transition *cannot* be pulled back to $a \to a_2$.

The function on the right is strictly preserving (trivially, because b' has no preimages) but is not bounded. The eligible transition $b \to b'$ cannot be pulled back to any preimage of b'.

Apart from these diverse notions of "functions that interact with structure",

there are several ways to generalise to *relations* that "interact with structure". The most essential is simulation.

Definition 3.3. Let Z be a relation $A \nrightarrow B$ (so $Z \subseteq A \times B$). We will say that Z is a *simulation from A to B* if the following condition holds:

$$\begin{array}{ll} \text{For any related pair } (a,b) \in Z \text{ and any transition } a \xrightarrow{\sigma} a' \text{ in } A & \\ \text{we can find a state } b' \in B \text{ with } b \xrightarrow{\sigma} b' \text{ and } (a',b') \in Z & \text{(Simulation)} \end{array}$$

It can be insightful to present this graphically. The relation Z is a simulation if we can take any partial square as in Fig. 3.3 and find a state b' that "fills in" the missing bottom right corner. The term "simulation" expresses the intended intuition: a simulation from A to B means that B can "simulate" the behaviour of A. If a and b are related, then any transition going from a can be "matched" by a transition in b, that leads to a related state.

$$\begin{tikzcd} a \arrow[r, "Z"] \arrow[d, "\sigma"'] & b \\ a' & \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} a \arrow[r, "Z"] \arrow[d, "\sigma"'] & b \arrow[d, dashed, "\sigma"] \\ a' \arrow[r, dashed, "Z"] & \exists b' \end{tikzcd}$$

Figure 3.3: The simulation condition

Just as with functions, the converse and bi-directional version of a simulation are also worth considering. If we swap the role of A and B, we can express the fact that A simulates B. And of course, it is possible for a relation to be a simulation *in both directions*.

Definition 3.4. A relation $Z : A \nrightarrow B$ is called a *co-simulation* if the converse of the simulation condition holds (depicted in Fig. 3.4).

$$\begin{array}{ll} \text{For any related pair } (a,b) \in Z \text{ and any transition } b \xrightarrow{\sigma} b' \text{ in } B & \\ \text{we can find a state } a' \in A \text{ with } a \xrightarrow{\sigma} a' \text{ and } (a',b') \in Z & \text{(Co-simulation)} \end{array}$$

We will say that Z is a *bisimulation* if it is a simulation and a co-simulation.

Note that when $f : A \to B$ is a function, the direct image relation $f_* : A \nrightarrow B$ is a simulation if and only if f is transition-preserving. When f_* is a *bisimulation*, we will call f a functional bisimulation.

$$
\begin{tikzcd}
a \arrow[r, "Z"] & b \arrow[d, "\sigma"] \\
 & b'
\end{tikzcd}
\quad \Rightarrow \quad
\begin{tikzcd}
a \arrow[r, "Z"] \arrow[d, dashed, "\sigma"'] & b \arrow[d, "\sigma"] \\
\exists a' \arrow[r, dashed, "Z"] & b'
\end{tikzcd}
$$

Figure 3.4: The co-simulation condition

There is an additional way to generalise simulations. All three notions we have seen (simulations, co-simulations and bisimulations) are *going forwards*—they demand the existence of certain transition *successors*. But we can flip this around, and think about simulation-like conditions that go backwards, and enforce the existence of transition *predecessors*. We shall call these things *reverse* simulations.

Definition 3.5. Let Z be a relation $A \nrightarrow B$.

1. We will call Z a *reverse simulation* if the following condition holds:

   For any related pair $(a', b') \in Z$ and any transition $a \xrightarrow{\sigma} a'$ in A
   we can find a state $b \in B$ with $b \xrightarrow{\sigma} b'$ and $(a, b) \in Z$ (Reverse simulation)

2. If the condition below holds, then Z is a *reverse co-simulation*.

   For any related pair $(a', b') \in Z$ and any transition $b \xrightarrow{\sigma} b'$ in B
   we can find a state $a \in A$ with $a \xrightarrow{\sigma} a$ and $(a, b) \in Z$
   (Reverse co-simulation)

3. If both conditions hold then Z is a *reverse bisimulation*.

4. If Z is a simulation and a reverse simulation then it is called a *double simulation*.

Note that each of these four simulation condition (forwards simulation and cosimulation, and reverse simulation and cosimulation) can hold independently for a given relation. Therefore there are $2^4 = 16$ possible combinations. These conditions are summarised in Fig. 3.5.

The following result shows that the standard notion of simulation (Definition 3.50) is sufficient to express all the variations.

$$\begin{tikzcd} a \arrow[r, "Z"] \arrow[d, "\sigma"'] & b \arrow[d, dashed, "\sigma"] \\ a' \arrow[r, dashed, "Z"] & \exists b' \end{tikzcd}$$

Simulation

$$\begin{tikzcd} a \arrow[r, "Z"] \arrow[d, dashed, "\sigma"'] & b \arrow[d, "\sigma"] \\ \exists a' \arrow[r, dashed, "Z"] & b' \end{tikzcd}$$

Co-simulation

$$\begin{tikzcd} a \arrow[r, dashed, "Z"] \arrow[d, "\sigma"'] & \exists b \arrow[d, dashed, "\sigma"] \\ a' \arrow[r, "Z"] & b' \end{tikzcd}$$

Reverse simulation

$$\begin{tikzcd} \exists a \arrow[r, dashed, "Z"] \arrow[d, dashed, "\sigma"'] & b \arrow[d, "\sigma"] \\ a' \arrow[r, "Z"] & b' \end{tikzcd}$$

Reverse co-simulation

Figure 3.5: All variants of the simulation condition

Lemma 3.6. *Let (A, R) and (B, S) be transition systems, and $Z : A \nrightarrow B$ a relation.*

1. *The relation Z is a co-simulation from (A, R) to (B, S) if and only if $Z^\dagger : B \nrightarrow A$ is a simulation from (B, S) to (A, R).*

2. *The relation Z is a reverse simulation from (A, R) to (B, S) if and only it is a simulation from $(A, R^\dagger)$ to $(B, S^\dagger)$.*

Proof. It is straightforward to verify this. □

There is another way to express the fact that a relation is a simulation. If R is a transition relation on A, we will write R_σ to denote the relation $\{(a, a') : (a, \sigma, a') \in R\}$ on A.

Proposition 3.7. *A relation $Z : A \nrightarrow B$ is a simulation if for every letter σ the following inclusion holds:*

$$Z^\dagger \fatsemi R_\sigma \subseteq S_\sigma \fatsemi Z^\dagger \tag{3.1}$$

Proof. Note that both sides are relations $B \nrightarrow A$.

Suppose Z is a simulation. We wish to show that for every state $b \in B$ and $a' \in A$, we have

$$(b, a') \in Z^\dagger \fatsemi R_\sigma \implies (b, a') \in S_\sigma \fatsemi Z^\dagger$$

If we have $(a', b) \in Z^\dagger \fatsemi R_\sigma$ then there exists an intermediate state a with $(b, a) \in Z^\dagger$ and $(a, a') \in R_\sigma$. That is, the hypothesis of the simulation condition is met. Since Z is indeed a simulation, we deduce that there exists a b' with $(a', b') \in Z$ (and therefore $(b', a') \in Z^\dagger$) and $(b, b') \in \xrightarrow{S}_\sigma$. This b' hence witnesses $(b, a') \in S_\sigma \fatsemi Z^\dagger$.

A symmetric argument will show that if Eq. 3.1 holds then Z is indeed a simulation. □

3.1 The category TS

We proceed to define a family of categories of labelled transition systems. Let Σ be a (usually finite) set of labels.

Definition 3.8 ([39])**.** There is a *category of (non-deterministic) transition systems over* Σ, denoted $\mathbf{TS}_\Sigma$.

- The objects of $\mathbf{TS}_\Sigma$ are pairs (A, R), where A is a set (the *set of states*, or the *carrier* or *underlying set*), and R is a relation $R \subseteq A \times \Sigma \times A$ called the *transition relation*.
- A morphism $f : (A, R) \to (B, S)$ consists of a function of underlying sets $f : A \to B$ that is transition preserving.

Remark 3.9. Note that the objects of $\mathbf{TS}_\Sigma$ are "sets with extra structure" (that of a transition relation), and that morphisms are "structure preserving functions". In particular, the identity morphism is *merely* the identity *function*, and composition of morphisms is performed by composition of functions in **Set**.

Moreover, there is a forgetful functor $U_\Sigma : \mathbf{TS}_\Sigma \to \mathbf{Set}$ that throws away the transition relation component of a system. Explicitly, we have on objects $U_\Sigma(A, R) = A$ and for $f : (A, R) \to (B, S)$ we have $U_\Sigma f = f : A \to B$. This functor is *faithful*. The category $\mathbf{TS}_\Sigma$ is *concrete over* ***Set***, in the sense of [3].

This property will save us some time when we are reasoning about equations of morphisms in $\mathbf{TS}_\Sigma$. For example, to show that $\mathrm{id}_A : (A, R) \to (A, R)$ genuinely is the identity morphism in $\mathbf{TS}_\Sigma$, we would need to verify that $f \circ \mathrm{id}_A = f$ for every morphism $f : (A, R) \to (B, S)$. But we know that this equation holds in **Set**, for the underlying functions $\mathrm{id}_A : A \to A$ and $f : A \to B$.

We will tend to denote transition systems $(A, R), (B, S)$ with caligraphic letters as $\mathcal{A} = (A, R)$, and so on. As a matter of convenience, when the transition systems

$\mathcal{A} = (A, R)$ and $\mathcal{B} = (B, S)$ are clear from context, we may write a morphism $f : \mathcal{A} \to \mathcal{B}$ as simply

$$f : A \to B, \text{ or even}$$
$$f : R \to S.$$

Note that we do not allow for multiple transitions (labelled by the same letter) between states. When Σ is a singleton set we may omit it entirely, for transition systems over a singleton set are essentially unlabelled systems. We can denote this category as **TS**.

Remark 3.10. Note that in the unlabelled case we may think of a transition relation $R \subseteq A \times \{\star\} \times A$ as a relation $R : A \nrightarrow A$.

Even in the case when Σ is not a singleton, we may think of R as a family $\{R_\sigma\}$ of relations on A indexed by Σ. This is due to the isomorphism:

$$\mathcal{P}(A \times \Sigma \times A) \cong \Sigma \to \mathcal{P}(A \times A).$$

Definition 3.11. Let (A, R) be a transition system over Σ.

1. If a is a state of A, and for all $\sigma \in \Sigma$ we have $a \xrightarrow{\sigma} a$, we call the state a *reflexive*. If every state in A is reflexive then we say that the entire transition system $\mathcal{A}$ is reflexive.

2. If the existence of a transition $a \xrightarrow{\sigma} a'$ implies the opposite transition $a' \xrightarrow{\sigma} a'$ then we call $\mathcal{A}$ *symmetric*.

3. If whenever we have $a \xrightarrow{\sigma} b$ and $b \xrightarrow{\sigma} c$ then also $a \xrightarrow{\sigma} c$, then we say that $\mathcal{A}$ is *transitive*.

Hence we may define the full subcategories of reflexive, symmetric, or transitive systems, as well as any combinations of these properties.

The reflexive subcategory will be denoted by $\mathbf{TS}^{\bullet}{}_{\Sigma}$, and the symmetric subcategory by $\mathbf{TS}^{\sim}{}_{\Sigma}$. Note that $\mathbf{TS}^{\sim}{}_{*}$ is isomorphic to the category of undirected graphs **Gph** [13].

In particular, the symmetric subcategory of **TS** (that is, unlabeled symmetric transition systems) is isomorphic to the category of undirected graphs and graph homomorphisms that is considered in [19, 17].

Demanding that all systems are reflexive has the interesting effect of allowing morphisms to "collapse" edges. The subcategory of reflexive *and* symmetric (unlabeled) transition systems is isomorphic to the category of undirected *simple* graphs of [4], with weak graph homomorphisms. A weak homomorphism of graphs $f : (V_1, E_1) \to (V_2, E_2)$ is a function $f : V_1 \to V_2$ of vertices such that for every edge $(u, v) \in E_1$, either $(f(u), f(v))$ is an edge of E_2, or in fact $f(u) = f(v)$.

An important notion is that of a *subsystem*.

Definition 3.12. Let $\mathcal{A}$ be a transition system in $\mathbf{TS}_\Sigma$. A *subsystem* of $\mathcal{A}$ is a *subobject*, in the categorical sense. Concretely, this is a transition system $\mathcal{B}$ and an injective morphism $i : \mathcal{B} \to \mathcal{A}$.

Now we recall a few constructions on transition systems.

Definition 3.13. Let $\mathcal{A} = (A, R)$ and $\mathcal{B} = (B, S)$ be two transition systems.

- The *disjoint union of* $\mathcal{A}$ *and* $\mathcal{B}$ is the transition system $\mathcal{A} \bigsqcup \mathcal{B}$ that has the set of states $A \bigsqcup B$, and we have

$$\mathcal{A} \bigsqcup B : u \to v$$

  if both u and v are in A and $\mathcal{A} : u \to V$, or likewise for $\mathcal{B}$.

- The *box product of* $\mathcal{A}$ *and* $\mathcal{B}$ is the transition system $\mathcal{A} \Box \mathcal{B}$ that has the set of states $A \times B$, and we have

$$\mathcal{A} \Box \mathcal{B} : (a, b) \to (a', b')$$

  if either $a' = a$ and $b \to b'$, or if $a \to a'$, and $b = b'$.

- The *tensor product of* $\mathcal{A}$ *and* $\mathcal{B}$ is the transition system $\mathcal{A} \times \mathcal{B}$ that has the set of states $A \times B$, and we have

$$\mathcal{A} \times \mathcal{B} : (a, b) \to (a', b')$$

  if $a \to a'$ and $b \to b'$.

Proposition 3.14. *These constructions have the following properties.*

- *The disjoint union* $\bigsqcup$ *is the categorical coproduct.*

- *The tensor product $\times$ is the categorical product.*
- *The box product $\Box$ is symmetric monoidal.*

The box product is not so important for our purposes, but it does generalise the well known box product for graphs.

We shall list some important transition systems.

- Let ε be the empty transition system $(\emptyset, \emptyset)$, with no states and no transitions.
- Let $\mathbf{0}$ be the systen (0) with a single state 0, and no transitions.
- Let $\mathbf{1}$ be the system $(\circlearrowleft 0)$ with a single state 0 and a transition $0 \xrightarrow{\sigma} 0$ for all $\sigma \in \Sigma$.
- Let $\mathbf{I}_\sigma$ (for $\sigma \in \Sigma$) be the transition system $(0 \xrightarrow{\sigma} 1)$, with two states 0 and 1, and a single transition $0 \xrightarrow{\sigma} 1$. In the unlabelled case we will simply write $\mathbf{I}$.

These systems enjoy the following properties.

- ε is the initial object.
- $\mathbf{1}$ is the terminal object.
- Morphisms $\overline{a} : \mathbf{0} \to \mathcal{A}$ pick out states $a = \overline{a}(0)$ of $\mathcal{A}$.
- Morphisms $e : \mathbf{I}_\sigma \to \mathcal{A}$ pick out transitions $e(0) \xrightarrow{\sigma} e(1)$ in $\mathcal{A}$.
- Global elements of $\mathcal{A}$, that is, morphisms $\overline{a} : \mathbf{1} \to \mathcal{A}$ are the *reflexive* states of $\mathcal{A}$.

Recall that a category is finitely complete if and only if it has a terminal object and all binary products and equalisers. Dually, a category is finitely cocomplete if it has an initial object, coproducts, and coequalisers. Hence, we can prove that

Proposition 3.15. *The category $\boldsymbol{TS}_\Sigma$ is finitely complete and finitely cocomplete.*

Proof. The initial object is the transition system ε. Equalisers and coequalisers are formed just as in **Set**. The verifications are routine. $\Box$

We can also characterize the monic and epic morphisms in this category, as follows.

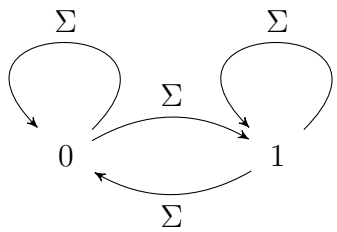


Figure 3.6: The system $\mathcal{C}$

Proposition 3.16. *A morphism in* $\boldsymbol{TS}_\Sigma$ *is monic if and only if it is an injective function. A morphism is epic if and only if it surjective.*

Proof. Let $f : \mathcal{A} \to \mathcal{B}$. Suppose that f is monic, and suppose further for the sake of contradiction that f is not injective, so we have $f(a) = f(a')$ for some $a \neq a'$. This means that $f \circ \overline{a} = f \circ \overline{a'}$, but $\overline{a} \neq \overline{a'}$. A contradiction. So every monic arrow is injective. The converse holds because $\mathbf{TS}_\Sigma$ is a concrete category.

Suppose that f is epic and not surjective. So there is a $b \in B$ that is not in the image of f. Let $\mathcal{C}$ be the transition system in Fig. 3.6. We define morphisms $g, h : \mathcal{B} \to \mathcal{C}$ by

$$g(x) = 0$$
$$h(x) = \begin{cases} 0 & \text{if } x = b \\ 1 & \text{otherwise} \end{cases}$$

It is quite clear that g and h are indeed morphisms (that is, they preserve transitions), and that $g \circ f = h \circ f$ yet $g \neq h$. So we see that every epimorphism must be surjective. Again, the converse holds because we are in a concrete category. □

3.2 Cartesian closed structure

In this section we will provide a definition of exponential objects in $\mathbf{TS}_\Sigma$. Hence $\mathbf{TS}_\Sigma$ is a cartesian closed category. Along the way, we will look at some examples of failed constructions. The fundamental construction of exponential objects is known in the case of undirected graphs. The extension to the directed and labelled case is not too hard, though to the best of my knowledge, it does not appear elsewhere in the literature. We shall examine the construction in quite a lot of

detail, and verify that all the properties hold. An exponential object $\mathcal{A} \Rightarrow \mathcal{B}$ is essentially an *internal function space*. It is characterised by the property that the functor $\mathcal{A} \Rightarrow -$ is right adjoint to the categorical product functor $- \times \mathcal{A}$.

A more concrete formulation is this.

Definition 3.17. Let $\mathcal{A}, \mathcal{B}$ be transition systems in $\mathbf{TS}_\Sigma$. An exponential object is a transition system $\mathcal{A} \Rightarrow \mathcal{B}$ with an *evaluation morphism* : $\mathrm{ev} : \mathcal{A} \times (\mathcal{A} \Rightarrow \mathcal{B}) \to \mathcal{B}$.

We require that for every morphism $g : \mathcal{C} \times \mathcal{A} \to \mathcal{B}$ there exists a unique morphism $\lambda g : \mathcal{C} \to \mathcal{A} \Rightarrow \mathcal{B}$ that makes the diagram below commute.

$$\begin{tikzcd}
\mathcal{A} \times (\mathcal{A} \Rightarrow \mathcal{B}) \arrow[r, "\mathrm{ev}"] & \mathcal{B} \\
\mathcal{A} \times \mathcal{C} \arrow[u, "\mathrm{id}_{\mathcal{A}} \times \lambda g"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Example 3.18. In the case of **Set**, the categorical product is the standard cartesian product of sets. The exponential B^A is simply the set of all functions $A \to B$.

The evaluation map is given by $\mathrm{ev}(f, a) = f(a)$, and for a function $g : C \times A \to B$, the curried function λg is defined by $\lambda g(c) = a \mapsto g(a, c)$.

As $\mathbf{TS}_\Sigma$ is concrete over **Set**, this seems like a good jumping off point. The name of the game will be to find an appropriate transition system structure on the set of morphisms $\mathcal{A} \to \mathcal{B}$ such that the **Set**-wise evaluation and currying maps are indeed *morphisms*. In [39], Malacaria identifies a *weak exponential*.

Definition 3.19. Let $\mathcal{A}, \mathcal{B}$ be two transition systems. The weak exponential $\mathcal{B}^{\mathcal{A}}$ has

- set of states given by the set of all *morphisms* $f : \mathcal{A} \to \mathcal{B}$.
- transition relation defined as follows. There is a transition $f \xrightarrow{\sigma} g$ if and only if for all letters σ and states a we have a transition $f(a) \xrightarrow{\sigma} g(a)$ in $\mathcal{B}$.

The problem with this transition structure is that the evaluation map is not a morphism. We require that

$$(a, f) \xrightarrow{\sigma} (a', g) \implies f(a) \xrightarrow{\sigma} g(a')$$

for all pairs of morphisms $f, g : \mathcal{A} \to \mathcal{B}$, and states a, a' of A. Recall that by the definition of the categorical product, we have a transition $(a, f) \xrightarrow{\sigma} (a', g)$ if

and only if we have component-wise transitions $f \xrightarrow{\sigma} g$ and $a \xrightarrow{\sigma} a'$. In general, this is not enough to deduce a transition $f(a) \xrightarrow{\sigma} g(a')$. Malacaria suggests the requirement that $\mathcal{B}$ is *transitive*. Then we could combine $f(a) \xrightarrow{\sigma} g(a)$ (from the fact that $f \xrightarrow{\sigma} g$, instantiated at a) with the transition $g(a) \xrightarrow{\sigma} g(a')$ (which we get from the fact that g is *transition preserving*) to deduce the required $f(a) \xrightarrow{\sigma} g(a')$.

Leaving aside that transitivity is a very strong (and unnaturally) property, this still would not solve the problem. There is the issue of currying. We require that for every morphism $g : \mathcal{A} \times \mathcal{C} \to \mathcal{B}$,

1. the partially applied $\lambda g(c)$ is a morphism $\mathcal{A} \to \mathcal{B}$ (so that it is an element of the exponential), and that

2. the mapping λg is transition-preserving in $\mathcal{C}$.

Neither of these conditions hold for the weak exponential.

1. in order for $\lambda g(c)$ to be a morphism we need

$$\begin{aligned} a \xrightarrow{\sigma} a' &\implies \lambda g(c)(a) \xrightarrow{\sigma} \lambda g(c)(a') \\ &\iff g(a, c) \xrightarrow{\sigma} g(a', c). \end{aligned}$$

   We cannot deduce this from g being a morphism, as we do not have $(a, c) \xrightarrow{\sigma} (a', c)$ in general, only when $c \xrightarrow{\sigma} c$.

2. in order for λg to be a morphism, we need

$$\begin{aligned} c \xrightarrow{\sigma} c' &\implies \lambda g(c) \xrightarrow{\sigma} \lambda g(c') \\ &\iff \text{for all } a \in A \;\; g(a, c) \xrightarrow{\sigma} g(a, c'). \end{aligned}$$

   We have the same problem as above.

The issue of the evaluation map can be solved by picking a different transition structure on the set of morphisms. We will say that $f \xrightarrow{\sigma} g$ if we have

$$a \xrightarrow{\sigma} a' \implies f(a) \xrightarrow{\sigma} g(a').$$

From this we immediately see that

$$\begin{aligned} (a, f) \xrightarrow{\sigma} (a', g) &\implies a \xrightarrow{\sigma} a' \text{ and } f \xrightarrow{\sigma} g \\ &\implies f(a) \xrightarrow{\sigma} g(a'), \end{aligned}$$

so the evaluation map is indeed transition preserving. We also get Condition 2, for if $c \xrightarrow{\sigma} c'$ and $a \xrightarrow{\sigma} a'$ then we have $(a, c) \xrightarrow{\sigma} (a', c')$. Because g is a morphism, we can deduce $g(a, c) \xrightarrow{\sigma} g(a', c')$, which is what we need.

Unfortunately, it is still not possible to deduce Condition 1. In general, $\lambda g(c)$ will be a morphism precisely when c is a reflexive state. One solution might be to throw up our hands and simply require, by fiat, that all of our transition systems are reflexive. This would indeed define a cartesian closed structure on the reflexive subcategory.

An alternative solution is to *expand* the exponential object $\mathcal{A} \Rightarrow \mathcal{B}$ to include *all functions* $A \to B$, and not merely the morphisms. This technique is used to construct the exponential graph, in the context of undirected graph theory [18, 24]. It is straightforward enough to adapt to the direct and labelled case.

Definition 3.20. Let $\mathcal{A}$ and $\mathcal{B}$ be two transition systems over Σ. The *exponential* object $\mathcal{A} \Rightarrow \mathcal{B}$ has

- set of states the collection of *all functions* $A \to B$, and
- a transition $f \xrightarrow{\sigma} g$ if, for all $a \xrightarrow{\sigma} a'$ we have $f(a) \xrightarrow{\sigma} g(a')$.

Evaluation and currying are defined as in **Set**.

Proposition 3.21. *With $\times$ and $\Rightarrow$, the category $\boldsymbol{TS}_\Sigma$ is cartesian closed.*

Proof. By the argument above, we can remain convinced that the evaluation map is a morphism, and that Condition 2 holds. Condition 1 becomes vacuous, as we do not need $\lambda g(c)$ to be a morphism, it is enough that it is a mere function $A \to B$. □

We may characterise the reflexive states of the function space in the following way.

Proposition 3.22. *Let f be a set-map $A \to B$. Then f is a fully reflexive state of $\mathcal{A} \Rightarrow \mathcal{B}$ (that is, $f \xrightarrow{\sigma} f$ for all σ) if and only if f is a morphism.*

Proof. Observe that $f \xrightarrow{\sigma} f$ if and only if $a \xrightarrow{\sigma} b$ implies $f(a) \xrightarrow{\sigma} f(b)$ (for all a, b). □

In particular, this means that the global elements of $\mathcal{A} \Rightarrow \mathcal{B}$, which are the morphisms $\mathbf{1} \to \mathcal{A} \Rightarrow \mathcal{B}$, are precisely the *morphisms* $\mathcal{A} \to \mathcal{B}$.

We will also look at the reflexive subcategory $\mathbf{TS}^\bullet{}_\Sigma$. The following result is not surprising.

Proposition 3.23. *The tensor product, box product, and disjoint union of two reflexive systems is itself reflexive.*

The tensor product is the categorical product in $\boldsymbol{TS}^{\bullet}{}_{\Sigma}$*, the box product is symmetric monoidal, and the disjoint union is the categorical coproduct.*

However, the category $\mathbf{TS}^{\bullet}{}_{\Sigma}$ is *not* closed under exponentiation.

Example 3.24. Let $\mathcal{A}$ and $\mathcal{B}$ be the following (unlabelled) systems. Note that they are both reflexive.

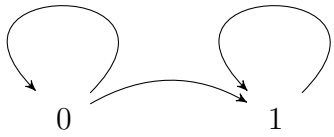


Figure 3.7: The system $\mathcal{A}$

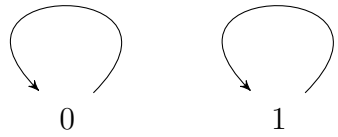


Figure 3.8: The system $\mathcal{B}$

Consider the identity function $A \to B$. This is not a morphism, for it doesn't preserve the transition $0 \xrightarrow{1}$ of A. So $\mathcal{A} \Rightarrow \mathcal{B}$ contains non-reflexive states.

We may define the following "reflexive" exponential.

Definition 3.25. Let $\mathcal{A}, \mathcal{B}$ be (not necessarily reflexive) transition systems. The *reflexive exponential* is the transition system $\mathcal{A} \Rightarrow^{\bullet} \mathcal{B}$ with

- set of states given by the collection of all *morphisms* $\mathcal{A} \to \mathcal{B}$
- a transition $f \xrightarrow{\sigma} g$ if for all $a \xrightarrow{\sigma} a'$,

$$a \xrightarrow{\sigma} a' \implies f(a) \xrightarrow{\sigma} g(a')$$

Equivalently, $\mathcal{A} \Rightarrow^{\bullet} \mathcal{B}$ is the subsystem of $\mathcal{A} \Rightarrow \mathcal{B}$ comprising of all the reflexive states.

At this point, we are left in a rather odd situation, with a cartesian closed subcategory of a CCC that has a *different structure*.

One particular point of confusion might be as follows: Suppose A, B are reflexive, and form the reflexive exponential $\mathcal{A} \Rightarrow^{\bullet} \mathcal{B}$ which satisfies certain universal properties. But $\mathcal{A}, \mathcal{B}$ exist in the super-category, so we may also form the exponential $\mathcal{A} \Rightarrow \mathcal{B}$, which satisfies the same universal properties but is decidedly *not* isomorphic to $\mathcal{A} \Rightarrow^{\bullet} \mathcal{B}$.

The solution is this. The universal properties satisfied by $\mathcal{A} \Rightarrow \mathcal{B}$ and $\mathcal{A} \Rightarrow^\bullet \mathcal{B}$ are *not* the same, for they are quantified over different domains. If $\mathcal{C}$ is not reflexive and we pick a $g : \mathcal{C} \times \mathcal{A} \to \mathcal{B}$ then we might end up with λg being a mere set-map, so we need $\mathcal{A} \Rightarrow \mathcal{B}$ to contain the set-maps. But, if $\mathcal{C}$ is reflexive then no such g will ever induce a non-morphism.

Recall that in both $\mathbf{TS}_\Sigma$ and $\mathbf{TS}^\bullet{}_\Sigma$ the terminal object is $\mathbf{1}$—the reflexive dot. Therefore, global elements of a system are fully reflexive states. So, while $\mathcal{A} \Rightarrow \mathcal{B}$ may contain "unwelcome" non-morphisms, they are, in some sense, "not really there", because we can't pick them out from inside the category.

It is worth examining the particular construction of $\mathcal{A} \Rightarrow^\bullet \mathcal{B}$ from $\mathcal{A} \Rightarrow \mathcal{B}$. We arrive at the reflexive exponential by deleting all the non-reflexive states. We can do this for *any* non-reflexive system, and obtain a reflexive sub-system comprising of the reflexive states. On the other hand, we can also make a system reflexive by *adding* reflexive transitions to all states. These two constructions are in fact functorial.

Let $\mathsf{I} : \mathbf{TS}^\bullet{}_\Sigma \hookrightarrow \mathbf{TS}_\Sigma$ be the inclusion functor.

Definition 3.26. We will define two functors in the opposite direction.

- Let $\mathsf{F} : \mathbf{TS}_\Sigma \to \mathbf{TS}^\bullet{}_\Sigma$ be the functor that deletes non-reflexive states.
- Let $\mathsf{G} : \mathbf{TS}_\Sigma \to \mathbf{TS}^\bullet{}_\Sigma$ be the functor that adds all reflexive transitions.

We should examine what these functors do to morphisms. Let $f : \mathcal{A} \to \mathcal{B}$. Then, F sends f to its restriction on $\mathsf{F}\mathcal{A}$. This has the right type (that is, f sends things in $\mathsf{F}A$ to $\mathsf{F}B$), for if a is reflexive $f(a)$ is also. The functor G doesn't touch f, and we only need to check that f preserves the new transitions in $\mathsf{G}\mathcal{A}$. But these are all of the form $a \xrightarrow{\sigma} a$, and we know that $f(a) \xrightarrow{\sigma} f(a)$ always, in $\mathsf{G}\mathcal{B}$.

We may think of $\mathsf{F}A$ as being the largest reflexive subsystem of A, and of $\mathsf{G}A$ as the smallest reflexive supersystem (with the natural inclusions). This can be formally expressed by the following proposition.

Proposition 3.27. *Every reflexive subsystem $f : \mathcal{C} \to \mathcal{A}$ of $\mathcal{A}$ factors through the inclusion $\mathsf{F}\mathcal{A} \hookrightarrow \mathcal{A}$.*

Proof. Let $f : \mathcal{C} \to \mathcal{A}$ be monic, where $\mathcal{C}$ is reflexive. This means every state in C must map to a reflexive state of $\mathcal{A}$—that is, a state of $\mathsf{F}\mathcal{A}$. The result follows. $\square$

The following result shows that the constructions F and G are dual, in a sense.

Proposition 3.28. *We have adjunctions*

$$I \dashv F \text{ and } G \dashv I$$

Proof. To show that F is right-adjoint to the inclusion we need to exhibit a bijection

$$\varphi_{\mathcal{A},\mathcal{B}} : \mathbf{TS}_\Sigma(\mathsf{I}\mathcal{A}, \mathcal{B}) \cong \mathbf{TS}^\bullet{}_\Sigma(\mathcal{A}, \mathsf{F}\mathcal{B})$$

that is natural in A and B. We shall not write this out fully, it is a bit tedious. The important fact is that every morphism $\mathsf{I}\mathcal{A} \to \mathcal{B}$ is actually a morphism $\mathcal{A} \to \mathsf{F}\mathcal{B}$ too (and vice versa). So all φ is doing is relabelling the domain/codomain. Naturality follows, because $\varphi(f)$ does precisely the same thing as f (and equality of morphisms in $\mathbf{TS}_\Sigma$ is extensional), so we may pre- and post-compose willy nilly.

The bijection we want can be rewritten as $\mathbf{TS}_\Sigma(\mathcal{A}, \mathcal{B}) \cong \mathbf{TS}_\Sigma(\mathcal{A}, \mathsf{F}\mathcal{B})$ (for $\mathbf{TS}^\bullet{}_\Sigma$ is full). Certainly every morphism from $\mathcal{A} \to \mathsf{F}\mathcal{B}$ extends to a morphism to $\mathcal{B}$. And on the other hand, every morphism $f : \mathcal{A} \to \mathcal{B}$ is actually a morphism $\mathcal{A} \to \mathsf{F}\mathcal{B}$, for it must send the states of the reflexive system $\mathcal{A}$ to states in $\mathsf{F}\mathcal{B}$.

Now we need to show that

$$\mathbf{TS}^\bullet{}_\Sigma(\mathsf{G}\mathcal{B}, \mathcal{A}) \cong \mathbf{TS}_\Sigma(\mathcal{B}, \mathsf{I}\mathcal{A}), \text{ or rather}$$
$$\mathbf{TS}_\Sigma(\mathsf{G}\mathcal{B}, \mathcal{A}) \cong \mathbf{TS}_\Sigma(\mathcal{B}, \mathcal{A})$$

where $\mathcal{A}$ is reflexive. Certainly every morphism $f : \mathsf{G}\mathcal{B} \to \mathcal{A}$ is also a morphism $\mathcal{B} \to \mathcal{A}$. Conversely, if $f : \mathcal{B} \to \mathcal{A}$ then we may extend f to $\mathsf{G}\mathcal{B}$ with no problems, because the new transitions in $\mathsf{G}\mathcal{B}$ are the reflexive $b \xrightarrow{\sigma} b$, and we know for a fact that $f(b) \xrightarrow{\sigma} f(b)$, for $\mathcal{A}$ is reflexive. □

Now we will look at some properties of these functors.

Proposition 3.29. *Both F and G preserve products.*

Furthermore, $F(\mathcal{A} \Rightarrow \mathcal{B}) = \mathcal{A} \Rightarrow^\bullet \mathcal{B}$.

So, if $\mathcal{A}, \mathcal{B}$ are in $\mathbf{TS}^\bullet{}_\Sigma$, we can form the exponential (in $\mathbf{TS}^\bullet{}_\Sigma$) by embedding them into $\mathbf{TS}_\Sigma$, forming $\mathcal{A} \Rightarrow \mathcal{B}$ there, and going back to $\mathbf{TS}^\bullet{}_\Sigma$ through F. It is unclear what sort of interpretation $\mathsf{G}(\mathcal{A} \Rightarrow \mathcal{B})$ might have.

Note that both of these functors are *retractions onto* $\mathbf{TS}^{\bullet}{}_{\Sigma}$, that is,

$$\mathsf{FI} = \mathsf{GI} = 1_{\mathbf{TS}^{\bullet}{}_{\Sigma}}. \tag{3.2}$$

Proposition 3.30. *We have natural transformations*

$$\begin{array}{lll} 1 \implies IG & \text{and} & 1 \implies GI, \\ IF \implies 1 & \text{and} & FI \implies 1 \end{array}$$

Proof. The transformations in the right hand column are equalities, on account of Eq. 3.2. The maps on the left are the canonical embeddings of $\mathcal{A}$ into $\mathsf{G}\mathcal{A}$, and of $\mathsf{F}\mathcal{A}$ into $\mathcal{A}$. □

A similar result holds for the symmetric subcategory $\mathbf{TS}^{\sim}{}_{\Sigma}$.

Definition 3.31. We will say that a transition $a \xrightarrow{\sigma} b$ in a system $\mathcal{A}$ is *opposed* if we also have $b \xrightarrow{\sigma} a$ (and *unopposed* otherwise).

Therefore a system $\mathcal{A}$ is symmetric if every transition is opposed.

Note that morphisms in $\mathbf{TS}_{\Sigma}$ preserve opposed transitions. It is a fact that the category of undirected graphs **Gph** is isomorphic to $\mathbf{TS}^{\sim}{}_{\Sigma}$. Therefore, we should be able to interpret everything in [13] as happening in $\mathbf{TS}^{\sim}{}_{\Sigma} \hookrightarrow \mathbf{TS}_{\Sigma}$.

Proposition 3.32. *The category* $\boldsymbol{TS}^{\sim}{}_{\Sigma}$ *is closed under taking product and exponentiation.*

In fact, $\mathcal{A} \Rightarrow \mathcal{B}$ is symmetric even when only $\mathcal{A}$ is.

Definition 3.33. We have two "symmetricisation" functors $\mathbf{TS}_{\Sigma} \to \mathbf{TS}^{\sim}{}_{\Sigma}$.

- Let U be the functor that deletes unopposed transitions.
- Let V be the functor that adds all missing opposing transitions.

Again, we may think of $\mathsf{U}\mathcal{A}$ as being the largest symmetric subsystem of $\mathcal{A}$, and $\mathsf{V}\mathcal{A}$ as the smallest symmetric supersystem.

Let J be the inclusion $\mathbf{TS}^{\sim}{}_{\Sigma} \hookrightarrow \mathbf{TS}_{\Sigma}$.

Note that U deletes *transitions*, whereas F deleted *states*.

Proposition 3.34. *We have the same sort of adjunctions as in Proposition 3.28*

$$J \dashv U \text{ and } V \dashv J$$

Proof. The idea, as before, is to show that morphisms $\mathcal{A} \to \mathsf{U}\mathcal{B}$ are the same as morphisms $\mathcal{A} \to \mathcal{B}$, where $\mathcal{A}$ is symmetric. But this is true, for if $f : \mathcal{A} \to \mathcal{B}$ and $a \xrightarrow{\sigma} b$ in $\mathcal{A}$ (and this transition is opposed), we will have that $f(a) \xrightarrow{\sigma} f(b)$ is an opposed transition in $\mathcal{B}$, and hence it is in $\mathsf{U}\mathcal{B}$ too. So f restricts to a morphism $f : \mathcal{A} \to \mathsf{U}\mathcal{B}$. The other direction is trivial.

For the second adjunction, we want to show that morphisms $\mathsf{G}\mathcal{B} \to \mathcal{A}$ are the same as those $\mathcal{B} \to \mathcal{A}$, where $\mathcal{A}$ is symmetric. Let $f : \mathcal{B} \to \mathcal{A}$. If we have $a \xrightarrow{\sigma} b$ in $\mathcal{B}$ then we know that $f(a) \xrightarrow{\sigma} f(b)$ in $\mathcal{A}$, and this latter transition is opposed. So adding the opposing $b \xrightarrow{\sigma} a$ in $\mathcal{B}$ doesn't break anything. Again, the other direction is simple—every morphism from $\mathsf{G}\mathcal{B}$ restricts to one from $\mathcal{B}$. $\square$

3.3 Homotopic interpretation

In this section, we will investigate the nature of *paths* in the exponential object. We will extend some notions of $\times$*-homotopy for undirected graphs* [17, 18, 13] to the directed, labeled case of $\mathbf{TS}_\Sigma$. At this point it is still unclear how much of this work can be extended to the case of $\mathbf{TS}_\Sigma$. There is significant opportunity for future work in this area. We culminate in the result (generalising [13]) that (under appropriate conditions) every transition $f \xrightarrow{\sigma} g$ of morphisms in the exponential object $\mathcal{A} \Rightarrow \mathcal{B}$ can be decomposed into a sequence of "primitive" homotopies known as *spider moves*.

We will begin with a brief introduction to the $\times$-homotopy theory of undirected graphs, largely following the presentation of [13].

Definition 3.35. A *path* in an undirected graph $\mathcal{G}$ is a graph morphism $\gamma : \mathbf{I}_n \to \mathcal{G}$, for some $n \in \mathbb{N}$. The notation $\mathbf{I}_n$ denotes the *n-interval*, the graph with $n+1$ nodes $\{0, 1, 2, \cdots, n\}$, and edges $(i, i+1)$ for all i.

$$\mathbf{I}_n = \quad 0 \text{ —— } 1 \text{ —— } \cdots \text{ —— } n$$

Similarly, we will let $\mathbf{I}^\bullet_n$ denote the *reflexive* interval. A morphism from $\mathbf{I}\bullet_n$ is called a *reflexive path.*

$$\mathbf{I}^\bullet_n = \quad \overset{\frown}{0} \text{ —— } \overset{\frown}{1} \text{ —— } \cdots \text{ —— } \overset{\frown}{n}$$

Concatenation of paths is defined in the natural way. If $\gamma : \mathbf{I}_n \to \mathcal{G}$ is a path of length n, and $\delta : \mathbf{I}_m \to \mathcal{G}$ is a path of length m, and furthermore $\gamma(n) = \delta(0)$,

then the concatenation is denoted $\gamma * \delta : \mathbf{I}_{n+m} \to \mathcal{G}$. Concatenation is associative, and the unit is the *empty path* $\varepsilon : \mathbf{I}_0 \to \mathcal{G}$. We can concatenate reflexive paths in exactly the same way.

When we move to the more general case of $\mathbf{TS}_\Sigma$, we do not have a single interval of length n. Rather, for every word $w \in \Sigma^\star$, we have the interval over the word w.

Definition 3.36. Let $w = w_1 w_2 \cdots w_n \in \Sigma^\star$. The *interval over* w is the system with $n+1$ nodes $\{0, 1, 2, \cdots, n\}$, and transitions $i \xrightarrow{w_{i+1}} i+1$ for all i. A *path over (or along)* w is a morphism $\gamma : \mathbf{I}_w \to \mathcal{A}$.

$$\mathbf{I}_w = \quad 0 \xrightarrow{w_1} 1 \xrightarrow{w_2} \cdots \xrightarrow{w_n} n$$

Similarly, we will let $\mathbf{I}^\bullet_w$ denote the *reflexive* interval over w. A morphism from $\mathbf{I}\bullet_w$ is called a *reflexive path (along* w*).*

$$\mathbf{I}^\bullet_w = \quad \overset{\Sigma}{\circlearrowleft} 0 \xrightarrow{w_1} \overset{\Sigma}{\circlearrowleft} 1 \xrightarrow{w_2} \cdots \xrightarrow{w_n} \overset{\Sigma}{\circlearrowleft} n$$

Similarly to the graph case, we may concatenate compatible paths. If $\gamma : \mathbf{I}_w \to \mathcal{A}$ and $\delta : \mathbf{I}_v \to \mathcal{A}$ (and the last state of γ matches the first state of δ), then the concatenated path is defined over the word wv, and denoted $\gamma * \delta : \mathbf{I}_{wv} \to \mathcal{A}$.

There have been several notions of "graph homotopy", that extend the topological notion of homotopy equivalence of functions to the case of morphisms of undirected graphs. Classically, a homotopy is defined the following way.

Definition 3.37. Let X, Y be topological spaces and $f, g : X \to Y$ two continuous maps. A *homotopy (between* f *and* g*)* is a continuous map $h : X \times [0,1] \to Y$ such that $h(-,0) = f$ and $h(-,1) = g$. When such a map h exists we write $h : f \simeq g$ or just $f \simeq g$, and say that f and g are *homotopic*.

This definition can generalised to the category of undirected graphs. Here, it is the systems $\mathbf{I}^\bullet_n$ that play the role of the interval.

Definition 3.38 ([13])**.** Let $\mathcal{G}, \mathcal{H}$ be undirected graphs, and $f, g : \mathcal{G} \to \mathcal{H}$ graph morphisms.

A *homotopy* between f and g is a graph morphism $h : \mathcal{G} \times \mathbf{I}^\bullet_n \to \mathcal{H}$ for some n, such that $h(-,0) = f$ and $h(-,n) = g$.

Because the category **Gph** is cartesian closed, we can interpret a morphism $\mathcal{G} \times \mathbf{I}_n^\bullet \to \mathcal{H}$ as a morphism $\mathbf{I}_n^\bullet \to \mathcal{G} \Rightarrow \mathcal{H}$. That is, homotopies correspond to reflexive paths in the exponential object. As in the topological case, the homotopy relation $\simeq$ is an equivalence relation on morphisms $\mathcal{G} \to \mathcal{H}$.

Note that the reflexive interval is chosen over the irreflexive. This is so that the the graph $\mathcal{G} \times \mathbf{I}_n^\bullet$ consists of n copies of $\mathcal{G}$. In the product $\mathcal{G} \times \mathbf{I}_n^\bullet$ we have $(a, i) \sim (a', i)$ for all i whenever there is an edge $a \sim a'$ in $\mathcal{G}$ (because $i \sim i$).

In the directed, labelled world, things become slightly more complicated. We may define a notion of homotopy using the intervals $\mathbf{I}_w^\bullet$.

Definition 3.39. Let $\mathcal{A}, \mathcal{B}$ be two transition systems, and $f, g : \mathcal{A} \to \mathcal{B}$ morphisms. A *homotopy* (over the word $w = w_1 w_2 \cdots w_n$) from f to g is a morphism $h : \mathcal{A} \times \mathbf{I}_w^\bullet \to \mathcal{B}$ with $h(-, 0) = f$ and $h(-, n) = g$.

We say $f \preceq_w g$ if there exists a homotopy from f to g along w.

Now, homotopy is still reflexive (along the empty word), and transitive (homotopies may be concatenated). But it is no longer symmetric. The existence of a homotopy $f \preceq g$ does not imply a homotopy $g \preceq f$.

Example 3.40. Let $\mathcal{A} = (0 \to 1)$ and $\mathcal{B} = (\circlearrowleft 0 \to \circlearrowleft 1)$. Let $f, g : \mathcal{A} \to \mathcal{B}$ be the constant maps to 0 and 1 respectively. There is a homotopy $f \preceq g$, for $f(0) \to g(1)$. However, we do not have $g \preceq f$ because $g(0) \not\to f(1)$.

Just like in **Gph**, when working in $\mathbf{TS}_\Sigma$ we need to take homotopies over the *reflexive* intervals. If $w = w_1 w_2 \cdots w_n$, then we can think of the product $\mathcal{A} \times \mathbf{I}_w^\bullet$ as being formed of $n + 1$ copies of $\mathcal{A}$, layered over each other. The transitions within each layer are exactly as in $\mathcal{A}$. There are also "diagonal" transitions going from every layer to the next. These are formed according to the following rule. Whenever there is a transition $a \xrightarrow{w_i} b$ in the ith layer, then there is a transition $(a, i) \xrightarrow{w_i} (b, i + 1)$ going from layer i to $i + 1$. This is depicted in Fig. 3.9.

The final result of this section will be following [13]. We will and define an "elementary" class of homotopies, that in some sense "generate" $A \Rightarrow B$. In general, f and g and so on will refer to *set-maps*, unless they are specified to be morphisms.

Definition 3.41. Let f, g be set-maps $\mathcal{A} \to \mathcal{B}$. We say that f *spiders to* g (along σ) if they agree except for perhaps at one state x. In which case, if $x \xrightarrow{\sigma} x$ then

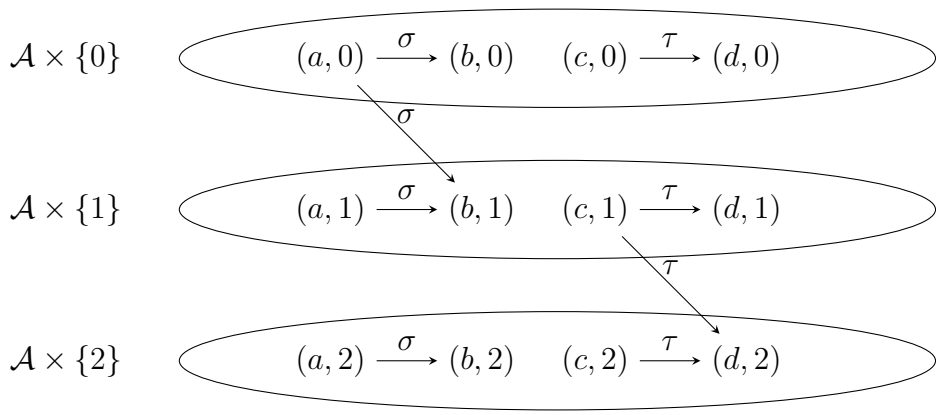


Figure 3.9: The product system $\mathcal{A} \times \mathbf{I}_{\sigma\tau}$

we will require that

$$f(x) \xrightarrow{\sigma} g(x).$$

If f spiders to g along σ we will write $f \overset{\sigma}{\rightsquigarrow} g$.

What is the intuition behind this definition? It simply says that f and g almost agree, and f homotopes to g. That is, assuming that f and g are functions that agree except for at one state, then we have $f \overset{\sigma}{\rightsquigarrow} g$ if and only if $f \xrightarrow{\sigma} g$.

It is therefore clear that

Proposition 3.42. *If $f \overset{\sigma}{\rightsquigarrow} g$ then $f \xrightarrow{\sigma} g$ (in $\mathcal{A} \Rightarrow \mathcal{B}$).*

We have a weak converse to this.

Proposition 3.43. *Let A be finite, and $f \xrightarrow{\sigma} g$ in $\mathcal{A} \Rightarrow \mathcal{B}$. Then there exists a sequence of set-maps $f_0, f_1, \cdots f_n$ such that*

$$f = f_0 \overset{\sigma}{\rightsquigarrow} f_1 \overset{\sigma}{\rightsquigarrow} \cdots \overset{\sigma}{\rightsquigarrow} f_{n-1} \overset{\sigma}{\rightsquigarrow} f_n = g$$

Proof. Enumerate the states of A as $x_0, \cdots, x_n$. Take f_i to be the set-map that acts as g on $x_0, \cdots x_{i-1}$ and f on $x_i, \cdots x_n$.

We need to check that $f_i \overset{\sigma}{\rightsquigarrow} f_{i+1}$. These maps disagree precisely at x_i, so if x_i does not transition to itself we are done. If we do have $x_i \xrightarrow{\sigma} x_i$, then we need $f_i(x_i) \xrightarrow{\sigma} f_{i+1}(x_i)$. But on the left we have $f(x_i)$ and on the right $g(x_i)$, and we know that $f(x_i) \xrightarrow{\sigma} g(x_i)$, for we have $f \xrightarrow{\sigma} g$.

So the result holds.

□

The following stronger result for **Gph** is due to Chih et al [13].

Proposition 3.44. *Let $\mathcal{A}$ be finite and symmetric, and f, g morphisms $\mathcal{A} \to \mathcal{B}$ with $f \xrightarrow{\sigma} g$. Then there is a sequence of* morphisms

$$f = f_0 \overset{\sigma}{\rightsquigarrow} f_1 \overset{\sigma}{\rightsquigarrow} \cdots \overset{\sigma}{\rightsquigarrow} f_{n-1} \overset{\sigma}{\rightsquigarrow} f_n = g$$

Proof. The proof is the same as that of Proposition 3.43. We just need to check that each f_i is a morphism.

So, pick an f_i and let $x_j \xrightarrow{\sigma} x_k$. We want $f_i(x_j) \xrightarrow{\sigma} f_i(x_k)$. There are several cases of i, j, k.

- j, k are on the same side of i. So f_i acts as either f or g on both x_j and x_k. So the required transition exists.
- $k < i \leq j$. Then we need $f(x_j) \xrightarrow{\sigma} g(x_k)$, but this happens because $f \xrightarrow{\sigma} g$.
- $j < i \leq k$. Here we need $g(x_j) \xrightarrow{\sigma} f(x_k)$. This is not a problem, because $\mathcal{A}$ is symmetric, so $\mathcal{A} \Rightarrow \mathcal{B}$ is too, and hence $g \xrightarrow{\sigma} f$.

□

If we have morphisms $f, g : \mathcal{A} \to \mathcal{B}$ where $\mathcal{A}$ is not symmetric, the f_i defined in the proof of Proposition 3.43 need not be morphisms, because we require symmetry in the third case above. However, the f_i are defined with reference to a specific enumeration of the states of A. It is possbile there might be a non-symmetric system $\mathcal{A}$, whereby one ordering of the states induces a sequence of non-morphisms, but a cleverly chosen ordering does give you a sequence of morphisms. This remains to be examined in the future.

We give a sufficient condition for when Proposition 3.44 holds in the general, non-symmetric case of $\mathbf{TS}_\Sigma$

Definition 3.45. Let $(\mathsf{N}, \geq)$ be the transition system with states in N, and a transition $n \xrightarrow{\sigma} m$ (for all σ) if and only if $n \geq m$.

Let $(N, \geq)$ be the full subsystem of $(\mathsf{N}, \geq)$ on the states $0, 1, \ldots N$.

Proposition 3.46. *Suppose $\mathcal{A}$ is finite and there is an embedding (i.e. a monomorphism) $h : \mathcal{A} \to (\mathsf{N}, \geq)$. Then any transition $f \xrightarrow{\sigma} g$ between parallel morphisms $f, g : \mathcal{A} \to \mathcal{B}$ factors into a spider sequence of* morphisms.

Proof. Label the vertices of $\mathcal{A}$ according to their image under h. This gives a total order on A that respects transitions. That is, if $x \xrightarrow{\sigma} y$ then $h(x) \geq h(y)$.

So we may take f_i to act as g on those x with $h(x) < i$, and as f on the x with $h(x) \geq i$. Each f_i will be a morphism, for, following the proof of Proposition 3.44, if $x \xrightarrow{\sigma} y$ then we never fall into the problematic $h(x) < i \leq h(y)$ case. □

3.4 Simulations and the category TSS

In this section, we will introduce the "simulation category" $\mathbf{TSS}_\Sigma$, and investigate some elementary properties. In particular, we will be thinking of this category as a *relational* verson of $\mathbf{TS}_\Sigma$, playing a role analogous to **Rel** and **Set**.

Definition 3.47. The category $\mathbf{TSS}_\Sigma$ has the same objects as $\mathbf{TS}_\Sigma$: transition systems over the alphabet Σ.

The morphisms $R : \mathcal{A} \to \mathcal{B}$ are the *simulations*.

The first line of agreement with **Rel** is the following result.

Proposition 3.48. *The product and coproduct are both given by the disjoint union.*

Proof. Let $\mathcal{A}, \mathcal{B}$ be transition systems. We are familiar with $\mathcal{A} \sqcup \mathcal{B}$. We have the obvious injections $i_A : \mathcal{A} \to \mathcal{A} \sqcup \mathcal{B}$ and projections $\pi_A : \mathcal{A} \sqcup \mathcal{B} \to \mathcal{A}$ (and likewise for $\mathcal{B}$). It is not hard to verify that these are indeed simulations.

Suppose we have maps $\mathcal{A} \xleftarrow{R} \mathcal{C} \xrightarrow{S} \mathcal{B}$. The pairing $\langle R, S\rangle : \mathcal{C} \to \mathcal{A} \sqcup \mathcal{B}$ is given by

$$c\langle R, S\rangle a \iff cRa \text{ for } a \in A$$
$$c\langle R, S\rangle b \iff cSb \text{ for } b \in B$$

This is a simulation, because if $c\langle R, S\rangle x$ and $c \xrightarrow{\sigma} c'$ then we may assume without loss of generality that $x = a \in A$, and hence cRa. Because R is a simulation, we can recreate the transition $c \xrightarrow{\sigma} c'$ in A. That is, there exists an a' with $a \xrightarrow{\sigma} a'$ and $c'Ra'$. But then $c'\langle R, S\rangle a'$. So we see that $\langle R, S\rangle$ is a simulation.

We also have that $\langle R, S\rangle \pi_A = R$. To see this, note that $c\langle R, S\rangle \pi_A a$ if and only if there exists an $x \in A \sqcup B$ with $c\langle R, S\rangle x$ and $x\pi_A a$. The second condition holds

if and only if $x = a$, so we deduce that

$$\begin{aligned} c\langle R, S\rangle \pi_A a &\iff c\langle R, S\rangle a \\ &\iff cRa \end{aligned}$$

It is not hard to see that $\langle R, S\rangle$ is the unique morphism that makes the product diagram commute. So the product of A, B is the disjoint union. We now look at the coproduct.

If we have maps $\mathcal{A} \xrightarrow{R} \mathcal{C} \xleftarrow{S} \mathcal{B}$ then we define the copairing $[R, S] : \mathcal{A} \sqcup \mathcal{B} \to \mathcal{C}$ by

$$\begin{aligned} a[R, S]c &\iff aRc \\ b[R, S]c &\iff bSc \end{aligned}$$

To check that this is a simulation, suppose that $a[R, S]c$ (that is, aRc) and $a \xrightarrow{\sigma} a'$. Because R is a simulation, we have a c' with $c \xrightarrow{\sigma} c'$ and $a'Rc'$—but this implies that $a'[R, S]c'$.

The argument that that $[R, S]$ is the unique morphism to make the coproduct diagram commute is very similar to the product case. □

Of course, we can still form the the tensor product of transition systems $\mathcal{A} \times \mathcal{B}$.

Proposition 3.49. *The tensor product is symmetric monoidal.*

Proof. Consider the map $\otimes : \mathbf{TSS}_\Sigma \times \mathbf{TSS}_\Sigma \to \mathbf{TSS}_\Sigma$ that sends a pair of systems $\mathcal{A}, \mathcal{B}$ to their tensor product $\mathcal{A} \otimes \mathcal{B}$ (this is the same as in $\mathbf{TS}_\Sigma$, but we will use $\otimes$ to avoid confusion with the categorical product, i.e. the disjoint union). We will first show that this extends to a functor. The proof is routine.

Let $R : \mathcal{A} \to \mathcal{X}$ and $S : \mathcal{B} \to \mathcal{Y}$ be simulations. The product relation is given by

$$(a, b)(R \otimes S)(x, y) \text{ if and only if } aRx \text{ and } bSy.$$

So suppose that $(a, b) \xrightarrow{\sigma} (a', b')$ (that is, we have transitions on both components), and $(a, b)(R \otimes S)(x, y)$. Since R and S are simulations we can find simulate $a \xrightarrow{\sigma} a'$ and $b \xrightarrow{\sigma} b'$ at x and y respectively, with x' and y'. Then of course, $(x, y) \xrightarrow{\sigma} (x', y')$ and $(a', b')(R \otimes S)(x'y')$.

The unit of the product is the reflexive dot.

The associator, swap, and unitors are precisely the same as in **Rel**. We simply need to verify that they are actually simulations. The required diagrams and naturalities will come for free.

Consider $S : \mathcal{A} \otimes \mathcal{B} \to \mathcal{B} \otimes \mathcal{A}$ defined by $(a, b)S(b, a)$ for all a, b. If we have $(a, b) \xrightarrow{\lceil} \sigma](a', b')$ and $(a, b)S(x, y)$ then it is clear that $(x, y) = (b, a)$, and we can complete the square with (b', a'). We should also verify that the inverse (in **Rel**) is a simulation. But the inverse to S is simply the reverse of S. This tells us that S is actually a bisimulation. It is similarly easy to check that the other relations are also simulations. $\square$

We know that **Rel** is actually symmetric monoidal *closed* under $\otimes$: the internal hom is also $\otimes$. This does not hold in $\mathbf{TSS}_\Sigma$. Consider the following counterexample.

Let $\mathcal{A} = 0 \to 1$, and $\mathcal{B} = \mathcal{C} = 0 \quad 1$ (all states are irreflexive). The products $A \otimes \mathcal{B}$ and $\mathcal{B} \otimes \mathcal{C}$ have no transitions. We see that the homset $A \to B \otimes C$ has exactly five members: the empty simulation, and for each of the four states x of $B \otimes C$ we have the simulation that matches 1 to x. On the other hand there are 2^8 simulations $A \otimes B \to C$: the source has no transitions, so every relation on the underlying sets is a simulation.

So the homsets $\mathbf{TSS}_\Sigma(\mathcal{A} \otimes \mathcal{B}, \mathcal{C})$ and $\mathbf{TSS}_\Sigma(\mathcal{A}, \mathcal{B} \otimes \mathcal{C})$ are not isomorphic.

A further point of disagreement is the fact that **Rel** is self-dual, whereas the opposite of a simulation will not in general be a simulation. In fact, R^{op} is a simulation precisely if R is a *bisimulation*. This suggests that the subcategory of transition systems and bisimulations might be a better analog for **Rel**.

3.5 Simulations via monads

In the previous section we defined the simulation category $\mathbf{TSS}_\Sigma$ from first principles. We shall now see how to construct it starting from the more primitive category $\mathbf{TS}_\Sigma$. The idea is this: we know that the objects of $\mathbf{TS}_\Sigma$ are sets with extra structure (a transition relation), and morphisms are functions that *preserve* that structure. Similarly, the objects of $\mathbf{TSS}_\Sigma$ are sets with structure and the morphisms are *relations* that preserve structure.

Thus the category $\mathbf{TS}_\Sigma$ is analogous to **Set**, while $\mathbf{TSS}_\Sigma$ is like **Rel**. One delightful connection between **Set** and **Rel** is that **Rel** is the Kleisli category of the (covariant) powerset monad $\mathcal{P}$ on **Set**. The goal of this section is to construct

a monad Sim on $\mathbf{TS}_\Sigma$ that will play the role of $\mathcal{P}$—the Kleisli category of Sim will be $\mathbf{TSS}_\Sigma$.

In fact, the monad Sim will have to be *very* similar to $\mathcal{P}$. If a Kleisli morphism $(A, R) \to \mathrm{Sim}(B, S)$ is to be equivalent to a simulation $(A, R) \nrightarrow (B, S)$ we can deduce (by passing through the forgetful functors) that the underlying set of $\mathrm{Sim}(B, S)$ must be $\mathcal{P}(B)$, and therefore the action of Sim on morphisms must also be that of $\mathcal{P}$.

But there is one piece of the puzzle remaining. A functor must have an action on objects, and an action on morphisms. In this case, the objects are sets with *structure*. Therefore the functor Sim shall have to transport this structure also.

$$\text{Structure on } A \longrightarrow \text{Structure on } \mathcal{P}(A)$$

In other words, the functor Sim is built by combining the powerset functor $\mathcal{P}$ with a "magic ingredient": a suitably well behaved mapping of transition relations on A to transition relations on $\mathcal{P}(A)$.

$$\mathrm{Sim} = \underbrace{\mathcal{P}}_{\text{sets and functions}} + \underbrace{\text{“magic ingredient”}}_{\text{structure}}$$

We examine the details and general theory of these sorts of *structural mappings* in Chapter 5 For now, we will concentrate on the concrete example of Sim. In this case, the type of mapping we are looking for is a function (that we will call $\overrightarrow{\mathcal{P}}$) from transition relations on A to transition relations on $\mathcal{P}(A)$. By "well behaved", we mean two things.

1. The Sim construction must not interfere with morphisms. That is, if $f : (A, R) \to (B, S)$ is a transition preserving function, then $\mathcal{P}f : (\mathcal{P}A, \overrightarrow{\mathcal{P}}(R)) \to (\mathcal{P}B, \overrightarrow{\mathcal{P}}(S))$ must also preserve the transitions of $\mathrm{Sim}\, A = (\mathcal{P}A, \overrightarrow{\mathcal{P}}(R))$.

2. The Kleisli morphisms must be simulations. A transition preserving function $A \to \mathrm{Sim}\, B$ must encode a simulation $A \nrightarrow B$, and vice versa.

To avoid notational confusion, we shall try to use Sim to refer only to the entire functor on $\mathbf{TS}_\Sigma$. The component actions on sets of states, morphisms, or transition relations will be denoted by $\mathcal{P}$, $\mathcal{P}$, and $\overrightarrow{\mathcal{P}}$ respectively. Hence we shall

have

$$\mathrm{Sim} = \mathcal{P} + \overrightarrow{\mathcal{P}}.$$

Definition 3.50 (Sim)**.** Let (A, R) be a transition system. The associated *simulation space* is a transition system where

- the set of states is $\mathcal{P}(A)$, and
- the transition transition relation $\overrightarrow{\mathcal{P}}(R) : \mathcal{P}(A) \nrightarrow \mathcal{P}(A)$ is defined labelwise by $U \xrightarrow{\sigma} V$ if and only if the following condition holds:

$$\text{for all } u \in U \text{ there exists a } v \in V \text{ such that } u \xrightarrow{\sigma} v. \qquad \text{(Sim rule)}$$

We can extend this construction to act on morphisms also. Let $f : A \to B$ be a morphism. We will take $\mathrm{Sim}\, f : \mathrm{Sim}(A, R) \to \mathrm{Sim}(B, S)$ to be the direct image $\mathcal{P}(f) : \mathcal{P}(A) \to \mathcal{P}(B)$. Therefore we have defined a functor Sim on $\mathbf{TS}_\Sigma$.

It is of course necessary to verify that $\mathcal{P}(f)$ is indeed a morphism (it preserves transitions). To that end, let $f : (A, R) \to (B, S)$ be a morphism of transition systems. Suppose we have $U \xrightarrow{\sigma} V$. We need to show that $\mathcal{P}f(U) \xrightarrow{\sigma} \mathcal{P}f(V)$. Let $x = f(u) \in \mathcal{P}f(U)$. By assumption, there is a $v \in V$ with $u \xrightarrow{\sigma} v$. Therefore $f(u) \xrightarrow{\sigma} f(v)$ (because f is a morphism), and certainly $f(v)$ is in $\mathcal{P}f(V)$.

Note that this is all we need to show to deduce that Sim is a functor. The other functor laws—$\mathrm{Sim}(\mathrm{id}) = \mathrm{id}$ and $\mathrm{Sim}(g \circ f) = \mathrm{Sim}(g) \circ \mathrm{Sim}(f)$—follow for free, by Remark 3.9. The action of Sim on the underlying sets and functions is precisely that of $\mathcal{P}$ on **Set** (in other words, the diagram in Fig. 3.10 commutes), which we know is a functor.

$$\begin{tikzcd}
\mathbf{TS}_\Sigma \arrow[r, "U_\Sigma"] \arrow[d, "\mathrm{Sim}"'] & \mathbf{Set} \arrow[d, "\mathcal{P}"] \\
\mathbf{TS}_\Sigma \arrow[r, "U_\Sigma"] & \mathbf{Set}
\end{tikzcd}$$

Figure 3.10

The intuitive interpretation of the lifted transition relation $\mathrm{Sim}(R) : \mathcal{P}A \nrightarrow \mathcal{P}A$ is that a subset U can transition to a subset V if V is "reachable" from U—no

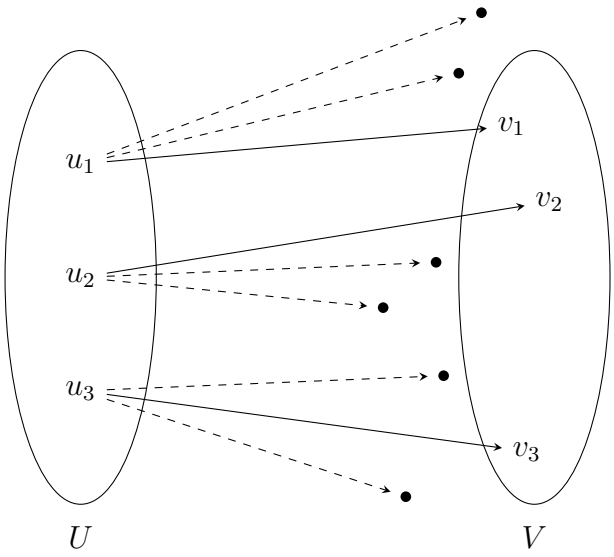


Figure 3.11: An instance of $U \to V$

matter which point in U you start at, you are able to make a transition that leads you to somewhere in V. This is depicted in Fig. 3.11.

Observing that the functorial properties of $\mathcal{P}$ can be lifted to Sim "for free", one might wonder whether we can do the same for any other properties of $\mathcal{P}$. In particular, we would like to give Sim the structure of a monad. There is really only one good choice.

Proposition 3.51. *The triple* $(\mathrm{Sim}, \eta^{\mathcal{P}}, \mu^{\mathcal{P}})$ *is a monad on* $\boldsymbol{TS}_{\Sigma}$.

Proof. We will first show that $\eta^{\mathcal{P}}$ and $\mu^{\mathcal{P}}$ are morphisms of transition systems. Let (A, R) be a transition system.

- $(\eta^{\mathcal{P}})$. Suppose that $a \xrightarrow{\sigma} a'$ in A. By definition, this implies that $\{a\} \xrightarrow{\sigma} \{a'\}$ in $\mathrm{Sim}(A)$. Therefore $\eta^{\mathcal{P}}$ preserves transitions.

- $(\mu^{\mathcal{P}})$. Suppose that $F \xrightarrow{\sigma} G$ in $\mathrm{Sim}(\mathrm{Sim}\, A)$. We wish to show that $\mu^{\mathcal{P}} F \xrightarrow{\sigma} \mu^{\mathcal{P}} G$ in $\mathrm{Sim}\, A$. So, let $a \in \mu^{\mathcal{P}} F$, which means $a \in U \in F$. By $F \xrightarrow{\sigma} G$, we can find a $V \in G$ with $U \xrightarrow{\sigma} V$. And, since $a \in U$, we can find an $a' \in V$ with $a \xrightarrow{\sigma} a'$. Now, because $a' \in V \in G$, we may deduce that $a' \in \mu^{\mathcal{P}} G$, and hence $\mu^{\mathcal{P}}$ also is a transition preserving function.

Note that the monad laws and the naturality of $\eta^{\mathcal{P}}$ and $\mu^{\mathcal{P}}$ come for free, by Remark 3.9. $\square$

The following result is a variation of a proposition found in [39]. Malacaria uses a slightly different definition of a simulation, which requires that every state in the source system be related to at least one state in the target system. Hence his version of the Sim monad is built on the non-empty powerset functor $\mathcal{P}^{\geq}$. The proof below provides some additional detail that is not found in [39], and a more abstract setting that will be generalised in Chapter 5.

Theorem 3.52. *The Kleisli category of* Sim *is equivalent to* $\boldsymbol{TSS}_\Sigma$.

Proof. Note that the categories $\mathbf{TS}_\Sigma$ and $\mathbf{TSS}_\Sigma$ have exactly the same objects: it is the morphisms that differ. We will show that for transition systems A and B, the morphisms $A \to \mathrm{Sim}\, B$ are in bijection with the simulations $A \to B$. This is all that will be necessary—we do not need to concern ourselves with showing that the Kleisli composition of morphisms is the same as composition in $\mathbf{TSS}_\Sigma$, or that the unit of the Sim monad is the identity—this all follows from the facts that:

1. $\mathbf{TS}_\Sigma$ is concrete over **Set**,
2. $\mathbf{TSS}_\Sigma$ is concrete over **Rel**, and
3. **Rel** is the Kleisli category of $\mathcal{P}$ on **Set**

And in fact, Item 3 tells us that the functions of underlying sets $A \to \mathcal{P}(B)$ are in bijection with the relations $A \nrightarrow B$. So all we need to verify is that this bijection restricts to edge-preserving functions and simulations.

The construction is as follows. For a function of carrier sets $f : A \to \mathcal{P}(B)$, the corresponding relation $\varphi(f) : A \nrightarrow B$ is given by $\varphi(f) = f_*$, with

$$(a, b) \in \varphi(f) \text{ iff } b \in f(a),$$

and for a relation $Z : A \nrightarrow B$, there is a function $\psi(Z) : A \to \mathcal{P}(B)$ defined as

$$\psi(Z)(a) = \{b \in B : (a, b) \in Z\}.$$

Let $f : A \to B$ be a morphism of transition systems. We require that $\varphi(f)$ is a simulation. So, let $a \xrightarrow{\sigma} a'$ and $(a, b) \in \varphi(f)$. This means that $b \in f(a)$. Since f preserves transitions, we can deduce that $f(a) \xrightarrow{\sigma} f(a')$ in Sim A, so every element of $f(a)$—and in particular, b—can transition to an element of $f(a')$. Hence there exists a b' with $b \xrightarrow{\sigma} b'$ and $(a', b') \in \varphi(f)$. Thus $\varphi(f)$ is a simulation.

On the other hand, let Z be a simulation $A \nrightarrow B$. We will show that $\psi(Z)$ preserves transitions. Suppose that $a \xrightarrow{\sigma} a'$, and let $b \in \psi(Z)(a)$, so $(a, b) \in Z$. By assumption we can find a b' with $b \xrightarrow{\sigma} b'$ in B, and $(a', b') \in Z$. Since we can find such a b' for any b, we can deduce $\psi(Z)(a) \xrightarrow{\sigma} \psi(Z)(z')$, as desired $\square$

We have seen that we can combine the powerset functor with an appropriate choice of "magic ingredient" (an action $\overrightarrow{\mathcal{P}}$ on transition relations) to produce a functor Sim on $\mathbf{TS}_\Sigma$. The mapping $\overrightarrow{\mathcal{P}}$ is sufficiently "nice" that not only do we get a functorial structure "for free", but we get a monad structure (using just the same monad components of $\mathcal{P}$) for free also. The Kleisli morphisms of Sim are relations (because the "base monad" is $\mathcal{P}$) that *preserve structure*, due to the behaviour of $\overrightarrow{\mathcal{P}}$.

As we have seen, there are two different ways to "flip" the notion of a simulation. We can flip *horizontally*, and end up with *cosimulations*, or *vertically* and end up with *reverse simulations*.

We might now ask: **Can we use the same recipe with a different "magic ingredient" to produce monads that encode some of these other types of simulations?**

And indeed, we can. The strategy is as follows:

1. Since a reverse simulation is just a simulation "in the opposite direction", we will define $\overleftarrow{\mathcal{P}}$ to be the opposite of $\overrightarrow{\mathcal{P}}$.

2. A double simulation is a simulation *and* a reverse simulation, we will combine $\overleftarrow{\mathcal{P}}$ and $\overrightarrow{\mathcal{P}}$ to produce DSim.

Just like before, we will see that the functoriality and the monad laws for RevSim and DSim come "for free". All we will need to verify is that the components are actually morphisms in $\mathbf{TS}_\Sigma$.

Definition 3.53 (RevSim, DSim)**.** Let R be a transition relation on a set A. We define the transition relation $\overleftarrow{\mathcal{P}} R$ labelwise on $\mathcal{P}(A)$ by saying $U \xrightarrow{\sigma} V$ if and only if:

$$\text{for all } v \in V \text{ there exists a } u \in U \text{ such that } u \xrightarrow{\sigma} v \qquad \text{(CoSim rule)}$$

Therefore there is a functor RevSim : $\mathbf{TS}_\Sigma \to \mathbf{TS}_\Sigma$ that sends

- transition systems (A, R) to $(\mathcal{P}A, \overleftarrow{\mathcal{P}} R)$, and

- morphisms $f : A \to B$ to $\mathcal{P}f$.

There is also a functor DSim on $\mathbf{TS}_\Sigma$ that sends

- transition systems (A, R) to $(\mathcal{P}A, \overrightarrow{\mathcal{P}}R \cap \overleftarrow{\mathcal{P}}R)$, and
- morphisms f to $\mathcal{P}f$.

As in the case of Sim, we need to verify that these functors preserve morphisms. So let $f : (A, R) \to (B, S)$ be a morphism in $\mathbf{TS}_\Sigma$. We will show that the direct image function $\mathcal{P}f$ is a morphism $\mathrm{RevSim}\, A \to \mathrm{RevSim}\, B$ and also $\mathrm{DSim}\, A \to \mathrm{DSim}\, B$.

The following result will be useful. It encodes the fact that a double simulation is merely a relation that is a simulation and a reverse simulation.

Lemma 3.54. *Suppose A and B are transition functions, and g is a function of underlying sets $A \to \mathcal{P}(B)$. If g is a* morphism *$A \to \mathrm{Sim}\, B$ and $A \to \mathrm{RevSim}\, B$ then it is also a morphism $A \to \mathrm{DSim}\, B$.*

- (RevSim). Let $U \xrightarrow{\sigma} V$ in $\mathrm{RevSim}\, A$. Do we have $\mathcal{P}f(U) \xrightarrow{\sigma} \mathcal{P}f(V)$? Well, let $y \in \mathcal{P}f(V)$. Therefore we can write $y = f(v)$ for some $v \in V$, and by assumption there is a $u \in U$ with $u \xrightarrow{\sigma} v$. Since f preserves transitions, we can take $x = f(u) \in \mathcal{P}f(U)$ and deduce that $f(u) \xrightarrow{\sigma} f(v)$, which is what we needed.
- (DSim). This follows from Lemma 3.54.

We can lift the monad structure of $\mathcal{P}$ to RevSim and DSim, just as we did for the functor Sim.

Proposition 3.55. *The triple $(\mathrm{RevSim}, \eta^{\mathcal{P}}, \mu^{\mathcal{P}})$ is a monad on $\boldsymbol{TS}_\Sigma$, as is $(\mathrm{DSim}, \eta^{\mathcal{P}}, \mu^{\mathcal{P}})$.*

Proof. As in Proposition 3.51, we will show that $\eta^{\mathcal{P}}$ and $\mu^{\mathcal{P}}$ are transition preserving for RevSim. The case of DSim will follow by Lemma 3.54.

Let (A, R) be a transition system.

- $(\eta^{\mathcal{P}})$. Suppose $a \xrightarrow{\sigma} a'$. We need to show that $\eta^{\mathcal{P}}(a) \xrightarrow{\sigma} \eta^{\mathcal{P}}(a')$ in $\mathrm{RevSim}\, A$. This is certainly true, as $\eta^{\mathcal{P}}(a') = \{a'\}$ is a singleton, and $a \in \{a\} = \eta^{\mathcal{P}}(a)$ witnesses $\eta^{\mathcal{P}}(a) \xrightarrow{\sigma} \eta^{\mathcal{P}}(a')$.

- $(\mu^{\mathcal{P}})$. Suppose $F \xrightarrow{\sigma} G$. Ge want $\mu^{\mathcal{P}} F \xrightarrow{\sigma} \mu^{\mathcal{P}} G$. So let $y \in \mu^{\mathcal{P}} G$, which means $y \in V$ for some $v \in G$. By $F \xrightarrow{\sigma} G$ we find a $U \in F$ with $U \xrightarrow{\sigma} V$ (in $\mathsf{RevSim}\, A$), which further implies that there is an $x \in U$ with $x \xrightarrow{\sigma} y$. Therefore $x \in \mu^{\mathcal{P}} F$, as desired.

□

The final step is to verify that Kleisli morphisms into RevSim really are reverse simulations. The proof will be symmetric to the Sim case.

Theorem 3.56. *The Kleisli categories of the* $\boldsymbol{TS}_\Sigma$*-monads* Sim*,* RevSim*, and* DSim *are:*

Sim	$\mathbf{TSS}_\Sigma$
RevSim	Transition systems with reverse simulations
DSim	Transition systems with double simulations

Proof. We shall prove that a Kleisli morphism for the RevSim monad is equivalent to a reverse simulation. The corresponding result for bisimulations will follow.

Let $f : A \to \mathsf{RevSim}\, B$ be a morphism of transition systems. We want to show that $\varphi(f)$ is a reverse simulation. Therefore let $a \xrightarrow{\sigma} a'$ in A, and $(a', b') \in \varphi(f)$. This means that $b' \in f(a')$. Since f is transition preserving we deduce $f(a) \xrightarrow{\sigma} f(a')$, and by the transition condition in $\mathsf{RevSim}\, B$, we can find an appropriate $b \in f(a)$ with $b \xrightarrow{\sigma} b'$. This completes the square.

On the other hand, let $Z : A \nrightarrow B$ be a reverse simulation. We show that $\psi(Z)$ is transition preserving $A \to \mathsf{RevSim}\, B$. Let $a \xrightarrow{\sigma} a'$. We need $\psi(Z)(a) \xrightarrow{\sigma} \psi(Z)(a')$ in $\mathsf{RevSim}\, B$. So let $b' \in \psi(Z)(a')$. This means that $(a', b') \in Z$. As Z is a reverse simulation, we can find an b with $b \in \psi(Z)(a)$ and $b \xrightarrow{\sigma} b'$. This completes the proof.

□

Chapter 4

In search of generality

In the preceding chapter we saw a detailed and comprehensive account of several related categories of "naïve" transition systems. We defined the category $\mathbf{TS}_\Sigma$, and performed some elementary constructions. In particular we exhibited a Cartesian closed structure, and an interpretation of "transitions between morphisms (or functions" as (discrete, directed) *homotopies*.

In the latter half of the chapter we switched focus to various categories of simulations (and reverse simulations and double simulations), and exhibited that the category of transitions systems with simulations $\mathbf{TSS}_\Sigma$ can be expressed as a Kleisli category of the Sim monad on $\mathbf{TS}_\Sigma$. The subsequent chapters of this thesis generalise this construction to different sorts of transition systems. There are, ultimately, two parts to this task.

1. Defining a more general sort of "category of transition systems"(with transition-preserving functions as morphisms) that subsumes the category $\mathbf{TS}_\Sigma$.

2. Constructing Sim-like monads on this category.

Firstly, we shall need to define the categories of transition systems we are interested in. Here we shall follow the "coalgebraic perspective" [2, 50, 37, 45].

A review of this material can be found in Section 4.1. In essence, an unlabelled non-deterministic transition system on a set of states A can be considered as a $\mathcal{P}$-coalgebra, $\alpha : A \to \mathcal{P}A$. By replacing the monad $\mathcal{P}$ with an arbitrary monad T, we can view T-coalgebras as transition systems *of transition type* T.

There is a natural notion of a morphism of coalgebras, and hence for every functor T there is a canonical category of T-coalgebras, $\mathbf{Coalg}(T)$. This turns

out to be too strict for our purposes. In the case of $T = \mathcal{P}$, the $\mathcal{P}$-coalgebra morphisms are in fact functional bisimulations of transition systems, not merely transition-preserving functions [29].

For this reason, we will need to consider instead the category of T-coalgebras with what are essentially the *lax cohomomorphisms* of Corradini et al. [15]. Although the presentation of [15] is in terms of "order-enriched" functors T, we shall instead consider the case of a monad T with a **Pos**-enriched Kleisli category $\mathcal{C}_T$. In Section 4.2 we define and elaborate upon some essential notions of **Pos**-enrichment (fundamentally according to Kelly [30]).

At this point, we are be able to define, for any suitable monad of transition T, a category of T-transition systems—the objects are T-coalgebras, and the morphisms are lax cohomomorphisms. In particular, we recover the category **TS** by taking $T = \mathcal{P}$.

The second task is understanding the construction of the Sim monad in terms of a category of $\mathcal{P}$-coalgebras.

The monad Sim can be decomposed in the following way.

1. on the set of states A it acts like $\mathcal{P}$
2. on the morphisms, $f : A \to B$, it acts like $\mathcal{P}$
3. on the transition relation $\alpha : A \nrightarrow A$, it lifts α to a transition relation $\mathcal{P}^+(\alpha)$ on $\mathcal{P}A$

Components 1 and 2 are simply the components of the standard covariant powerset functor on objects (sets) and morphisms (functions). It is component 3 that is interesting, because by keeping components 1 and 2 constant and swapping out component 3 for a *different* action $\mathbf{Rel}(A, A) \to \mathbf{Rel}(\mathcal{P}A, \mathcal{P}A)$, we arrive at the related monads RevSim and DSim.

The three monads Sim, RevSim and DSim are built on the same basic *monad of simulation*, which in this case is also $\mathcal{P}$. The final "magic ingredient" is a sufficiently well-behaved mapping of transition relations on A to transition relations on $\mathcal{P}(A)$. The construction of these monads is "modular", because a different choice of magic ingredient will lead to a different type of simulation

monad.

Component	Type	Action of Sim
set of states	$\mathbf{Set} \to \mathbf{Set}$	$A \mapsto \mathcal{P}(A)$
morphisms	$\mathbf{Set}(A, B) \to \mathbf{Set}(\mathcal{P}A \to \mathcal{P}B)$	$f \mapsto \mathcal{P}(f)$
transition relations	$\mathbf{Rel}(A, A) \to \mathbf{Rel}(\mathcal{P}A, \mathcal{P}A)$ $\mathbf{Set}(A, \mathcal{P}A) \to \mathbf{Set}(\mathcal{P}A, \mathcal{P}\mathcal{P}A)$	$\alpha \mapsto \mathcal{P}^+(\alpha)$

In a more general setting, we might try something like this. Let (T, η^T, μ^T) be a monad of transition on a base category $\mathcal{C}$. If the Kleisli category $\mathcal{C}_T$ is **Pos**-enriched, we have a decent notion of a category of T-transition systems with transition preserving morphisms. Let (S, η^S, μ^S) be another monad on $\mathcal{C}$. A "magic ingredient" in this context would be a mapping $\underline{S}$ that turns a system $\alpha : A \to TA$ into a system $\underline{S}\alpha : SA \to TSA$.

When $\underline{S}$ can be endowed with the structure of a monad $\underline{S}, \eta^{\underline{S}}, \mu^{\underline{S}})$ on the Kleisli category $\mathcal{C}_T$ it is known as an *extension of the monad* (S, η^S, μ^S) *to* $\mathcal{C}_T$. These are known to be in correspondence with *monad distributive laws of* S *over* T, which are natural transformations $\lambda : ST \to TS$ that satisfy four conditions: the η^T and μ^T laws, and the η^S and μ^S laws. It is the former pair of T-laws that make the extension $\underline{S}$ a functor. The latter S-laws correspond to the naturality conditions for $(\underline{S}, \eta^{\underline{S}}, \mu^{\underline{S}})$. This idea is essentially due to Beck [7], but a good presentation of this correspondence can be found in [25].

Various weakenings of these conditions exist in the literature. Garner and Goy [25, 23] have studied *weak distributive laws*, which ignore the η^S condition. The consequence is that the extension $\underline{S}$ does not have a natural transformation $\eta^{\underline{S}}$. Garner gives a correspondence theorem between weak extensions and weak distributive laws in [23].

A further weakening is that of a functorial extension (the terminology of a "functorial extension" is novel to this thesis, but such examples have been studied before, by [11] e.g.). A functorial extension of S to $\mathcal{C}_T$ is simply a functor $\underline{S}$ on $\mathcal{C}_T$ that satisfies the requirement that $\underline{S}(\eta^T \circ f) = \eta^T \circ Sf$ for any $\mathcal{C}$-morphism f. These are in correspondence with a suitably weakened version of distributive laws (ignoring the η^S and μ^S laws).

However, the mapping $\mathcal{P}^+$ is emphatically *not* an extension of the powerset monad to **Rel**, nor is it even a functorial extension of $\mathcal{P}$. The proper notion, it

turns out, is that of a *lax extension*, which consists of a lax functor $\underline{S}$ on $\mathcal{C}_T$, and on the other hand, lax distributive laws. Lax distributive laws of the form $SQ \to QS$, where Q is the monad of a quantale, have been studied by Tholen and others in the context of *monoidal topology*. Tholen provides a lax correspondence theorem in [53]. In Section 4.4 a generalisation of this result is given (Theorem 4.28), providing a correspondencence theorem for the case where T is merely a monad with a **Pos**-enriched Kleisli category.

At this point, we have a slight mismatch. The construction of Sim comes from a lax extension of $\mathcal{P}$ to **Rel**. The extension $\mathcal{P}^+$ is not a monad on **Rel**, because the components $\eta^{\mathcal{P}} \circ \eta^{\mathcal{P}}$ and $\eta^{\mathcal{P}} \circ \mu^{\mathcal{P}}$ are not natural transformations. But $(\mathrm{Sim}, \eta^{\mathcal{P}}, \mu^{\mathcal{P}})$ is a monad on $\mathbf{TS}_\Sigma$. There is something else going on here.

In the general case of a monad (T, η^T, μ^T) and a lax extension $\underline{S}$ of a monad (S, η^S, μ^S), we will construct a monad $\mathbb{S}$ on the lax category of T-coalgebras (these are T-transition systems). The action of $\mathbb{S}$ on a coalgebra $\alpha : A \to TA$ will be

$$\underline{S}\alpha : SA \to TSA,$$

whereas for lax cohomomorphisms $f : \alpha \to \beta$, $\mathbb{S}$ will act as S:

$$\mathbb{S}(f) = Sf : SA \to SB.$$

Mirroring the definition of Sim, the components of the monad $\mathbb{S}$ will simply be the components of S: $(\mathbb{S}, \eta^S, \mu^S)$. In order for this to actually be a monad, we require that η^S and μ^S are natural transformations, and that the monad laws hold. But as the lax T-coalgebra category is *concrete*—that is, morphisms of T-coalgebras are merely morphisms in the underlying category that satisfy an extra condition—we can deduce these for free.

The only thing we have to worry about is whether the components of $(\mathbb{S}, \eta^{\mathbb{S}}, \mu^{\mathbb{S}})$ are *actually lax cohomomorphisms*. Strikingly, this happens precisely when a *laxened* version of the η^S and μ^S axioms for distributive laws holds.

One might conjecture a correspondence theorem between lax monad distributive laws and monads on the lax category of coalgebras. Unfortunately we cannot deduce the $\Leftarrow$ direction. The category of coalgebras is not large enough. A monad $(\mathbb{S}, \eta^S, \mu^S)$ on this category will only give us the values of $\underline{S}$ on the *endomorphisms* of $\mathcal{C}_T$, because these are the transition morphism components of T-coalgebras. In order to recover the full data of the lax functor $\underline{S}$ on $\mathcal{C}_T$

from a transition system monad $(\mathbb{S}, \eta^S, \mu^S)$ we need to consider a horizontal categorification of the the category of coalgebras.

The categorical formulation borrows from the realisation ([21]) that deterministic and partial transition systems with labels in Σ may be considered as *actions* of the word monoid $\Sigma^\star$. In particular, the natural notion of a morphism of monoid actions corresponds exactly to a transition preserving function. We take the liberty of generalising the notion of a monoid action—a monoid morphism into the endomorphism monoid $\mathbf{Set}(A, A)$— as monoid morphisms into the *Kleisli endomorphism* monoid $\mathcal{C}_T(A, A)$. Importantly, a morphism of T-actions remains in the base category, and not in the Kleisli category.

From here a horizontal categorification into the category of *lax functors* into $\mathcal{C}_T$ is the next step. As this construction is arbitrary in the *source category*, we actually have a (contravariant) functor T-$\mathbf{Act}(-)$ from the category of $\mathbf{Pos}$-enriched categories into $\mathbf{Cat}$. We are then able to state and prove a correspondence theorem identifying monads $\mathbb{S}$ on this functor with lax monad distributive laws $ST \to TS$.

A crucial intermediate step is a restricted correspondence theorem, which is a sort of Yoneda result. There is a correspondence between mere natural transformations on the T-action functor and *functorial* lax distributive laws $ST \to TS$.

4.1 Thinking with coalgebras

Up till now, we have been working with non-deterministic transition systems with labels in a set Σ. We have been presenting such transition systems as comprising

- a *base set* A, and
- a *transition relation* $R \subseteq A \times \Sigma \times A$

An example of a non-deterministic system over the alphabet $\Sigma = \{a, b, c\}$ is found in Fig. 4.2a (with set of states $A = \{x, y, z\}$). The first problem we encounter is that the explicit notion of simulation (and morphism) is, in many contexts, too strict.

Example 4.1. This example is adapted from [34]. Consider the two *probablistic* transition systems depicted in Fig. 4.1. The edges are labelled by elements of the

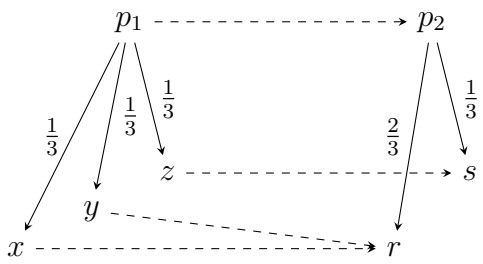


Figure 4.1: A simulation of probablistic systems?

interval $[0, 1]$, such that the sum of the outgoing edges is equal to 1 (by convention, the lack of any outgoing edges at x indicates that $x \xrightarrow{1} x$, and likewise for the other leaf states).

The dashed lines depict a *probabilistic bisimulation*—a notion introduced in [34]. The essence is that the x and y are both related to r, and hence the *cumulative probability* of starting at p_1 and ending up at either x or y must be equal to the probability of transitioning from p_2 to r:

$$\frac{1}{3} + \frac{1}{3} = \frac{2}{3}.$$

But, if we were to envision these systems in the simple framework described above, as transition systems with labels in the set $\Sigma = [0, 1]$, the relation depicted is certainly *not* a bisimulation—it is not even a simulation!

To see this, we may form the corner

$$\begin{tikzcd} p_1 \arrow[r, dashed] \arrow[d, "\frac{1}{3}"'] & p_2 \\ x & \end{tikzcd}$$

Now, if we wish to complete the square, we will need to find a state q with $x \dashrightarrow q$ and $p_2 \xrightarrow{\frac{1}{3}} q$. But no such state exists: the only state that is related to x is r, and we do not have a transition $p_2 \xrightarrow{\frac{1}{3}} r$, rather we have $p_2 \xrightarrow{\frac{2}{3}} r$.

The point of this example is that the presented notion of simulation is too strict—it does not allow us to express all of the types of simulations that exist and have found use "in the wild". We can see that probabilistic transition systems (and probabilistic bisimulations) are not *merely* non-deterministic systems with labels

in $\Sigma = [0, 1]$. The desired *behaviour* of probablistic systems and bisimulations—encoding notions of *cumulative probability*—cannot be expressed in the naïve framework. In order to have a good categorical understanding of these sorts of systems, we need a different formulation.

One approach is to view a transition system as a *coalgebra* for a functor of transition T. Traditionally, T is a functor on **Set**, but there is, in principle, nothing wrong with considering functors on a different base category $\mathcal{C}$ (for example, a category of *domains* [46, 16], or Stone spaces [22, 8]).

Some functors that appear in the literature are listed below.

Example 4.2. The trivial functor $T = \mathrm{Id}$, which is the type of deterministic transition systems

Example 4.3 ([16, 6])**.** When $T = \mathcal{D}$, we get probablistic transition systems.

Example 4.4 ([29, 2])**.** We can take $T = \mathcal{P}$, as well as variations like $\mathcal{P}^f$, or $\mathcal{P}^{\geq}$. These functors encode non-deterministic branching. They can also be combined with other functors, as in Examples 4.5 and 4.6.

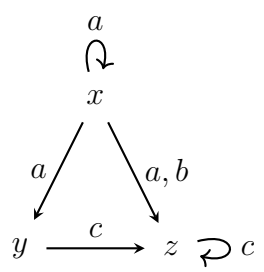
Example 4.5 ([37])**.** The functor $T = \mathcal{P}\,\mathrm{Maybe}$ is the type of transition systems with divergence.

Example 4.6 ([25, 29])**.** The functor $T\mathcal{P}(\Sigma \times -)$ is the type of *labelled* non-deterministic systems.

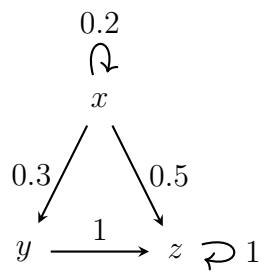
Example 4.7 ([12, 10, 43])**.** If $T = \mathcal{P}_S$, then T-coalgebras are weighted automata with weights in a semiring S.

When T is a transition functor on **Set**, a T-coalgebra consists of a *set of states* A, and a *transition function* $\alpha : A \to TA$. For every state $a \in A$, the value $\alpha(a) \in TA$ is called the *successor* of a.

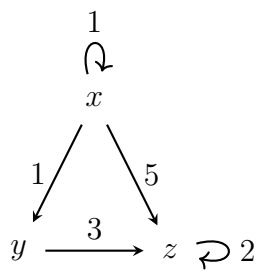
- When $T = \mathrm{Id}$, the successor of a is a unique state $\alpha(a) \in A$.
- When $T = \mathcal{D}$, the successor of a is a distribution on A
- When $T = \mathcal{P}$, the successor of a is in fact a set of successors, $\alpha(a) \subseteq A$.
- When $T = \mathcal{P}_S$, the successor of a is an S-valued subset of A.



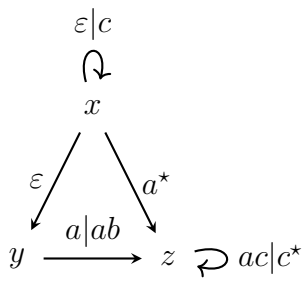
(a) Non-deterministic



(b) Probablistic



(c) Weighted (in $\mathbb{N}$, perhaps)



(d) Generalised NFA

Figure 4.2: Some diverse transition systems

The category of coalgebras provides a natural notion of a morphism of transition systems. Recall Definition 2.12. If $\alpha : A \to TA$ and $\beta : B \to TB$ are T-coalgebras, then a coalgebra morphism $f : \alpha \to \beta$ consists of a function $f : A \to B$ such that the diagram below commutes.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\alpha"'] & B \arrow[d, "\beta"] \\ TA \arrow[r, "Tf"] & TB \end{tikzcd}$$

This has an intuitive interpretation in terms of transition systems. The diagram depicts two functions $A \to TB$, i.e. from states of A to successors in B. In order for it to commute, they must have equal values for all states a. The first is $Tf \circ \alpha$, which is essentially "f applied to all the successors of a". The other is $\beta \circ f$, which we can think of as "the successor of $f(a)$".

- When $T = \mathrm{Id}$, f is a coalhebra morphism precisely when $f(\alpha(a)) = \beta(f(a))$. That is, when the unique successor of $f(a)$ is equal to the image under f of the successor of a.

- When $T = \mathcal{D}$, the condition for f becomes $\mathcal{D}f(\alpha(a)) = \beta(f(a))$. The successors are distributions over states of B. In order for this equality to hold, we require that for every state b,

  $$\sum_{a' \in f^{-1}(b)} \alpha(a, a') = \beta(f(a), b).$$

  This is a very strict condition. It says that the probability of transitioning from $f(a)$ to b is equal to the combined probability of transitioning from a to a preimage of b. A more natural condition would be

  $$\sum_{a' \in f^{-1}(b)} \alpha(a, a') \leq \beta(f(a), b),$$

  which is equivalent to

  $$\alpha(a, a') \leq \beta(f(a), f(a')) \tag{4.1}$$

  for all a, a'. This expresses the requirement that the image of every transition $f(a) \to f(a')$ must be *at least as likely* as the original transition $a \to a'$ in A.

- When $T = \mathcal{P}$, the coalgebra morphisms are in fact the functional bisimulations. That is, they are the functions that preserve *and reflect* transitions.

Remark 4.8. A standard approach to handling *labelled transition systems* (with labels in a set Σ), is to treat them as coalgebras for the functor $\mathcal{P}(- \times \Sigma)$. According to this understanding, a transition system comprises a set A and a transition function $\alpha : A \to \mathcal{P}(A \times \Sigma)$. For every state a, the successor value $\alpha(a) \subseteq A \times \Sigma$ consists of pairs (a', σ), indicating transitions $a \xrightarrow{\sigma} a'$.

However, we find the alternate type signature of $\Sigma \to (A \to \mathcal{P}A)$ more useful. This perspective essentially abstracts away the labelling, recognising a labelled system as a *family* of $\mathcal{P}$-systems. In fact, this family of transition relations (indexed by the elements of Σ) induces a monoid homomorphism $\Sigma^\star \to \mathbf{Rel}(A, A)$. We will expand on this perspective in Chapter 6.

4.2 A lax setting

The discussion above shows that the equality of morphisms that is expressed by the coalgebra morphism condition is too strict. Rather than expressing "edge preserving functions", we end up with functions that preserve and reflect transitions. In order to be able to express just one of these conditions, we require the structure of a category where the morphisms are "comparable", that is, instead of a mere hom-"set" of morphisms $A \to B$, we have a hom-"poset". The setting of *enriched category theory* [30] will be suitable.

Let $\mathcal{C}$ be a category. We briefly recall the cateogorical product in **Pos**. If $(A, \leq)$ and $(B, \leq)$ are two posets, the product $A \wedge B$ is given by the following ordering on $A \times B$:

$$(a, b) \leq (a', b') \iff a \leq a' \text{ and } b \leq b'$$

Definition 4.9. A ***Pos**-category* $(\mathcal{C}, \leq)$ consists of a category $\mathcal{C}$ and and a partial order $\leq$ on every every hom-set of $\mathcal{C}$. We also require that composition is a morphism (i.e. monotone function) $\mathcal{C}(B, C) \wedge \mathcal{C}(A, B) \to \mathcal{C}(A, C)$ in **Pos**.

If $(\mathcal{C}, \leq)$ is a **Pos**-category, we can say that $\leq$ is a ***Pos**-enrichment*, or an *enrichment in the category **Pos*** of $\mathcal{C}$. When the enrichment component $\leq$ is clear

from context, we may refer to the **Pos**-category $(\mathcal{C}, \leq)$ as merely $\mathcal{C}$. This is a mild abuse of notation.

Explicitly, the requirement that composition is monotone means that if we have two pairs of parallel morphisms $f, f' : A \to B$ and $g, g' : B \to C$ with $f \leq f'$ and $g \leq g'$ we may deduce that

$$g \circ f \leq g' \circ f'. \tag{4.2}$$

Remark 4.10. If we have two parallel morphisms $f, f' : A \to B$ in a **Pos**-enriched category $\mathcal{C}$ with $f \leq f'$, we can say that f is *included in* f', or that *there is an inclusion* $f \leq f'$, or that f is *extended by* f'.

We may also draw diagrams. For example, the diagram below on the left indicates that there is an inclusion of $g \circ p$ into $q \circ f$.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "p"'] \arrow[dr, phantom, "\Nearrow"] & B \arrow[d, "q"] \\ C \arrow[r, "g"] & D \end{tikzcd} \qquad \begin{tikzcd} B \arrow[r, "s"] \arrow[d, "q"'] \arrow[dr, phantom, "\Nearrow"] & E \arrow[d, "r"] \\ D \arrow[r, "t"] & F \end{tikzcd}$$

One consequence of working in a **Pos**-enriched category is that we are able to combine such diagrams by pasting. If we have additionally the diagram above on the right, witnessing the inclusion $t \circ q \leq r \circ s$, we can paste the two side-by-side to form the composite diagram

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "p"'] \arrow[dr, phantom, "\Nearrow"] & B \arrow[r, "s"] \arrow[d, "q"] \arrow[dr, phantom, "\Nearrow"] & E \arrow[d, "r"] \\ C \arrow[r, "g"] & D \arrow[r, "t"] & F \end{tikzcd} \quad = \quad \begin{tikzcd} A \arrow[r, "s \circ f"] \arrow[d, "p"'] \arrow[dr, phantom, "\Nearrow"] & E \arrow[d, "r"] \\ C \arrow[r, "t \circ g"] & F \end{tikzcd}$$

The correctness of this diagram (i.e. the existence of the inclusion $t \circ g \circ p \leq r \circ s \circ f$ follows from the component diagrams and Eq. 4.2.

$$\begin{aligned} t \circ (g \circ p) &\leq t \circ (q \circ f) \\ &= (t \circ q) \circ f \\ &\leq (r \circ s) \circ f \end{aligned}$$

Naturally, we may also paste inclusion diagrams vertically.

Let $(\mathcal{C}, \leq_C)$ and $(\mathcal{D}, \leq_D)$ be two **Pos**-categories. We will consider the structure of functors between these categories. There are several interesting variations of what a "functor between **Pos**-categories" could be. In every case, a functor $F : (\mathcal{C}, \leq_C)$ and $(\mathcal{D}, \leq_D)$ will consist of the same data as a functor on the underlying categories $F : \mathcal{C} \to \mathcal{D}$, that is:

- for every object A of $\mathcal{C}$, an object FA of $\mathcal{D}$,
- for every morphism $f : A \to B$ of $\mathcal{C}$, a morphism $Ff : FA \to FB$ in $\mathcal{D}$,

Now, there are three conditions that may or may not be satisfied.

1. The lax F-id condition says that for every object A, there is an inclusion $\mathrm{id}_{FA} \leq_D F(\mathrm{id}_A)$. If in fact there is an equality $\mathrm{id}_{FA} = F(\mathrm{id}_A)$ we say that F-id. holds *strictly*.
2. The lax F-comp condition says that for every composable pair of morphisms $f : A \to B$ and $g : B \to C$, we have $F(g) \circ F(f) \leq_D F(g \circ f)$. Again, if the inclusion is an equality, then we say that this condition holds strictly.
3. The F-mono condition says that if we have $f \leq f' : A \to B$ then $Ff \leq Ff' : FA \to FB$.

Definition 4.11. Let F be a mapping of objects and morphisms $\mathcal{C} \to \mathcal{D}$ that satisfies F-mono.

1. If F also satisfies (F-id) and (F-comp.) laxly then we will call F a *lax* ***Pos****-functor*. This is the most relaxed sort of functor we will consider.
2. If F satisfies (F-id) laxly and (F-comp.) *strictly*, then we will say that F is a *semilax* **Pos**-functor.
3. If F satisfies both (F-id) and (F-comp.) strictly, then F is a *strict* **Pos**-functor.

Note that it is possible for a lax functor to satisfy (F-id) strictly and (F-comp.) only *laxly*. But this is not of particular interest to us, so we will not give this case a special name.

These naming conventions are summarised in Table 4.1.

F-id	F-comp	Type of **Pos**-functor
lax	lax	lax
lax	strict	semilax
strict	strict	strict

Table 4.1: A taxonomy of **Pos** functors

Example 4.12. Let $\mathcal{C}$ be any category. Then we can endow $\mathcal{C}$ with the *trivial*, or *discrete* **Pos**-enrichment, $=$. The only inclusions of f into g are when f and g are identical.

The collection of small **Pos**-categories forms a category itself. In the most general setting the morphisms will be lax **Pos**-functors, but we can pick out the wide subcategories of semilax and strict **Pos**-functors.

Definition 4.13. There is a category $\mathbf{PosCat}^{\mathsf{lax}}$ whose

- objects are categories enriched in **Pos**, and
- morphisms are lax **Pos**-functors.

There are two interesting wide subcategories:

1. The category $\mathbf{PosCat}^{\mathsf{semi}}$ has the same objects as $\mathbf{PosCat}^{\mathsf{lax}}$, and the morphisms are the semilax **Pos**-functors.
2. The category **PosCat** has as morphisms the strict **Pos**-functors.

We can endow $\mathbf{PosCat}^{\mathsf{lax}}$ with higher categorical structure ([36]), but for our purposes, the 1-categorical structure will be sufficient.

The particular case that will be considered in this thesis is that of a monad (T, η^T, μ^T) with a **Pos**-enriched Kleisli category.

Example 4.14. The following monads admit enriched Kleisli categories.

The Id monad on $\mathcal{C}$ admits the trivial **Pos**-enrichment.

The enrichment for the powerset monad $\mathcal{P}$ is essentially given pointwise by $\subseteq$. For parallel morphisms $f, g : A \to \mathcal{P}(B)$, we say $f \leq g$ if, for all $a \in A$ we have

$$f(a) \subseteq g(a) \subseteq B.$$

There is a **Pos**-enrichment for the Maybe monad. When $f, g : A \to \mathrm{Maybe}\, B$, we have $f \leq g$ if for all $a \in A$, either $f(a)$ and $g(a)$ are both defined and equal, or $f(a) = \bot$.

When S is a semiring ordered by $\leq$, this extends pointwise to a **Pos**-enrichment on $\mathbf{FinRel}_S$.

4.3 Interaction via distributive laws

The simulation monads on $\mathbf{TS}_\Sigma$ consist of the powerset functor $\mathcal{P}$ applied to states and morphisms, and a "magic ingredient" that transforms relations on A to relations on $\mathcal{P}(A)$. We shall see that this ingredient is actually a lax distributive law of type $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$. We begin by recollecting the theory of strict distributive laws.

Definition 4.15. Let T and S be monads on $\mathcal{C}$. A *distributive law of S over T* consists a natural transformation $\lambda : ST \to TS$ such that the equations below hold. These are also depicted graphically in Fig. 4.3.

$$\eta^T_{SA} = \lambda_A \circ S\eta^T_A \tag{η^T}$$

$$\eta^S_{TA} = \lambda_A \circ T\eta^S_A \tag{η^S}$$

$$\mu^T_{SA} \circ T\lambda_A \circ \lambda_{TA} = \lambda_A \circ S\mu^T_A \tag{μ^T}$$

$$T\mu^S_A \circ \lambda_{SA} \circ S\lambda_A = \lambda_A \circ \mu^S_{TA} \tag{μ^S}$$

The naturality condition for λ further requires that for every $f : A \to B$ in $\mathcal{C}$ we have

$$TSf \circ \lambda_A = \lambda_B \circ STf \tag{nat λ}$$

Definition 4.16. Let (S, η^S, μ^S) be a monad on $\mathcal{C}$. Recall that a *(strict) monad extension of S over T* is a strict monad $(\underline{S}, \eta^{\underline{S}}, \mu^{\underline{S}})$ on the Kleisli category $\mathcal{C}_T$ such that the *extension condition* holds:

$$F_T S = \underline{S} F_T \tag{ext.}$$

More concretely, this means that S and $\underline{S}$ must agree on objects: for all A in

$$\begin{tikzcd} STA \arrow[r, "STf"] \arrow[d, "\lambda_A"'] & STB \arrow[d, "\lambda_B"] \\ TSA \arrow[r, "TSf"] & TSB \end{tikzcd}$$

strict (nat λ)

$$\begin{tikzcd} S \arrow[r, "S\eta^T"] \arrow[dr, "\eta^T S"'] & ST \arrow[d, "\lambda"] \\ & TS \end{tikzcd}$$

strict (η^T) law

$$\begin{tikzcd} T \arrow[r, "T\eta^S"] \arrow[dr, "\eta^S T"'] & ST \arrow[d, "\lambda"] \\ & TS \end{tikzcd}$$

strict (η^S) law

$$\begin{tikzcd} STT \arrow[r, "S\mu^T"] \arrow[d, "\lambda T"'] & ST \arrow[dd, "\lambda"] \\ TST \arrow[d, "T\lambda"'] & \\ TTS \arrow[r, "\mu^T S"'] & TS \end{tikzcd}$$

strict (μ^T) law

$$\begin{tikzcd} SST \arrow[r, "\mu^S T"] \arrow[d, "S\lambda"'] & ST \arrow[dd, "\lambda"] \\ STS \arrow[d, "\lambda S"'] & \\ TSS \arrow[r, "T\mu^S"'] & TS \end{tikzcd}$$

strict (μ^S) law

Figure 4.3: A strict distributive law

$\mathcal{C}$ we have

$$SA = \underline{S}A,$$

and that for morphisms $f : A \to B$ we have:

$$\eta^T_{SB} \circ Sf = \underline{S}(\eta^T_B \circ f)$$

Diagrammatically, we require that the diagram below commutes

$$\begin{tikzcd} \mathcal{C} \arrow[r, "F_T"] \arrow[d, "S"'] & \mathcal{C}_T \arrow[d, "\underline{S}"] \\ \mathcal{C} \arrow[r, "F_T"'] & \mathcal{C}_T \end{tikzcd}$$

And as a further condition, we require that the unit and multiplication of $\underline{S}$ are extensions of the unit and multiplication of S:

$$\eta^{\underline{S}} = \eta^T \circ \eta^S$$
$$\mu^{\underline{S}} = \eta^T \circ \mu^S$$

The following is a useful lemma relating mixed compositions in a Kleisli

category and in the base category.

Lemma 4.17. *Let T be a monad on $\mathcal{C}$. Let $f : X \to A$ and $h : B \to Y$ be morphisms in $\mathcal{C}$, and $g : A \nrightarrow B$ be a morphism in $\mathcal{C}_T$. Then the following equations hold:*

$$g \bullet F_T f = g \circ f : X \to TB \tag{4.3}$$

$$F_T h \bullet g = Th \circ g : A \to TY \tag{4.4}$$

Proof. These simply follow from the monad laws.

$$\begin{aligned} g \bullet F_T f &= \mu^T_B \circ Tg \circ \eta^T_A \circ f \\ &= \mu^T_B \circ \eta^T_{TB} \circ g \circ f \\ &= g \circ f \\ F_T h \bullet g &= \mu^T_Y \circ T\eta^T_Y \circ Th \circ g \\ &= Th \circ g \end{aligned}$$

□

The following result is well known [23, 25]. We will not give the details of the proof yet, but we will sketch the constructions. Later on we shall examine precisely how the various conditions on distributive laws and extensions correspond to each other in the lax setting.

Lemma 4.18. *There is a correspondence between distributive laws λ of S over T and extensions $\underline{S}$ of S over T.*

Sketch of proof. Let λ be a distributive law $ST \to TS$. We define a functor $\underline{S}$ on $\mathcal{C}_T$ in the following way:

- on objects A, we have $\underline{S}A = SA$, and
- on morphisms $f : A \nrightarrow B$ we have $\underline{S}f = \lambda \circ Sf : SA \nrightarrow SB$.

The monad components must be

- $\eta^{\underline{S}} = \eta^T \circ \eta^S$, and
- $\mu^{\underline{S}} = \eta^T \circ \mu^S$.

It is necessary to show that $\underline{S}$ is indeed a functor, that it is an extension of S, and that the transformations $\eta^{\underline{S}}$ and $\mu^{\underline{S}}$ are natural and satisfy the monad laws. All this does follow from the requirement that λ is a distributive law, but we will examine how these properties interact in a more granular way in Chapter 5, when we look at the lax setting.

On the other hand, if $\underline{S}$ is an extension of S, then we define a distributive law λ by $\lambda_A = \underline{S}(\varepsilon_A) : STA \nrightarrow SA$. Again, the required equations can be deduced from the extension laws, but we will omit this verification for now.

We will, however, verify that these two constructions (from distributive laws to extensions, and the other way around) are actually inverse to each other. That is, we will show that

$$\lambda_A \circ S(\varepsilon_A) = \lambda_A \tag{4.5}$$

$$\underline{S}(\varepsilon_B) \circ Sf = \underline{S}f. \tag{4.6}$$

The first equation follows from the fact that $\varepsilon_A = \mathrm{id}_A : A \to A$ (in $\mathcal{C}$), and S is a functor (and hence $S(\mathrm{id}_A) = \mathrm{id}_{SA}$). We can deduce the second equation from Lemma 4.17 and the fact that $\underline{S}$ is an extension of S.

$$\begin{aligned}
\underline{S}f &= \underline{S}(\varepsilon_B \circ f) \\
&= \underline{S}(\varepsilon_B \bullet F_T f) && \text{(Lemma 4.17)} \\
&= \underline{S}(\varepsilon_B) \bullet \underline{S}(F_T f) && (\underline{S} \text{ is a strict functor}) \\
&= \underline{S}(\varepsilon_B) \bullet F_T(Sf) && (\underline{S} \text{ extends } S) \\
&= \underline{S}(\varepsilon_B) \circ Sf && \text{(Lemma 4.17)}
\end{aligned}$$

□

Remark 4.19. There is a third side to this correspondence: liftings of T to the *Eilenberg-Moore* category of S. This is further discussed in [25], but it is not relevant to the work of this thesis.

4.4 Lax distributive laws

The conditions of a distributive law (and of a monad extension) are quite strong. Recent work [54, 57] has exhibited "no-go theorems" that prove that there can be no strict distributive laws of type $\mathcal{PP} \to \mathcal{PP}$, nor $\mathcal{DP} \to \mathcal{PD}$, or several other

types. Nonetheless, there are several opportunities for weakening the conditions of distributive laws.

The approach to lax extensions that we shall take differs slightly from what is found in the literature. The first divergence is that we shall not be interested in making sure that $\underline{S}$ is a monad on $\mathcal{C}_T$, so we will be ignoring the η^S and μ^S laws entirely (these correspond to the monad properties of $\underline{S}$). We will also be interested in the case where S is not a monad at all, but a mere functor on $\mathcal{C}$. This will be useful when we define certain functors on categories of transition systems in Chapter 6.

An alternative approach is to allow some of the laws to hold only *laxly*. Starting with such a laxened distributive law $\lambda : ST \to TS$, we could follow the procedure described in Lemma 4.18 and end up with a mapping $\underline{S}$ on $\mathcal{C}_T$. This mapping might have various properties:

1. it could be a strict, semilax, or lax functor $\mathcal{C}_T$
2. one or both of $\eta^{\underline{S}}$ and $\mu^{\underline{S}}$ could be natural transformations
3. one or more of the monad laws for $(\underline{S}, \eta^{\underline{S}}, \mu^{\underline{S}})$ could be satisfied

There are also various possibilities for how $\underline{S}$ interacts with S. The *extension condition* can be satisfied strictly, or only laxly. This means that instead of requiring that for every morphism f we have $F_T S f$ exactly equal to the image $\underline{S} F_T f$, we have only a lax inclusion. This by itself is quite a weak condition, and we shall call a functor $\underline{S}$ with this property a *lax pseudo-extension*, to emphasize that this is insufficient for good behaviour. We will introduce two new conditions that express some further interaction between $\underline{S}$ and S.

Definition 4.20. Let S be a functor on $\mathcal{C}$, and $\underline{S}$ a lax functor on $\mathcal{C}_T$.

1. We say that $\underline{S}$ satisfies the *lax pseudo-extension condition (with respect to S)* if for every morphism $f : A \to B$ in $\mathcal{C}$

$$F_T S f \leq \underline{S} F_T f \tag{lax ext}$$

This amounts to the lax inclusion in the diagram below.

$$\begin{tikzcd} \mathcal{C} \arrow[r, "F_T"] \arrow[d, "S"'] \arrow[dr, phantom, "\Nearrow"] & \mathcal{C}_T \arrow[d, "\underline{S}"] \\ \mathcal{C} \arrow[r, "F_T"] & \mathcal{C}_T \end{tikzcd}$$

2. We will say that $\underline{S}$ satisfies the *lax left whiskering condition* if we have, for every Kleisli morphism $p : A \nrightarrow B$ and $\mathcal{C}$-morphism $f : B \to Y$ we have

$$F_T S f \bullet \underline{S} p \leq \underline{S}(F_T f \bullet p) \qquad \text{(lax left whisk.)}$$

3. The *lax right whiskering condition* requires that for $p : A \nrightarrow B$ and $g : X \to A$ we have

$$\underline{S} p \bullet F_T S g \leq \underline{S}(p \bullet F_T g) \qquad \text{(lax right whisk.)}$$

These conditions may also hold strictly, if the inclusion is actually an equality.

Note that by Lemma 4.17 the left and right whiskering conditions can be written as:

$$T S f \circ \underline{S} p \leq \underline{S}(T f \circ p) \qquad \text{(left)}$$
$$\underline{S} p \circ S g \leq \underline{S}(p \circ g) \qquad \text{(right)}$$

Proposition 4.21. *If $\underline{S}$ is a strict extension of S then $\underline{S}$ strictly preserves identities.*

If $\underline{S}$ strictly preserves identities and satisfies one of the whiskering conditions strictly, then $\underline{S}$ is a strict extension.

Proof. Suppose $\underline{S}$ is a strict extension. Take $f = \mathrm{id}_A$ to see that

$$F_T(S\mathrm{id}_A) = \underline{S}(F_T \mathrm{id}_A)$$
$$\eta^T_{SA} = \underline{S}(\eta^T_A)$$

Now suppose that it is the strict right whiskering condition that holds. Let $f : A \to B$ be a morphism. We see that

$$\begin{aligned} S F_T f &= S F_T f \bullet \eta^T_{SA} \\ &= S F_T f \bullet \underline{S}(\eta^T_A) && \text{(strict id.)} \\ &= \underline{S}(F_T f \bullet \eta^T_A) && \text{(strict right whisk.)} \\ &= \underline{S}(F_T f) \end{aligned}$$

If instead we have the left whiskering condition, the proof proceeds symmetrically,

introducing η^T_{SA} on the left hand side of SF_Tf. $\square$

The following result is a variation of [53, Proposition 6.3]. It rather simplifies the situation. In the presence of the lax identity and composition conditions, it turns out that left-whiskering, right-whiskering, and being a lax pseudo-extension are all equivalent.

Proposition 4.22. *Let $\underline{S}$ be a lax functor on $\mathcal{C}_T$ and S a functor on $\mathcal{C}$. The following are equivalent.*

1. *$\underline{S}$ is a lax pseudo-extension*
2. *$\underline{S}$ is lax left whiskering*
3. *$\underline{S}$ is lax right whiskering*

Suppose furthermore that $\underline{S}$ satisfies one of the whiskering conditions strictly*. Then $\underline{S}$ is a strict extension of S if and only if $\underline{S}$ strictly preserves identities.*

Proof.

- $(1 \implies 2,3)$ Suppose that $\underline{S}$ is a lax pseudo-extension. Let p be a Kleisli morphism, and f and g pre- and post-composable $\mathcal{C}$-morphisms. We then calculate

$$\begin{aligned} SF_Tf \bullet \underline{S}p &\leq \underline{S}(F_Tf) \bullet \underline{S}p && \text{(pseudo-ext)} \\ &\leq \underline{S}(F_Tf \bullet p) && \text{(lax comp.)} \\ \underline{S}p \bullet SF_Tf &\leq \underline{S}p \bullet \underline{S}(F_Tf) && \text{(pseudo-ext)} \\ &\leq \underline{S}(p \bullet F_Tf), && \text{(lax comp.)} \end{aligned}$$

  so the left and right whiskering conditions hold laxly.

- $(2 \implies 1)$ Suppose that lax left whiskering holds. By taking $p = \eta^T$ we deduce

$$\begin{aligned} SF_Tf &= SF_Tf \bullet \eta^T \\ &\leq SF_Tf \bullet \underline{S}\eta^T && \text{(lax id.)} \\ &\leq \underline{S}(F_Tf \bullet \eta^T) && \text{(lax left whisk.)} \\ &= \underline{S}F_Tf, \end{aligned}$$

  which shows that $\underline{S}$ is a pseudo-extension.

• (3 $\implies$ 1) This case proceeds as above.

If the lax left whiskering condition holds, then we have for any $f : B \to Y$ and $p : A \nrightarrow B$ that

$$F_T S f \bullet \underline{S} p \leq \underline{S}(F_T f \bullet p)$$

By taking $p = \eta^T_B$ and using the fact that $\underline{S}$ laxly preserves identities we deduce

$$\begin{aligned} F_T S f &= F_T S f \bullet \eta^T_{SB} && \text{(def } \bullet) \\ &\leq F_T S f \bullet \underline{S}\eta^T_B && \text{(lax id.)} \\ &\leq \underline{S}(F_T f \bullet \eta^T_B) && \text{(lax left whisk)} \\ &= \underline{S}(F_T f) && \text{(def } \bullet) \end{aligned}$$

The case of the lax right whiskering condition proceeds symmetrically.

For the second claim, suppose without loss of generality that it is the left whiskering condition that holds strictly. One direction is easy—if $\underline{S}$ is a strict extension then by Proposition 4.21 it preserves identities strictly.

On the other hand, suppose that $\underline{S}$ strictly preserves identities:

$$\eta^T_{SA} = \underline{S}(\eta^T_A).$$

We use the strict left whiskering condition. Let $f : A \to B$ be a morphism. Then we have that

$$\begin{aligned} F_T(Sf) &= F_T(Sf) \bullet \eta^T_{SA} \\ &= F_T(Sf) \bullet \underline{S}\eta^T_A && \text{(strict id.)} \\ &= \underline{S}(F_T f \bullet \eta^T_A) && \text{(strict left whisk.)} \\ &= \underline{S}(F_T f), \end{aligned}$$

which is the strict extension condition.

□

There are two other conditions that may be of interest. These may be formulated when S is not just a functor, but is endowed with the structure of a monad (S, η^S, μ^S). These final two conditions will express some interaction between a lax functor $\underline{S}$ and the monad components of S.

Definition 4.23. Let (S, η^S, μ^S) be a monad on $\mathcal{C}$, and $\underline{S}$ a lax functor on $\mathcal{C}_T$.

1. The *lax η^S condition* is satisfied if for every $p : A \nrightarrow B$ we have

$$F_T\eta^S \bullet p \leq \underline{S}p \bullet F_T\eta^S \tag{lax η^S}$$

2. The *lax μ^S condition* says that for every $p : A \nrightarrow B$ we have

$$F_T\mu^S \bullet \underline{SS}p \leq \underline{S}p \bullet F_T\mu^S \tag{lax μ^S}$$

Definition 4.24. Let (T, η^T, μ^T) be a monad on $\mathcal{C}$, $\leq$ a **Pos**-enrichment of $\mathcal{C}_T$, and S a functor on $\mathcal{C}$. Let $\underline{S}$ be a lax **Pos**-functor on $\mathcal{C}_T$ such that the right whiskering condition holds strictly We will call $\underline{S}$ a *lax extension of the functor* S, or say that S *laxly extends to the lax **Pos**-functor* $\underline{S}$.

Furthermore, if S is given the structure of a monad (S, η^S, μ^S), we will call $\underline{S}$ a *lax monad extension*, or a *lax extension of the monad* (S, η^S, μ^S) if additionaly the lax η^S and lax μ^S conditions hold.

Remark 4.25. Demanding the right whiskering condition holds strictly seems ungainly. But this will be necessary to guarantee a good *correspondence theorem*. Strict right whiskering means that $\underline{S}$ is fully determined by the corresponding distributive law λ: that we have

$$\underline{S}p = \lambda \circ Sp$$

The left whiskering condition—which we get for free by Proposition 4.22, in the presence of strict right whiskering—will correspond to *naturality* of λ. When λ is strictly natural, we shall in fact have strict left whiskering.

Note that by Proposition 4.22 a lax extension always satisfies the *pseudo-extension* property.

Definition 4.26. Let (T, η^T, μ^T) be a monad on $\mathcal{C}$, $\leq$ a **Pos**-enrichment of $\mathcal{C}_T$, and S a functor on $\mathcal{C}$. Let λ be a family of morphisms $\lambda_A : STA \to TSA$.

If λ is a lax natural transformation, and the lax η^T and lax μ^T laws hold, then we will call λ a *lax functorial distributive law*

If (S, η^S, μ^S) is a monad and the lax η^S and μ^S laws hold, then we call λ a *lax monadic distributive law*.

Of course, a lax distributive law may satisfy some of these laws strictly. There are two additional conditions that are worth considering also.

Definition 4.27. Let λ be a lax distributive law $ST \to TS$.

1. The *oplax composition* condition requires that for every Kleisli morphism $p : A \nrightarrow B$ there is an inclusion:

$$\mu^T_{SB} \circ T\lambda_B \circ \lambda_{TB} \circ STg \leq \mu^T_{SB} \circ T\ \lambda_B \circ TSg \circ \lambda_A$$

2. We say that λ is monotone if whenever $p, q : A \to TB$ are Kleisli morphisms with $p \leq q$ (in $\mathcal{C}_T$) we have

$$\lambda_B \circ Sp \leq \lambda B \circ Sq$$

The oplax composition condition is rather peculiar. It is used to guarantee that the resulting functor $\underline{S}$ will have *strict composition*. Strict composition will always require the strict μ^T law, and in addition strict naturality of λ would be *sufficient* to give strict composition of $\underline{S}$. But there are examples of lax distributive laws λ that induce strictly composing functors and yet do not satisfy strict naturality. This the weakest condition on λ that, in the presence of the strict μ^T law, will induce strict composition.

The following result and proof is a generalisation of a result that is very lucidly presented by Tholen in [53, Proposition 6.4].

The result given by Tholen accounts only for case when $T = \mathcal{P}_Q$ is the monad of a quantale, and the proof uses the dagger structure of the correspond Kleisli category. Happily, it can be modified to work in the general case quite straightforwardly.

Theorem 4.28. *Let S be a functor on $\mathcal{C}$. There is a bijective correspondence between lax monotone distributive laws $\lambda : ST \to TS$ and lax extensions $\underline{S}$ of the functor S.*

Furthermore, additional properties of λ and $\underline{S}$ correspond in the following way.

strict nat. λ		*strict left whiskering*
strict η^T	*strict id.* $\iff$	*strict extension*
strict μ^T and oplax comp.		*strict comp.*

$$\begin{tikzcd} STA \arrow[r, "STf"] \arrow[d, "\lambda_A"'] \arrow[dr, phantom, "\Nearrow"] & STA \arrow[d, "\lambda_B"] \\ TSA \arrow[r, "TSf"'] & TSA \end{tikzcd}$$

lax (nat. λ)

$$\begin{tikzcd} S \arrow[r, "S\eta^T"] \arrow[dr, "\eta^T S"'] & ST \arrow[d, "\lambda"] \arrow[dl, phantom, "\Nearrow"] \\ & TS \end{tikzcd}$$

lax (η^T) law

$$\begin{tikzcd} T \arrow[r, "T\eta^S"] \arrow[dr, "\eta^S T"'] & ST \arrow[d, "\lambda"] \arrow[dl, phantom, "\Nearrow"] \\ & TS \end{tikzcd}$$

lax (η^S) law

$$\begin{tikzcd} STT \arrow[r, "S\mu^T"] \arrow[d, "\lambda T"'] & ST \arrow[dd, "\lambda"] \\ TST \arrow[d, "T\lambda"'] \arrow[ur, phantom, "\Nearrow"] & \\ TTS \arrow[r, "\mu^T S"'] & TS \end{tikzcd}$$

lax (μ^T) law

$$\begin{tikzcd} SST \arrow[r, "\mu^S T"] \arrow[d, "S\lambda"'] & ST \arrow[dd, "\lambda"] \\ STS \arrow[d, "\lambda S"'] \arrow[ur, phantom, "\Nearrow"] & \\ TSS \arrow[r, "T\mu^S"'] & TS \end{tikzcd}$$

lax (μ^S) law

Figure 4.4: The lax conditions

If (S, η^S, μ^S) is a monad then this extends to a correspondence between lax monadic distributive laws and lax monadic extensions in the following way.

lax/strict η^S law	*lax/strict η^S condition*
lax/strict μ^S law	*lax/strict μ^S condition*

Proof. The construction is as detailed in Lemma 4.18.

Note that this correspondence remains bijective even though the proof in Lemma 4.18 uses the fact that $\underline{S}$ has strict composition and strictly extends S to show that $\underline{S}f = \underline{S}(\varepsilon_B) \circ Sf$. In fact, these two steps can be combined into one. It is the right whiskering condition that allows us to go from $\underline{S}(\varepsilon_B \bullet F_T f)$ to $\underline{S}(\varepsilon_B) \bullet F_T(Sf)$.

We begin with a lax distributive law λ, and construct $\underline{S}$ by defining, for $f : A \nrightarrow B$

$$\underline{S}f = \lambda_B \circ Sf$$

We will show that $\underline{S}$ satisfies the strict right whiskering condition, the lax left whiskering condition, the lax identity, and the lax composition conditions, and is monotone. This will make $\underline{S}$ a lax functor that is a lax extension of S.

- (strict right whisk.) We get this for free, by definition of $\underline{S}$ in terms of λ. For any p and g we have

$$\begin{aligned}\underline{S}(p \circ g) &= \lambda \circ S(p \circ g) \\ &= (\lambda \circ Sp) \circ Sg \\ &= \underline{S}p \circ Sg\end{aligned}$$

- (left whisk.) Note that the lax left whiskering condition corresponds to lax naturality of λ. Let $f : A \to B$ and $p : X \nrightarrow A$.

$$\begin{aligned}TSf \circ \underline{S}p &= TSf \circ \lambda \circ Sp \\ &\leq \lambda \circ STf \circ Sp && \text{(lax nat)} \\ &= \underline{S}(Tf \circ p)\end{aligned}$$

  If we have strict naturality of λ, then we may replace the inclusion above with an equality, and hence deduce strict left whiskering.

- (lax id.) This directly corresponds to the lax η^T law. If the η^T law holds strictly then $\underline{S}$ strictly preserves identities. And, by Proposition 4.22, it is a strict extension of $\underline{S}$.

- (lax comp.) Let $p : A \nrightarrow B$ and $q : B \nrightarrow C$.

$$\begin{aligned}\underline{S}q \bullet \underline{S}p &= \mu^T \circ T\underline{S}q \circ \underline{S}p \\ &= \mu^T \circ T\lambda \circ TSp \circ \underline{S}q \\ &\leq \mu^T \circ T\lambda \circ \underline{S}(Tp \circ q) && \text{(lax left whisk.)} \\ &= \mu^T \circ T\lambda \circ \lambda \circ S(Tp \circ q) \\ &\leq \lambda \circ S\mu^T \circ STp \circ Sq && \text{(lax } \mu^T\text{)} \\ &= \underline{S}(\mu^T \circ Tp \circ q) \\ &= \underline{S}(p \bullet q)\end{aligned}$$

- (strict comp.) Again, let $p : A \nrightarrow B$ and $q : B \nrightarrow C$. Suppose we have the strict μ^T law, and the oplax comp condition. As we already have lax composition of $\underline{S}$, it is sufficient to show that $\underline{S}(q \bullet p) \leq \underline{S}q \bullet \underline{S}p$. This follows

from the oplax comp. condition, by taking $g = q$

$$\begin{aligned}
\underline{S}(q \bullet p) &= \underline{S}(\mu^T \circ Tq \circ p) \\
&= (\lambda \circ S\mu^T) \circ STq \circ Sp \\
&= (\mu^T \circ T\lambda \circ \lambda \circ STq) \circ Sp && \text{(strict } \mu^T\text{)} \\
&\leq \mu^T \circ (T\lambda \circ TSq) \circ (\lambda \circ Sp) && \text{(oplax comp.)} \\
&= \mu^T \circ T\underline{S}q \circ \underline{S}p \\
&= \underline{S}q \bullet \underline{S}p
\end{aligned}$$

- (η^S) Suppose now that (S, η^S, μ^S) is a monad, and that the lax η^S and μ^S laws hold. We will show first that the η^S condition holds for $\underline{S}$. Let $p : A \nrightarrow B$ be a Kleisli morphism in $\mathcal{C}_T$. We require the inclusion

$$\begin{aligned}
&F_T\eta^S \bullet p \leq \underline{S}p \bullet F_T\eta^S \text{ or equivalently,} \\
&T\eta^S \circ p \leq \underline{S}p \circ \eta^S
\end{aligned}$$

This follows from the lax η^S law,

$$T\eta^S \leq \lambda \circ \eta^S_T$$

in the following way:

$$\begin{aligned}
T\eta^S \circ p &= T\eta^S \bullet F_T p \\
&\leq (\lambda \circ \eta^S) \bullet F_T p && \text{(lax } \eta^S \text{ law)} \\
&= \lambda \circ (\eta^S \circ p) \\
&= \lambda \circ Sp \circ \eta^S && (\eta^S \text{ nat.}) \\
&= \underline{S}p \circ \eta^S
\end{aligned}$$

Note that if the η^S law holds *strictly* for λ, we get the corresponding strict η^S condition on $\underline{S}$.

- (μ^S). The case of μ^S proceeds similarly. Suppose the lax μ^S law holds:

$$T\mu^S \circ \lambda \circ S\lambda \leq \lambda \circ \mu^S$$

Let $p : A \nrightarrow B$. We need to show that

$$T\mu^S \circ \underline{SS}p \leq \underline{S}p \circ \mu^S.$$

The proof of this inclusion is below.

$$\begin{aligned}
T\mu^S \circ \underline{SS}p &= T\mu^S \circ \lambda \circ S(\lambda \circ Sp) \\
&= T\mu^S \circ \lambda \circ S\lambda \bullet F_T SSp \\
&\leq (\lambda \circ \mu^S) \bullet F_T SSp && \text{(lax } \mu^S \text{ law)} \\
&= \lambda \circ (\mu^S \circ SSp) \\
&= \lambda \circ Sp \circ \mu^S && (\mu^S \text{ nat.}) \\
&= \underline{S}p \circ \mu^S
\end{aligned}$$

And again, we can see that the strict μ^S law induces the strict μ^S condition on $\underline{S}$.

- (monotone) Let $p, q : A \nrightarrow B$ be Kleisli morphisms with $p \leq q$. Monotonicity of λ means that $\lambda \circ Sp \leq \lambda \circ Sq$. But this is exactly

$$\underline{S}p \leq \underline{S}q,$$

which is what we need for $\underline{S}$ to be monotone.

On the other hand, let $\underline{S}$ be a lax extension of S. Define $\lambda_A = \underline{S}(\varepsilon_A)$. We will show that λ is lax natural, and satisfies the lax η^T and μ^T conditions.

- (lax nat.) Let $f : A \to B$. This follows from lax left whiskering, with $p = \varepsilon_A$.

$$\begin{aligned}
TSf \circ \lambda_A &= TSf \circ \underline{S}(\varepsilon_A) \\
&\leq \underline{S}(Tf \circ \varepsilon_A) && \text{(lax left whisk.)} \\
&= \underline{S}(\varepsilon_B \circ Tf) \\
&= \underline{S}(\varepsilon_B) \circ STf && \text{(strict right whisk.)} \\
&= \lambda_B \circ STf
\end{aligned}$$

If we have strict left whiskering, then we get strict naturality of λ.

- (lax η^T) This is clear, by above.

- (lax μ^T) Lax composition gives us $\underline{S}(\varepsilon_A) \bullet \underline{S}(\varepsilon_A) \leq \underline{S}(\varepsilon_A \bullet \varepsilon_A)$. Hence:

$$\begin{aligned}
\mu^T \circ T\lambda_A \circ \lambda_{TA} &= \lambda_A \bullet \lambda \\
&= \underline{S}(\varepsilon) \bullet \underline{S}(\varepsilon) \\
&\leq \underline{S}(\varepsilon \bullet \varepsilon_A) && \text{(lax comp.)} \\
&= \underline{S}(\mu^T) \\
&= \lambda \circ S\mu^T
\end{aligned}$$

  Certainly, strict composition of $\underline{S}$ will lead to the strict μ^T law.

- (η^S) Suppose now that (S, η^S, μ^S) is a monad, and $\underline{S}$ is a *monadic* extension—it satisfies the η^S and μ^S conditions at least laxly. This means that for any morphism $p : A \nrightarrow B$ we have the inclusion

$$T\eta^S \circ p \leq \underline{S}p \circ \eta^S.$$

  By taking $p = \varepsilon_A : TA \nrightarrow A$ we deduce that

$$\begin{aligned}
T\eta^S \circ \varepsilon_A &\leq \underline{S}p \circ \eta^S \\
T\eta^S &\leq \lambda_A \circ \eta^S
\end{aligned}$$

  Of course, if the η^S condition holds *strictly*, then so does the corresponding η^S law for λ.

- (μ^S) The case of μ^S is similar. We apply the μ^S condition taking $p = \varepsilon_A$. Hence

$$\begin{aligned}
T\mu^S \circ (\lambda \circ S\lambda) &= T\mu^S \circ \underline{S}\lambda && \text{(def. } \lambda\text{)} \\
&= T\mu^S \circ \underline{S}\underline{S}\varepsilon_A && \text{(def. } \lambda\text{)} \\
&\leq \underline{S}\varepsilon_A \circ \mu^S && \text{(lax } \mu^S \text{ cond.)} \\
&= \lambda \circ \mu^S. && \text{(def. } \lambda\text{)}
\end{aligned}$$

  If the μ^S condition holds strictly, then this inclusion is an equality.

- By the discussion above, if $\underline{S}$ is monotone then λ will be monotone also.

□

Example 4.29. Recall the Sim monad of Section 3.5. The action of Sim on transition relations is given by the construction $\mathcal{P}^+$, which (in the unlabelled case) lifts a transition relation $\alpha : A \nrightarrow A$ to a transition relation $\mathcal{P}^+(\alpha) : \mathcal{P}A \nrightarrow \mathcal{P}A$.

Transition relations must be homogenous (i.e. endomorphisms), but the construction of $\mathcal{P}^+$ can be generalised non-homogenous relations $R : A \nrightarrow B$ by the rule

$$(U, V) \in \mathcal{P}^+(R) \iff \forall u \in U \ \exists v \in V \ (u, v) \in R. \tag{4.7}$$

We may verify that $\mathcal{P}^+()$ is a lax extension of $\mathcal{P}$ to **Rel** with the following properties:

- lax left whiskering
- lax identity/extension
- strict composition
- lax $\eta^{\mathcal{P}}$ condition
- strict $\mu^{\mathcal{P}}$ condition

The corresponding lax distributive law $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$ is denoted ℓ^+, and given by the equation

$$\ell^+(F) = \{X : \forall U \in F.\, U \cap X \neq \emptyset\}.$$

The distributive law ℓ^+ of Example 4.29, as well as some other laws of type $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$ will be examined in further detail in Chapter 5.

Chapter 5

Lax distributive laws of type $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$

In this chapter we shall examine the particular case of lax distributive laws $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$—which we know are in bijective correspondence to lax functorial extensions of $\mathcal{P}$ to **Rel**—and the two different ways they can be used to define simulations of transition systems. We will show that there are three essential examples of such extensions: $\mathcal{P}^+, \mathcal{P}^-$, and the conjunction $\mathcal{P}^+ \cap \mathcal{P}^-$.

In the first section, we provide some insight to the construction of $\mathcal{P}^+ \cap \mathcal{P}^-$ as the *unique* strict extension of $\mathcal{P}$ to **Rel**. Then we verify that $\mathcal{P}^+$ and $\mathcal{P}^-$ are indeed lax extensions.

The last part of this chapter is a comparison of previous work on extending $\mathcal{P}$ (and other functors) to **Rel** with the techniques of this thesis. Lax extensions of $\mathcal{P}$ have been studied before by Thijs [52] and others, under the name of $\mathcal{P}$-relators. Relators provide a way of constructing simulations of transition systems that is rather different to the Kleisli approach, using the monad of simulations that we saw in Chapter 3. Relators and the Kleisli approach both use the same "ingredient": lax distributive laws of type $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$, but in very different ways.

The case of $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$ is particularly interesting, because the monad $\mathcal{P}$ wears two very distinct hats here. We saw earlier that the powerset monad $\mathcal{P}$ is the *monad of transition* for (unlabelled) *non-deterministic* transition systems. By this, we mean that a non-deterministic transition system (a $\mathcal{P}$-system) is simply a $\mathcal{P}$-coalgebra, a function of type $\alpha : A \to \mathcal{P}A$. But the powerset monad has another role to play, for it is also the *monad of simulation*. This latter statement can be interpreted in two different ways.

- The first interpretation, which is the primary focus of this thesis, is what we shall call the *Kleisli approach*. According to the Kleisli approach, a simulation of $\mathcal{P}$-systems is expressed as a Kleisli morphism for a suitable monad Sim on **TS**. The essential data of Sim is given by a *relation lifting*, that transforms transition relations α on A to relations $\mathcal{P}^+(\alpha)$ on $\mathcal{P}A$. In other words, a (lax) extension of $\mathcal{P}$ to **Rel**, the Kleisli category of the monad of *transition*, $\mathcal{P}$. Hence a simulation from (A, α) to (B, β) consists of a morphism $R : (A, \alpha) \to (\mathcal{P}B, \mathcal{P}^+(\beta))$, which is simply a function $R : A \to \mathcal{P}B$ that satisfies the morphism condition Item 1.

- A second way to capture simulations is to consider relations as the primitive morphisms between transition systems. The simulation condition is again expressed in terms of a well-behaved relation lifting, which is called a *relator*. But rather than lifting the *transition relation*, it is the *simulation relation* that is lifted.

These two ways of arriving at simulations are shown in the diagram below. The diagram on the left depicts the morphism condition for $R : \alpha \to \mathcal{P}^+(\beta)$. Note that the relation lifting $\mathcal{P}^+$ appears vertically, acting on the transition relation. Whereas on the right the action of $\mathcal{P}^+$ is horizontal, acting on the relation of the simulation R.

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Nearrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}^+(\beta)"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}R"'] & \mathcal{P}\mathcal{P}B \end{tikzcd} \quad \Longleftrightarrow \quad \begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Swarrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}\beta"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}^+(R)"'] & \mathcal{P}\mathcal{P}B \end{tikzcd}$$

It is a *marvelous coincidence* that these two notions are the same—for when we seek to generalise from the specific case of forwards simulations of $\mathcal{P}$-systems, we see a divergence.

More specifically, these two perspectives above can be compared across two orthogonal axes of generalisation. On one hand, we wish to vary the *monad of transition* T, and have a robust theory of simulations for T-transition systems.

- On the relator side, this generalisation is well understood. There is a general notion of a T-relator, that corresponds essentially to lax extensions of T to **Rel**, and provides a way of *horizontally* lifting relations $A \nrightarrow B$ to relations on transition sets $TA \nrightarrow TB$.

- In the Kleisli world, generalising to a monad of transition T means that instead of a *relation lifting*, we need a lax extension of $\mathcal{P}$ to the Kleisli category of T. This will let us turn T-transition systems on a set of states A into systems of type $\mathcal{P}A \to T\mathcal{P}A$.

The other sort of generalisation we are interested in is varying the notion of simulation. We would like to additionally express related concepts like cosimulations, reverse simulations, and bisimulations.

1. Different relators directly provide different "simulation types". Thijs [52] demonstrates that "$\mathcal{P}^-$-simulations" are co-simulations, and that simulations with respect to $\mathcal{P}^+ \cap \mathcal{P}^-$ are in fact bisimulations.

2. When it comes to the Kleisli perspective, it takes a bit more work, and we sometimes have to fiddle with the underlying category. Simply replacing $\mathcal{P}^+$ with $\mathcal{P}^-$ gives us the monad of *reverse simulations*, not co-simulations. To encode co-simulations we need to consider the simulation monad of $\mathcal{P}^+$ on the category of transition systems where the morphism condition is reversed. Or equivalently, we can take the dual of the Kleisli category of $\mathcal{P}^+$ on **TS**—this is because a relation R is a co-simulation precisely if the converse relation $R^\dagger$ is a simulation.

In both cases of generalisation the theory of relators is relatively well-understood. In this chapter we provide a brief overview of the literature, essentially following the thesis of Thijs [52]. Expanding on the Kleisli perspective is a novel contribution of this thesis. Only the specific case of $\mathcal{P}^+$ providing simulations can be found in the literature [39].

5.1 Extending functors to Rel

The category **Rel** is very special, and relations $R : A \nrightarrow B$ can be regarded in several different ways. The first, and perhaps most "concrete" way is to consider R as a set of ordered pairs $R \subseteq A \times B$. In this accounting, the identity relation on A is given by the set $\{(a, a) : a \in A\}$, and the composite $R \fatsemi S$ (where $S : B \nrightarrow C$) is defined as

$$\{(a, c) : \exists b \in B.\ \ (a, b) \in R \text{ and } (b, c) \in S\}\,.$$

A slightly more abstract perspective is to consider the set of ordered pairs R (also known as the *graph* of R) as a span from A to B, with components given by the projections onto the domain (d), and the codomain (c).

$$\begin{tikzcd} & R \arrow[dl, "d"'] \arrow[dr, "c"] & \\ A & & B \end{tikzcd}$$

In this way, we can think of *any* span as encoding a relation. Let $(C, f : C \to A, g : C \to B)$ be a span $A \to B$.

$$\begin{tikzcd} & C \arrow[dl, "f"'] \arrow[dr, "g"] & \\ A & & B \end{tikzcd}$$

The corresponding relation R is given by the rule

$$\begin{aligned} (a, b) \in R \iff \exists c \in C.\ & f(c) = a \text{ and} \\ & g(c) = b. \end{aligned}$$

Intuitively, the set C is an abstract set of "arrows", and the morphisms f and g assign a "source" and a "target" to each arrow. The same relation can be represented by more than one span. We call such a representation a *tabulation* of the relation R. Note that the identity relation may be tabulated by the trivial span below.

$$\begin{tikzcd} & A \arrow[dl, equal] \arrow[dr, equal] & \\ A & & A \end{tikzcd}$$

An alternative perspective on relations is the Kleisli construction. We know that **Rel** is the Kleisli category of the powerset functor $\mathcal{P}$ on **Set**. Hence a relation R is identified with a Kleisli morphism $r : A \to \mathcal{P}(B)$, with

$$(a, b) \in R \iff b \in r(a).$$

We will tend to elide this distinction, writing both $(a, b) \in R$ and $b \in R(a)$, depending on which form is more convenient.

One benefit of viewing relations as spans is that the symmetry becomes rapidly apparent. There is no direction inherent in a span, any relation $R : A \nrightarrow B$ has a *converse* relation $R^\dagger : B \nrightarrow A$.

$$\begin{tikzcd} & C \arrow[dl, "f"'] \arrow[dr, "g"] & \\ A & & B \end{tikzcd} \qquad \begin{tikzcd} & C \arrow[dl, "g"'] \arrow[dr, "f"] & \\ B & & A \end{tikzcd}$$

Figure 5.1: The same relation?

Thus **Rel** has the structure of a dagger category. It is admittedly difficult to arrive at this conclusion from the Kleisli construction.

Recall that the free functor $F_{\mathcal{P}} : \mathbf{Set} \to \mathbf{Rel}$ sends a function $f : A \to B$ to the relation $F_{\mathcal{P}} f = \eta^{\mathcal{P}}_B \circ f : A \nrightarrow B$. Note that this relation can be tabulated as the following *semi-trivial* span:

$$\begin{tikzcd} & A \arrow[dl, equal] \arrow[dr, "f"] & \\ A & & B \end{tikzcd}$$

We will introduce f_* as an alternate notation for $F_{\mathcal{P}} f$, which is defined by the rule

$$(a, b) \in f_* \iff b = f(a).$$

There is another way to embed f into **Rel**, that is dual to the free functor. We will write f^* for the *converse* relation $B \nrightarrow A$ given by

$$(b, a) \in f^* \iff b = f(a).$$

This corresponds to the opposite tabulation.

$$\begin{tikzcd} & A \arrow[dl, "f"'] \arrow[dr, equal] & \\ B & & A \end{tikzcd}$$

These two embeddings of a **Set**-morphism f are related in the following important way, that makes use of the **Pos**-enriched structure of **Rel**.

Proposition 5.1 ([32, 47, Def. 4.2.1])**.** *Let $f : A \to B$ be a function. There is an adjunction $f_* \dashv f^*$. Concretely, this means that*

$$\eta^{\mathcal{P}}_A \subseteq f^* \bullet f_* \text{ and} \tag{5.1}$$

$$f_* \bullet f^* \subseteq \eta^{\mathcal{P}}_B. \tag{5.2}$$

Moreover, these are the only *adjunctions in* ***Rel****. If a relation $R : A \nrightarrow B$ is a left adjoint (with $R \dashv S$) then it must be of the form $R = f_*$ for some function $f : A \to B$, and furthermore $S = f^*$.*

We will also note that the relation corresponding to a span (C, f, g) can be expressed via the composition

$$g_* \bullet f^*.$$

The fact that morphisms of **Rel** can be tabulated in this way provides a significant restriction to the structure of extensions of a functor $S : \mathbf{Set} \to \mathbf{Set}$ to **Rel**. The following construction is due to Barr ([5]), but the result is most lucidly stated (without proof) by Garner in [23].

Lemma 5.2. *Let S be a functor on* ***Set****. There is at most one strict extension of S to a strict functor $\hat{S}$ on* ***Rel****, which exists precisely when S preserves weak pullbacks.*

A partial proof—that $\hat{S}$ preserving composition corresponds to S preserving weak pullbacks—can be found in [32]. We will not spend too much time on this part. However we will elaborate on the uniqueness of $\hat{S}$. Garner hints that the key is the adjunction between f_* and f^*. We will examine this in some more detail.

Proof. The extension $\hat{S}$ is defined in the following way. Let $R : A \nrightarrow B$ be a relation, tabulated as the span below.

$$\begin{tikzcd} & R \arrow[dl, "d"'] \arrow[dr, "c"] & \\ A & & B \end{tikzcd}$$

The relation $\hat{S}(R)$ is constructed by applying S to the tabulation of R. This will encode a relation $SA \nrightarrow SB$.

$$\begin{tikzcd} & SR \arrow[dl, "Sd"'] \arrow[dr, "Sc"] & \\ SA & & SB \end{tikzcd}$$

Explicitly, we can express $\hat{S}(R)$ as the composition

$$\hat{S}(R) = (Sc)_* \bullet (Sd)^*.$$

Thus we can compute

$$(X, Y) \in \hat{S}(R) \iff \exists Z \in SR. \begin{cases} Sd(Z) = X \\ Sc(Z) = Y. \end{cases} \tag{5.3}$$

Now, we have claimed that $\hat{S}$ is a monotone strict functor on **Rel** that strictly extends S. Barr shows in [5] that for *any functor* S the resulting $\hat{S}$ is monotone, preservies identities strictly, and composition *op-laxly*. That is, for composable relations Q, R we have

$$\hat{S}(Q \bullet R) \subseteq \hat{S}Q \bullet \hat{S}R.$$

Barr also shows that if S preserves weak pullbacks then $\hat{S}$ does indeed satisfy lax composition. The converse of this result is proven in more detail in [32].

It is straightforward to verify that $\hat{S}$ is indeed a strict extension of S. The strict extension condition corresponds to the equality

$$\hat{S}(f_*) = (Sf)_*,$$

which clearly holds by construction.

We now turn our attention to the uniqueness of $\hat{S}$. We will first show that $\hat{S}(f_*) = (Sf)_*$ is left adjoint to $\hat{S}(f^*)$. This will mean that $\hat{S}(f^*)$ must be equal to $(Sf)^*$, by Proposition 5.1. We derive the required inclusions in the following way.

$$\begin{aligned}
\eta^{\mathcal{P}}_{SA} &= \hat{S}(\eta^{\mathcal{P}}_{A}) && \text{(strict id. } \hat{S}) \\
&\subseteq \hat{S}(f^* \bullet f_*) && (f_* \dashv f^*,\ \hat{S} \text{ mono.}) \\
&= \hat{S}(f^*) \bullet \hat{S}(f_*) && \text{(strict comp. } \hat{S}) \\
\hat{S}(f_*) \bullet \hat{S}(f^*) &= \hat{S}(f_* \bullet f^*) && \text{(strict comp. } \hat{S}) \\
&\subseteq \hat{S}(\eta^{\mathcal{P}}_{B}) && (f_* \dashv f^*,\ \hat{S} \text{ mono.}) \\
&= \eta^{\mathcal{P}}_{SB} && \text{(strict id. } \hat{S})
\end{aligned}$$

So, suppose that $\underline{S}$ is a strict functor on **Rel** that also strictly extends S. This means that $\hat{S}$ and $\underline{S}$ agree on left adjoints: they map f_* to $(Sf)_*$. The argument above shows that $\hat{S}$ and $\underline{S}$ must also agree on *right* adjoints—we have $\hat{S}(f^*) = \underline{S}(f^*) = (Sf)^*$.

Since every relation R can be expressed as the composition of a right and left adjoint, this is enough to guarantee that $\underline{S}$ and $\hat{S}$ have the same action on *all* relations. Let R be tabulated as $R = c_* \bullet d^*$. Then we have

$$\begin{aligned} \underline{S}(R) &= \underline{S}(c_*) \bullet \underline{S}(d^*) && \text{(strict comp. } \underline{S}) \\ &= \hat{S}(c_*) \bullet \hat{S}(d^*) \\ &= \hat{S}(R). && \text{(strict comp. } \hat{S}) \end{aligned}$$

So $\hat{S}$ is in fact the unique, strict monotone extension of S. □

Example 5.3. When $S = \mathcal{P}$, we calculate (by Eq. 5.3) that

$$(U, V) \in \hat{\mathcal{P}}(R) \iff \exists C \subseteq R. \begin{cases} \mathcal{P}(d)(C) = U \\ \mathcal{P}(c)(C) = V \end{cases} \tag{5.4}$$

An intuitive interpretation is this. Think of $R \subseteq A \times B$ as a set of arrows, each with a source in A and a target in B (provided by the maps $d : R \to A$ and $c : R \to B$). The relation $\hat{\mathcal{P}}(R)$ is formed by taking all possible subsets of arrows $C \subseteq R$. Every such subset witnesses a relation between the corresponding set of sources and the set of targets.

Note that condition 5.4 can also be expressed as

$$(U, V) \in \hat{\mathcal{P}}(R) \iff \begin{cases} \forall u \in U.\ \exists v \in V.\ (u, v) \in R & (5.7) \\ \forall v \in V.\ \exists u \in U.\ (u, v) \in R, & (5.8) \end{cases}$$

or equivalently, considering R as a function $A \to \mathcal{P}B$,

$$(U, V) \in \hat{\mathcal{P}}(R) \iff \begin{cases} \forall u \in U.\ R(u) \cap V \neq \emptyset & (5.11) \\ V \subseteq \bigcup \mathcal{P}(R)(U). & (5.12) \end{cases}$$

In particular, the unique extension of $\mathcal{P}$ provides the action of the double simulation monad on transition relations. We have

$$\mathrm{DSim}(A, R_\sigma) = (\mathcal{P}A, \hat{\mathcal{P}}(R_\sigma)).$$

5.2 Two lax distributive laws

There is a connection between distributive laws of type $\mathcal{PP} \to \mathcal{PP}$ and the monads of simulation that we saw in Chapter 4. Recall that such a monad consists of an action on states A, on morphisms (certain functions of sets $A \to B$), and on transition *relations* $A \nrightarrow A$.

The novel contribution is to realise that λ can be expressed as the conjunction of two *lax* distributive laws:

$$\lambda = \ell^+ \cap \ell^-$$

The extension corresponding to the ℓ^+ law is the monad of simulations Sim, while ℓ^- encodes the reverse simulation monad.

The construction of $\hat{\mathcal{P}}$ in Example 5.3 induces a corresponding law $\lambda : \mathcal{PP} \to \mathcal{PP}$ given by $\lambda_A = \hat{\mathcal{P}}(\varepsilon_A)$. As $\hat{\mathcal{P}}$ is a strict functor on **Rel** that strictly extends $\mathcal{P}$, by Theorem 4.28, the law λ is what we would call a strict functorial distributive law. Furthermore, it can be shown that λ satisfies the $\mu^{\mathcal{P}}$ condition strictly, and the $\eta^{\mathcal{P}}$ condition laxly. This means that it is not a strict monad distributive law. It is however, what Garner and Goy call a (monotone) "weak" distributive law [23, 25]

The following expression for λ can be found in [25].

$$\lambda_A(F) = \left\{ X : \forall U \in F.\ U \cap X \neq \emptyset \text{ and } X \subseteq \bigcup F \right\}$$

The conjunction is very suggestive. One might wonder what would happen if we were to split this law into two components, ℓ^+ and ℓ^-.

$$\begin{aligned} {\ell^+}_A(F) &= \{X : \forall U \in F.\ \exists x \in X.\ \ x \in U\} && (\ell^+ \text{ law}) \\ &= \{X : \forall U \in F.\ \ X \cap U \neq \emptyset\} \\ {\ell^-}_A(F) &= \{X : \forall x \in X.\ \exists U \in F.\ \ x \in U\} && (\ell^- \text{ law}) \\ &= \left\{ X : X \subseteq \bigcup F \right\} \end{aligned}$$

These each have the right type of a distributive law $\mathcal{PP} \to \mathcal{PP}$, and hence the corresponding extensions are defined in the following way. Let $R : A \nrightarrow B$ be

a relation. Then we set

$$\mathcal{P}^+(R) = \ell^+{}_B \circ \mathcal{P}(R)$$
$$\mathcal{P}^-(R) = \ell^-{}_B \circ \mathcal{P}(R).$$

We can compute this lifted relation explicitly. For $U \subseteq A$ and $V \subseteq B$ we have

$$(U, V) \in \mathcal{P}^+(R) \iff \forall u \in U.\ \exists v \in V.\ (u, v) \in R$$
$$(U, V) \in \mathcal{P}^-(R) \iff \forall v \in V.\ \exists u \in U.\ (u, v) \in R.$$

Equivalently, we can consider $\mathcal{P}^+(R)$ and $\mathcal{P}^-(R)$ as Kleisli morphisms $\mathcal{P}A \to \mathcal{P}\mathcal{P}B$, defined by

$$\begin{aligned}
\mathcal{P}^+(R)(U) &= \{V : \forall u \in U.\ \exists v \in V.\ (u, v) \in R\} \\
&= \{V : \forall u \in U.\ R(u) \cap V \neq \emptyset\} \\
\mathcal{P}^-(R)(U) &= \{V : \forall v \in V.\ \exists u \in U.\ \ v \in R(u)\} \\
&= \left\{V : V \subseteq \bigcup \mathcal{P}(R)(U)\right\}
\end{aligned}$$

We will show that $\mathcal{P}^+$ and $\mathcal{P}^-$ are indeed lax extensions of $\mathcal{P}$ to **Rel**. Hence ℓ^+ and ℓ^- will be lax distributive laws by Theorem 4.28.

Proposition 5.4. *The mappings $\mathcal{P}^+$ and $\mathcal{P}^-$ are lax extensions of $\mathcal{P}$ to **Rel**.*

Proof. We need to show that both $\mathcal{P}^+$ and $\mathcal{P}^-$ are monotone, laxly preserve identities and composition (in fact, they both preserve composition *strictly*), and are lax left whiskering.

1. (mono.) Suppose that $R \subseteq S : A \nrightarrow B$. We need to show that if $(U, V) \in \mathcal{P}^+(R)$ then $(U, V) \in \mathcal{P}^+(S)$ also (and likewise for $\mathcal{P}^-$.) So suppose that $(U, V) \in \mathcal{P}^+(R)$. This means that for all $u \in U$ we can find a v in V with $(u, v) \in R$. Since $R \subseteq S$, we also have $(u, v) \in S$. This means that $(U, V) \in \mathcal{P}^+(S)$, as desired. The case of $\mathcal{P}^-$ is symmetric.

2. (lax id.) We need to show that $\eta^{\mathcal{P}}_{\mathcal{P}A} \subseteq \mathcal{P}^+(\eta^{\mathcal{P}}_A)$, and likewise for $\mathcal{P}^-$. We will evaluate $\mathcal{P}^+(\eta^{\mathcal{P}}_A)$ and $\mathcal{P}^-(\eta^{\mathcal{P}}_A)$.

   Let $U, V \subseteq A$. When is $(U, V) \in \mathcal{P}^+(\eta^{\mathcal{P}}_A)$? Precisely when for all $u \in U$ we

can find a $v \in V$ with

$$(u, v) \in \eta_A^{\mathcal{P}}.$$

As $\eta_A^{\mathcal{P}}$ is the identity relation on A, this happens only when $v = u$. So we see that $(U, V) \in \mathcal{P}^+(\eta_A^{\mathcal{P}})$ when every $u \in U$ is also contained in V. In other words,

$$\mathcal{P}^+(\eta_A^{\mathcal{P}})(U) = \{V : U \subseteq V\}.$$

The case of $\mathcal{P}^-$ is symmetric. We require that every $v \in V$ is contained in u in order to have $(U, V) \in \mathcal{P}^-(\eta_A^{\mathcal{P}})$. Hence we can deduce that

$$\mathcal{P}^-(\eta_A^{\mathcal{P}})(U) = \{V : U \supseteq V\}.$$

Naturally we have $\eta_{\mathcal{P}A}^{\mathcal{P}}(U) = \{U\}$, which is a subset of both $\mathcal{P}^+(\eta_A^{\mathcal{P}})(U)$ and $\mathcal{P}^-(\eta_A^{\mathcal{P}})(U)$. So it follows that the lax identity law holds for $\mathcal{P}^+$ and $\mathcal{P}^-$.

3. (strict comp.) Let $R : A \nrightarrow B$ and $S : B \nrightarrow C$ be composable relations. We will first show that lax composition holds for both $\mathcal{P}^+$ and $\mathcal{P}^-$. This means that

$$\mathcal{P}^+(R) \fatsemi \mathcal{P}^+(S) \subseteq \mathcal{P}^+(R \fatsemi S) \text{ and}$$
$$\mathcal{P}^-(R) \fatsemi \mathcal{P}^-(S) \subseteq \mathcal{P}^-(R \fatsemi S)$$

So suppose that $(U, V) \in \mathcal{P}^+(R) \fatsemi \mathcal{P}^+(S)$. By definition of relational composition there must be an intermediate set $W \subseteq B$ with $(U, W) \in \mathcal{P}^+(R)$ and $(W, V) \in \mathcal{P}^+(S)$. We will use the elements of W to construct witnesses of $(U, V) \in \mathcal{P}^+(R \fatsemi S)$.

We need to show that for every $u \in U$ there is a $v \in V$ with $(u, v) \in R \fatsemi S$. Since we have $(U, W) \in \mathcal{P}^+(R)$ we can find a w with $(u, w) \in R$. Then, we can start at $w \in W$ and use the fact that $(W, V) \in \mathcal{P}^+(S)$ to find a v with $(w, v) \in S$. Putting this together we get $(u, v) \in R \fatsemi S$, as desired. The case of $\mathcal{P}^-$ is symmetric.

Now, we will show that the opposite inclusion also exists. So suppose that $(U, V) \in \mathcal{P}^+(R \fatsemi S)$. We need to find an intermediate set W to witness $(U, V) \in \mathcal{P}^+(R) \fatsemi \mathcal{P}^+(S)$.

We will construct W in the following way. For every $u \in U$, we know that there is at least one v with $(u, v) \in R \fatsemi S$. This means there must be a mediating $w \in B$ with $(u, w) \in R$ and $(w, v) \in S$. We will take W to be the set of all such w. By construction, we will have $(U, W) \in \mathcal{P}^+(R)$, because every u corresponds to a mediating w. We also have $(W, V) \in \mathcal{P}^+(S)$, because a mediating w is, by definition, related to something in V. Again, the case of $\mathcal{P}^-$ is symmetric, and not substantially different.

4. (lax left whisk.) Let $R : A \nrightarrow B$ be a relation, and $g : B \to Y$ a function. We need to show that

$$\mathcal{P}(\mathcal{P}g) \circ \mathcal{P}^+(R) \subseteq \mathcal{P}^+(\mathcal{P}g \circ R) \text{ and}$$
$$\mathcal{P}(\mathcal{P}g) \circ \mathcal{P}^-(R) \subseteq \mathcal{P}^-(\mathcal{P}g \circ R).$$

Let U be a subset of A, and suppose that $W \subseteq Y$ is in $\mathcal{P}(\mathcal{P}g)(\mathcal{P}^+(R)(U))$. This means that $W = \mathcal{P}g(V)$ for some $V \in \mathcal{P}^+(R)(U)$. We need to show that $W \in \mathcal{P}^+(\mathcal{P}g \circ R)(U)$ also.

Since $V \in \mathcal{P}^+(R)(U)$ we deduce that for all $u \in U$ there is a $v \in V$ with $v \in R(u)$. By taking $w = g(v) \in W$, we can find, for every $u \in U$, an element $w \in W$ with $w \in \mathcal{P}g(R(u))$. This proves that $W \in \mathcal{P}^+(\mathcal{P}g \circ R)(U)$, as desired.

The case of $\mathcal{P}^-$ is similar. It is sufficient to show that for any $V \in \mathcal{P}^-(R)(U)$, we have $\mathcal{P}g(V) \in \mathcal{P}^-(\mathcal{P}g \circ R)(U)$. Let $g(v) \in \mathcal{P}g(V)$, so that $v \in V$. Hence we can find a $u \in U$ with $v \in R(u)$. But this implies that $g(v) \in \mathcal{P}g(R(u))$, so it follows that $\mathcal{P}g(V) \in \mathcal{P}^-(\mathcal{P}g \circ R)(U)$, as desired.

□

Note that by Proposition 4.22, the fact that both $\mathcal{P}^+$ and $\mathcal{P}^-$ satisfy the identity condition only *laxly* implies that they cannot be strict extensions.

5.3 Relators and lax extensions to Rel

An alternative approach to lax extension of functors to **Rel** is that of relation liftings, or "relators". In particular, relators can be used to provide a method for expressing different notions of simulation of diverse transition systems of general type T.

A pivotal text is the thesis of Thijs [52]. Relators have also been studied significantly by Levy et al [37], and are well-treated in the survey [32]. In this section we shall provide a brief overview of relators and how they encode simulations, and a comparison to our lax extensions of Chapter 4.

The idea goes like this. Suppose $\alpha : A \to \mathcal{P}A$ and $\beta : B \to \mathcal{P}B$ are two non-deterministic transition systems. A simulation from α to β consists of a *relation* of states, $R : A \to B$, that satisfies a certain condition.

But what about the general case, where rather than $\mathcal{P}$ we take $T : \mathbf{Set} \to \mathbf{Set}$ to be any (mere) *functor of transitions*, for example, $T = \mathcal{D}$, or any of the other possibilities described in Section 4.1? Transition systems of type T can be modelled as T-coalgebras, $\alpha : A \to TA$. A T-relator is essentially a well-behaved (we shall explain precisely what this means shortly) way of turning a relation $R : A \to B$ of base sets, to a relation $\underline{T}R : TA \to TB$. Given a T-relator $\underline{T}$, we are able to form the diagram below.

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha_*"'] & B \arrow[d, "\beta_*"] \\ TA \arrow[r, "\underline{T}R"] & TB \end{tikzcd}$$

Note that this diagram lives in **Rel**—the arrows designate relations, and the composition is relational composition. The relation R is considered a simulation—*with respect to the relator* $\underline{T}$—if there is an op-lax inclusion of

$$\beta_* \bullet R \subseteq \hat{T}R \bullet \alpha_*, \tag{5.13}$$

although this is usually formulated in a slightly different way. This is represented in Fig. 5.2.

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha_*"'] & B \arrow[d, "\beta_*"] \\ TA \arrow[r, "\underline{T}R"] \arrow[ur, phantom, "\Swarrow"] & TB \end{tikzcd}$$

Figure 5.2: A $\underline{T}$-simulation

Hence any choice of relator $\underline{T}$ induces a notion of $\underline{T}$-simulation of T-transition systems. When we take $T = \mathcal{P}$, it turns out that $\mathcal{P}^+, \mathcal{P}^-$, and the conjunction $\mathcal{P}^+ \cap \mathcal{P}^- = \hat{\mathcal{P}}$ are all $\mathcal{P}$-relators. In particular, $\mathcal{P}^+$ encodes the standard notion

of simulations. Whereas $\mathcal{P}^-$-simulations correspond to co-simulations, and $\hat{\mathcal{P}}$-simulations are *bisimulations*.

A significant list is presented in [37], including

Definition 5.5. Let $R : A \nrightarrow B$ be a relation, and $f : Z \to A$ and $g : W \to B$ functions. The *inverse image* is a relation $Z \nrightarrow W$, denoted as $(f, g)^{-1}R$ and defined by the rule

$$(z, w) \in (f, g)^{-1}R \iff (f(z), g(w)) \in R \tag{5.14}$$

We can also form the direct image of a relation. Let f and g be as above, and Q a relation $Z \nrightarrow W$. Then $(f \times g)[Q]$ is a relation $A \nrightarrow B$ defined by

$$(f \times g)[Q] = \{(f(z), g(w)) : (z, w) \in Q\} \tag{5.15}$$

We could also write $(f \times g)[Q]$ as $\mathcal{P}(f \times g)(Q)$, since Q is a subset of $Z \times W$. But the former notation is in line with [52].

Remark 5.6. We may rewrite the expressions above, in line with the notation of the previous sections. Let f, g, R, Q be as in Eq. 5.14. Then we have that

$$(f, g)^{-1}R = g^* \bullet R \bullet f_*$$
$$(f \times g)[Q] = g_* \bullet Q \bullet f^*$$

The following definition is slightly adapted from Thijs in [52].

Definition 5.7 ([52])**.** Let T be a functor on **Set**. An *T-relator* is a mapping $\underline{T}$ of relations $A \nrightarrow B$ to relations $TA \nrightarrow TB$ such that

1. If $Q \subseteq R : A \nrightarrow B$ then $\underline{T}(Q) \subseteq \underline{T}(R)$
2. For every set A we have $\eta^{\mathcal{P}}_{TA} \subseteq \underline{T}(\eta^{\mathcal{P}}_A)$
3. For every $Q : A \nrightarrow B$ and $R : B \nrightarrow C$ we have $\underline{T}R \bullet \underline{T}Q \subseteq \underline{T}(R \bullet Q)$
4. For every relation $Q : Z \nrightarrow W$ and functions $f : Z \to A$ and $g : W \to B$ we have

$$(Tf \times Tg)[\underline{T}Q] \subseteq \underline{T}(f \times g)[Q] \tag{5.16}$$

Note that Thijs uses "monotone relators" to refer to relators that satisfy the conditions above, and in addition preserve composition *strictly*. We shall instead consider the more general case.

The fourth condition is a sort of "coherence condition" that expresses some interaction between the relator $\underline{T}$ and the original functor T. Sometimes it is helpful to express this coherence condition in terms of the inverse image. This fact is stated in [32]. For convenience, we provide some more detail.

Proposition 5.8. *Let T be a functor on* ***Set****, and $\underline{T}$ a lax functor on* ***Rel****. Let $f : Z \to A$ and $g : W \to B$ be functions. Then the following are equivalent:*

1. *For every relation $Q : Z \nrightarrow W$ we have*

$$(Tf \times Tg)[\underline{T}Q] \subseteq \underline{T}\left((f \times g)[Q]\right)$$

2. *For every relation $R : A \nrightarrow B$ we have*

$$\underline{T}((f,g)^{-1}R) \subseteq (Tf, Tg)^{-1}\underline{T}(R)$$

Proof. We will find the alternate expression of Remark 5.6 more useful. The conditions above become

$$(Tf)_* \bullet \underline{T}Q \bullet (Tg)^* \subseteq \underline{T}(f_* \bullet Q \bullet g^*) \tag{5.17}$$

$$\underline{T}(f^* \bullet R \bullet g_*) \subseteq (Tf)^* \bullet \underline{T}R \bullet (Tg)_* \tag{5.18}$$

We will first show $1 \implies 2$. Let $R : A \nrightarrow B$ be a relation. Take $Q = f^* \bullet R \bullet g_*$. Then we see that

$$\begin{aligned}
\underline{T}Q &\subseteq (Tf)^* \bullet (Tf)_* \bullet \underline{T}Q \bullet (Tg)^* \bullet (Tg)_* && \text{(Eq. 5.1)}\\
&\subseteq (Tf)^* \bullet \underline{T}(f_* \bullet Q \bullet g^*) \bullet (Tg)_* && \text{(Eq. 5.17)}\\
&= (Tf)^* \bullet \underline{T}(f_* \bullet f^* \bullet R \bullet g_* \bullet g^*) \bullet (Tg)_* \\
&\subseteq (Tf)^* \bullet \underline{T}R \bullet (Tg)_* && \text{(Eq. 5.2 and } \underline{T} \text{ mono.)}
\end{aligned}$$

which is the desired inclusion. The case of $2 \implies 1$ is symmetric. □

The second form of the "coherence condition" is used in [37]. .

Now, from the expressions of Eqs. 5.17 and 5.18 we can deduce that every T-relator corresponds to a lax extension of T.

Proposition 5.9. *Let T be a functor on* ***Set****, and $\underline{T}$ an T-relator. Then $\underline{T}$ is also a lax extension of T.*

Proof. The first three conditions of Definition 5.7 tell us that $\underline{T}$ is indeed a lax functor on **Rel**—it laxly preserves identities and composition, and is monotone.

Now, from Eq. 5.17 we may deduce (by taking $g = \mathrm{id}$) that

$$(Tf)_* \bullet \underline{T}Q \subseteq \underline{T}(f_* \bullet Q),$$

which is the lax left whiskering condition. By Proposition 4.22 this implies the lax right whiskering condition also, as we are in the presence of lax identity and lax composition rules.

Similarly, Eq. 5.18 with $f = \mathrm{id}$ gives us

$$\underline{T}(R \bullet g_*) \subseteq \underline{T}R \bullet (Tg)_*,$$

the *op-lax* right whiskering condition. As we also have lax right whiskering, we may deduce that right whiskering holds strictly, and so $\underline{T}$ is indeed a lax extension of T. □

Note that if Eq. 5.17 held *strictly* then we could deduce the *strict* left whiskering condition.

Levy defines relators ([37]) as satisfying Eq. 5.18 strictly. This does *not* imply the strict version of Eq. 5.17.

There is a partial converse to Proposition 5.9.

Proposition 5.10. *If $\underline{T}$ is a lax extension of T that also satisfies*

$$(Tg)^* \subseteq \underline{T}(g^*)$$

then $\underline{T}$ is a relator.

Proof. It is necessary to show that the inclusion of Eq. 5.17 holds. Hence we see that

$$\begin{aligned}
(Tf)_* \bullet \underline{T}Q \bullet (Tg)^* &\subseteq \underline{T}(f_*) \bullet \underline{T}Q \bullet \underline{T}(g^*) && \text{(assumption and lax ext. } \underline{T}\text{)} \\
&\subseteq \underline{T}(f_* \bullet Q \bullet g^*) && \text{(lax comp. } \underline{T}\text{)}
\end{aligned}$$

which is the desired result. □

An important result of Thijs is a structure/correspondence theorem for relators. We give a brief overview. First, it is demonstrated that if T is a functor that preserves weak pullbacks, then the constructed $\hat{T}$ of Lemma 5.2 is a relator. Thijs calls this the *minimal T-relator*, because every T-relator $\underline{T}$ must contain the minimal relator as a subset:

$$\hat{T}(R) \subseteq \underline{T}(R) \text{ for every relation } R : A \nrightarrow B.$$

Further, Thijs shows that every T-relator factors through the minimal relator in the following way:

$$\underline{T}(R) = \underline{T}(\eta_B^{\mathcal{P}}) \bullet \hat{T}(R) \bullet \underline{T}(\eta_A^{\mathcal{P}}) \text{ for every relation } R : A \nrightarrow B. \tag{5.19}$$

Moreover, the relation $\underline{T}(\eta_B^{\mathcal{P}}) : TB \nrightarrow TB$ is always a pre-order on TB, and every family of pre-orders defines a relator. Whereas we have shown that there is a correspondence between extensions $\underline{T}$ and the distributive laws given by $\underline{T}(\varepsilon)$.

Example 5.11. When $T = \mathcal{P}$, we see that T-coalgebras are essentially (unlabelled) non-deterministic transition systems. The minimal relator is $\hat{\mathcal{P}}$, as detailed in Example 5.3.

Note that $\mathcal{P}^+$ and $\mathcal{P}^-$ are lax extensions that further satisfy the inclusions

$$(\mathcal{P}g)^* \subseteq \mathcal{P}^+(()g^*) \text{ and}$$
$$(\mathcal{P}g)^* \subseteq \mathcal{P}^-(()g^*).$$

Hence by Proposition 5.10 they must also be $\mathcal{P}$-relators. Thijs demonstrates in [52] that the corresponding preorders on $\mathcal{P}A$ are $\subseteq$ and $\supseteq$.

That means we have

$$\mathcal{P}^+(R) = \subseteq \bullet \hat{\mathcal{P}}(R) \bullet \subseteq \tag{5.20}$$
$$\mathcal{P}^-(R) = \supseteq \bullet \hat{\mathcal{P}}(R) \bullet \supseteq \tag{5.21}$$

The following definition is essentially due to Thijs.

Definition 5.12. Let T be a functor on **Set**, and let $\alpha : A \to TA$ and $\beta : B \to TB$ be T-coalgebras. Let $\underline{T}$ be an T-relator.

A relation $R : A \nrightarrow B$ is called an $\underline{T}$-*simulation* if the following inclusion holds:

$$(\alpha \times \beta)[R] \subseteq \underline{T}R, \text{ or equivalently,} \tag{5.22}$$

$$R \subseteq (\alpha, \beta)^{-1}(\underline{T}R) \tag{5.23}$$

Now, recalling that, by Lemma 4.17, we can rewrite the inclusion of Eq. 5.13 as

$$\underline{T}R \circ f \subseteq \mathcal{P}\beta \circ R. \tag{5.24}$$

Thus we get the diagram of Fig. 5.3 below, which consists of functions and composition in the category **Set**.

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Swarrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}\beta"] \\ TA \arrow[r, "\underline{T}R"'] & \mathcal{P}TB \end{tikzcd}$$

Figure 5.3: A $\underline{T}$-simulation, expressed in **Set**

Example 5.13. Let us examine the particular case of standard simulations of $\mathcal{P}$-systems. Let $\alpha : A \to \mathcal{P}A$ and $\beta : B \to \mathcal{P}B$ be $\mathcal{P}$-coalgebras. In [52] it is shown that a relation $R : A \nrightarrow B$ is a simulation (in the sense of Chapter 3) if and only if it is a $\mathcal{P}^+$-simulation (in the sense of Definition 5.12).

Furthermore, the $\mathcal{P}^-$-simulations are precisely the *co-simulations*, and the simulations with respect to the minimal relator $\hat{\mathcal{P}}$ are *bisimulations*

On the other hand, we also know that simulations are Kleisli morphisms of the simulation monad $\mathcal{P}^+$ on **TS**. The action of $\mathcal{P}^+$ is to send a transition system $\beta : B \to \mathcal{P}B$ on B to the system on $\mathcal{P}B$ with the "lifted" transition relation:

$$\mathcal{P}^+(\beta) : \mathcal{P}B \to \mathcal{P}\mathcal{P}B.$$

Recall that R (considered as a *function* $A \to \mathcal{P}B$) is a morphism precisely if it preserves transitions. This can be expressed as the lax inclusion of

$$\mathcal{P}R \bullet \alpha \subseteq \mathcal{P}^+(\beta) \bullet R,$$

or represented diagramatically as below. The diagram of Fig. 5.4 depicts these two equivalent characterisations of simulations.

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Nearrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}^+(\beta)"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}R"'] & \mathcal{P}\mathcal{P}B \end{tikzcd} \quad \Longleftrightarrow \quad \begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Nwarrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}\beta"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}^+(R)"'] & \mathcal{P}\mathcal{P}B \end{tikzcd}$$

Figure 5.4: Two equivalent ways of expressing the fact that R is a simulation

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Nearrow"] & B \arrow[d, "\beta"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}f"'] & \mathcal{P}B \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Nearrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}^+(\beta)"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}R"'] & \mathcal{P}\mathcal{P}B \end{tikzcd}$$

Figure 5.5: A simulation $\alpha \to \beta$ is simply a *morphism* $\alpha \to \mathcal{P}^+(\beta)$

When it comes to other sorts of simulations the situation diverges. On the relator side, cosimulations and bisimulations simply correspond to *simulations with respect to different relators*. It is the dual relator $\mathcal{P}^-$ that expresses cosimulations. A relation $R : A \nrightarrow B$ is a cosimulation precisely if the inclusion of Eq. 5.13 holds for $\mathcal{P}^-$. This corresponds to the diagram on the right in Fig. 5.6

On the Kleisli side, the situation is more complicated. On one hand, we may simply replace the *monad* of simulations with the reverse simulation monad, the action of which is to map a system $\beta : B \to \mathcal{P}B$ to the system $\mathcal{P}^-(\beta) : \mathcal{P}B \to \mathcal{P}\mathcal{P}B$. As we saw in Section 3.5, the Kleisli morphisms of this monad are the relations $R : A \nrightarrow B$ such that the inclusion in the diagram below holds.

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Nearrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}^-(\beta)"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}R"'] & \mathcal{P}\mathcal{P}B \end{tikzcd}$$

But these are not cosimulations, they are reverse simulations! The problem is that the relation lifting is being used in two very different ways in these two approaches. The relator approach is to apply the lifting *horizontaly*, turning a *relation of states* $A \nrightarrow B$ into a lifted relation of successor sets $\mathcal{P}A \nrightarrow \mathcal{P}B$. On the other hand, the Kleisli approach is to apply the lifting *vertically* to lift the *transition relation* of the target system, turning a relation $B \nrightarrow B$ into a relation $\mathcal{P}B \nrightarrow \mathcal{P}B$.

In the case of simulations, it is really a remarkable coincidence that these two notions coincide. But when we move to cosimulations, the correspondence breaks down.

Intuitively, a cosimulation is a simulation that has been dualised "horizontally". Rather than beginning with a transition in A and finding a transition in B, we do the opposite, assume a transition in B and find a matching transition in A. The dagger structure of **Rel** gives an alternative characterisation. The relation $R : A \nrightarrow B$ is a cosimulation precisely when the converse relation $R^\dagger : B \nrightarrow A$ is a simulation. This gives us a correspondence between the relators $\mathcal{P}^+$ and $\mathcal{P}^-$. For a relation $R : A \nrightarrow B$ we have

$$\mathcal{P}^- R = (\mathcal{P}^+(R^\dagger))^\dagger : \mathcal{P}A \nrightarrow \mathcal{P}B. \tag{5.25}$$

On the Kleisli side, if we replace $\mathcal{P}^+$ with $\mathcal{P}^-$ we get the *reverse simulation* monad RevSim. This is because a reverse simulation is the *vertical dual* of a simulation. In Fig. 5.6 we see two equivalent formulations of the fact that R is a co-simulation—as an *oplax* Kleisli morphism for Sim on the left, and as a $\mathcal{P}^-$-relator on the right. To encode bisimulations, we need the *strict* morphisms $R : A \nrightarrow \mathrm{Sim}\, B$ (that is, the functional bisimulations). This is depicted in Fig. 5.7

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Swarrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}^+(\beta)"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}R"'] & \mathcal{P}\mathcal{P}B \end{tikzcd} \quad \Longleftrightarrow \quad \begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Swarrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}\beta"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}^-(R)"'] & \mathcal{P}\mathcal{P}B \end{tikzcd}$$

Figure 5.6: Two equivalent ways of expressing the fact that R is a cosimulation

$$\begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "//"] & \mathcal{P}B \arrow[d, "\mathcal{P}^+(\beta)"] \\ \mathcal{P}A \arrow[r, "\mathcal{P}R"'] & \mathcal{P}\mathcal{P}B \end{tikzcd} \quad \Longleftrightarrow \quad \begin{tikzcd} A \arrow[r, "R"] \arrow[d, "\alpha"'] \arrow[dr, phantom, "\Swarrow"] & \mathcal{P}B \arrow[d, "\mathcal{P}\beta"] \\ \mathcal{P}A \arrow[r, "\hat{\mathcal{P}}(R)"'] & \mathcal{P}\mathcal{P}B \end{tikzcd}$$

Figure 5.7: Two equivalent ways of expressing the fact that R is a bisimulation

Chapter 6

Generalised monoid actions

In the previous chapter we saw that different types of transition systems can be represented as coalgebras for diverse *transition monads*. We also saw that the there is a correspondence between lax distributive laws $ST \to TS$, and lax extensions $\underline{S}$ of S to $\mathcal{C}_T$. We would like to replicate the results in Chapter 3—which was specific to non-deterministic systems (the $\mathcal{P}$ monad)—in the case of other transition monads. That is, we would like to

- define an appropriate category of T-transition systems, with "edge preserving functions" as morphisms
- define a monad $\mathbb{S}$ on this category, so that the Kleisli morphisms correspond to simulations of T-transition systems.

Given that we are encoding transition systems as coalgebras, and there is an existing notion of a "morphism of coalgebras" (and hence a category of T-coalgebras $\mathbf{Coalg}(T)$), we might consider taking this to be the canonical category of T-transition systems.

There are two problems with this approach. The first, discussed in the previous chapter, is that coalgebra morphisms are *too strict*, and that a more relaxed notion is required: a *lax cohomomorphism* [15]. To proceed in this way, we will need to have some additional structure, that of a **Pos**-enrichment on the Kleisli category $\mathcal{C}_T$, as described in Chapter 4. If we take the natural **Pos**-enrichment on $\mathbf{Set}_{\mathcal{P}}$ (pointwise subset inclusion), we will see that the resulting category of $\mathcal{P}$-coalgebras with lax cohomomorphisms is indeed equivalent to the category of (unlabelled) non-deterministic transition systems.

The second issue is to do with the construction of monads on this category. The category of coalgebras is too small. We cannot hope to start with a monad $\mathbb{S}$ on the category of T-coalgebras and construct a lax monad extension $\underline{S}$ on $\mathcal{C}_T$. This is because $\mathbb{S}$ will only contain the action of $\underline{S}$ on *endomorphisms* $A \nrightarrow A$. In order to prove a correspondence theorem, we need the monad $\mathbb{S}$ to be on a *larger category* that includes all the morphisms of $\mathcal{C}_T$ as objects.

6.1 Classical monoid actions

First, a word about contravariance and opposite monoids, and a clarification of notation. The driving example is the case of a non-deterministic finite automaton over an alphabet Σ. That is, a set A and a family of relations indexed by Σ, $R_\sigma : A \nrightarrow A$. We can formally extend this to a family of relations over all of $\Sigma^\star$, by defining

$$R_{\sigma_1\sigma_2\cdots\sigma_n} = R_{\sigma_1} \fatsemi R_{\sigma_2} \fatsemi \cdots R_{\sigma_n},$$

and this composite has the effect of performing a transition along σ_1, then along σ_2, and so on.

So this encodes a monoid morphism from $\Sigma^\star$ to the monoid of relations on A (with multiplication given by $\fatsemi$), which is in fact the endomorphism monoid $\mathbf{Rel}(A, A)$. We know that a relation on A may equivalently be viewd as a function $A \to \mathcal{P}(A)$, and so we can form the endomorphism monoid $\mathbf{Set}_{\mathcal{P}}(A, A)$. But multiplication here is given by Kleisli composition of relations, which is opposite to the relational composition $\fatsemi$.

$$\begin{aligned} R \fatsemi S &= S \bullet R \\ &= \mu \circ \mathcal{P}(S) \circ R \end{aligned}$$

The point, therefore, is that the structure we are considering here is that of a *contravariant* monoid morphism from the monoid $\Sigma^\star$, to a Kleisli endomorphism monoid. The mismatch is because strings are consumed from the left-most end, while compositions of morphisms are consumed (applied) from the right-most end.

Now, let us recall the definition of a (right) monoid action (presented in [31], for example). If M is a monoid then an action of M consists of a base set A and a function $\bullet : A \times M \to A$ such that

1. $a \bullet 1 = a$

2. $(a \bullet \sigma) \bullet \tau = a \bullet (\sigma\tau)$.

A morphism of actions is a function $f : A \to B$ that interacts nicely with the two actions: for every $a \in A$ and $\sigma \in M$ we have

$$f(a \bullet \sigma) = f(a) \bullet \sigma.$$

The collection of actions of M and morphisms forms a category, which is usually denoted $\mathbf{Act}(M)$. In order to generalise this construction, we will curry the type signature of the monad action. So let $\delta : M \to (A \to A) = \mathrm{End}(A)$, where δ is defined by $\delta(\sigma)(a) = a \bullet \sigma$ We can rewrite the two conditions above in terms of δ.

1. $\delta(1) = \mathrm{id}_A$

2. $\delta(\sigma\tau) = \delta(\tau) \circ \delta(\sigma)$

If we reverse the order of composition, writing $\delta(\sigma) \fatsemi \delta(\tau) = \delta(\tau) \circ \delta(\sigma)$, we see that a (right) monoid action is equivalent to a (contravariant) monoid homomorphism into an endomorphism monoid. Left actions are equivalent to covariant homomorphisms.

A more *categorical* way of thinking about this situation is to remember that a monoid M is equivalently a category $\underline{M}$ with a single object. It is due to the contravariance exhibited above that we find it easier to consider the contravariant category $\underline{M}$. We will show that the category of right actions of M is in fact isomorphic to the category of functors from $\underline{M}$ into **Set** (a similar result will hold for left actions). Recall that a functor $\Delta : \underline{M} \to \mathbf{Set}$ is specified by

- a mapping of objects, that is, a set $A = \Delta\star$, and

- a mapping of morphisms, that is, for every element σ of M, a *function* $\Delta\sigma : A \to A$.

The two properties of a functor are that it preserves identies and composition. That is,

1. $\Delta(1) = \mathrm{id}_A$, and

2. $\Delta(\sigma\tau) = \Delta(\sigma) \fatsemi \Delta(\tau)$

Let $\Delta, \Gamma : \underline{M}^{\mathrm{op}} \to \mathbf{Set}$ be two functors. A natural transformation $f : \Delta \to \Gamma$ is given by a function $f_\star : \Delta(\star) \to \Gamma(\star)$ such that for every element σ in M we have

$$\begin{tikzcd} \Delta(\star) \arrow[r, "f_\star"] \arrow[d, "\Delta(\sigma)"'] & \Gamma(\star) \arrow[d, "\Gamma(\sigma)"] \\ \Delta(\star) \arrow[r, "f_\star"] & \Gamma(\star) \end{tikzcd}$$

This means that for every σ in M and every $a \in \Delta(\star)$ we have

$$f_\star(\Delta(\sigma)(a)) = \Gamma(\sigma)(f_\star(a)),$$

which will hold if and only if $f_\star$ is a right M-action morphism.

Monoid action	Monoid morphism	Functor
$* : A \times M \to A$	$\delta : M \to (A \to A)$	$\Delta : \mathcal{B}M^{\mathrm{op}} \to \mathbf{Set}$
$a = a * 1$	$\mathrm{id}_A = \delta(1)$	$\mathrm{id}_{\Delta(\star)} = \Delta(1)$
$a * \sigma * \tau = a * \sigma\tau$	$\delta(\sigma) \fatsemi \delta(\tau) = \delta(\sigma\tau)$	$\Delta(\sigma) \fatsemi \Delta(\tau) = \Delta(\sigma\tau)$
$f : A \to B$	$f : A \to B$	$f_\star : \Delta(\star) \to \Gamma(\star)$
$f(a *_A \sigma) = f(a) *_B \sigma$	$f(\delta_A(\sigma)(a)) = \delta_B(\sigma)(f(a))$	$f_\star(\Delta(\sigma)(a)) = \Gamma(\sigma)(f_\star(a))$

Table 6.1: Three characterisations of monoid actions

So, we have seen three equivalent ways of presenting the same data: a carrier set A, and a structured mapping from monoid elements to transformations of A, and a coherent notion of morphisms. See Table 6.1 for a summary of this information. We are therefore justified in treating the categories $\mathbf{Act}(M)$ and $[\underline{M}^{\mathrm{op}}, \mathbf{Set}]$ as equivalent. The first connection to automata is as follows.

Proposition 6.1. *For an alphabet Σ, the category $\boldsymbol{Act}(\Sigma^\star)$ is equivalent to $\boldsymbol{DTS}_\Sigma$, the category of* deterministic *transition systems over Σ, with edge preserving functions as morphisms.*

Proof. The monoid $\Sigma^\star$ is freely generated by Σ, so any monoid morphism from $\Sigma^\star$ is uniquely determined by its action on the generators, i.e. by a family indexed by Σ. So it is clear that the objects of the two categories are in bijection.

In both categories, morphisms consist of a function of carrier sets. It is the "preservation condition" that is worded slightly differently.

- In $\mathbf{Act}(\Sigma^\star)$, we require that for every *word* $w \in \Sigma\star$, that $f(\delta(w)(a)) = \delta(w)(f(a))$.
- In $\mathbf{DTS}_\Sigma$, we need $f(\delta(\sigma)(a)) = \delta(\sigma)(f(a))$ for *every letter* $\sigma \in \Sigma$.

Obviously, the first condition implies the second. The converse holds also, because if we assume that f preserves σ and τ transitions, we can form the two commutative diagrams below:

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\delta(\sigma)"'] & B \arrow[d, "\delta(\sigma)"] \\ A \arrow[r, "f"] & B \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\delta(\tau)"'] & B \arrow[d, "\delta(\tau)"] \\ A \arrow[r, "f"] & B. \end{tikzcd}$$

Because the squares commute, we can paste them together vertically, and use the composition rule to combine the transition morphisms running vertically.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\delta(\sigma)"'] & B \arrow[d, "\delta(\sigma)"] \\ A \arrow[r, "f"] \arrow[d, "\delta(\tau)"'] & B \arrow[d, "\delta(\tau)"] \\ A \arrow[r, "f"] & B \end{tikzcd} \implies \begin{tikzcd} A \arrow[r, "f"] \arrow[dd, "\delta(\sigma\tau)"'] & B \arrow[dd, "\delta(\sigma\tau)"] \\ & \\ A \arrow[r, "f"] & B \end{tikzcd}$$

So in fact, we only need to verify that a given function $f : A \to B$ preserves the letters (the generators of $\Sigma^\star$). If it does, it will preserve the other words in $\Sigma^\star$ *for free*. This concludes the proof. $\square$

But deterministic automata are a very special case. We would like to generalise this to non-deterministic automata, and other transition types. We will also need to keep in mind that we have seen characterisations of *unlabelled* automata with transition type T as T-coalgebras.

Our first guess might be that we should replace **Set** with a different category, and consider the category of (contravariant) functors $\underline{M} \to \mathcal{C}$. Non-deterministic transition functions are *relations* (i.e. morphisms in **Rel**), so we might wager that the category $\mathbf{TS}_\Sigma$ is equivalent to the category of functors $\underline{\Sigma^\star} \to \mathbf{Rel}$.

This doesn't quite work. Let Δ, Γ be contravariant functors into **Rel**, with carrier sets A, B. A morphism $f : \Delta \to \Gamma$ is a natural transformation. It consists of a *morphism in* **Rel**—a *relation* $f : A \to B$. But the morphisms in $\mathbf{TS}_\Sigma$ are functions from A to B.

They are not Kleisli morphisms in $\mathbf{Rel} \cong \mathbf{Set}_{\mathcal{P}}$, rather they live in the *base category* **Set**.

The point is that a successful generalisation cannot be so crude as simply "functors into a different category" (the Kleisli category of the transition monad). We will need knowledge of the base category as well.

6.1.1 Semigroup actions

Before we carry on with that train of thought, we will take a small diversion and look at an interesting generalisation of a monoid action. Recall that a *semigroup* is a monoid without identity—simply a set S with an associative multiplication.

Therefore we can say that a semigroup action should be a monoid action without condition 1 (the identity rule).

Definition 6.2. Let S be a semigroup. The category $\mathbf{SemiAct}(S)$ has as objects *(right) semigroup actions of* S. A semigroup action [21] consists of a set A and a contravariant semigroup morphism $\delta : S \to \mathrm{End}(A)$ such that the following condition holds:

$$\delta(\sigma) \fatsemi \delta(\tau) = \delta(\sigma\tau) \tag{6.1}$$

A morphism of semigroup actions $f : (A, \delta) \to (B, \gamma)$ is a function $f : A \to B$ such that the diagram below commutes for every element σ in S.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\delta(\sigma)"'] & B \arrow[d, "\gamma(\sigma)"] \\ A \arrow[r, "f"] & B \end{tikzcd}$$

If M is a monoid we can view it as a semigroup by simply "forgetting" the identity element. Therefore a semigroup action of M is a map from M into an endomorphism *monoid* $\mathrm{End}(A)$ that preserves multiplication. It need not send $1 \in M$ to the identity on A. But we are able to say *something* about the image of 1.

Remark 6.3. Let δ be a semigroup action of M on a set A. Note that $\delta(1)$ may

not be the unit of $\mathrm{End}(A)$, but by condition 6.1 we have that for every $\sigma \in M$

$$\begin{aligned}\delta(\sigma) \fatsemi \delta(1) &= \delta(\sigma \cdot 1) \\ &= \delta(\sigma) \\ &= \delta(1) \fatsemi \delta(\sigma)\end{aligned}$$

This means that the image $\delta(M) \subseteq \mathrm{End}(A)$ is always a monoid, with unit $\delta(1)$. But it is not necessarily a *submonoid* of $\mathrm{End}(A)$.

In terms of transition systems, a semigroup action $\delta : \Sigma^\star \to \mathbf{Rel}(A, A)$ corresponds to a transition system where $\delta(\varepsilon)$ is not necessarily the identity. That is, it is a transition system with non-trivial ε *transitions* [49].

6.2 Monadic actions

We said earlier in Section 6.1 that it is inappropriate to think of transition systems as "actions into **Rel**". Even though the transition functions are relations, the *morphisms* are not: they live in **Set**. In order to succesfully generalise this idea to different transition types, we need to realise that there is a connection between **Set** and **Rel**: the category **Rel** is the Kleisli category of $\mathcal{P}$ on **Set**.

A deterministic transition system over Σ is an action of $\Sigma^\star$ on a set A. For every element $w \in \Sigma^\star$ we have an honest endomorphism (in **Set**) of A, a transition function $\Delta(w) : A \to A$.

In the non-deterministic case, the transition functions are "effectful", they are relations $\Delta(w) : A \nrightarrow A$. This is the same as a function $A \to \mathcal{P}(A)$ in **Set**. To generalise, we will consider a monad T on a category $\mathcal{C}$. We will define a category of monoid actions that are in some sense "twisted through", or "have effects in" the monad T. This means they send monoid elements σ to Kleisli endomorphisms $A \to TA$. Composition happens in the Kleisli category $\mathcal{C}_T$. And yet, the morphisms of T-actions will remain morphisms in the base category, that interact well with these "effectful" action.

Definition 6.4. Let M be a monoid, and T a monad on $\mathcal{C}$. A (right) action of M *through* T (we may also call this a T-action of M) consists of an object A of $\mathcal{C}$ and a contravariant monoid morphism $\Delta : M \to \mathcal{C}_T(A, A)$.

A morphism of T-actions $(A, \Delta) \to (B, \Gamma)$ consists of a $\mathcal{C}$-morphism $f : A \to B$ of carriers such that for every element $\sigma \in M$ the following square commutes.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\Delta(\sigma)"'] & B \arrow[d, "\Gamma(\sigma)"] \\ TA \arrow[r, "Tf"] & TB \end{tikzcd}$$

The collection of T-actions of M and morphisms forms a category, which we will call T-**Act**(M). The following result relates the notion of T-actions to some familiar categories.

Proposition 6.5. *We have the following isomorphisms of categories.*

1. *For any monoid M, the category Id-**Act**(M) is isomorphic to **Act**(M).*

2. *For any monad T, the category T-**Act**$(\mathbb{N})$ is isomorphic to* $\operatorname{Coalg} T$.

Proof. The first result clearly follows from the definition.

To see the second isomorphism, note that a T-action on $\mathbb{N}$ (i.e. a contravariant monoid morphism $\Delta : \mathbb{N} \to \operatorname{End}_{\mathcal{C}_T}(A)$) is *fully determined* by the image of 1—the Kleisli morphism $\Delta(1) : A \nrightarrow A = A \to TA$. This is the same data as a T-coalgebra.

We simply need to verify that T-action morphisms $(A, \Delta) \to (B, \Gamma)$ correspond to coalgebra morphisms $\Delta(1) \to \Gamma(1)$. The data is the same: morphisms $f : A \to B$ in $\mathcal{C}$. However the conditions are expressed differently. The morphism f is a morphism of coalgebras $\Delta(1) \to \Gamma(1)$ if the single diagram below on the left commutes. On the other hand, f is a morphism of T-actions if the diagram below on the right commutes *for every natural number n*.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\Delta(1)"'] & B \arrow[d, "\Gamma(1)"] \\ TA \arrow[r, "Tf"] & TB \end{tikzcd}$$

Coalgebra morphism

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\Delta(n)"'] & B \arrow[d, "\Gamma(n)"] \\ TA \arrow[r, "Tf"] & TB \end{tikzcd}$$

T-action morphism commutes for *every n*

It is clear that a T-action morphism is a coalgebra morphism. And in fact the converse holds also—a coalgebra morphism extends to a morphism of T-actions on $\mathbb{N}$ by pasting (as in the proof of Proposition 6.1).

□

We are getting closer to our goal of expressing the category of transition systems. But unfortunately, we are not quite there yet. Simply taking $T = \mathcal{P}$,

the category $\mathcal{P}$-**Act**$(\Sigma^\star)$ is not equivalent to $\mathbf{TS}_\Sigma$. The objects are correct, and the morphisms have the appropriate type (being *functions* of carrier sets). But as we saw earlier, the preservation condition is too strict. Rather than transition preserving functions, they encode the *functional bisimulations*.

So if we want to capture the entire category $\mathbf{TS}_\Sigma$ in the language of $\mathcal{P}$-actions, we will need the morphisms to be functions $f : A \to B$ such that the inclusion in the diagram below exists.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\Delta(\sigma)"'] \arrow[dr, phantom, "\subseteq"] & B \arrow[d, "\Gamma(\sigma)"] \\ \mathcal{P}(A) \arrow[r, "\mathcal{P}(f)"'] & \mathcal{P}(B) \end{tikzcd}$$

6.2.1 Lax morphisms

Let T be a monad on the category $\mathcal{C}$, and let $\leq$ be a **Pos**-enrichment of the Kleisli category $\mathcal{C}_T$. We will define a category of T-actions with *lax* morphisms (the laxness is controlled by the enrichment of $\mathcal{C}_T$).

Definition 6.6. Let M be a monoid. There is a category T-$\mathbf{Act}_{\mathsf{lax}}(M)$, whose objects are T-actions of M.

A morphism $f : (A, \Delta) \to (B, \Gamma)$ is a $\mathcal{C}$-morphism $f : A \to B$ such that the lax inclusion in the diagram below exists for every element σ in M.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "\Delta(\sigma)"'] \arrow[dr, phantom, "\Nearrow"] & A \arrow[d, "\Gamma(\sigma)"] \\ TA \arrow[r, "Tf"] & TA \end{tikzcd}$$

We will call this the *category of T-actions over M with lax morphisms*, or simply the *lax category of T-actions (over M)*.

Proposition 6.7. *The category $\mathcal{P}$-$\boldsymbol{Act}_{\mathsf{lax}}(\Sigma^\star)$ is isomorphic to $\boldsymbol{TS}_\Sigma$.*

Proof. The objects of $\mathcal{P}$-$\mathbf{Act}_{\mathsf{lax}}(\Sigma^\star)$ are strict $\mathcal{P}$-actions of $\Sigma^\star$. These consist of a base set A and a monoid morphism $\Delta : \Sigma^\star \to \mathbf{Rel}(A, A)$. As $\Sigma^\star$ is a free monoid, the morphism Δ is completely determined by the relations $\Delta(\sigma) : A \nrightarrow A$ for all $\sigma \in \Sigma$. So Δ carries precisely the same information as a classical transition relation $R \subseteq A \times \Sigma \times A$.

The condition for morphisms is straightforward to verify. $\square$

This notion of a lax category of T-actions subsumes the strict category we defined previously. The category T-$\mathbf{Act}(M)$ of Definition 6.4 is a wide subcategory of T-$\mathbf{Act}_{\mathsf{lax}}(M)$.

We have at this point reached a significant milestone: a satisfactory and general definition of a category of transition systems.

The next step will be to redefine the monad of simulations Sim for the categories $\mathcal{P}$-$\mathbf{Act}(\Sigma^\star)$, and then to generalise this construction to "simulation-like" monads for different transition types. There are two further generalisations of the T-action notion that will be helpful.

The first is to investigate semigroup T-actions. These, it turns out, will be too poorly-behaved for our liking. What is more useful is a *relaxed* form of T-action over a monoid, where the identity condition is not *completely removed*, but holds *laxly* instead of strictly. We will also consider T-actions where both the identity and composition conditions hold laxly.

The second generalisation will be a type of "horizontal categorification". Rather than thinking of T-actions as contravariant monoid morphisms from $\Sigma^\star$ into the Kleisli endomorphism monoid $\mathcal{C}_T(A, A)$, we will view them as functors $(\underline{\Sigma^\star}) \to \mathcal{C}_T$. And of course, we can think about functors from an arbitrary category $\mathcal{D}$ into $\mathcal{C}_T$. This is rather more general than we actually *need*, but the fuller categorical setting will be useful when it comes time to define the simulation monads. Rather than a mere *family* of categories T-$\mathbf{Act}(\mathcal{D})$, we will have a $\mathbf{Cat}$-valued *functor* T-$\mathbf{Act}(-)$. The Sim monad will be a natural transformation on this functor.

6.3 Lax and semilax actions

We saw in Section 6.1.1 that a semigroup action over a monoid M is simply a map δ from M to an endomorphism monoid $\mathrm{End}(A)$. The map δ must preserve composition, but we drop the identity condition entirely. We could replicate this notion in the setting of T-actions, but a slightly stronger type of action will be more useful.

Definition 6.8. Let M be a monoid. A *lax* T-action over M consists of a map $\delta : M \to \mathcal{C}_T(A, A)$ such that

1. $\eta_A^T \leq \delta(1)$, and
2. $\delta(\sigma) \fatsemi \delta(\tau) \leq \delta(\sigma\tau)$

It is also worth considering T-actions where the identity rule holds laxly, but the composition is strict. Such actions will be called *semilax*.

1. $\eta_A^T \leq \delta(1)$, and
2. $\delta(\sigma) \fatsemi \delta(\tau) = \delta(\sigma\tau)$

The collections of lax and semilax T-actions with strict morphisms form categories, which we will denote $T\text{-}\mathbf{Act}^{\mathsf{lax}}(M)$ and $T\text{-}\mathbf{Act}^{\mathsf{semi}}(M)$. There are also the corresponding categories with *lax* morphisms—$T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(M)$ and $T\text{-}\mathbf{Act}^{\mathsf{semi}}_{\mathsf{lax}}(M)$.

6.4 Actions based over a category

Up till now we have been considering various types of T-actions (strict, semilax, and lax), and various notions of morphisms (strict and lax). We saw earlier that we can think of monoid actions equivalently as maps $M \times A \to A$, or $M \to \mathrm{End}(A)$, or as functors $(\underline{M})^{\mathrm{op}} \to \mathbf{Set}$. We will now switch over to this third, functorial perspective, and reframe the various categories of T-actions as categories of functors from $(\underline{M})^{\mathrm{op}}$. Then we will generalise, and consider T-actions as categories of functors over arbitrary categories.

Let δ be a strict T-action of M on A. There is a corresponding functor $\Delta : \underline{M}^{\mathrm{op}} \to \mathcal{C}_T$ defined by

$$\Delta(\bullet) = A$$
$$\Delta(\sigma) = \delta(\sigma)$$

The strict identity and composition laws for δ mean that Δ is a strict functor. On the other hand, a strict $\mathbf{Pos}$-functor of the type $(\underline{M})^{\mathrm{op}} \to \mathcal{C}_T$ induces a strict T-action of M, and this correspondence is bijective. Moreover, this holds when we generalise to semilax T-actions (which correspond precisely to semilax $\mathbf{Pos}$-functors), and generalise again to lax T-actions and lax $\mathbf{Pos}$-functors.

T-actions on M	$\mathbf{Pos}$-functors $(\underline{M})^{\mathrm{op}} \to \mathcal{C}_T$
strict action	strict functor
semilax action	semilax functor
lax action	lax functor

At this point, one might hypothesize that there is an isomorphism of categories. On one hand, the category of (strict) T-actions on M, and on the other hand, the category of strict **Pos**-functors from $\underline{M}$ into $\mathcal{C}_T$.

$$T\text{-}\mathbf{Act}(M) \cong \mathbf{PosCat}((\underline{M}), \mathcal{C}_T) \qquad (\textbf{Incorrect})$$

This is **incorrect**, because the *morphisms* do not match up. Morphisms of T-actions have components in the *base category* $\mathcal{C}$, whereas **Pos**-natural transformations have components in the *source category*, which is $\mathcal{C}_T$. So we shall have to invent a new type of morphism between functors that mirrors the morphisms of T-actions. At the same time, we will generalise from simply considering actions over M (functors from the category $(\underline{M})^{\mathrm{op}}$) to actions over *an arbitrary* ***Pos****-category* $\mathcal{D}$ (these will be **Pos**-functors $\mathcal{D} \to \mathcal{C}_T$).

Suppose that $\mathcal{D}$ is a **Pos**-enriched category. We will simultaneously define six different categories of strict, semilax, and lax T-actions over $\mathcal{D}$, with strict or lax morphisms.

Definition 6.9. We begin with the most general case. There is a category $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$, whose objects are lax **Pos**-enriched functors $\mathcal{D} \to \mathcal{C}_T$. If Δ and Γ are two lax **Pos**-functors $\mathcal{D} \to \mathcal{C}_T$, then a (lax) morphism $f : \Delta \to \Gamma$ consists of a family of $\mathcal{C}$-morphisms indexed by the objects of $\mathcal{D}$.

For every D in $\mathcal{D}$ there is a morphism $f_D : \Delta(D) \to \Gamma(D)$ in $\mathcal{C}$. Note carefully that the components of f are in $\mathcal{C}$, and *not* in the Kleisli category $\mathcal{C}_T$.

In addition, the family f of $\mathcal{C}$-morphisms must satisfy the following lax morphism condition. For every morphism $\sigma : D \to E$ in $\mathcal{D}$, the lax square below must exist:

$$\begin{tikzcd}
\Delta(D) \arrow[r, "f_D"] \arrow[d, "\Delta(\sigma)"'] \arrow[dr, phantom, "\Nearrow"] & \Gamma(D) \arrow[d, "\Gamma(\sigma)"] \\
T\Delta(E) \arrow[r, "Tf_E"] & T\Gamma(E)
\end{tikzcd}$$

If $f : \Delta \to \Gamma$ is a morphism, and for every σ the lax inclusion above is actually an *equality*, we will say that f is *strict*.

Furthermore:

- There is a wide subcategory of lax T-actions with *strict* morphisms, which we will denote $T\text{-}\mathbf{Act}^{\mathsf{lax}}(\mathcal{D})$.

- There is a full subcategory consisting of the semilax T-actions, denoted $T\text{-}\mathbf{Act}^{\mathsf{semi}}_{\mathsf{lax}}(\mathcal{D})$.
- There is a full subcategory consisting of the strict T-actions, denoted $T\text{-}\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$
- There are corresponding subcategories of semilax and strict actions with strict morphisms, $T\text{-}\mathbf{Act}^{\mathsf{semi}}(\mathcal{D})$ and $T\text{-}\mathbf{Act}(\mathcal{D})$.

These six categories and the corresponding inclusions are depicted in the diagram below.

$$\begin{tikzcd}
T\text{-}\mathbf{Act}(\mathcal{D}) \arrow[r, hook, "\mathsf{wide}"] \arrow[d, hook, "\mathsf{full}"'] & T\text{-}\mathbf{Act}_{\mathsf{lax}}(\mathcal{D}) \arrow[d, hook, "\mathsf{full}"] \\
T\text{-}\mathbf{Act}^{\mathsf{semi}}(\mathcal{D}) \arrow[r, hook, "\mathsf{wide}"] \arrow[d, hook, "\mathsf{full}"'] & T\text{-}\mathbf{Act}^{\mathsf{semi}}_{\mathsf{lax}}(\mathcal{D}) \arrow[d, hook, "\mathsf{full}"] \\
T\text{-}\mathbf{Act}^{\mathsf{lax}}(\mathcal{D}) \arrow[r, hook, "\mathsf{wide}"] & T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})
\end{tikzcd}$$

All of these collections are indeed categories—we will provide some brief verification. The identity morphism $\mathrm{id}_\Delta : \Delta \to \Delta$ has components $\mathrm{id}_{\Delta,D} = \mathrm{id}_{\Delta(D)} : \Delta(D) \to \Delta(D)$. The lax inclusion specified by the diagram exists, because both compositions are equal to $\Delta(\sigma)$—so the identity is actually a *strict* morphism $\Delta \to \Delta$.

The composition of two lax morphisms $f : \Delta \to \Gamma$ and $g : \Gamma \to \Theta$ is computed pointwise: $(g \circ f)_D = g_D \circ f_D$. This will be a valid morphism, because we can horizontally paste the characteristic diagrams of f and g to form a lax inclusion for the composite (by Remark 4.10).

$$\begin{tikzcd}
\Delta(D) \arrow[r, "f_D"] \arrow[d, "\Delta(\sigma)"'] & \Gamma(D) \arrow[r, "g_D"] \arrow[d, "\Gamma(\sigma)"] \arrow[dl, Rightarrow] & \Theta(D) \arrow[d, "\Theta(\sigma)"] \\
T\Delta(E) \arrow[r, "Tf_E"] & T\Gamma(E) \arrow[r, "Tg_E"] & T\Theta(E)
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
\Delta(D) \arrow[r, "(g\circ f)_D"] \arrow[d, "\Delta(\sigma)"'] & \Theta(D) \arrow[d, "\Theta(\sigma)"] \arrow[dl, Rightarrow] \\
T\Delta(E) \arrow[r, "T(g\circ f)_E"] & T\Theta(E)
\end{tikzcd}$$

The composition of strict morphisms will also be strict, because the pasting above will be a pasting of *commutative* diagrams.

The construction of $T\text{-}\mathbf{Act}(\mathcal{D})$ (and the other categories) is "universal", it doesn't really depend on the details of $\mathcal{D}$. We have a map (rather, several maps) $\mathbf{PosCat} \to \mathbf{Cat}$—to every $\mathbf{Pos}$-enriched category $\mathcal{D}$ we assign the category

T-$\mathbf{Act}(\mathcal{D})$. At this point, we might ask: "Can this map of objects be extended to a *functor* T-$\mathbf{Act}(-)$?"

And indeed it can be. We shall sketch the intuition behind this construction. Let $\mathcal{D}, \mathcal{E}$ be $\mathbf{Pos}$-categories, and $F : \mathcal{D} \to \mathcal{E}$ a $\mathbf{Pos}$-functor. How can we turn F into a functor between T-$\mathbf{Act}(\mathcal{D})$ and T-$\mathbf{Act}(\mathcal{E})$? The objects of one category are $\mathbf{Pos}$-functors $\mathcal{D} \to \mathcal{C}_T$, and the objects of the other are $\mathbf{Pos}$-functors $\mathcal{E} \to \mathcal{C}_T$. The solution is to start with a functor $\mathcal{E} \to \mathcal{C}_T$, and precompose F to end up with a functor from $\mathcal{D}$. Hence we end up with a contravariant functor. In order to really complete the definition however, we shall need to say what happens to the *morphisms* of the category T-$\mathbf{Act}(\mathcal{E})$. We will also need to consider the various cases of strict, semilax, and lax T-actions, and the categories $\mathbf{PosCat}$, $\mathbf{PosCat}^{\mathsf{semi}}$, and $\mathbf{PosCat}^{\mathsf{lax}}$.

Definition 6.10. There is a contravariant functor T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-) : (\mathbf{PosCat}^{\mathsf{lax}})^{\mathrm{op}} \to \mathbf{Cat}$, defined in the following way.

- the action on objects is to map a $\mathbf{Pos}$-category $\mathcal{D}$ to the strict category T-$\mathbf{Act}(\mathcal{D})$,
- the action on morphisms in $\mathbf{PosCat}^{\mathsf{lax}}$ (lax $\mathbf{Pos}$-functors $F : \mathcal{D} \to \mathcal{E}$) will be the strict functor T-$\mathbf{Act}(F)$, a morphism in $\mathbf{Cat}$.

The strict functor T-$\mathbf{Act}(F)$ has type T-$\mathbf{Act}(\mathcal{E}) \to T$-$\mathbf{Act}(\mathcal{D})$. The action of T-$\mathbf{Act}(F)$ on T-$\mathbf{Act}(\mathcal{E})$ is:

- to send an object Δ, which is a lax $\mathbf{Pos}$-functor $\mathcal{E} \to \mathcal{C}_T$, to the composite lax $\mathbf{Pos}$-functor $\Delta \circ F : \mathcal{D} \to \mathcal{C}_T$,
- to send a morphism $f : \Delta \to \Gamma$, which has components (morphisms in $\mathcal{C}$) $f_E : \Delta(E) \to \Gamma(E)$ indexed by the objects E of $\mathcal{E}$, to a morphism T-$\mathbf{Act}(F)(f)$, which is a morphism T-$\mathbf{Act}(F)(\Delta) \to T$-$\mathbf{Act}(F)(\Gamma)$, or equivalently, $\Delta \circ F \to \Gamma \circ F$. The components T-$\mathbf{Act}(F)(f)_D$ are given by $f_{FD} : \Delta(FD) \to \Gamma(FD)$. We can write f_F for T-$\mathbf{Act}(F)(f)$.

Note that T-$\mathbf{Act}(F)$ is indeed a functor. It is clear that the composition and identity rules hold. We just need to verify that it sends morphisms to morphisms. If $f : \Delta \to \Gamma$ is a morphism of T-actions over $\mathcal{E}$, this means that the lax inclusion below holds for every morphism $\sigma : E \to E'$ in $\mathcal{E}$.

$$\begin{tikzcd} \Delta E \arrow[r, "f_E"] \arrow[d, "\Delta(\sigma)"'] \arrow[dr, phantom, "\Nearrow"] & \Gamma E \arrow[d, "\Gamma(\sigma)"] \\ T\Delta E' \arrow[r, "f_{E'}"] & T\Gamma E' \end{tikzcd}$$

In order to show that f_F is a morphism, let τ be a morphism $D \to D'$ in $\mathcal{D}$. We form the required square below

$$\begin{tikzcd} \Delta FD \arrow[r, "f_{FD}"] \arrow[d, "\Delta(\sigma)"'] \arrow[dr, phantom, "\Nearrow"] & \Gamma FD \arrow[d, "\Gamma(\sigma)"] \\ T\Delta FD' \arrow[r, "f_{FD'}"] & T\Gamma FD' \end{tikzcd}$$

By taking $E = FD$ and $E' = FD'$, we see that the lax inclusion exists. Therefore T-$\mathbf{Act}(F)$ is indeed a functor.

There are corresponding functors for all the other categories we have defined.

Definition 6.11. We have the family of functors, defined in accordance with Definition 6.10:

- T-$\mathbf{Act}^{\mathsf{semi}}_{\mathsf{lax}}(-) : (\mathbf{PosCat}^{\mathsf{semi}})^{\mathrm{op}} \to \mathbf{Cat}$
- T-$\mathbf{Act}_{\mathsf{lax}}(-) : \mathbf{PosCat}^{\mathrm{op}} \to \mathbf{Cat}$

There are also the three functors that give the categories with *strict* morphisms. By the argument above, T-$\mathbf{Act}(F)$ (and the semilax and lax variations) preserves strict morphisms.

- T-$\mathbf{Act}^{\mathsf{lax}}(-) : (\mathbf{PosCat}^{\mathsf{lax}})^{\mathrm{op}} \to \mathbf{Cat}$
- T-$\mathbf{Act}^{\mathsf{semi}}(-) : (\mathbf{PosCat}^{\mathsf{semi}})^{\mathrm{op}} \to \mathbf{Cat}$
- T-$\mathbf{Act}(-) : \mathbf{PosCat}^{\mathrm{op}} \to \mathbf{Cat}$

We saw earlier that $\mathcal{P}$-$\mathbf{Act}_{\mathsf{lax}}(\underline{\Sigma^\star})$ is isomorphic to the category $\mathbf{TS}_\Sigma$.

Proposition 6.12. *Let (T, η^T, μ^T) be a monad on $\mathcal{C}$ such that $\mathcal{C}_T$ is* ***Pos****-enriched. Let 1 denote the trivial monoid. The following categories are isomorphic:*

1. T-$\boldsymbol{Act}(\underline{1})$
2. T-$\boldsymbol{Act}_{\mathsf{lax}}(\underline{1})$
3. $\mathcal{C}$

Proof. Note that the objects of both T-$\mathbf{Act}(\underline{1})$ and T-$\mathbf{Act}_{\mathsf{lax}}(\underline{1})$ are strict $\mathbf{Pos}$-functors $\Delta : \underline{1} \to \mathcal{C}_T$. Such a functor consists of two pieces of data:

1. an object $\Delta(*)$ of $\mathcal{C}_T$, and
2. for every morphism in $\underline{1}$, a morphism in $\mathcal{C}_T$.

Since there is only one morphism in $\underline{1}$, the identity $\mathrm{id} : * \to *$, this second component is *uniquely determined* by the object $\Delta(*)$. It is forced to be the identity morphism in $\mathcal{C}_T$, which is the unit of the monad T:

$$\Delta(\mathrm{id}) = \eta^T_{\Delta(*)} : \Delta(*) \to T\Delta(*)$$

On the other hand, every object A induces the functor $\underline{A} : \underline{1} \to \mathcal{C}_T$, with

$$\begin{aligned} \underline{A}(*) &= A \\ \underline{A}(\mathrm{id}) &= \eta^T_A : A \to TA \end{aligned}$$

Hence there is a bijective correspondence between strict $\mathbf{Pos}$-functors $\underline{1} \to \mathcal{C}_T$ and objects of $\mathcal{C}_T$ (which are the same as objects of $\mathcal{C}$). Every T-action over $\underline{1}$ is of the form $\underline{A}$ for some object A of $\mathcal{C}$.

We will verify that this correspondence extends to morphisms also. A strict morphism of T-actions $f : \underline{A} \to \underline{B}$ consists simply of a $\mathcal{C}$-morphism $f_* : A \to B$ such that the diagram below commutes.

$$\begin{tikzcd} A \arrow[r, "f_*"] \arrow[d, "\eta^T_A"'] & B \arrow[d, "\eta^T_B"] \\ TA \arrow[r, "Tf_*"] & TB \end{tikzcd}$$

But this is the naturality square of η^T! So in fact, *every* morphism in $\mathcal{C}$ will be a strict morphism of T-actions over $\underline{1}$. From this we deduce that T-$\mathbf{Act}(\underline{1}) \cong \mathcal{C}$. For a given morphism $f : A \to B$ in $\mathcal{C}$, we write $\underline{f}$ for the corresponding morphism of T-actions $\underline{A} \to \underline{B}$.

The case of T-$\mathbf{Act}_{\mathsf{lax}}(\underline{1})$ holds also because there are *no* properly lax morphisms of T-actions. The diagram above can never be a proper lax inclusion, since η^T is a natural transformation it will always be a strict equality. Hence we have T-$\mathbf{Act}_{\mathsf{lax}}(\underline{1}) \cong T$-$\mathbf{Act}(\underline{1})$. $\square$

6.5 Higher categorical structure

Now that we are dealing not with just a single category of transition systems $\mathbf{TS}_\Sigma$, nor a mere *family* of categories indexed by Σ in $\mathbf{Set}$, but rather a *functorial assignment* of $\mathbf{Pos}$-categories $\mathcal{D}$ to categories $T\text{-}\mathbf{Act}(\mathcal{D})$ of transition systems *over* $\mathcal{D}$.

We can realise the simple category of transition systems $\mathbf{TS}_\Sigma$ as $\mathcal{P}\text{-}\mathbf{Act}_{\mathsf{lax}}(\underline{\Sigma^\star})$. But how can we define the monad Sim in terms of these functors from $\mathbf{PosCat}$? Note that the construction of $\mathrm{Sim} : \mathbf{TS}_\Sigma \to \mathbf{TS}_\Sigma$ makes *no reference* to the actual label set Σ—it is generic and universal. This suggests that rather than thinking of Sim as a family of monads on the categories $\mathbf{TS}_\Sigma$, we could have something stronger: a monad on the functor $\mathcal{P}\text{-}\mathbf{Act}_{\mathsf{lax}}(-)$.

What exactly is a *monad on a functor* to be? If a monad on a category $\mathcal{C}$ consists of

1. a functor (a 1-morphism in $\mathbf{Cat}$) $T : \mathcal{C} \to \mathcal{C}$ and
2. natural transformations (2-morphisms in $\mathbf{Cat}$) η and μ that satisfy the monad laws,

then a monad on the functor $\mathcal{P}\text{-}\mathbf{Act}_{\mathsf{lax}}(-)$ could consist of

1. a natural transformation $\mathbb{S} : \mathcal{P}\text{-}\mathbf{Act}_{\mathsf{lax}}(-) \to \mathcal{P}\text{-}\mathbf{Act}_{\mathsf{lax}}(-)$, and
2. higher morphisms η, μ that satisfy the monad laws.

In the simple case, we defined a monad Sim on an object $\mathbf{TS}_\Sigma$ that lived in the 2-category $\mathbf{Cat}$. To generalise, we will need to come up with an appropriate 2-category for the functor $\mathcal{P}\text{-}\mathbf{Act}_{\mathsf{lax}}(-)$ to live in. Then we shall examine what the 1 and 2-morphisms look like, and determine what it means for Sim to be a monad on $\mathcal{P}\text{-}\mathbf{Act}_{\mathsf{lax}}(-)$.

The 2-category in question will be the functor 2-category $[\mathbf{PosCat}^{\mathrm{op}}, \mathbf{Cat}]$ (as defined in [36]). We will examine the case of the strict functor $T\text{-}\mathbf{Act}(-)$ first—the following remarks will hold for all the lax and semilax variations. Much of this material can be found in [36], but it is worth examining the specific details.

First, we recall that the category $[\mathbf{PosCat}^{\mathrm{op}}, \mathbf{Cat}]$ is defined as follows

- the objects are strict functors $\mathsf{P} : \mathbf{PosCat}^{\mathrm{op}} \to \mathbf{Cat}$,
- the morphisms (1-cells) $\mathbb{S} : \mathsf{P} \to \mathsf{Q}$ are strict natural transformations, and

- the 2-cells $\alpha : \mathbb{S} \to \mathbb{R}$ are strict *modifications*

Explicitly, the components of natural transformation $\mathbb{S} : \mathsf{P} \to \mathsf{Q}$ are morphisms in **Cat** (strict functors), indexed by the objects $\mathcal{D}$ of **PosCat** (**Pos**-categories), $\mathbb{S}_{\mathcal{D}} : \mathsf{P}\mathcal{D} \to \mathsf{Q}\mathcal{D}$. The naturality condition requires that for every morphism (**Pos**-functor) $F : \mathcal{D} \to \mathcal{E}$ of **PosCat**, the square below (which lives in **Cat**) commutes.

$$\begin{tikzcd} \mathsf{P}\mathcal{E} \arrow[r, "\mathsf{P}(F)"] \arrow[d, "\mathbb{S}_{\mathcal{E}}"'] & \mathsf{P}\mathcal{D} \arrow[d, "\mathbb{S}_{\mathcal{D}}"] \\ \mathsf{Q}\mathcal{E} \arrow[r, "\mathsf{Q}(F)"] & \mathsf{Q}\mathcal{D} \end{tikzcd}$$

Figure 6.2: Naturality condition

Furthermore, if $\mathbb{S}, \mathbb{R}$ are parallel natural transformations $\mathsf{P} \to \mathsf{Q}$, then a 2-morphism (a strict modification) $\alpha : \mathbb{S} \to \mathbb{R}$ also has components indexed by the objects of **PosCat**. For every poset-enriched category $\mathcal{D}$, we have a strict natural transformation $\alpha_{\mathcal{D}}$ between the strict functors $\mathbb{S}_{\mathcal{D}}$ and $\mathbb{R}_{\mathcal{D}}$.

The coherence condition (found in [36]) is that for every $F : \mathcal{D} \to \mathcal{E}$ the diagram below commutes.

$$\begin{tikzcd} \mathsf{Q}(F) \circ \mathbb{S}_{\mathcal{E}} \arrow[r, "1*\alpha_{\mathcal{D}}"] \arrow[d, "\mathbb{S}_F"'] & \mathsf{Q}(F) \circ \mathbb{R}_{\mathcal{E}} \arrow[d, "\mathbb{R}_F"] \\ \mathbb{S}_{\mathcal{D}} \circ \mathsf{P}(F) \arrow[r, "\alpha_{\mathcal{E}}*1"] & \mathbb{R}_{\mathcal{D}} \circ \mathsf{P}(F) \end{tikzcd}$$

Note that this diagram is in $\mathbf{Cat}(\mathsf{P}\mathcal{E}, \mathsf{Q}\mathcal{D})$, the objects are functors $\mathsf{P}\mathcal{E} \to \mathsf{Q}\mathcal{D}$, and the arrows are natural transformations. The operator $*$ refers to the horizontal composition of natural transformations. Finally, the vertical arrows $\mathbb{S}_F$ and $\mathbb{R}_F$ refer to the 2-cell components of Fig. 6.2. But since these are strict equalities, the coherence condition boils down to the following equality of natural transformations

$$\begin{tikzcd} \mathsf{Q}F \circ \mathbb{S}_{\mathcal{E}} \arrow[r, bend left, "1*\alpha_{\mathcal{D}}"] \arrow[r, bend right, "\alpha_{\mathcal{E}}*1"'] \arrow[r, phantom, "\Vert"] & \mathbb{R}_{\mathcal{D}} \circ \mathsf{P}F. \end{tikzcd}$$

Let us pick this equality apart further. Equality of natural transformations is componentwise, so in order to have $1 * \alpha_{\mathcal{D}} = \alpha_{\mathcal{E}} * 1 : \mathsf{P}\mathcal{E} \to \mathsf{Q}\mathcal{D}$, we require that for every object Y of $\mathsf{P}\mathcal{E}$, we have the equality of morphisms in $\mathsf{Q}\mathcal{D}$

$$(1 * \alpha_{\mathcal{D}})_Y = (\alpha_{\mathcal{E}} * 1)_Y : \mathbb{S}_{\mathcal{D}} \to \mathbb{R}_{\mathcal{D}}.$$

In order to express the components of the natural transformations $1 * \alpha_{\mathcal{D}}$ and $\alpha_{\mathcal{E}} * 1$, it is helpful to construct the diagrams below.

$$\begin{tikzcd} \mathsf{P}\mathcal{E} \arrow[r, "\mathsf{P}F"] & \mathsf{P}\mathcal{D} \arrow[r, bend left, "\mathbb{S}_{\mathcal{D}}"] \arrow[r, bend right, "\mathbb{R}_{\mathcal{D}}"'] \arrow[r, phantom, "\Downarrow\alpha_{\mathcal{D}}"] & \mathsf{Q}\mathcal{D}. \end{tikzcd}$$

(a) $1 * \alpha_{\mathcal{D}}$

$$\begin{tikzcd} \mathsf{P}\mathcal{E} \arrow[r, bend left, "\mathbb{S}_{\mathcal{E}}"] \arrow[r, bend right, "\mathbb{R}_{\mathcal{E}}"'] \arrow[r, phantom, "\Downarrow\alpha_{\mathcal{E}}"] & \mathsf{Q}\mathcal{E} \arrow[r, "\mathsf{P}F"] & \mathsf{Q}\mathcal{D}. \end{tikzcd}$$

(b) $\alpha_{\mathcal{E}} * 1$

Figure 6.3: The two natural transformations

By the definition of horizontal composition, the components are given by

$$\begin{aligned} (1 * \alpha_{\mathcal{D}})_Y &= \alpha_{\mathcal{D},\mathsf{P}F(Y)} : \mathbb{S}_{\mathcal{D}}(\mathsf{P}F(Y)) \to \mathbb{R}_{\mathcal{D}}(\mathsf{P}F(Y)) \\ (\alpha_{\mathcal{E}} * 1)_Y &= \mathsf{Q}F(\alpha_{\mathcal{E},Y}) : \mathsf{Q}F(\mathbb{S}_{\mathcal{E}}(Y)) \to \mathsf{Q}F(\mathbb{R}_{\mathcal{E}}(Y)) \end{aligned}$$

Naturality of $\mathbb{S}$ and $\mathbb{R}$ tells us that the two type signatures are actually equal, they are both $\mathsf{Q}F(\mathbb{S}_{\mathcal{E}}(Y)) \to \mathbb{R}_{\mathcal{D}}(\mathsf{P}F(Y))$, which is the correct type.

This is summarised by the following result.

Proposition 6.13. *Let P, Q be parallel functors* ***PosCat***$^{op} \to$ ***Cat****, and $\mathbb{S}, \mathbb{R}$ be natural transformations $\mathsf{P} \to \mathsf{Q}$. Let α be a family of morphisms indexed by* ***Pos****-categories $\mathcal{D}$ and objects Y of $\mathsf{P}\mathcal{D}$,*

$$\alpha_{\mathcal{D},Y} : \mathbb{S}_{\mathcal{D}}Y \to \mathbb{R}_{\mathcal{D}}Y \text{ in the category } \mathsf{Q}\mathcal{D}.$$

The family α is a (strict) modification *if and only if the following conditions hold.*

1. *For all morphisms $f : X \to Y$ in $\mathsf{P}\mathcal{D}$ the diagram below commutes,*

$$\begin{tikzcd} \mathbb{S}_{\mathcal{D}}X \arrow[r, "\mathbb{S}_{\mathcal{D}}f"] \arrow[d, "\alpha_{\mathcal{D},X}"'] & \mathbb{S}_{\mathcal{D}}Y \arrow[d, "\alpha_{\mathcal{D},Y}"] \\ \mathbb{R}_{\mathcal{D}}X \arrow[r, "\mathbb{R}_{\mathcal{D}}f"] & \mathbb{R}_{\mathcal{D}}Y \end{tikzcd}$$

2. For every ***Pos****-functor* $F : \mathcal{D} \to \mathcal{E}$ *we have the equality,*

$$\alpha_{\mathcal{D},PF(Y)} = QF(\alpha_{\mathcal{E},Y})$$

Proof. The first condition encodes naturality of every $\alpha\mathcal{D} : \mathbb{S}_{\mathcal{D}} \to \mathbb{R}_{\mathcal{D}}$. The second is the modification condition. □

The final contribution of this section is a Yoneda-style result for the T-action functors. We will show that the structure of natural transformations on a T-action functor is very limited: they are all correspond to post-composition by a **Pos**-functor $\mathcal{C}_T \to \mathcal{C}_T$.

The (contravariant) Yoneda lemma is reproduced below, as well as a brief proof.

Lemma 6.14 (Yoneda)**.** *Let* $\mathcal{C}$ *be a category, and* $G : \mathcal{C}^{op} \to$ ***Set*** *a contravariant functor. Let* A *be an object of* $\mathcal{C}$*.*

There is a bijective correspondence between natural transformations from the contravariant hom-functor $\mathrm{Hom}(-, A)$ *to* G *and elements of* $G(A)$*.*

$$\mathrm{Nat}(\mathrm{Hom}(-, A), G) \cong G(A)$$

Proof. The correspondence looks like this. Given a natural transformation $\mathbb{S} : \mathrm{Hom}(-, A) \to G$ we construct an element of GA by applying $\mathbb{S}_A : \mathrm{Hom}(A, A) \to GA$ to the only element of $\mathrm{Hom}(A, A)$ that we can take for granted: the identity id_A.

$$\mathbb{S} \mapsto \mathbb{S}_A(\mathrm{id}_A) \in GA$$

On the other hand, if $x \in GA$ we construct a natural transformation $\mathbb{S}$ in the following way. For an object B of $\mathcal{C}$ and a morphism $f : B \to A$ we define $\mathbb{S}_B : \mathrm{Hom}(B, A) \to GB$ by

$$\mathbb{S}_B(f) = Gf(x) \in GB$$

There are two final things we need to confirm. First, that this correspondence is bijective. This is a routine verification. On one hand, let $\mathbb{S}$ be a natural

transformation. Let $x = \mathbb{S}_A(\mathrm{id}_A) \in GA$ and let $\mathbb{R} : \mathrm{Hom}(-, A) \to G$ be given by

$$\mathbb{R}_B(f) = Gf(x).$$

We wish to show that $\mathbb{R}$ is actually the same as $\mathbb{S}$. So let $f : B \to A$ be a morphism in $\mathrm{Hom}(B, A)$. We have:

$$\begin{aligned}
\mathbb{R}_B(f) &= Gf(x) \\
&= Gf(\mathbb{S}_A(\mathrm{id}_A)) \\
&= \mathbb{S}_B(f \circ \mathrm{id}_A)) && \text{(by nat. } \mathbb{S}) \\
&= \mathbb{S}_B(f)
\end{aligned}$$

The penultimate equation follows from chasing f around the naturality diagram.

$$\begin{tikzcd}
\mathrm{Hom}(B, A) \arrow[r, "\mathrm{Hom}(f,A)"] \arrow[d, "\mathbb{S}_B"'] & \mathrm{Hom}(A, A) \arrow[d, "\mathbb{S}_A"] \\
GB \arrow[r, "Gf"] & GA
\end{tikzcd}$$

On the other hand, if $x \in GA$, we can form the natural transformation $\mathbb{S}$ as above, and consider the element $\mathbb{S}_A(\mathrm{id}_A) \in GA$. By definition, we have

$$\begin{aligned}
\mathbb{S}_A(\mathrm{id}_A) &= G\mathrm{id}_A x \\
&= x
\end{aligned}$$

because G is a functor.

The second consideration is that the transformation induced by an element x must actually be natural. To see that it is, take a morphism $g : B \to C$ in $\mathcal{C}$ and form the naturality square below.

$$\begin{tikzcd}
\mathrm{Hom}(C, A) \arrow[r, "\mathrm{Hom}(g,A)"] \arrow[d, "\mathbb{S}_C"'] & \mathrm{Hom}(B, A) \arrow[d, "\mathbb{S}_B"] \\
GC \arrow[r, "Gg"] & GB
\end{tikzcd}$$

We need to show that

$$Gg \circ \mathbb{S}_B = \mathbb{S}_C \circ \mathrm{Hom}(g, A) : \mathrm{Hom}(C, A) \to GB.$$

So let $f : C \to A$ and consider both sides:

$$\begin{aligned} Gg(\mathbb{S}_B(f) &= Gg(Gf(x)) \\ &= G(f \circ g)(x) \\ \mathbb{S}_C(\mathrm{Hom}(g, A)(f)) &= \mathbb{S}_C(f \circ g) \\ &= G(f \circ g)(x) \end{aligned}$$

and hence we see that $\mathbb{S}$ is indeed natural. □

A particularly useful corollary is when we take the functor G to *also* be a hom-functor.

Corollary 6.15. *For every object A, B of $\mathcal{C}$, there is a bijection of sets*

$$\mathrm{Nat}(\mathrm{Hom}(-, A), \mathrm{Hom}(-, B)) \cong \mathrm{Hom}(A, B). \tag{6.2}$$

The details of the construction are as follows. A natural transformation $\mathbb{S} : \mathrm{Hom}(-, A) \to \mathrm{Hom}(-, B)$ is assigned to the morphism $\mathbb{S}_A(id_A) : A \to B$. And on the other hand, a morphism $f : A \to B$ induces the natural transformation $\mathbb{S}$ where

$$\mathbb{S}_C(g) = f \circ g.$$

The fact that this correspondence is bijective means that the behaviour of any natural transformation $\mathbb{S}$ is totally determined *by the value $\mathbb{S}_A(id_A)$. For any other object C and any morphism $g : C \to A$ we have (by the above equation)*

$$\mathbb{S}_C(g) = \mathbb{S}_A(id_A) \circ g$$

Now, since the functor T-**Act**$(-)$: **PosCat**$^{\mathrm{op}} \to$ **Cat** is "almost" a hom-functor—the objects of T-**Act**$(\mathcal{D})$ are **Pos**-functors $\mathcal{D} \to \mathcal{C}_T$—it seems reasonable to suggest that a similar result will hold for natural transformations $\mathbb{S}$ on T-**Act**$(-)$. We will attempt to generalise the constructions of Corollary 6.15 first.

So, let $\mathbb{S}$ be a natural transformation T-**Act**$(-) \to T$-**Act**$(-)$. We will define $\underline{S}$ to be $\mathbb{S}_{\mathcal{C}_T}(\mathrm{Id}_{\mathcal{C}_T})$, which is a strict **Pos**-functor $\mathcal{C}_T \to \mathcal{C}_T$.

On the other hand, a mere **Pos**-functor $\underline{S} : \mathcal{C}_T \to \mathcal{C}_T$ is *insufficient* information to define a natural transformation $\mathbb{S}$. A component $\mathbb{S}_{\mathcal{D}}$ must be a *functor*

T-$\mathbf{Act}(\mathcal{D}) \to T$-$\mathbf{Act}(\mathcal{D})$. We can define the action on *objects* by the rule

$$\mathbb{S}_{\mathcal{D}}(\Delta) = \underline{S} \circ \Delta,$$

but we also need an action on *morphisms*. That is, for a morphism $f : \Delta \to \Gamma$, which has components $f_D : \Delta D \to \Gamma D$ in the *base category* $\mathcal{C}$, we need to give components $\underline{S}\Delta D \to \underline{S}\Gamma D$. So we need both a functor $\underline{S}$ on $\mathcal{C}_T$, and a functor S on $\mathcal{C}$. This leads us to the following result.

Lemma 6.16. *Let F be one of the six T-action functors defined in Definition 6.11. There is a bijective correspondence between natural transformations $\mathbb{S}$ on F, and lax extension pairs of $\underline{S}$ and S . The properties of F (strict/semilax/lax objects, and strict/lax morphisms) correspond to properties of the extension of $\underline{S}$ over S in the following way.*

T-$\boldsymbol{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-)$	*lax $\underline{S}$, lax ext.*
T-$\boldsymbol{Act}^{\mathsf{semi}}_{\mathsf{lax}}(-)$	*semilax $\underline{S}$, lax ext.*
T-$\boldsymbol{Act}_{\mathsf{lax}}(-)$	*strict $\underline{S}$, lax ext. ($\Longleftrightarrow$ strict ext.)*
T-$\boldsymbol{Act}^{\mathsf{lax}}(-)$	*lax $\underline{S}$, lax ext. + strict left whisk.*
T-$\boldsymbol{Act}^{\mathsf{semi}}(-)$	*semilax $\underline{S}$, lax ext. + strict left whisk.*
T-$\boldsymbol{Act}(-)$	*strict $\underline{S}$, lax ext. ($\Longleftrightarrow$ strict ext.) + strict left whisk.*

Proof. We begin by giving the details of the construction. In the most general case we will take $F = T$-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-)$. Let $\mathbb{S}$ be a natural transformation $F \to F$. The extending functor will be $\underline{S} = \mathbb{S}_{\mathcal{C}_T}(\mathrm{Id}_{\mathcal{C}_T})$, a lax $\mathbf{Pos}$-functor on $\mathcal{C}_T$. The base functor S will be the restriction of $\mathbb{S}_{\underline{1}}$, a functor on T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{1})$, to the subcategory T-$\mathbf{Act}(\underline{1}) \cong \mathcal{C}$. More explicitly, for an object A in $\mathcal{C}$ we take

$$SA = \underline{S}A$$

and for a morphism $f : A \to B$, we take

$$Sf = \mathbb{S}_{\underline{1}}(\underline{f})_*$$

where $\underline{f} : \underline{A} \to \underline{B}$ is the morphism f considered as a T-action morphism in T-$\mathbf{Act}(\underline{1})$.

On the other hand, given a base functor S on $\mathcal{C}$ and a lax extension $\underline{S}$ to $\mathcal{C}_T$, we define $\mathbb{S}$ componentwise by

$$\begin{aligned}\mathbb{S}_{\mathcal{D}}(\Delta) &= \underline{S} \circ \Delta \\ (\mathbb{S}_{\mathcal{D}} f)_D &= S(f_D).\end{aligned}$$

The first step of the proof is to show that these two constructions define a bijection.

Let us begin with a lax extension of the functor S to the lax **Pos**-functor $\underline{S}$. We will define $\mathbb{S}$ as above. We expect, that when constructing the corresponding extension from $\mathbb{S}$, to arrive back at $\underline{S}$ and S. And indeed, we do, for

$$\begin{aligned}\mathbb{S}_{\mathcal{C}_T}(\mathrm{Id}_{\mathcal{C}_T}) &= \underline{S} \circ \mathrm{Id}_{\mathcal{C}_T} \\ &= \underline{S}\end{aligned}$$

and for $\mathbb{S}_{\underline{1}}$ we have

$$\begin{aligned}(\mathbb{S}_{\underline{1}} \underline{f})_* &= S(\underline{f}_*) \\ &= Sf\end{aligned}$$

On the other hand, suppose we start with a natural transformation $\mathbb{S}$. In order for the construction to be bijective we need for every object Δ and every morphism $f : \Delta \to \Gamma$ in T-$\mathbf{Act}(\mathcal{D})$:

$$\begin{aligned}\mathbb{S}_{\mathcal{D}}(\Delta) &= \mathbb{S}_{\mathcal{C}_T}(\mathrm{Id}_{\mathcal{C}_T}) \circ \Delta \\ (\mathbb{S}_{\mathcal{D}} f)_D &= \mathbb{S}_{\underline{1}} \underline{(f_D)}_*\end{aligned}$$

The first equality follows as in the standard Yoneda proof, by considering Δ as a morphism in $\mathbf{PosCat}^{\mathsf{lax}}$ and following $\mathrm{Id}_{\mathcal{C}_T}$ around.

To derive the second we need to fix an object D of $\mathcal{D}$. Let $\underline{D}$ be the characteristic strict **Pos**-functor $\underline{1} \to \mathcal{D}$ that picks out D. Hence we form the naturality square

$$\begin{tikzcd} T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D}) \arrow[r, "T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})"] \arrow[d, "\mathbb{S}_{\mathcal{D}}"'] & T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{1}) \arrow[d, "\mathbb{S}_{\underline{1}}"] \\ T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D}) \arrow[r, "T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})"] & T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{1}) \end{tikzcd}$$

The effect of T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})$ is to pick out the component at D. For a lax

Pos-functor $\Delta : \mathcal{D} \to \mathcal{C}_T$ we have

$$T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})(\Delta)(*) = \Delta D$$

and for a morphism $f : \Delta \to \Gamma$, we have

$$\begin{aligned} T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})(f)_* &= f_D : \Delta D \to \Gamma D \text{ or equivalently,} \\ T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})(f) &= \underline{f_D} \end{aligned}$$

Hence, following a given morphism f through the naturality square, we deduce the equality of morphisms $\underline{\mathbb{S}(\Delta)D} \to \underline{\mathbb{S}(\Gamma)D}$ in $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{1})$

$$T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})(\mathbb{S}_{\mathcal{D}} f) = \mathbb{S}_{\underline{1}}(T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})f)$$

which corresponds to the equality of morphisms in $\mathcal{C}$

$$T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})(\mathbb{S}_{\mathcal{D}} f)_* = \mathbb{S}_{\underline{1}}(T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})f)_*$$

Unpicking the definitions of both sides, we discover that

$$\begin{aligned} T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})(\mathbb{S}_{\mathcal{D}} f)_* &= (\mathbb{S}_{\mathcal{D}} f)_{\underline{D}(*)} \\ &= (\mathbb{S}_{\mathcal{D}} f)_D \\ \mathbb{S}_{\underline{1}}(T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{D})f)_* &= \mathbb{S}_{\underline{1}}(\underline{f_D})_* \end{aligned}$$

which is the desired equality.

We can now proceed to the second part of the proof. Let $\mathbb{S}$ be a natural transformation, and construct $\underline{S}$ and S. We shall need to show that $\underline{S}$ and S satisfy the strict left whiskering and lax right whiskering conditions. The technique here is to encode the whiskering conditions as a morphism of T-actions that is preserved by a suitable component of $\mathbb{S}$ (being a functor).

We will introduce some helpful notation. Let $\mathbf{2}$ denote the category with two objects 0 and 1, and a single non-identity morphism $e : 0 \to 1$. We will consider $\mathbf{2}$ as a **Pos**-category with the discrete **Pos**-enrichment. A strict T-action on $\mathbf{2}$ is simply a morphism in $\mathcal{C}_T$. A lax morphism from $(p : A \nrightarrow B)$ to $(q : C \nrightarrow D)$ takes the form of a pair of $\mathcal{C}$ morphisms $(f, g) : (A, B) \to (C, D)$ such that the inclusion in the diagram below exists.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "p"'] & C \arrow[d, "q"] \\ TB \arrow[r, "Tg"] & TD \end{tikzcd}$$

1. (lax left whisk.) We need to show that $TSf \circ \underline{S}p \leq \underline{S}(f \circ p)$ for all morphisms $p : A \nrightarrow B$ and $f : B \to C$.

   We form a pair T-actions over $\mathbf{2}$ corresponding to the morphisms $p : A \nrightarrow B$ and $TF \circ p : A \nrightarrow C$. There is a (strict) morphism $(\mathrm{id}_A, Tf) : p \to TF \circ p$, corresponding to the trivial equality

   $$Tf \circ p = (Tf \circ p) \circ \mathrm{id}_A.$$

   Applying $\mathbb{S}_{\mathbf{2}}$ induces a morphism $(S\mathrm{id}_A, STf) : \underline{S}p \to \underline{S}(Tf \circ p)$, which witnesses

   $$TSf \circ \underline{S}p \leq \underline{S}(Tf \circ p) \circ S(\mathrm{id}_A) \qquad = \underline{S}(Tf \circ p), \tag{6.3}$$

   the lax left whiskering condition. This is all expressed in the diagram below.

   $$\begin{tikzcd} A \arrow[r, equal] \arrow[d, "p"'] & A \arrow[d, "Tf \circ p"] \\ TB \arrow[r, "Tf"'] & TC \end{tikzcd} \quad \Longrightarrow \quad \begin{tikzcd} SA \arrow[r, equal] \arrow[d, "\underline{S}p"'] \arrow[dr, phantom, "\Nearrow"] & SA \arrow[d, "\underline{S}(Tf \circ p)"] \\ TSB \arrow[r, "TSf"'] & TSC \end{tikzcd}$$

   Note that despite the source morphism (id_A, Tf) being *strict*, we do not claim that the image $(S\mathrm{id}_A, STf)$ is a strict morphism $\underline{S}p \to \underline{S}(Tf \circ p)$ (if it was, it would witness the *strict* left whiskering condition). This is because $\underline{S}_{\mathbf{2}}$ is a functor $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathbf{2}) \to T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathbf{2})$, and all we can guarantee is that it transforms a lax morphism into a lax morphism—we cannot guarantee that when we give it a strict morphism it will preserve the strictness.

   If $\mathbb{S}$ *did* have this property—that every component $\mathbb{S}_{\mathcal{D}}$ restricts to a functor on the strict morphism subcategory $T\text{-}\mathbf{Act}^{\mathsf{lax}}(\mathcal{D})$—then we could derive the strict left whiskering condition.

2. (strict right whisk.) We need to show that $\underline{S}p \circ Sg = \underline{S}(p \circ g)$ for all morphisms $g : A \to B$ and $p : B \nrightarrow C$.

   We use a similar trick as in the lax left whiskering case to derive the *oplax* right whiskering condition.

$$
\begin{tikzcd}
A \arrow[r, "g"] \arrow[d, "p\circ g"'] & B \arrow[d, "p"] \\
TC \arrow[r, equal] & TC
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
SA \arrow[r, "Sg"] \arrow[d, "\underline{S}(p\circ g)"'] \arrow[dr, phantom, "\Nearrow"] & SB \arrow[d, "\underline{S}p"] \\
TSC \arrow[r, equal] & TSC
\end{tikzcd}
$$

Equationally, this corresponds to the inclusion

$$
\begin{aligned}
T\mathrm{id}_{SC} \circ \underline{S}(p \circ g) &\leq \underline{S}p \circ Sg \\
\underline{S}(p \circ g) &\leq \underline{S}p \circ Sg
\end{aligned}
$$

To finish the job, we will need Proposition 4.22. Because $\underline{S}$ satisfies the lax left whiskering condition, it also satisfies the lax pseudo-extension condition. By construction, it is a lax functor and hence laxly preserves composition. So we may deduce

$$
\begin{aligned}
\underline{S}p \bullet F_T Sg &\leq \underline{S}p \bullet \underline{S}F_T g && \text{(lax pseudo-ext.)} \\
&\leq \underline{S}(p \bullet F_T g) && \text{(lax comp.)}
\end{aligned}
$$

which is the lax right whiskering condition. In conjunction with the oplax inequality, we get the strict condition.

And thirdly, we will verify that if we start with a lax extension $\underline{S}$ of a functor S, the constructed $\mathbb{S}$ is actually a natural transformation. This means that:

1. for every $\mathcal{D}$, the map $\mathbb{S}_{\mathcal{D}}$ is a functor $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D}) \to T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$ (it preserves morphisms)

2. $\mathbb{S}$ is natural in $\mathcal{D}$.

3. when $\underline{S}$ is semilax that $\mathbb{S}_{\mathcal{D}}$ restricts to a functor on $T\text{-}\mathbf{Act}^{\mathsf{semi}}_{\mathsf{lax}}(\mathcal{D})$

4. when $\underline{S}$ is strict, $\mathbb{S}_{\mathcal{D}}$ restricts to a functor on $T\text{-}\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$

We will show that each of these statements holds.

1. ($\mathbb{S}_{\mathcal{D}}$ functorial). The action of $\mathbb{S}_{\mathcal{D}}$ on $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$ on objects $\Delta : \mathcal{D} \to \mathcal{C}_T$ is post-composition by $\underline{S}$. On morphisms $f : \Delta \to \Gamma$, it is to apply S componentwise. Since S is a functor on $\mathcal{C}$, we get the composition and

identity rules for $\mathbb{S}_{\mathcal{D}}$ for free. All we have to check is that it preserves the morphism property.

It will be sufficient to show that for every possible "morphism square", the image under $\mathbb{S}_{\mathcal{D}}$ is valid.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "p"'] & C \arrow[d, "q"] \\ TB \arrow[r, "Tg"'] \arrow[ur, phantom, "\Nearrow"] & TD \end{tikzcd} \quad \Longrightarrow \quad \begin{tikzcd} SA \arrow[r, "Sf"] \arrow[d, "\underline{S}p"'] & SC \arrow[d, "\underline{S}q"] \\ TSB \arrow[r, "TSg"'] \arrow[ur, phantom, "\Nearrow"] & TSD \end{tikzcd}$$

Equationally, this means we have an implication of inequalities:

$$Tg \circ p \leq q \circ f \quad \Longrightarrow \quad TSg \circ \underline{S}p \leq \underline{S}q \circ Sf$$

So suppose that $Tg \circ p \leq q \circ f$. We then have that

$$\begin{aligned} TSg \circ \underline{S}p &\leq \underline{S}(Tg \circ p) && \text{(lax left whisk.)} \\ &\leq \underline{S}(q \circ f) && (\underline{S} \text{ monotone}) \\ &= \underline{S}q \circ Sf && \text{(strict right whisk.)} \end{aligned}$$

which is the desired result.

If we want $\mathbb{S}_{\mathcal{D}}$ to preserve *strict morphisms* we require that $\underline{S}$ satisfies the *strict left whiskering* condition. In that case, a strict morphism square is simply a commuting square.

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "p"'] & C \arrow[d, "q"] \\ TB \arrow[r, "Tg"'] & TD \end{tikzcd} \quad \Longrightarrow \quad \begin{tikzcd} SA \arrow[r, "Sf"] \arrow[d, "\underline{S}p"'] & SC \arrow[d, "\underline{S}q"] \\ TSB \arrow[r, "TSg"'] & TSD \end{tikzcd}$$

From the equality $Tg \circ p = q \circ f$ we deduce

$$\begin{aligned} TSg \circ \underline{S}p &= \underline{S}(Tg \circ p) && \text{(strict left whisk.)} \\ &= \underline{S}(q \circ f) \\ &= \underline{S}q \circ Sf && \text{(strict right whisk.)} \end{aligned}$$

This means that every component $\mathbb{S}_{\mathcal{D}}$ restricts to a functor on $T\text{-}\mathbf{Act}^{\mathsf{lax}}(\mathcal{D}) \hookrightarrow T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$

2. ($\mathbb{S}$ natural) Let $G : \mathcal{D} \to \mathcal{E}$ be a lax **Pos**-functor. In order for $\mathbb{S}$ to be natural we require that the following diagram commutes.

$$\begin{tikzcd}
T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{E}) \arrow[r, "T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G)"] \arrow[d, "\mathbb{S}_{\mathcal{E}}"'] & T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D}) \arrow[d, "\mathbb{S}_{\mathcal{D}}"] \\
T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{E}) \arrow[r, "T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G)"] & T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})
\end{tikzcd}$$

The action of the two functors

$$\begin{aligned}
&\mathbb{S}_{\mathcal{D}} \circ T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G) \text{ and} \\
&T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G) \circ \mathbb{S}_{\mathcal{E}}
\end{aligned}$$

on objects $\Delta : \mathcal{E} \to \mathcal{C}_T$ is identical, as in the standard Yoneda proof. This follows from associativity of composition in $\mathbf{PosCat}^{\mathsf{lax}}$.

$$\begin{aligned}
(\mathbb{S}_{\mathcal{D}} \circ T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G))(\Delta) &= \mathbb{S}_{\mathcal{D}}(\Delta \circ G) \\
&= \underline{S} \circ (\Delta \circ G) \\
(T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G) \circ \mathbb{S}_{\mathcal{E}})(\Delta) &= T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G)(\underline{S} \circ \Delta) \\
&= (\underline{S} \circ \Delta) \circ G
\end{aligned}$$

We also need to verify that these functors have the same action on morphisms. So let $f : \Gamma \to \Delta$ be a lax morphism of lax T-actions over $\mathcal{E}$. This means that f has components indexed by objects E of $\mathcal{E}$. We will end up with a morphism indexed by the objects of $\mathcal{D}$. Since equality of morphisms is determined componentwise, let D be an object in $\mathcal{D}$. We wish to show that

$$(\mathbb{S}_{\mathcal{D}} \circ T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G))(f)_D = (T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G) \circ \mathbb{S}_{\mathcal{E}})(f)_D.$$

On the left hand side we have

$$\begin{aligned}
(\mathbb{S}_{\mathcal{D}} \circ T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G))(f)_D &= S(T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G)(f)_D) \\
&= S(f_{GD}),
\end{aligned}$$

and on the right side we see that

$$\begin{aligned}(T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(G) \circ \mathbb{S}_{\mathcal{E}})(f)_D &= \mathbb{S}_{\mathcal{E}}(f)_{GD} \\ &= S(f_{GD}),\end{aligned}$$

so the two compositions are equal on morphisms also. Hence we see that $\mathbb{S}$ is natural.

3. (semilax and strict restrictions). In the case where $\underline{S}$ is not just a lax **Pos**-functor but in fact has strict composition (semilax), or strict composition and strict identity (fully strict), we can restrict $\mathbb{S}_{\mathcal{D}} : T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D}) \to T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$ to the subcategory of semilax (resp. strict) T-actions.

   This holds simply because when $\underline{S}$ is semilax, and Δ is a semilax T-action (a semilax functor $\mathcal{D} \to \mathcal{C}_T$), the composition $\underline{S} \circ \Delta : \mathcal{D} \to \mathcal{C}_T$ is semilax—an object of $T\text{-}\mathbf{Act}^{\mathsf{semi}}_{\mathsf{lax}}(\mathcal{D})$. The strict case works similarly.

□

Note that the more general case of Lemma 6.14, which concerns only natural transformations *out* of a hom-functor and into any arbitrary functor G, does not hold in this more general setting. It is worth examining what goes wrong.

Let G be a contravariant functor $\mathbf{PosCat}^{\mathsf{lax}\mathrm{op}} \to \mathbf{Cat}$. We would like to associate natural transformations $\mathbb{S} : T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-) \to G$ to objects of the category $G\mathcal{C}_T$. Following the proof of Lemma 6.14, one direction is straight forward. For every natural transformation $\mathbb{S}$, we can construct the object $\mathbb{S}_{\mathcal{C}_T}(\mathrm{Id}_{\mathcal{C}_T})$ of $G\mathcal{C}_T$.

On the other hand, suppose that X is an object of $G\mathcal{C}_T$. In order to define $\mathbb{S}$, we shall need to say what the component functors $\mathbb{S}_{\mathcal{D}} : T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D}) \to G\mathcal{D}$ are. We can start by setting

$$\mathbb{S}_{\mathcal{D}}(\Delta) = G\Delta(x).$$

The trick here is to view an object Δ of $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$ as a morphism $\mathcal{D} \to \mathcal{C}_T$ in $\mathbf{PosCat}^{\mathsf{lax}}$. However, this is as far as we can go. We need to also give an action of $\mathbb{S}_{\mathcal{D}}$ on *morphisms*. A lax morphism of T-actions $f : \Delta \to \Gamma$ would need to be mapped to a morphism $G\Delta x \to G\Gamma x$ in $G\mathcal{C}_T$, and we have no way to construct one. So the proof fails at this point. It is currently unclear whether any more general result than Lemma 6.16 holds.

6.6 Monads on T-$\mathbf{Act}(-)$

Now that we have looked at the structure of modifications in $\left[\mathbf{PosCat}^{\mathsf{lax}\mathrm{op}}, \mathbf{Cat}\right]$, we shall examine what a monad on the object T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-)$ looks like. Such a monad will consist of three components:

1. a natural transformation $\mathbb{S} : T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-) \to T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-)$, or equivalently, a functor S on $\mathcal{C}$ and a lax extension to a lax functor $\underline{S}$ on $\mathcal{C}_T$, as well as

2. a modification $\eta^{\mathbb{S}} : 1 \to \mathbb{S}$, and

3. a modification $\mu^{\mathbb{S}} : \mathbb{S}^2 \to \mathbb{S}$.

The natural transformation $\mathbb{S}$ has components indexed by $\mathbf{PosCat}^{\mathrm{op}}$. For every $\mathbf{Pos}$-category $\mathcal{D}$, we have a functor $\mathbb{S}_{\mathcal{D}} : T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D}) \to T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$. The action of $\mathbb{S}_{\mathcal{D}}$ on objects is post-composition by $\underline{S}$, and on morphisms it is pointwise application of S. Note that 1 and $\mathbb{S}^2$ denote the natural transformations $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-) \to T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-)$ with components (functors $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D}) \to T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$) given by

$$\begin{aligned} 1_{\mathcal{D}} &= \mathrm{Id}_{T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})} \\ (\mathbb{S}^2)_{\mathcal{D}} &= \mathbb{S}_{\mathcal{D}} \circ \mathbb{S}_{\mathcal{D}}. \end{aligned}$$

Hence the modification $\eta^{\mathbb{S}}$ will have components $\eta^{\mathbb{S}}_{\mathcal{D}}$ that are strict natural transformations $\mathrm{Id}_{T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})} \to \mathbb{S}_{\mathcal{D}}$, which themselves will have components

$$\eta^{\mathbb{S}}_{\mathcal{D},\Delta} : \Delta \to \mathbb{S}_{\mathcal{D}}\Delta$$

which are morphisms in $T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$, for every $\mathbf{Pos}$-category $\mathcal{D}$ and every T-action Δ over $\mathcal{D}$. We can in fact expand this definition further, since a morphism of T-actions has components over the objects of $\mathcal{D}$.

We see therefore that $\eta^{\mathbb{S}}$ is defined by the following data. For every $\mathbf{Pos}$-category $\mathcal{D}$, and every T-action Δ, and every object D of $\mathcal{D}$, there is a $\mathcal{C}$-morphism

$$\eta^{\mathbb{S}}_{\mathcal{D},\Delta,D} : \Delta D \to \mathbb{S}_{\mathcal{D}}\Delta D = S\Delta D.$$

All this is summarised in Table 6.2.

Component	Indexed by	Type
$\eta^{\mathbb{S}}$	—	modification $1 \to \mathbb{S}$
$\eta^{\mathbb{S}}_{\mathcal{D}}$	**Pos**-categories $\mathcal{D}$	natural transformation $\mathrm{Id} \to \mathbb{S}_{\mathcal{D}}$
$\eta^{\mathbb{S}}_{\mathcal{D},\Delta}$	T-actions Δ	T-action morphism $\Delta \to \underline{S} \circ \Delta$
$\eta^{\mathbb{S}}_{\mathcal{D},\Delta,D}$	objects D of $\mathcal{D}$	$\mathcal{C}$-morphism $\Delta D \to S\Delta D$
$\mu^{\mathbb{S}}$	—	modification $\mathbb{S}^2 \to \mathbb{S}$
$\mu^{\mathbb{S}}_{\mathcal{D}}$	**Pos**-categories $\mathcal{D}$	natural transformation $\mathbb{S}^2_{\mathcal{D}} \to \mathbb{S}_{\mathcal{D}}$
$\mu^{\mathbb{S}}_{\mathcal{D},\Delta}$	T-actions Δ	T-action morphism $\underline{S}^2 \circ \Delta \to \underline{S} \circ \Delta$
$\mu^{\mathbb{S}}_{\mathcal{D},\Delta,D}$	objects D of $\mathcal{D}$	$\mathcal{C}$-morphism $SS\Delta D \to S\Delta D$

Table 6.2: The components of $\eta^{\mathbb{S}}$ and $\mu^{\mathbb{S}}$

Theorem 6.17. *Let $\mathcal{C}$ be a category, and (T, η^T, μ^T) a monad on $\mathcal{C}$ such that $\mathcal{C}_T$ is **Pos**-enriched.*

There is a bijective correspondence between monads on T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-)$ and lax extensions of monads on $\mathcal{C}$.

Proof. Let (S, η^S, μ^S) be a monad on $\mathcal{C}$, and let $\underline{S}$ be a lax extension of the functor S. We construct the natural transformation $\mathbb{S}$ according to Lemma 6.16, and define the modifications $\eta^{\mathbb{S}}$ and $\mu^{\mathbb{S}}$ pointwise by

$$\eta^{\mathbb{S}}_{\mathcal{D},\Delta,D} = \eta^S_{\Delta D} : \Delta D \to \mathbb{S}_{\mathcal{D}}\Delta D$$
$$\mu^{\mathbb{S}}_{\mathcal{D},\Delta,D} = \mu^S_{\Delta D} : \mathbb{S}_{\mathcal{D}}\mathbb{S}_{\mathcal{D}}\Delta D \to \mathbb{S}_{\mathcal{D}}\Delta D$$

We shall need to verify that $\eta^{\mathbb{S}}$ and $\mu^{\mathbb{S}}$ are actually modifications. This means that the components $\Delta \to \mathbb{S}\Delta$ must be morphisms, the components T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-) \to T$-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-)$ must be natural transformations, and the modification condition of Proposition 6.13) needs to hold. We also need to show that the corresponding monad laws for modifications hold. Only the first of these (the morphism condition) is significant—the rest are rather tedious exercises in component-wise rewriting.

1. ($\eta^{\mathbb{S}}_{\mathcal{D},\Delta}$ morphism) We need to show that $\eta^{\mathbb{S}}_{\mathcal{D},\Delta}$ is indeed a lax morphism $\Delta \to \underline{S} \circ \Delta$ for every lax T-action Δ. This will be the case if, for every

Kleisli morphism $p : A \nrightarrow B$ in $\mathcal{C}_T$ we have the lax inclusion in the diagram on the left below.

$$\begin{tikzcd} A \arrow[r, "\eta^S_A"] \arrow[d, "p"'] \arrow[dr, phantom, "\Nearrow"] & SA \arrow[d, "\underline{S}p"] \\ TB \arrow[r, "T\eta^S_B"'] & TSB \end{tikzcd}$$

Equationally, this is the inclusion

$$T\eta^S_B \circ p \leq \underline{S}p \circ \eta^S_A.$$

But this is precisely the η^S condition on the monadic extension $\underline{S}$! We know this holds at least laxly. If it holds strictly then $\eta^{\mathbb{S}}_{\mathcal{D},\Delta}$ is not just a lax morphism $\Delta \to \underline{S} \circ \Delta$, but a *strict morphism* that lives in the subcategory T-$\mathbf{Act}^{\mathsf{lax}}(\mathcal{D})$.

2. ($\mu^{\mathbb{S}}_{\mathcal{D},\Delta}$ morphism) The situation for $\mu^{\mathbb{S}}_{\mathcal{D},\Delta}$ is similar. It will be a morphism precisely when the inclusion depicted in the square below exists for all Kleisli morphisms $p : A \nrightarrow B$.

$$\begin{tikzcd} SSA \arrow[r, "\mu^S_A"] \arrow[d, "\underline{SS}p"'] \arrow[dr, phantom, "\Nearrow"] & SA \arrow[d, "\underline{S}p"] \\ TSSB \arrow[r, "T\mu^S_B"'] & TSB \end{tikzcd}$$

This corresponds exactly to the lax μ^S condition on $\underline{S}$. If $\underline{S}$ satisfies both the μ^S and η^S conditions strictly, then the components $\eta^{\mathbb{S}}$ and $\mu^{\mathbb{S}}$ restrict to the *strict* functor T-$\mathbf{Act}^{\mathsf{lax}}(-)$, and $(\mathbb{S}, \eta^{\mathbb{S}}, \mu^{\mathbb{S}}$ will restrict to a monad on this functor.

3. ($\eta^{\mathbb{S}}_{\mathcal{D}}$ and $\mu^{\mathbb{S}}_{\mathcal{D}}$ nat.) We will begin with the case of $\eta^{\mathbb{S}}_{\mathcal{D}}$, which is to be a natural transformation $\mathrm{Id} \to \mathbb{S}_{\mathcal{D}}$. By Proposition 6.13 we need the naturality square below on the left to commute for any lax morphism $f : \Delta \to \Gamma$ in T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$.

$$\begin{tikzcd} \Delta \arrow[r, "f"] \arrow[d, "\eta^{\mathbb{S}}_{\mathcal{D},\Delta}"'] & \Gamma \arrow[d, "\eta^{\mathbb{S}}_{\mathcal{D},\Gamma}"] \\ \mathbb{S}_{\mathcal{D}}\Delta \arrow[r, "\mathbb{S}_{\mathcal{D}}f"'] & \mathbb{S}_{\mathcal{D}}\Gamma \end{tikzcd}$$

As this is an equality of lax T-action morphisms, it must hold componentwise. Hence we form the corresponding diagram in $\mathcal{C}$, for any object D of $\mathcal{D}$.

$$\begin{tikzcd} \Delta D \arrow[r, "f_D"] \arrow[d, "\eta^{\mathbb{S}}_{\mathcal{D},\Delta,D}"'] & \Gamma D \arrow[d, "\eta^{\mathbb{S}}_{\mathcal{D},\Gamma,D}"] \\ \mathbb{S}_{\mathcal{D}}\Delta D \arrow[r, "(\mathbb{S}_{\mathcal{D}}f)_D"'] & \mathbb{S}_{\mathcal{D}}\Gamma D \end{tikzcd} \quad = \quad \begin{tikzcd} \Delta D \arrow[r, "f_D"] \arrow[d, "\eta^{S}_{\Delta D}"] & \Gamma D \arrow[d, "\eta^{S}_{\Gamma D}"] \\ S\Delta D \arrow[r, "S(f_D)"'] & S\Gamma D \end{tikzcd}$$

This is the naturality square for η^S, so it must commute. Hence we see that naturality of $\eta^{\mathbb{S}}_{\mathcal{D}}$ corresponds to component-wise naturality of η^S. The case of $\mu^{\mathbb{S}}_{\mathcal{D}}$ is similar.

4. ($\eta^{\mathbb{S}}$ and $\mu^{\mathbb{S}}$ mod.) The modification condition for $\eta^{\mathbb{S}} : 1 \to \mathbb{S}$ is that for all categories lax **Pos**-functors $F : \mathcal{D} \to \mathcal{E}$ and objects Δ in T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{E})$, we have an equality of morphisms T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)(\mathbb{S}_{\mathcal{E}}(\Delta)) \to \mathbb{S}_{\mathcal{D}}(T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)(\Delta))$

$$\eta^{\mathbb{S}}_{\mathcal{D},T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)\Delta} = T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)(\eta^{\mathbb{S}}_{\mathcal{E},\Delta})$$

Note that this is a lax morphism of T-actions that lives in the category T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\mathcal{D})$, so this boils down to a component-wise equality over objects D of $\mathcal{D}$:

$$\eta^{\mathbb{S}}_{\mathcal{D},T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)\Delta,D} = T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)(\eta^{\mathbb{S}}_{\mathcal{E},\Delta})_D.$$

On the left side we see that

$$\begin{aligned} \eta^{\mathbb{S}}_{\mathcal{D},T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)\Delta,D} &= \eta^{S}_{T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)\Delta D} \\ &= \eta^{S}_{\Delta F D}, \end{aligned}$$

whereas on the right hand side we have

$$\begin{aligned} T\text{-}\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(F)(\eta^{\mathbb{S}}_{\mathcal{E},\Delta})_D &= (\eta^{\mathbb{S}}_{\mathcal{E},\Delta})_{FD} \\ &= \eta^{S}_{\Delta F D}, \end{aligned}$$

the desired equality. The case for $\mu^{\mathbb{S}}$ proceeds similarly.

5. (Monad laws) The left $\eta^{\mathbb{S}}$ law is the equality of modifications on $\mathbb{S}$:

$$\mu^{\mathbb{S}} \circ \eta^{\mathbb{S}}\mathbb{S} = \mathrm{Id}_{\mathbb{S}} \tag{6.4}$$

By unravelling this into something more concrete, we end up with the equality (quantified over all **Pos**-categories $\mathcal{D}$, T-actions Δ, and objects D of $\mathcal{D}$) of $\mathcal{C}$-morphisms:

$$\mu^S_{\Delta D} \circ \eta^S_{\Delta D} = \mathrm{id}_{\Delta D},$$

which is merely the left η^S law for the monad (S, η^S, μ^S) evaluated at ΔD. In fact, the same thing happens with the other monad laws for $(\mathbb{S}, \eta^{\mathbb{S}}, \mu^{\mathbb{S}})$—they correspond exactly to the monad laws for (S, η^S, μ^S).

On the other hand, suppose we start with a monad $(\mathbb{S}, \eta^{\mathbb{S}}, \mu^{\mathbb{S}})$ on T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(-)$. We construct a monad on $\mathcal{C}$ in two steps. First, by Lemma 6.16, we let S denote the functor on $\mathcal{C}$ corresponding to the natural transformation $\mathbb{S}$, and $\underline{S}$ the corresponding lax functorial extension. Then, we set

$$\eta^S_A = \eta^{\mathbb{S}}_{\underline{1},\underline{A},*} : A \to SA$$
$$\mu^S_A = \mu^{\mathbb{S}}_{\underline{1},\underline{A},*} : SSA \to SA.$$

Recall that $\underline{A}$ is the characteristic T-action over $\underline{1}$ corresponding to the object A in $\mathcal{C}$.

We first need to check that η^S and μ^S are natural and that they satisfy the monad laws. This will mean that (S, η^S, μ^S) is indeed a monad on $\mathcal{C}$. Then, since we know that $\underline{S}$ is a lax functorial extension, we will just need to verify that the lax η^S and μ^S conditions hold.

1. (nat. η^S, μ^S) Let $f : A \to B$ be a morphism in $\mathcal{C}$. Let $\underline{f} : \underline{A} \to \underline{B}$ be the corresponding morphism in T-$\mathbf{Act}(\underline{1})$, which is a subcategory of T-$\mathbf{Act}^{\mathsf{lax}}_{\mathsf{lax}}(\underline{1})$.

   Naturality of the component $\eta^{\mathbb{S}}_{\underline{1}}$ tells us that the square below commutes.

$$\begin{tikzcd}
\underline{A} \arrow[r, "\underline{f}"] \arrow[d, "\eta^{\mathbb{S}}_{\underline{1},\underline{A}}"'] & \underline{B} \arrow[d, "\eta^{\mathbb{S}}_{\underline{1},\underline{B}}"] \\
\eta^{\mathbb{S}}_{\underline{1}}\underline{A} \arrow[r, "\eta^{\mathbb{S}}_{\underline{1}}\underline{f}"'] & \eta^{\mathbb{S}}_{\underline{1}}\underline{B}
\end{tikzcd}$$

   This is a diagram of morphisms in T-$\mathbf{Act}(\underline{1})$, and by considering the sole component (at $*$) we derive the naturality square of η^S in $\mathcal{C}$. The case of μ^S is similar.

2. (Monad laws) We use a similar trick to prove the monad laws for (S, η^S, μ^S). Let A be an object of $\mathcal{C}$. Each of the monad laws for $(\mathbb{S}, \eta^{\mathbb{S}}, \mu^{\mathbb{S}})$, applied at $(\underline{1}, \underline{A}, *)$ results in the corresponding equation in $\mathcal{C}$.

3. (η^S cond.) As we saw earlier, the η^S condition on the monadic extension $\underline{S}$ is used to show that every $\eta^{\mathbb{S}}_{\mathcal{D},\Delta}$ is indeed a lax morphism $\Delta \to \underline{S} \circ \Delta$. The converse holds also.

   Suppose that $(\mathbb{S}, \eta^{\mathbb{S}}, \mu^{\mathbb{S}})$ is a monad. This means that every component $\eta^{\mathbb{S}}_{\mathcal{D},\Delta}$ is a lax morphism, for every category $\mathcal{D}$ and every lax T-action Δ.

   Now, let $p : A \nrightarrow B$ be a Kleisli morphism. We wish to show that the inclusion below exists (the lax (η^S) condition for $\underline{S}$):

$$T\eta^S \circ p \leq \underline{S}p \circ \eta^S$$

   We will consider p as a T-action over $\mathbf{2}$. By assumption, the component $\eta^{\mathbb{S}}_{\mathbf{2},\underline{p}}$ is a lax morphism $\underline{p} \to \underline{S} \circ \underline{p}$. Applying the lax morphism condition at the single base morphism e in $\mathbf{2}$, we get the lax inclusion below

$$\begin{tikzcd} A \arrow[r, "\eta^S_A"] \arrow[d, "p"'] \arrow[dr, phantom, "\Nearrow"] & SA \arrow[d, "\underline{S}p"] \\ TB \arrow[r, "T\eta^S_B"'] & TSB \end{tikzcd}$$

   And this is precisely the lax η^S condition on $\underline{S}$. If $(\mathbb{S}, \eta^{\mathbb{S}}, \mu^{\mathbb{S}})$ can be restricted to a monad on the subfunctor of strict T-action morphisms T-$\mathbf{Act}^{\mathsf{lax}}(-)$, then this inclusion holds strictly and we have the strict η^S condition.

4. (μ^S cond.) The case of μ^S is similar.

$\square$

Chapter 7

Distributive laws for lattice monads

In this chapter we will be investigating the case of T-actions when $T = \mathcal{P}_S$, for a semiring S. In order to have a suitable category of $\mathcal{P}_S$-actions, we need a **Pos**-enrichment of the Kleisli category. This corresponds to the imposition of *order axioms* on the semiring S. Since $\mathcal{P}$ is a semiring monad (over the boolean semiring $\mathbb{B}$), we might try to generalise the lax laws $\mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$ to laws $\mathcal{P}_S\mathcal{P}_S \to \mathcal{P}_S\mathcal{P}_S$. Unfortunately, the structure of a semiring with order is not enough.

We will see that it is the structure of a *bounded distributive monoidal lattice* $(S, \leq, \bot, \top, \wedge, \vee, 1, *)$ that allows us to define lax distributive laws $\ell^+, \ell^-, \ell^+ \wedge \ell^- : \mathcal{P}_S\mathcal{P}_S \to \mathcal{P}_S\mathcal{P}_S$.

Furthermore, we will see that meet and join of S induce the categorical product and coproduct in the categories $\mathcal{P}_S$-$\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$, and the multiplication induces a monoidal structure. Furthermore, when S is residuated, the left and right residuals induce left and right residual $\mathcal{P}_S$-actions, giving $\mathcal{P}_S$-$\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$ the structure of a *residuated category* [51]. When S is in fact complete (a quantale), then we may form an exponential object which makes $\mathcal{P}_S$-$\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$ cartesian closed.

When we take $S = \mathbb{B}$, the residuated and cartesian structure are the same. In more interesting cases, these residual spaces provide a notion of a "valued homotopy" between morphisms of S-valued transition systems, that extends the ideas of Section 3.3.

Definition 7.1. An *ordered semiring* consists of a semiring $(S, 0, 1, +, *)$ with a partial order $\leq$ that satisfies the following axioms:

- If $a \leq b$ then $a + c \leq b + c$
- If $a \leq b$ then $c * a \leq c * b$ and $a * c \leq b * c$

Example 7.2. The following semirings are ordered.

- $(\mathbb{B}, 0, 1, \vee, \wedge)$, with the standard order of $0 \leq 1$.
- $(\mathcal{P}(B), \emptyset, B, \cup, \cap)$, with the subset ordering.
- The min-plus semiring $(\mathbb{N} \cup \{\infty\}, \infty, 0, \min, +)$ with the ordering $\geq$. When we think of this semiring as being embedded in $\mathcal{P}(\mathbb{N}, +)$, the order $\geq$ is the subset ordering. This is because $\{n, n+1, \cdots\} \subseteq \{m, m+1, \cdots\}$ if and only if $n \geq m$. The extra point ∞ is maximal with respect to this order.
- The semiring of regular languages is ordered with language containment.

However $\mathbb{R}$ is not. The second condition only holds for $c \geq 0$, while we require it for all c.

Recall that the Kleisli morphisms $R : A \nrightarrow B$ of this category are essentially functions $R : A \times B \to S$ such that for every $a \in A$ there exist only finitely many $b \in B$ with $R(a, b)$ non-zero.

The enrichment on homsets is hence pointwise. For $R, S : A \nrightarrow B$ we say $R \leq S$ if for all $a \in A$ and $b \in B$ we have the inequality in S

$$R(a, b) \leq S(a, b) \tag{7.1}$$

This leads us to the following proposition.

Proposition 7.3. *Let $(S, 0, 1, +, *, \leq)$ be an ordered semiring. There is a canonical* ***Pos****-enrichment on the Kleisli category of $\mathcal{P}_S$.*

Proof. We require that composition is monotone. Let $R \leq R' : A \nrightarrow B$ and $S \leq S' : B \nrightarrow C$. We may calculate

$$\begin{aligned}(R \fatsemi S)(a, c) &= \sum_{b \in B} R(a, b) * S(b, c) \\ &\leq \sum_{b \in B} R'(a, b) * S'(b, c) \\ &= (r' \fatsemi S')(a, c).\end{aligned}$$

□

Note that when $S = \mathbb{B}$ this agrees with the standard **Pos**-enrichment of **Rel** (set containment of relations). For the remainder of this chapter, we shall only be considering the case where $(S, \leq, \bot, \top, \vee, \wedge, 1, *)$ is a bounded, distributive, monoidal lattice. We will let $\mathcal{P}_S$ denote the semiring monad of $(S, \bot, 1, \vee, *)$. The **Pos**-enrichment for the Kleisli category of $\mathcal{P}_S$ is given pointwise by $\leq$.

Another interesting example that we will be paying particular attention to is the min-plus semiring $\mathcal{N}$. We will begin by elaborating on a *representation result*. Recall that the min-plus semiring has elements $\mathbb{N} \cup \{\infty\}$. The semiring addition is min, and the multiplication is $+$. The unit of min is ∞ (for $\min(n, \infty) = n$ for all n, as $n \leq \infty$), while the unit of $+$ is 0.

Remark 7.4. The min-plus semiring can be embedded into the powerset lattice $\mathcal{P}(\mathbb{N})$ in the following way. We define a mapping $[\![-]\!] : \mathcal{N} \to \mathcal{P}(\mathbb{N})$ by

$$\begin{aligned} [\![n]\!] &= \{n, n+1, \cdots\} \\ [\![\infty]\!] &= \emptyset. \end{aligned}$$

Note that we may define an addition on subsets of natural numbers elementwise:

$$U + V = \{u + v : u \in U, v \in V\}.$$

Hence we see that

$$\begin{aligned} [\![n]\!] + [\![m]\!] &= [\![n+m]\!] \\ [\![n]\!] \cup [\![m]\!] &= [\![\min(n,m)]\!] \\ [\![n]\!] \cap [\![m]\!] &= [\![\max(n,m)]\!] \\ [\![n]\!] \subseteq [\![m]\!] &\iff n \geq m. \end{aligned}$$

This suggests that the min-plus semiring is an instance of a bounded, distributive lattice, ordered by $\geq$, and with meet and join given by max and min respectively. Addition is monoidal with respect to this structure.

An intuitive interpretation of this lattice is that it models *cost minimisation*. A transition system for this lattice is one where every transition has a cost that must be paid. A cost of ∞ is one that can *never* be paid—it represents impossibility. The join and meet of the lattice model *alternation*, or *simultaneous choice*. When presented with an choice of transitions, each with cost x_i, the join $\bigvee x_i = \min\{x_i\}$

represents the *best choice*, it is the minimal cost that must be paid. On the other hand, the meet $\bigwedge x_i = \max\{x_i\}$ is the *worst case scenario*. Addition models *sequencing*. If you have to pay a cost of x, and then a cost of y, then the total cost is $x + y$. The interpretation of $x \geq y$ is that y is *at least as possible* as x—if you can pay the cost of x, then one can certainly pay the cost of y.

7.1 Lax laws $\mathcal{P}^f\mathcal{P}_S \to \mathcal{P}_S\mathcal{P}^f$

The strategy here is to generalise the lax distributive laws of the powerset functor. We recall that there are two dual laws ℓ^+ and ℓ^- defined by the following rules.

$$\ell^+{}_A(F) = \{X : \forall U \in F.\ \exists x \in X.\ \ x \in U\}$$
$$\ell^-{}_A(F) = \{X : \forall x \in X.\ \exists U \in F.\ \ x \in U\}$$

In the case of $\mathbb{B}$, we can think of the universal quantification $\forall$ as being a "generalised conjunction", and the existential $\exists$ as implementing a "generalised disjunction". This suggests the following definition.

Definition 7.5. The lax laws $\ell^+, \ell^- : \mathcal{P}^f\mathcal{P}_S \to \mathcal{P}_S\mathcal{P}^f$ are defined as follows.

$$\ell^+_A(F)(V) = \bigwedge_{U \in F} \bigvee_{x \in V} U(x) \tag{7.2}$$
$$\ell^-_A(F)(V) = \bigwedge_{x \in V} \bigvee_{U \in F} U(x). \tag{7.3}$$

We shall elaborate on the meaning of Eq. 7.2. First of all, we can see that the type of F is $\mathcal{P}\mathcal{P}_S(A)$, so F is a finite set of S-valued subsets of A. The notation $\bigwedge_{U \in F}$ is well-defined, as there are only finitely many S-valued sets U contained in F. On the other hand, V is a finite subset of A.

We can alternatively express these as lax extensions of $\mathcal{P}^f S$ to $\mathbf{FinRel}_S$.

Proposition 7.6. *There are two lax extensions of $\mathcal{P}^f$ to $\boldsymbol{FinRel}_S$, denoted $\mathcal{P}^{S+}$*

and $\mathcal{P}^{S-}$. They are defined on S-valued relations $R : A \nrightarrow B$ in the following way:

$$\mathcal{P}^{S+}(R)(U,V) = \bigwedge_{a\in U} \bigvee_{b\in V} R(a,b) \tag{7.4}$$

$$\mathcal{P}^{S-}(R)(U,V) = \bigwedge_{b\in V} \bigvee_{a\in U} R(a,b) \tag{7.5}$$

Proof. Note that by definition of the meet and join, for any S-relation $R : A \nrightarrow B$, the lifted relation $\mathcal{P}^{S+}(R)$ is monotone in the second argument and anti-monotone in the first. The case of $\mathcal{P}^{S-}(R)$ is dual. That is, for $U \subseteq U' \subseteq A$ and $V \subseteq V' \subseteq B$ we have

$$\begin{aligned}
\mathcal{P}^{S+}(R)(U,V) &\leq \mathcal{P}^{S+}(R)(U,V') \\
\mathcal{P}^{S+}(R)(U,V) &\geq \mathcal{P}^{S+}(R)(U',V) \\
\mathcal{P}^{S-}(R)(U,V) &\geq \mathcal{P}^{S-}(R)(U,V') \\
\mathcal{P}^{S-}(R)(U,V) &\leq \mathcal{P}^{S-}(R)(U',V).
\end{aligned}$$

We will verify that $\mathcal{P}^{S+}$ and $\mathcal{P}^{S-}$ satisfy the conditions of a lax extension.

1. (mono.) Suppose that $R \leq Q : A \nrightarrow B$. This is pointwise, so for all $a \in A$ and $b \in B$ we have $R(a,b) \leq Q(a,b)$. As meet and join in a lattice are monotone, we may deduce that

$$\bigwedge_{a\in U} \bigvee_{b\in V} R(a,b) \leq \bigwedge_{a\in U} \bigvee_{b\in V} Q(a,b).$$

   Hence $\mathcal{P}^{S+}(R)(U,V) \leq \mathcal{P}^{S+}(Q)(U,V)$ for all $U \subseteq A$ and $V \subseteq B$. The case of $\mathcal{P}^{S-}$ is similar.

2. (lax id.) Note that

$$\eta^S_{\mathcal{P}_S A}(U,V) = \begin{cases} 1 & U = V \\ \bot & U \neq V \end{cases}.$$

   We know that $\mathcal{P}^{S+}(\eta^S_A)(U,V) = \bigwedge_{a\in U} \bigvee_{b\in V} \eta^S_A(a,b)$. Observe that

$$\bigvee_{b\in V} \eta^S_A(a,b) = \begin{cases} 1 & a \in V \\ \bot & a \notin V \end{cases}.$$

Therefore we deduce that

$$\mathcal{P}^{S+}(\eta^S_A)(U,V) = \begin{cases} 1 & U \subseteq V \\ \bot & \text{otherwise} \end{cases}.$$

From this it follows that $\eta^S_{\mathcal{P}_S A} \leq \mathcal{P}^{S+}(\eta^S_A)$. A similar argument shows that

$$\mathcal{P}^{S-}(\eta^S_A)(U,V) = \begin{cases} 1 & U \supseteq V \\ \bot & \text{otherwise} \end{cases},$$

and hence $\mathcal{P}^{S-}$ also fulfils the lax identity law.

3. (lax comp.) Let $R : A \nrightarrow B . Q : B \nrightarrow C$, and $U \subseteq A, V \subseteq C$. We recall that

$$(R \fatsemi Q)(a,c) = \bigvee_{b \in B} R(a,b) * Q(b,c)$$
$$\mathcal{P}^{S+}(R \fatsemi Q)(U,V) = \bigwedge_{a \in U} \bigvee_{c \in V, b \in B} R(a,b) * Q(b,c)$$
$$\mathcal{P}^{S+}(R) \fatsemi \mathcal{P}^{S+}(Q)(U,V) = \bigvee_{W \subseteq B} \left(\bigwedge_{a \in U} \bigvee_{b \in W} R(a,b) * \bigwedge_{b \in W} \bigvee_{c \in V} Q(b,c) \right)$$

We will show that

$$\mathcal{P}^{S+}(R) \fatsemi \mathcal{P}^{S+}(Q)(U,V) \leq \mathcal{P}^{S+}(R \fatsemi Q)(U,V).$$

We approach this inequality step by step. As the left hand side is a join, it is enough to show that every term is bounded above by the right hand side (as the join is the least upper bound): for all $W \subseteq B$ we require

$$\left(\bigwedge_{a \in U} \bigvee_{b \in W} R(a,b) * \bigwedge_{b \in W} \bigvee_{c \in V} Q(b,c) \right) \leq \bigwedge_{a \in U} \bigvee_{c \in V, b \in B} R(a,b) * Q(b,c).$$

Now the right hand side is a meet, so we will show that the left hand side is a lower bound for every term. That is, for all W and for all $a \in U$,

$$\left(\bigwedge_{a' \in U} \bigvee_{b \in W} R(a',b) * \bigwedge_{b \in W} \bigvee_{c \in V} Q(b,c) \right) \leq \bigvee_{c \in V, b \in B} R(a,b) * Q(b,c). \tag{7.6}$$

Note that the right hand side is equal to

$$\bigvee_{b \in B} R(a,b) * \bigvee_{b \in B, c \in V} Q(b,c).$$

One of the factors on left is a join over all the $a' \in A$, so this is a lower bound for the a term.

$$\bigwedge_{a' \in U} \bigvee_{b \in W} R(a',b) \leq \bigvee_{b \in B} R(a,b)$$

Similarly, we have that

$$\bigwedge_{b \in W} \bigvee_{c \in V} Q(b,c) \leq \bigvee_{c \in V, b \in B} Q(b,c).$$

We can put these last two inequalities together to get the desired inequality of Eq. 7.6. The proof for $\mathcal{P}^{S-}$ is symmetric.

4. (lax left whisk.) In this setting, the lax left whiskering condition amounts to the requirement that, for any $R : A \nrightarrow B$ and $g : B \to C$ we have

$$\mathcal{P}^{S+}(R) \fatsemi (\mathcal{P}g)_* \leq \mathcal{P}^{S+}(R \fatsemi g_*),$$

where $g_* : B \nrightarrow C$ is the S-valued relation given by

$$g_*(b,c) = \begin{cases} 1 & c = g(b) \\ \bot & \text{otherwise} \end{cases}.$$

So we can compute

$$\begin{aligned}
\left(\mathcal{P}^{S+}(R) \fatsemi (\mathcal{P}g)_*\right)(U,V) &= \bigvee_{W \subseteq B} \mathcal{P}^{S+}(R)(U,W) * \mathcal{P}g_*(W,V) \\
&= \bigvee_{W \in (\mathcal{P}g)^{-1}(V)} \mathcal{P}^{S+}(R)(U,W) \\
\mathcal{P}^{S+}(R \fatsemi g_*)(U,V) &= \bigwedge_{a \in U} \bigvee_{c \in V, b \in g^{-1}(c)} R(a,b) \\
&= \bigwedge_{a \in U} \bigvee_{b \in B} R(a,b) \\
&= \mathcal{P}^{S+}(R)(U,B)
\end{aligned}$$

The result follows from monotonicity of $\mathcal{P}^{S+}(R)$. The case of $\mathcal{P}^{S-}$ is similar.

□

We shall examine what happens in the min-plus case. Let $R : A \nrightarrow B$ be a $\mathcal{N}$-valued relation, and $U \subseteq A, V \subseteq B$. For every $a \in A$ and $b \in B$ the value $R(a,b) \in \mathcal{N}$ represents the *cost* of going from a to b. The expression of Eq. 7.4 amounts to

$$\mathcal{P}^{\mathcal{N}+}(R)(U,V) = \max_{a \in U} \min_{b \in V} R(a,b).$$

This indicates that the cost of going from a subset U to a subset V is calculated in two steps. First, for every possible starting point $a \in U$, the minimal cost of landing anywhere in V is calculated. This is $\min_{b \in V} R(a,b)$. The second step is to find the maximal such value.

This can be interpreted as an adversarial game. Your opponent makes the first move, choosing a state $a \in U$. But the second move is yours, and you may choose the "cheapest" $b \in V$ to move to.

An example is depicted in Fig. 7.1. The solid lines depict the cheapest transition from a given point in U to any point in V. The ultimate value of $\mathcal{P}^{\mathcal{N}+}(R)(U,V)$ would be 4, as this is the smallest value that will be able to pay for a transition to V, no matter where in U we start at.

It is worth mentioning that in this case the extension $\mathcal{P}^{\mathcal{N}+}$ is properly lax with respect to composition. Let $R : A \nrightarrow B$ and $S : B \nrightarrow C$ be as depicted in

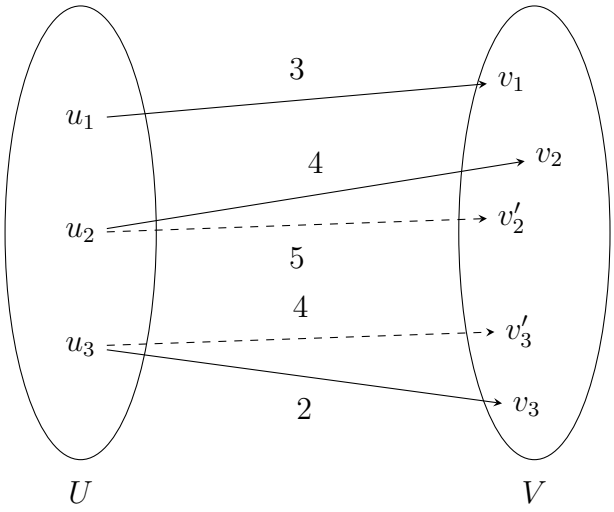


Figure 7.1: In this case, $\mathcal{P}^{\mathcal{N}+}(R)(U,V) = 4$.

Fig. 7.2. Clearly, we have that

$$\mathcal{P}^{\mathcal{N}+}(()R \fatsemi S)(A,C) = 3.$$

However, we may calculate that

$$(\mathcal{P}^{\mathcal{N}+}(R) \fatsemi \mathcal{P}^{\mathcal{N}+}(S))(A,C) = 4.$$

The interpretation is that having to play two rounds of the game gives your opponent an advantage. On the first round, she forces you to the top row, making you pick the most expensive transition $A \to B$, for a cost of 2. In the second round, she moves you to the bottom row, and again you have to take the transition with a cost of 2.

Calculating the composite $R \fatsemi S$ first gives you an advantage, as your opponent is not able to interfere in the middle of your move from A to C.

7.2 Cartesian closed structure

We shall see how to define the product and coproduct in every category $\mathcal{P}_S$-$\mathbf{Act}_{\mathsf{lax}}(\mathcal{D})$. To begin with, we take $\mathcal{D}$ to be an arbitrary $\mathbf{Pos}$-enriched category.

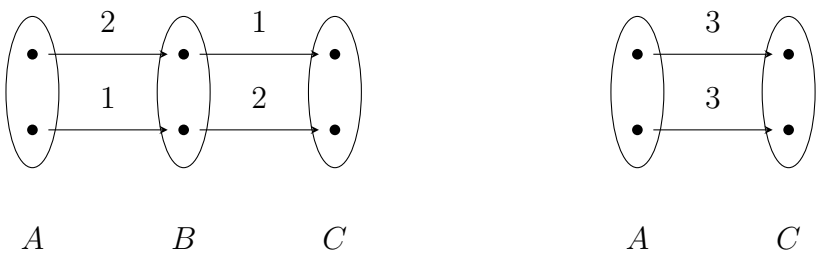


Figure 7.2: Two $\mathcal{N}$-relations, $R : A \nrightarrow B$ and $S : B \nrightarrow C$, and their composite.

Let $\Delta, \Gamma : \mathcal{D} \to \mathbf{FinRel}_S$ be two $\mathcal{P}_S$-actions on $\mathcal{D}$. This means that for every object D of $\mathcal{D}$ we have sets $\Delta D, \Gamma D$. And, for every morphism $\sigma : D \to E$ in $\mathcal{D}$, there are functions $\Delta_\sigma : \Delta D \to \mathcal{P}_S(\Delta E), \Gamma_\sigma : \Gamma D \to \mathcal{P}_S(\Gamma D)$. Note that we move the σ to subscript for readability.

We will, however, find the type signatures below more convenient.

$$\Delta_\sigma : \Delta D \times \Delta E \to S$$
$$\Gamma_\sigma : \Gamma D \times \Gamma E \to S.$$

This means that we can think of a $\mathcal{P}_S$ action Δ as providing, for every morphism $_\sigma : D \to E$ and every pair of elements $d \in \Delta D$ and $e \in \Delta E$, a value $\Delta_\sigma(d, e)$ in S.

The following result will help us reason about morphisms in this category.

Proposition 7.7. *Let $\{f_D : \Delta D \to \Gamma D\}_{\mathcal{D}}$ be a family of functions, indexed by the objects of $\mathcal{D}$. The family f is a lax morphism $\Gamma \to \Delta$ if and only if, for every $_\sigma : D \to E$ and every pair of elements $d \in \Delta D, e \in \Delta E$ the inequality below holds*

$$\Delta_\sigma(d, d') \leq \Gamma_\sigma(f(d), f(d')). \tag{7.7}$$

Proof. Recall that f is a lax morphism if for every$_\sigma : D \to E$, the inclusion in the diagram below exists.

$$\begin{tikzcd}
\Delta D, \arrow[r, "f_D"] \arrow[d, "\Delta_\sigma"'] \arrow[dr, phantom, "\Nearrow"] & \Gamma D \arrow[d, "\Gamma_\sigma"] \\
\mathcal{P}_S(\Delta E) \arrow[r, "\mathcal{P}_S f_E"] & \mathcal{P}_S(\Gamma E)
\end{tikzcd}$$

Concretely, this means that for all $d \in \Delta D$ and $e' \in \Gamma E$ we have

$$\mathcal{P}_S f_E(\Delta_\sigma(d))(e') \leq \Gamma_\sigma(f_D(d), e').$$

Rewriting the left hand side we end up with

$$\bigvee_{e \in f_E^{-1}(e')} \Delta_\sigma(d, d') \leq \Gamma_\sigma(f) D(d), e'), \tag{7.8}$$

quantified over all d, e'.

Suppose we have Eq. 7.7. Pick a d and e'. Then every term $\Delta_\sigma(d, d')$ in the join on the left hand side of Eq. 7.8 is bounded above by the value $\Gamma_\sigma(f_D(d), e')$ (by Eq. 7.7, since for each d' we have $f_E(d') = e'$).

On the other hand, suppose we have Eq. 7.8. Pick d, d', and let $e' = f_E(d')$. We have $\Delta_\sigma(d, d') \leq \mathcal{P}_S f_E(\Delta_\sigma(d))(e')$, so by Eq. 7.8 we deduce $\Delta_\sigma(d, d') \leq \Gamma_\sigma(f_D(d), e')$, as desired.

□

When $S = \mathbb{B}$, this is telling us that a morphism of transition systems is a function that preserves σ-transitions for all σ. In the S-valued case, this corresponds to preserving *order*.

Definition 7.8. The categorical product of Δ and Γ is denoted $\Delta \wedge \Gamma$. The action of $\Delta \wedge \Gamma$ on objects D is the product of sets, $\Delta D \times \Gamma D$. For morphisms ${}_\sigma : D \to E$ we define

$$(\Delta \wedge \Gamma)_\sigma : (\Delta D \times \Gamma D) \times (\Delta E \times \Gamma E) \to S$$
$$(\Delta \wedge \Gamma)_\sigma(d, d', e, e') = \Delta_\sigma(d, e) \wedge \Gamma_\sigma(d', e')$$

Explicitly, the definition above is saying that the product of two transition systems $\alpha : A \to \mathcal{P}A$ and $\beta : B \to \mathcal{P}B$ is the transition system on the set $A \times B$, with transitions given by the rule

$$(a, b) \to (a', b') \text{ if and only if } a \to a' \text{ and } b \to b',$$

which is the familiar definition of the product in **TS**.

Definition 7.9. The categorical coproduct is denoted $\Delta \vee \Gamma$. On objects this is

also the product of sets. For a morphism ${}_\sigma : D \to E$ we have

$$(\Delta \vee \Gamma)_\sigma : (\Delta D \times \Gamma D) \times (\Delta E \times \Gamma E) \to S$$
$$(\Delta \vee \Gamma)_\sigma(d, d', e, e') = \Delta_\sigma(d, e) \vee \Gamma_\sigma(d', e')$$

The pairing and copairing morphisms are constructed (pointwise) in the same way as in **Set**. We omit the proof that they are lax morphisms, as it is tedious and does not reveal anything interesting.

The next step is to construct an exponential object. Again, we will take the underlying set of $\Delta \Rightarrow \Gamma$ to be (pointwise) the set of all functions, not merely the morphisms.

Definition 7.10. The $\mathcal{P}_S$-action $\Delta \Rightarrow \Gamma$ is defined in the following way. For objects D we have

$$\Delta \Rightarrow \Gamma(D) = \{f : \Delta D \to \Gamma D\},$$

which is the exponential $\Delta D \Rightarrow \Gamma D$ in **Set**.

Let ${}_\sigma : D \to E$ be a morphism in $\mathcal{D}$. Then we will define

$$\Delta \Rightarrow \Gamma_\sigma(f, g) = n,$$

where $n \in S$ is the greatest element with

$$n \wedge \Delta_\sigma(d, d') \leq \Gamma_\sigma(f(d), g(d')), \tag{7.9}$$

for all $d, d' \in \Delta D$.

An alternative expression for the value of Eq. 7.9 is

$$(\Delta \Rightarrow \Gamma)_\sigma(f, g) = \bigvee \{n \in S \mid n \wedge \Delta_\sigma(d, d') \leq \Gamma_\sigma(f(d), g(d')) \text{ for all } d, d' \in \Delta D\}. \tag{7.10}$$

When S has all joins (i.e. when the semiring $(S, \perp, 1, \vee, *)$ is *complete* in the sense of Definition 2.13), the join of Eq. 7.10 is guaranteed to exist. It is at this point unclear if the condition of Eq. 7.9 is well-defined even in the case of a semiring that is *not* complete.

Proposition 7.11. *The category $\mathcal{P}_S$-$\boldsymbol{Act}_{lax}(\mathcal{D})$ is cartesian closed.*

Proof. Let $\Delta, \Gamma : \mathcal{D} \to \mathbf{FinRel}_S$ be $\mathcal{P}_S$-actions. We saw in Definition 7.8 that the product is given by $\vee$, and we described the construction of $\Delta \Rightarrow \Gamma$ in Definition 7.10.

We will now describe the evaluation and currying maps, and prove that they are morphisms. The type of ev will be $\Delta \wedge (\Delta \Rightarrow \Gamma) \to \Gamma$. The components of ev are

$$\begin{aligned} \mathrm{ev}_D &: \Delta D \times (\Delta D \Rightarrow \Gamma D) \to \Gamma D \\ \mathrm{ev}_D(d, f) &= f(d) \end{aligned}$$

We need to verify that this is a morphism. We will check that it "preserves transitions" in the way of Proposition 7.7. This means that the inequality below must hold

$$\Delta_\sigma(d, e)(\Delta \Rightarrow \Gamma)_\sigma(f, g) \leq \Gamma_\sigma(f(d), g(e)).$$

But this holds precisely by Eq. 7.9!

Now, let $g : \Theta \wedge \Delta \to \Gamma$ be a morphism. We define $\lambda g : \Theta \to \Delta \Rightarrow \Gamma$ by currying componentwise,

$$\begin{aligned} (\lambda g)_D &: \Theta D \to \Delta D \Rightarrow \Gamma D \\ (\lambda g)_D &= \lambda(g_D). \end{aligned}$$

This certainly has the right type. All we need to worry about is whether the curried λg is indeed a morphism. We will check that it preserves transitions. We require that for every $_\sigma : D \to E$ and every $d \in \Theta D$ and $e \in \Theta E$ that

$$\Theta_\sigma(d, e) \leq (\Delta \Rightarrow \Gamma)_\sigma(\lambda g d, \lambda g e).$$

Note that for any $d' \in \Delta D$, $e' \in \Delta E$ we have

$$\Delta_\sigma(d', e') \wedge \Theta_\sigma(d, e) \leq \Gamma_\sigma(g(d, d'), g(e, e')),$$

because g is a lax morphism. The right hand side of this inequality is

$$\Gamma_\sigma(\lambda g d(d'), \lambda g e(e')).$$

So by taking $n = \Theta_\sigma(d, e)$ we see that

$$n \wedge \Delta(d', e') \leq \Gamma_\sigma(\lambda g d(d'), \lambda g e(e'))$$

for all d', e'. Thus the value of $(\Delta \Rightarrow \Gamma)_\sigma(\lambda g d, \lambda g e)$ must be greater than or equal to n, which gives the desired inequality

$$\Theta_\sigma(d, e) \leq (\Delta \Rightarrow \Gamma)_\sigma(\lambda g d, \lambda g e).$$

□

Example 7.12. Consider the following example in the min-plus semiring. Let f be the *function* $A \to B$ depicted in Fig. 7.3

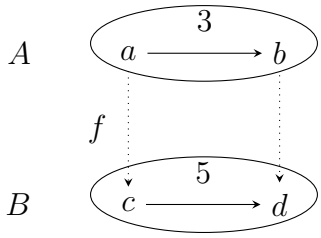


Figure 7.3: The function $f : A \to B$.

The function f is *not* a morphism. It is not transition preserving, as $3 \not\geq 5$. The "resource" intuition is helpful here. In A, we can make the transition $a \to b$ for a cost of 3. The corresponding transition $c \to d$ in B meanwhile, requires a cost of 5 to be paid. So being able to make $a \to b$ does not guarantee that we can also make $c \to d$.
By applying the formula of Eq. 7.9 we can deduce that the value of $A \Rightarrow B(f, f)$ is 5, as we have

$$\max(5, 3) \geq 5$$

and 5 is minimal (that is, greatest with respect to $\geq$) with this property.

7.3 Additional residuated structure

Following the previous section, we may also lift the multiplicative structure of S to the level of $\mathcal{P}_S$-actions.

Definition 7.13. Let Δ, Γ be $\mathcal{P}_S$-actions on $\mathcal{D}$. The action $\Delta * \Gamma$ is defined by

$$(\Delta * \Gamma)D = \Delta D \times \Gamma D$$
$$(\Delta * \Gamma)_\sigma(d, d', e, e') = \Delta_\sigma(d, e) * \Gamma_\sigma(d', e').$$

Proposition 7.14. *The construct $\Delta * \Gamma$ can be extended (pointwise) to a monoidal functor on $\mathcal{P}_S$-$\boldsymbol{Act}_{\mathsf{lax}}(\mathcal{D})$. The unit is given by $\mathbf{1}$, where*

$$\mathbf{1} : \mathcal{D} \to \boldsymbol{FinRel}_S$$
$$\mathbf{1}D = \{\star\}$$
$$\mathbf{1}(\sigma : D \to E) = \eta^S(1),$$

where 1 is the monoidal unit of S.

When $*$ is commutative on S, this functor is symmetric monoidal.

Furthermore, suppose that the lattice S is actually residuated, with left and right residuals given by $c \diagup a$ and $b \diagdown c$. We are able to define the residuals of $\mathcal{P}_S$-actions. These will satisfy the isomorphisms of homsets

$$\mathrm{Hom}(\Delta * \Gamma, \Theta) \cong \mathrm{Hom}(\Gamma, \Delta \diagdown \Theta) \cong \mathrm{Hom}(\Delta, \Theta \diagup \Gamma).$$

Definition 7.15. Let Δ, Γ be $\mathcal{P}_S$ actions on $\mathcal{D}$. On objects D, the left and right residuals are equal. They are (pointwise) the set of all functions $\Delta D \to \Gamma D$.

$$(\Gamma \diagup \Delta)D = \Delta D \Rightarrow \Gamma D$$
$$(\Delta \diagdown \Gamma)D = \Delta D \Rightarrow \Gamma D$$

Let $\sigma : D \to E$ be a morphism. Let $f : \Delta D \to \Gamma D$ and $g : \Delta E \to \Gamma E$. We define the residuals componentwise by

$$(\Gamma \diagup \Delta)_\sigma(f, g) = \bigwedge_{d,e} (\Gamma_\sigma(f(d), g(e)) \diagup \Delta_\sigma(d, e))$$
$$(\Delta \diagdown \Gamma)_\sigma(f, g) = \bigwedge_{d,e} (\Delta_\sigma(d, e) \diagdown \Gamma_\sigma(f(d), g(e)))\,,$$

where d ranges over ΔD and e over ΔE.

Note that, by the definition of residuation in S, the residual spaces have the property that for every $\sigma : D \to E$, $d \in \Delta D$, $e \in \Delta E$, $f : \Delta D \to \Gamma D$, and $g : \Delta E \to \Gamma E$, we have

$$(\Gamma \diagup \Delta)_\sigma(f, g) * \Delta_\sigma(d, e) \leq \Gamma_\sigma(f(d), g(e)) \tag{7.11}$$

$$\Delta_\sigma(d, e) * (\Delta \diagdown \Gamma)_\sigma(f, g) \leq \Gamma_\sigma(f(d), g(e)), \tag{7.12}$$

and $(\Gamma \diagup \Delta)_\sigma(f, g)$ and $(\Delta \diagdown \Gamma)_\sigma(f, g)$ are maximal in S with regard to this property.

Proposition 7.16. *The multiplication $- * -$ and residuals $- \diagup -$ and $- \diagdown -$ give $\mathcal{P}_S$-$\boldsymbol{Act}_{\mathsf{lax}}(\mathcal{D})$ the structure of a* residuated category*.*

Proof. We construct left and right evaluation morphisms $\mathrm{lev} : (\Gamma \diagup \Delta) * \Delta \to \Gamma$ and $\mathrm{rev} : \Delta * (\Delta \diagdown \Gamma) \to \Gamma$ by

$$\mathrm{lev}_D(f, d) = f(d)$$
$$\mathrm{rev}_D(d, f) = f(d).$$

Similarly, we define left and right currying. Let $g : \Theta * \Delta \to \Gamma$ and $h : \Delta * \Theta \to \Gamma$. Then we define the morphisms $\lambda g : \Theta \to \Gamma \diagup \Delta$ and $\rho h : \Theta \to \Delta \diagdown \Gamma$ pointwise by

$$(\lambda g)_D(c) = d \mapsto g(c, d)$$
$$(\rho h)_D(c) = d \mapsto h(d, c).$$

We need to verify that $\mathrm{ev}, \mathrm{lev}$ and $\lambda g, \rho h$ are lax morphisms. Let $\sigma : D \to E$, $d \in \Delta D, e \in \Delta E, f : \Delta D \to \Gamma D, g : \Delta E \to \Gamma E$. Then we calculate

$$\begin{aligned} ((\Gamma \diagup \Delta) * \Delta)_\sigma(f, d, g, e) &= (\Gamma \diagup \Delta)_\sigma(f, g) * \Delta_\sigma(d, e) \\ &\leq \Gamma_\sigma(f(d), g(e)) && \text{(by Eq. 7.11)} \end{aligned}$$

The case of rev proceeds symmetrically.

Now we need to show that for $c \in \Theta D$ and $c' \in \Theta E$ we have

$$\Theta_\sigma(c, c') \leq (\Gamma \diagup \Delta)_\sigma(\lambda g c, \lambda g c').$$

This will hold if we have

$$\begin{aligned}\Theta_\sigma(c, c') * \Delta_\sigma(d, e) &\leq \Gamma_\sigma(\lambda gc(d), \lambda gc'(e)) \\ &= \Gamma_\sigma(g(c, d), g(c', e)),\end{aligned}$$

because $(\Gamma \diagup \Delta)_\sigma(\lambda gc, \lambda gc')$ must be minimal with respect to this property. But this inequality holds because g is a morphism! The case of ρh is similar. □

The following result is a generalisation of Proposition 3.22.

Proposition 7.17. *Let $\{f_D : \Delta D \to \Gamma D\}_{\mathcal{D}}$ be a family of functions. The following are equivalent.*

1. *f is a lax morphism $\Delta \to \Gamma$*
2. *$1 \leq (\Gamma \diagup \Delta)_\sigma(f, f)$ for all $\sigma : D \to E$ in $\mathcal{D}$*
3. *$1 \leq (\Delta \diagdown \Gamma)_\sigma(f, f)$ for all $\sigma : D \to E$ in $\mathcal{D}$*

Proof.

- (1 $\implies$ 2). Suppose that f is a lax morphism. This means that $\Delta_\sigma(d, e) \leq \Gamma_\sigma(f_D(d), f_E(e))$ for all σ, d, e. As 1 is the unit of $*$, we deduce

$$1 * \Delta_\sigma(d, e) \leq \Gamma_\sigma(f_D(d), f_E(e)).$$

  This immediately implies $1 \leq (\Gamma \diagup \Delta)_\sigma(f, f)$, as that is the maximal x with $x * \Delta_\sigma(d, e) \leq \Gamma_\sigma(f_D(d), f_E(e))$.

- (2 $\implies$ 1). Suppose that $1 \leq (\Gamma \diagup \Delta)_\sigma(f, f)$ for all σ. By multiplying on the right with $\Delta_\sigma(d, e)$ (for arbitrary d, e, we deduce

$$\Delta_\sigma(d, e) \leq (\Gamma \diagup \Delta)_\sigma(f, f) * \Delta_\sigma(d, e).$$

  Now, the right hand side is bounded above by $\Gamma_\sigma(f_D(d), f_E(e))$, by Eq. 7.11. So it follows that f is a morphism.

- The case of 1 $\iff$ 3 is symmetric.

Example 7.18. Let us consider the case of Fig. 7.3 again. Recall that in the min-plus semiring, the left and right residuals are both given by truncated subtraction.

Hence we may compute that the value of $A \diagup B(f, f) = B \diagdown A(f, f)$ is 2, as we have

$$2 + 3 \geq 5,$$

and 2 is minimal (maximal with respect to $\geq$) with this property.

$\square$

Chapter 8

Conclusion

In this thesis we have seen a construction of T-actions that unifies many existing notions of "categories of transition systems". The categories that we are able to express include

- the category of labeled transition systems with edge-preserving functions, $\mathbf{TS}_\Sigma$
- as a special case, the categories of directed and undirected graphs
- via a monad on $\mathbf{TS}_\Sigma$, categories of transition systems with simulations and other relational morphisms
- categories of T-coalgebras, with both coalgebra morphisms and *lax* cohomomorphisms. These can be thought of as transition systems for the transition type T, with transition preserving functions as morphisms.
- in particular categories of S-valued transition systems (as coalgebras for the $\mathcal{P}_S$ monad)
- categories of monoid actions $\mathbf{Act}(M)$ with morphisms of actions.

We have examined the particular structure of $\mathbf{TS}_\Sigma$ in some detail. In particular, we have generalised the cartesian closed structure of $\mathbf{Gph}$ to the directed and labelled case. Following in this vein, we have taken the "homotopic view" of transitions $f \xrightarrow{\sigma} g$ in the exponential object $\mathcal{A} \Rightarrow \mathcal{B}$, and extended results of $\times$-homotopy theory for undirected graphs to $\mathbf{TS}_\Sigma$, in particular the "spider moves" decomposition of [13].

We presented a generalisation of the result of [39], that simulations of transition systems are in fact Kleisli morphisms of the Sim monad on $\mathbf{TS}_\Sigma$. We showed that this monad is formed from a *lax extension* of $\mathcal{P}$ to **Rel**, and that there is a dual extension which induces the monad of *reverse simulations*, RevSim.

We formulated a notion of lax distributive laws in a **Pos**-enriched setting, and present a well-behaved correspondence theorem (Theorem 4.28) between lax extensions $\underline{S}$ of S to $\mathcal{C}_T$ and lax distributive laws $\lambda : ST \to TS$. We take the approach of [53]— which presents a similar result for the case of $T = \mathcal{P}_Q$ the monad of a quantale—of examining precisely which axioms of distributive laws correspond to which properties of a lax extension. In particular, we show that the T-*relators* of Thijs [52] can be understood as lax extensions of T to **Rel**. The lax extension $\mathcal{P}^+$ of $\mathcal{P}$ induces both the $\mathcal{P}$ *relator* of simulations, and the *monad* Sim of simulations.

The latter half of the thesis is concerned with generalising the construction of Sim to construct monads on categories of transition systems for other transition types T. We take a higher categorical approach, defining a notion of a (lax) T-action on a **Pos**-category $\mathcal{D}$. These are essentially lax functors $\mathcal{D} \to \mathcal{C}_T$. We showed that the appropriate notion of morphism generalises *lax cohomomorphisms* of T-coalgebras and morphisms of monoid actions. The collection of all such T-actions forms a category T-$\mathbf{Act}(\mathcal{D})$. We showed further that this construction is *functorial* in $\mathcal{D}$, and hence formulate a contravariant functor T-$\mathbf{Act}(-)$: $\mathbf{PosCat}^{\mathrm{op}} \to \mathbf{Cat}$.

We presented a Yoneda-style result in Lemma 6.16, showing that natural transformations $\mathbb{S} : T\text{-}\mathbf{Act}(-) \to T\text{-}\mathbf{Act}(-)$ are in correspondence with *lax functorial extensions* $\underline{S}$ of S to $\mathcal{C}_T$. This holds because T-$\mathbf{Act}(-)$ is almost a hom-functor. When S is a monad, we proved (Theorem 6.17) that there is a correspondence between lax monad extensions and lax monad distributive laws. We exhibited Sim as an instance of this construction on $\mathcal{P}$-$\mathbf{Act}_{\mathsf{lax}}(-)$, induced by the extension $\mathcal{P}^+$.

Finally, we took a closer look at the case of $\mathcal{P}_S$-valued transition systems, where S is a distributive monoidal lattice. The case of $\mathcal{P}_Q$ for Q a quantale is relatively well-explored, but many good results can be recovered in this weaker setting. We present a pair of lax distributive law $\ell^+, \ell^- : \mathcal{P}^f\mathcal{P}_S \to \mathcal{P}_S\mathcal{P}^f$ that generalise the laws $\ell^+, \ell^- : \mathcal{P}\mathcal{P} \to \mathcal{P}\mathcal{P}$. The ℓ^- law is a *finitary* version of a well known lax distributive law $\mathcal{P}\mathcal{P}_Q \to \mathcal{P}_Q\mathcal{P}$ [33].

We demonstrated that when S has all joins, there is a cartesian closed structure on the category of $\mathcal{P}_S$-transition systems. Furthermore, when S is a residuated lattice, the residuated structure lifts to the category of transition systems also. Both of these constructions agree with the cartesian closed structure of $\mathbf{TS}_\Sigma$ for $S = \mathbb{B}$, and provide novel notions of homotopy for functions of S-valued transition systems.

8.1 Future Work

There are several opportunities for future work.

- There is much more work to be done in extending results of $\times$-homotopy for undirected graphs to the directed, labelled case of $\mathbf{TS}_\Sigma$. Desirable results could include finding a good notion of homotopy equivalence of systems, defining higher path categories, and understanding what properties of transition systems are preserved by homotopy. It would be good to see whether the "spider moves" results of Section 3.3 could be generalised to the particular case of homotopies for $\mathcal{P}_S$-systems, introduced in Chapter 7.

- The categorical structure of $\mathbf{TSS}_\Sigma$ is not fully understood. It is not a perfect analog to **Rel** as it does not have a dagger structure (the converse of a simulation is not, in general a simulation). Furthermore, the tensor product is not symmetric monoidal *closed*. It might also be worth investigating the subcategories of $\mathbf{TSS}_\Sigma$ comprising of the reflexive or symmetric systems, and seeing if there are any adjunctions like those of Propositions 3.28 and 3.34. One could also try to develop a homotopy theory for *simulations* and relational morphisms.

- We have explicitly considered only a small number of lax distributive laws. There are several more lax laws known in the literature. An area of future work would be to apply the techniques of this thesis and see what sort of monads of transition systems they correspond to. Recent work [20] suggests a notion of S-valued bisimulations for S-valued systems—this could correspond to a lax law $\mathcal{P}_S\mathcal{P}_S \to \mathcal{P}_S\mathcal{P}_S$.

- In particular, it would be interesting to explore the connections between the work of this thesis and the field of *monoidal topology* [42, 48, 53, 33], which provides another use-case for lax distributive laws.

- It may be possible to generalise the setting of lax distributive laws even further. The work of this thesis is framed in terms of **Pos**-enrichment, but it may be possible to enrich in *preordered sets* (like the *order-endowed functors* of [15]), or even work in an arbitrary 2-category. It is at this stage unclear what conditions on a 2-category are required to formulate the correspondence of lax distributive laws and lax extensions.

Bibliography

[1] Samson Abramsky and Bob Coecke. "A Categorical Semantics of Quantum Protocols". In: (5th Mar. 2007). arXiv: quant-ph/0402130.

[2] Peter Aczel and Nax Mendler. "A Final Coalgebra Theorem". In: *Category Theory and Computer Science*. Lecture Notes in Computer Science. 1989, pp. 357–365.

[3] Jiří Adámek, Horst Herrlich and George E. Strecker. *Abstract and Concrete Categories*. Reprints in Theory and Applications of Categories 17. 1990.

[4] E. Babson et al. "A Homotopy Theory for Graphs". In: *arXiv:math/0403146* (Mar. 2004). eprint: math/0403146.

[5] Michael Barr. "Relational Algebras". In: *Reports of the Midwest Category Seminar*. 1969.

[6] Falk Bartels, Ana Sokolova and Erik Vink. "A Hierarchy of Probabilistic System Types". In: *Electronic Notes in Theoretical Computer Science* 327 (Oct. 2004), pp. 57–75.

[7] Jon Beck. "Distributive Laws". In: *Seminar on Triples and Categorical Homology Theory*. Lecture Notes in Mathematics. 1969, pp. 119–140.

[8] N. Bezhanishvili, G. Fontaine and Y. Venema. "Vietoris Bisimulations". In: *Journal of Logic and Computation* 20.5 (2010), pp. 1017–1040.

[9] Patrick Blackburn, Maarten de Rijke and Yde Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science 53. 2001.

[10] Filippo Bonchi et al. "A Coalgebraic Perspective on Linear Weighted Automata". In: *Information and Computation* 211 (2012), pp. 77–105.

[11] Marcello M. Bonsangue et al. "Presenting Distributive Laws". In: *Logical Methods in Computer Science* Volume 11, Issue 3 (7th Aug. 2015), p. 1578. ISSN: 1860-5974. arXiv: 1503.02447 [cs]. URL: http://arxiv.org/abs/1503.02447.

[12] Peter Buchholz. "Bisimulation Relations for Weighted Automata". In: *Theoretical Computer Science* 393.1 (2008), pp. 109–123.

[13] Tien Chih and Laura Scull. "A Homotopy Category for Graphs". In: (May 2020). arXiv: 1901.01619 [math].

[14] Corina Cirstea. *Linear Time Logics – A Coalgebraic Perspective*. Sept. 2021. arXiv: 1612.07844 [cs].

[15] Andrea Corradini, Martin Große-Rhode and Reiko Heckel. "A Coalgebraic Presentation of Structured Transition Systems". In: *Theoretical Computer Science* 260.1-2 (June 2001), pp. 27–55.

[16] E. P. de Vink and J. J. M. M. Rutten. "Bisimulation for Probabilistic Transition Systems: A Coalgebraic Approach". In: *Theoretical Computer Science* 221.1 (1999), pp. 271–293.

[17] Anton Dochtermann. "Hom Complexes and Homotopy Theory in the Category of Graphs". In: *arXiv:math/0605275* (July 2008). eprint: math/0605275.

[18] Anton Dochtermann. *Homotopy Groups of Hom Complexes of Graphs*. July 2008. arXiv: arXiv:0705.2620.

[19] Jean-Marie Droz. "Quillen Model Structures on the Category of Graphs". In: *arXiv:1209.2699 [math]* (Sept. 2012). arXiv: 1209.2699.

[20] Yibin Du and Ping Zhu. "Fuzzy Approximations of Fuzzy Relational Structures". In: *International Journal of Approximate Reasoning* 98 (July 2018), pp. 1–10.

[21] Samuel Eilenberg. *Automata, Languages, and Machines*. Pure and Applied Mathematics. Academic Press, 1974.

[22] Sebastian Enqvist and Sumit Sourabh. "Bisimulations for coalgebras on Stone spaces". In: *Journal of Logic and Computation* 28.6 (2018), pp. 991–1010.

[23] Richard Garner. "The Vietoris Monad and Weak Distributive Laws". In: *Applied Categorical Structures* 28.2 (Apr. 2020), pp. 339–354. arXiv: 1811.00214 [math].

[24] Chris Godsil and Gordon Royle. *Algebraic Graph Theory*. Vol. 207. Graduate Texts in Mathematics. 2001.

[25] Alexandre Goy. "On the Compositionality of Monads via Weak Distributive Laws." PhD thesis. Université Paris-Saclay, 2021.

[26] Marco Grandis. *Higher Dimensional Categories: From Double to Multiple Categories*. World Scientific, Sept. 2019.

[27] Rachel Hardeman. "The Lifting Properties of A-Homotopy Theory". In: *arXiv:1904.12065 [math]* (Oct. 2019). arXiv: 1904.12065 [math].

[28] P. Johnstone, J. Power and T. Tsujishita. "An Axiomatics for Categories of Transition Systems as Coalgebras". In: *Proceedings. Thirteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No.98CB36226)*. 1998, pp. 207–213.

[29] Peter Johnstone et al. "On the Structure of Categories of Coalgebras". In: *Theoretical Computer Science* 260.1-2 (June 2001), pp. 87–117.

[30] G M Kelly. "Basic Concepts of Enriched Category Theory". In: *Reprints in Theory and Applications of Categories* 10 (2005).

[31] M. Kil'p, U. Knauer and A. V. Mikhalev. *Monoids, Acts, and Categories: With Applications to Wreath Products and Graphs: A Handbook for Students and Researchers*. De Gruyter Expositions in Mathematics 29. 2000.

[32] Alexander Kurz and Jiří Velebil. "Relation Lifting, a Survey". In: *Journal of Logical and Algebraic Methods in Programming* 85.4 (June 2016), pp. 475–499.

[33] Hongliang Lai and Walter Tholen. *Quantale-Valued Approach Spaces via Closure and Convergence*. 2016. eprint: arXiv:1604.08813.

[34] Kim G. Larsen and Arne Skou. "Bisimulation through Probabilistic Testing". In: *Information and Computation* 94.1 (Sept. 1991), pp. 1–28.

[35] F. William Lawvere. "Metric Spaces, Generalized Logic, and Closed Categories". In: *Rendiconti del Seminario Matematico e Fisico di Milano* 43.1 (Dec. 1973), pp. 135–166.

[36] Tom Leinster. *Basic Bicategories*. Oct. 1998. arXiv: math/9810017.

[37] Paul Blain Levy. "Similarity Quotients as Final Coalgebras". In: *Foundations of Software Science and Computational Structures*. Vol. 6604. 2011, pp. 27–41.

[38] Saunders Mac Lane. *Categories for the Working Mathematician*. 2nd ed. Graduate Texts in Mathematics 5. 1998.

[39] Pasquale Malacaria. "Equivalences of Transition Systems in an Algebraic Framework". In: *Algebraic Methodology and Software Technology (AMAST'93)*. 1994, pp. 263–270.

[40] Takahiro Matsushita. "Box Complexes and Homotopy Theory of Graphs". In: *arXiv:1605.06222 [math]* (July 2017).

[41] Eugenio Moggi. "Notions of Computation and Monads". In: *Information and Computation* 93.1 (1991), pp. 55–92.

[42] "Monoidal Topology: A Categorical Approach to Order, Metric, and Topology". In: July 2014.

[43] José N. Oliveira. "Weighted Automata as Coalgebras in Categories of Matrices". In: *International Journal of Foundations of Computer Science* 24.06 (Sept. 2013), pp. 709–728.

[44] Gordon D Plotkin. "A Structural Approach to Operational Semantics". In: *The Journal of Logic and Algebraic Programming* 60–61 (2004), pp. 17–139.

[45] J.J.M.M. Rutten. "Universal Coalgebra: A Theory of Systems". In: *Theoretical Computer Science* 249.1 (Oct. 2000), pp. 3–80.

[46] Jan J. M. M. Rutten and Daniele Turi. "On the Foundations of Final Semantics: Non-standard Sets, Metric Spaces, Partial Orders". In: *Semantics: Foundations and Applications*. Vol. 666. 1993, pp. 477–530.

[47] Gunther Schmidt and Thomas Ströhlein. "Relations and Graphs". In: *EATCS Monographs on Theoretical Computer Science*. 1993.

[48] C. Schubert. "Lax Algebras: A Scenic Approach". PhD thesis. Universität Bremen, 2006.

[49] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 2013.

[50] Sam Staton. *Relating Coalgebraic Notions of Bisimulation*. arXiv: 1101.4223v1.

[51] M. E. Szabo. *Algebra of Proofs*. Studies in Logic and the Foundations of Mathematics v. 88. 1978.

[52] Albert Marchienus Thijs. "Simulation and Fixpoint Semantics". PhD thesis. University of Groningen, 1996.

[53] Walter Tholen. *Lax Distributive Laws for Topology, I*. Aug. 2016. arXiv: 1603.06251 [math].

[54] Daniele Varacca. "Probability, Nondeterminism and Concurrency: Two Denotational Models for Probabilistic Computation". PhD thesis. Université Paris Diderot, Apr. 2004.

[55] Sebastiano Vigna. "A Guided Tour in the Topos of Graphs". In: (June 2003). arXiv: math/0306394.

[56] Kim Ritter Wagner. "Liminf Convergence in Ω-categories". In: *Theoretical Computer Science* 184.1-2 (1997), pp. 61–104.

[57] Maaike Zwart and Dan Marsden. "No-Go Theorems for Distributive Laws". In: *Logical Methods in Computer Science* Volume 18, Issue 1 (Jan. 2022).