Chapter 7

Presenting distributive laws

In the current chapter, we study distributive laws of monads over functors. These capture interaction between algebraic structure and observable behaviour in a systematic way. There are several benefits of this approach, recalled in more detail in Section 3.5: a distributive law canonically induces an algebra on the final coalgebra, provides a compositional semantics, and yields solutions to recursive equations. Moreover, distributive laws play a central role in the framework of up-to techniques introduced in the first part of this thesis.

However, concretely describing a distributive law of a monad over a functor and proving the associated axioms can be rather complicated. Instead, one may try to use general methods for constructing distributive laws from simpler ingredients. An important example of this is given by abstract GSOS, where distributive laws are represented by plain natural transformations. Further, in [HK11] it was shown how an abstract GSOS specification for a functor B can be lifted to one for the functor $(B-)^A$ which describes B-systems with input in A. Another method, which works for all monads on Set but only for certain polynomial behaviour functors B, produces a distributive law inducing a "pointwise lifting" of the algebra structure to B-behaviours [Jac06b, SBBR13].

But many examples do not fit into the above mentioned settings. A motivating example for the current chapter is that of context-free grammars, where sequential composition is not a pointwise operation and whose formal semantics satisfies the axioms of idempotent semirings, which is not a free monad. More generally, one may be interested in distributive laws involving a monad that arises as the quotient of a free monad with respect to some equations.

We give a general approach for constructing a distributive law $\lambda^{\mathcal{E}}$ for a monad $\mathcal{T}^{\mathcal{E}}$, which is presented as a quotient of a monad $\mathcal{T}$ by some equations $\mathcal{E}$, from a distributive law λ for the monad $\mathcal{T}$. In the typical application of our result, $\mathcal{T}$ is a free monad, so that λ can in turn be defined in terms of an abstract GSOS specification. Then $\lambda^{\mathcal{E}}$ is obtained as a certain quotient of λ by the equations $\mathcal{E}$, hence we say that $\lambda^{\mathcal{E}}$ *is presented by* λ *and the equations* $\mathcal{E}$. We show that such quotients exist when the distributive law *preserves the equations* $\mathcal{E}$, which roughly means that congruences generated by the equations are bisimulations. We also discuss how these quotients of distributive laws give rise to quotients of bialgebras, thereby giving a concrete operational interpretation. As an illustration and application, we show the existence of a distributive law of the monad for idempotent semirings over the deterministic automata functor. This result yields the equivalence between the representation of context-free languages via grammars in Greibach normal form and the coalgebraic representation via context-free expressions given in [WBR13].

Outline. In the next section, we describe in detail how to construct the quotient of a monad with respect to some given equations. In Section 7.2, we prove our main results on quotients of distributive laws. Then, in Section 7.3 we show that such quotients induce quotients of bialgebras. Finally, in Section 7.4 we discuss related work, and provide some directions for future work.

7.1 Quotients of monads

Let $\mathcal{T} = (T, \eta, \mu)$ be a monad on a category $\mathcal{C}$. For a general notion of equations on a monad, we define *$\mathcal{T}$-equations* or *equations for $\mathcal{T}$* as a 3-tuple $\mathcal{E} = (E, l, r)$ where E is an endofunctor on $\mathcal{C}$ and $l, r \colon E \Rightarrow T$ are natural transformations. The intuition is that E models the arity of the equations, i.e., the (number of) variables occurring in each equation, and l and r give the left and right-hand side. The advantage of using natural transformations (over, say, a subset of $TV \times TV$ for some set of variables V, or a generalization thereof) is that this approach defines equations on TX uniformly over any set X.

Example 7.1.1. Consider the Set functor $\Sigma X = X \times X + 1$, modelling a binary operation and a constant, which we call $+$ and 0 respectively. The (underlying functor of the) free monad Σ^* for Σ sends a set X to the terms over X built from $+$ and 0. The equations $x + 0 = x$, $x + y = y + x$ and $(x+y)+z = x+(y+z)$ can be modelled as follows. The functor E is defined as $EX = X + (X \times X) + (X \times X \times X)$. The natural transformations $l, r \colon E \Rightarrow \Sigma^*$ are given by $l_X(x) = x + 0$ and $r_X(x) = x$ for all $x \in X$; $l_X(x, y) = x + y$ and $r_X(x, y) = y + x$ for all $(x, y) \in X \times X$; $l_X(x, y, z) = x + (y + z)$ and $r_X(x, y, z) = (x + y) + z$ for all $(x, y, z) \in X \times X \times X$. This defines l_X and r_X uniformly for any set X, which makes naturality of l and r easy to prove.

A $\mathcal{T}$-algebra (X, α) is said to *satisfy* $\mathcal{E}$ if $\alpha \circ l_X = \alpha \circ r_X$:

$$\begin{tikzcd} EX \arrow[r, shift left, "l_X"] \arrow[r, shift right, "r_X"'] & TX \arrow[r, "\alpha"] & X . \end{tikzcd}$$

We denote the full subcategory of $\mathcal{T}$-algebras that satisfy $\mathcal{E}$ by $(\mathcal{T}, \mathcal{E})$-Alg.

Throughout this chapter we need assumptions on $\mathcal{C}$, $\mathcal{T}$, and $\mathcal{E}$. This involves regular epis: an epi is regular if it is the coequalizer of a pair of morphisms.

Assumption 7.1.2. We assume that $\mathcal{T} = (T, \eta, \mu)$ is a monad on $\mathcal{C}$, and $E \colon \mathcal{C} \to \mathcal{C}$ is a functor such that:

1. $\mathcal{T}$-Alg has coequalizers.
2. U maps regular epis in $\mathcal{T}$-Alg to epis in $\mathcal{C}$.
3. EU and TU map regular epis in $\mathcal{T}$-Alg to epis in $\mathcal{C}$.

The first condition is needed to construct quotients of free algebras modulo equations. The second condition relates quotients of algebras (regular epis) with quotients in the base category (epis). The last condition is satisfied if condition (2) holds and E and T preserve epimorphisms in $\mathcal{C}$. If $\mathcal{C} = \mathsf{Set}$ the conditions are satisfied for any monad $\mathcal{T}$ and endofunctor E. In that case, the first condition holds since $\mathcal{T}$-Alg is cocomplete if $\mathcal{C} = \mathsf{Set}$ (see, e.g., [BW05, Proposition 3.4]), the second condition holds since U preserves regular epis if $\mathcal{C} = \mathsf{Set}$ (see the proof of [BW05, Proposition 4.6]), and the third follows from the second, since any Set functor preserves epis.

Any $\mathcal{T}$-algebra (X, α) can be turned into an algebra that satisfies the equations, by taking the coequalizer s_α of $\alpha \circ l_X^\sharp$ and $\alpha \circ r_X^\sharp$ in $\mathcal{T}$-Alg, as depicted in the following diagram:

$$\begin{tikzcd} (TEX, \mu_{EX}) \arrow[r, shift left, "l_X^\sharp"] \arrow[r, shift right, "r_X^\sharp"'] & (TX, \mu_X) \arrow[r, "\alpha"] & (X, \alpha) \arrow[r, "s_\alpha"] & (X/\mathcal{E}, \alpha_{\mathcal{E}})\,. \end{tikzcd} \tag{7.1}$$

Since coequalizers are unique only up to isomorphism, we choose $s_\alpha = \mathsf{id}$ for every algebra in $(\mathcal{T}, \mathcal{E})$-Alg.

In the case $\mathcal{C} = \mathsf{Set}$, the definition of s_α (7.1) implies that $\mathsf{ker}(s_\alpha)$ is the *congruence* generated by the set $E_\alpha = \{(\alpha(l_X(e)), \alpha(r_X(e)) \mid e \in EX\}$, i.e., it is the least equivalence relation on X that includes E_α and is a subalgebra of $(X, \alpha) \times (X, \alpha)$. In this sense, the kernel pair of a morphism always yields a congruence, and conversely, every congruence relation on an algebra (X, α) is the kernel of the corresponding quotient homomorphism.

In general, the coequalizer (7.1) in $\mathcal{T}$-Alg differs from the one obtained by applying the forgetful functor U and then computing the coequalizer of $\alpha \circ l_X^\sharp$ and $\alpha \circ r_X^\sharp$ in Set. The coequalizers in $\mathcal{T}$-Alg and Set coincide if the equations are reflexive in the sense that the two parallel maps $\alpha \circ l_X$ and $\alpha \circ r_X$ from EX to X have a common section, and the forgetful functor U preserves reflexive coequalizers (sections and reflexive coequalizers are recalled in Section 4.5, above Theorem 4.5.4). If T is finitary, then U preserves reflexive coequalizers. Moreover, if U preserves reflexive coequalizers then T preserves them too, but not every Set-functor preserves reflexive coequalizers [AKV00, Example 4.3].

The main step to obtain the quotient monad is to show that $(\mathcal{T}, \mathcal{E})$-Alg is a reflective subcategory of $\mathcal{T}$-Alg, meaning that the inclusion functor has a left adjoint. This left adjoint uses the coequalizer in (7.1) to map an algebra to its quotient.

Lemma 7.1.3. *The inclusion* $V\colon (\mathcal{T},\mathcal{E})\text{-}\mathsf{Alg} \to \mathcal{T}\text{-}\mathsf{Alg}$ *has a left adjoint* $H\colon \mathcal{T}\text{-}\mathsf{Alg} \to (\mathcal{T},\mathcal{E})\text{-}\mathsf{Alg}$ *with unit* $\overline{\eta}_\alpha = s_\alpha\colon (X,\alpha) \to (X/\mathcal{E},\alpha_\mathcal{E})$ *for all* $\alpha\colon X \to TX$ *in* $\mathcal{T}\text{-}\mathsf{Alg}$*, and counit* $\overline{\epsilon}_\alpha = \mathsf{id}$ *the identity for all* $\alpha \in (\mathcal{T},\mathcal{E})\text{-}\mathsf{Alg}$*.*

Proof. We first show that for any (X,α) in $\mathcal{T}$-Alg, $(X/\mathcal{E},\alpha_\mathcal{E})$ is indeed an object in $(\mathcal{T},\mathcal{E})$-$\mathsf{Alg}$, i.e., it satisfies the equations. Consider the following diagram:

$$\begin{tikzcd}
TEX \arrow[dr, bend left=40, "l^\sharp_X"] \arrow[dr, bend left=20, "r^\sharp_X"'] & & \\
EX \arrow[u, "\eta_{EX}"] \arrow[r, shift left, "l_X"] \arrow[r, shift right, "r_X"'] \arrow[d, "Es_\alpha"'] & TX \arrow[r, "\alpha"] \arrow[d, "Ts_\alpha"] & X \arrow[d, "s_\alpha"] \\
E(X/\mathcal{E}) \arrow[r, shift left, "l_{X/\mathcal{E}}"] \arrow[r, shift right, "r_{X/\mathcal{E}}"'] & T(X/\mathcal{E}) \arrow[r, "\alpha_\mathcal{E}"'] & X/\mathcal{E}
\end{tikzcd}$$

The right-hand square commutes by the definition of s_α as a coequalizer in $\mathcal{T}$-Alg, see (7.1). The left-hand squares (for l and r respectively) commute by naturality of l and r. The upper two paths from TEX to $X/\mathcal{E}$ commute by definition of s_α. From the above diagram we obtain $\alpha_\mathcal{E} \circ l_{X/\mathcal{E}} \circ E(s_\alpha) = \alpha_\mathcal{E} \circ r_{X/\mathcal{E}} \circ E(s_\alpha)$. Since s_α is a regular epi, by Assumption 7.1.2 it follows that $E(s_\alpha)$ is an epi, and thus $\alpha_\mathcal{E} \circ l_{X/\mathcal{E}} = \alpha_\mathcal{E} \circ r_{X/\mathcal{E}}$.

It remains to show that if $f\colon X \to Y$ is an algebra homomorphism from (X,α) to an algebra (Y,β) in $(\mathcal{T},\mathcal{E})$-$\mathsf{Alg}$, then there is a unique algebra homomorphism $g\colon X/\mathcal{E} \to Y$ such that $g \circ s_\alpha = f$. Since (Y,β) satisfies the equations we know $\beta \circ l_Y = \beta \circ r_Y$, and thus the following diagram commutes:

$$\begin{tikzcd}
TEX \arrow[dr, bend left=40, "l^\sharp_X"] \arrow[dr, bend left=20, "r^\sharp_X"'] & & & \\
EX \arrow[u, "\eta_{EX}"] \arrow[r, shift left, "l_X"] \arrow[r, shift right, "r_X"'] \arrow[d, "Ef"'] & TX \arrow[r, "\alpha"] \arrow[d, "Tf"] & X \arrow[r, "s_\alpha"] \arrow[d, "f"] & X/\mathcal{E} \arrow[dl, dashed, "g"] \\
EY \arrow[r, shift left, "l_Y"] \arrow[r, shift right, "r_Y"'] & TY \arrow[r, "\beta"'] & Y &
\end{tikzcd}$$

In particular, we have $f \circ \alpha \circ l_X = f \circ \alpha \circ r_X$. Thus $f \circ \alpha \circ l^\sharp_X = f \circ \alpha \circ r^\sharp_X$, hence the desired homomorphism g arises from the universal property of the coequalizer $s_\alpha\colon (X,\alpha) \to (X/\mathcal{E},\alpha_\mathcal{E})$.

By defining $H\colon \mathcal{T}\text{-}\mathsf{Alg} \to (\mathcal{T},\mathcal{E})\text{-}\mathsf{Alg}$ as $H(X,\alpha) = (X/\mathcal{E},\alpha_\mathcal{E})$, H is left adjoint to V, and the unit of the adjunction is $\overline{\eta} = s$. For the counit, we have $V(\epsilon_\alpha) \circ s_{V\alpha} = \mathsf{id}_{V\alpha}$, and since $s_{V\alpha} = \mathsf{id}_{V\alpha}$ then $V(\epsilon_\alpha) = \mathsf{id}_{V\alpha} = V(\mathsf{id}_\alpha)$, which implies that $\epsilon_\alpha = \mathsf{id}_\alpha$ (V is an inclusion). □

By composition of adjoints, the functor $UV\colon (\mathcal{T},\mathcal{E})\text{-}\mathsf{Alg} \to \mathcal{T}\text{-}\mathsf{Alg} \to \mathcal{C}$ has a left adjoint given by $X \mapsto (TX/\mathcal{E},(\mu_X)_\mathcal{E})$. In what follows, we will write $T^\mathcal{E}X$ for $TX/\mathcal{E}$.

Definition 7.1.4 (Quotient monad)**.** Given a monad $\mathcal{T} = (T, \eta, \mu)$ on $\mathcal{C}$ and $\mathcal{T}$-equations $\mathcal{E}$, we define the *quotient monad* $\mathcal{T}^{\mathcal{E}} = (T^{\mathcal{E}}, \eta^{\mathcal{E}}, \mu^{\mathcal{E}})$ as the monad on $\mathcal{C}$ arising from the composition of the adjunction $(H, V, \overline{\eta} = s, \overline{\epsilon} = \mathsf{id})$ of Lemma 7.1.3 and the Eilenberg-Moore adjunction (G, U, η, ϵ) of $\mathcal{T}$:

$$\begin{tikzcd}
(\mathcal{T},\mathcal{E})\text{-}\mathsf{Alg} \arrow[r, bend left, "V"] \arrow[r, phantom, "\top"] & \mathcal{T}\text{-}\mathsf{Alg} \arrow[l, bend left, "H"] \arrow[r, bend left, "U"] \arrow[r, phantom, "\top"] & \mathcal{C} \arrow[l, bend left, "G"] \arrow[loop right, "T^{\mathcal{E}}"]
\end{tikzcd}$$

We define the natural transformation $q\colon T \Rightarrow T^{\mathcal{E}}$ as the family of underlying $\mathcal{C}$-arrows of s for free algebras:

$$q_X = Us_{GX} = Us_{(TX,\mu_X)}\colon TX \to T^{\mathcal{E}}X \tag{7.2}$$

The next result summarizes what we need to know about q and the quotient monad.

Theorem 7.1.5. *Let $\mathcal{T}^{\mathcal{E}} = (T^{\mathcal{E}}, \eta^{\mathcal{E}}, \mu^{\mathcal{E}})$ be the quotient monad associated to a monad $\mathcal{T} = (T, \eta, \mu)$ on $\mathcal{C}$ with $\mathcal{T}$-equations $\mathcal{E}$. Define the natural transformation q as in* (7.2)*, so $q\colon T \Rightarrow \mathcal{T}^{\mathcal{E}}$ is defined on an object X as the coequalizer of $\mu_X \circ l^\sharp_{TX}$ and $\mu_X \circ r^\sharp_{TX}$:*

$$\begin{tikzcd}
(TETX, \mu_{ETX}) \arrow[r, shift left, "l^\sharp_{TX}"] \arrow[r, shift right, "r^\sharp_{TX}"'] & (TTX, \mu_{TX}) \arrow[r, "\mu_X"] & (TX, \mu_X) \arrow[r, "q_X"] & (T^{\mathcal{E}}X, (\mu_X)_{\mathcal{E}})\,.
\end{tikzcd}$$

Then

1. *the components of q (as well as Tq and Eq) are epimorphisms in the underlying category $\mathcal{C}$,*

2. *the unit of the quotient monad is given by $\eta^{\mathcal{E}} = q \circ \eta$ and*

3. *q is a monad morphism from $\mathcal{T}$ to $\mathcal{T}^{\mathcal{E}}$.*

Proof. The first item follows from Assumption 7.1.2. For the second item, we have $\eta^{\mathcal{E}} = UsG \circ \eta = q \circ \eta$. The third item is proved below in Corollary 7.1.7. □

Next, we show that q is indeed a monad morphism from $\mathcal{T}$ to $\mathcal{T}^{\mathcal{E}}$. One way of doing so is to show that q is a coequalizer in the category of monads and monad morphisms. Kelly studied colimits in categories of monads, and proved their existence in the context of a certain adjunction [Kel80, Proposition 26.4]; with a bit of effort one can instantiate this to the adjunction constructed above. For a self-contained presentation in this section, we do not invoke Kelly's results but instead prove directly the part that shows the existence of a monad morphism. This is instantiated below to the adjunction of the quotient monad.

Lemma 7.1.6. *Let $\mathcal{A}$ be any subcategory of $\mathcal{T}$-Alg, and suppose the forgetful functor $U\colon \mathcal{A} \to \mathcal{C}$ has a left adjoint F, with unit and counit denoted by η' and ϵ' respectively. Then*

1. *F induces a natural transformation $\kappa\colon TUF \Rightarrow UF$ so that $\kappa \circ T\eta'\colon T \Rightarrow UF$ is a monad morphism.*

2. *Precomposing the functor UF-$\mathsf{Alg} \to \mathcal{T}$-$\mathsf{Alg}$ induced by this monad morphism with the comparison functor $\mathcal{A} \to UF$-Alg yields the inclusion $\mathcal{A} \to \mathcal{T}$-$\mathsf{Alg}$.*

The relevant categories and functors are summarized in the diagram below, where the functors in the right-hand triangle are given by 2 above (hence, this triangle commutes).

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{A} \arrow[l, bend left, "U"] \arrow[r] \arrow[d, hook] & UF\text{-}\mathsf{Alg} \arrow[dl] \\
& \mathcal{T}\text{-}\mathsf{Alg} \arrow[ul] &
\end{tikzcd}$$

Proof. The functor F sends any $\mathcal{C}$-object X to a T-algebra structure on UFX; we define κ_X to be that algebra structure. Naturality of κ is immediate since Ff is a $\mathcal{T}$-algebra homomorphism for any $\mathcal{C}$-arrow f. To see that $\kappa \circ T\eta'$ is a monad morphism, we have to prove that the outside of the following diagram commutes:

$$\begin{tikzcd}
TT \arrow[r, Rightarrow, "TT\eta'"] \arrow[d, Rightarrow, "\mu"'] & TTUF \arrow[r, Rightarrow, "T\kappa"] \arrow[d, Rightarrow, "\mu UF"] & TUF \arrow[r, Rightarrow, "T\eta' UF"] \arrow[d, Rightarrow, "\kappa"] & TUFUF \arrow[d, Rightarrow, "\kappa UF"] \\
T \arrow[r, Rightarrow, "T\eta'"] & TUF \arrow[r, Rightarrow, "\kappa"] & UF & UFUF \arrow[l, Rightarrow, "\mu'"'] \\
\mathsf{Id} \arrow[u, Rightarrow, "\eta"] \arrow[r, Rightarrow, "\eta'"'] & UF \arrow[u, Rightarrow, "\eta UF"'] \arrow[ur, equal] & &
\end{tikzcd}$$

Here $\mu' = U\epsilon' F$ is the multiplication of the monad that arises from the adjunction $F \dashv U$. The top left square commutes by naturality and the middle square since any component of κ is a $\mathcal{T}$-algebra. For the right-hand square we have

$$\kappa = \kappa \circ TU\epsilon' F \circ T\eta' UF = U\epsilon' F \circ \kappa UF \circ T\eta' UF = \mu' \circ \kappa UF \circ T\eta' UF$$

where the first equality follows from the triangle identity $\mathsf{id}UF = U\epsilon' F \circ \eta' UF$ (and functoriality), and the second from the fact that, for any X, ϵ'_{FX} is a $\mathcal{T}$-algebra homomorphism from κ_{UFX} to κ_X. The bottom left square commutes by naturality, and the triangle since κ is an $\mathcal{T}$-algebra.

For item 2 of the statement of the theorem, we first note that the composite functor under consideration maps any T-algebra (A, α) in $\mathcal{A}$ to $U\epsilon'_{(A,\alpha)} \circ \kappa_A \circ T\eta'_A$.

But the following diagram commutes:

$$\begin{tikzcd}
TA \arrow[r, "T\eta'_A"] \arrow[dr, equal] & TUFA \arrow[r, "\kappa_A"] \arrow[d, "TU\epsilon'_{(A,\alpha)}"] & UFA \arrow[d, "U\epsilon'_{(A,\alpha)}"] \\
& TA \arrow[r, "\alpha"'] & A
\end{tikzcd}$$

by a triangle identity and the fact that $\epsilon'_{(A,\alpha)}$ is an algebra morphism. Hence $U\epsilon'_{(A,\alpha)} \circ \kappa_A \circ T\eta'_A = \alpha$, which means that the composite functor under consideration indeed coincides with the inclusion. □

Corollary 7.1.7. *Item 3 of Theorem 7.1.5 holds: $q\colon T \Rightarrow T^{\mathcal{E}}$ is a monad morphism.*

Proof. By Lemma 7.1.6, we only need to show that q coincides with $\kappa \circ T\eta^{\mathcal{E}}$, where $\eta^{\mathcal{E}}$ is the unit of the quotient monad. To this end, consider the following diagram:

$$\begin{tikzcd}
T \arrow[r, Rightarrow, "T\eta^{\mathcal{E}}"] \arrow[dr, Rightarrow, "T\eta"'] & TT^{\mathcal{E}} \arrow[r, Rightarrow, "\kappa"] & T^{\mathcal{E}} \\
& TT \arrow[u, Rightarrow, "Tq"'] \arrow[r, Rightarrow, "\mu"'] & T \arrow[u, Rightarrow, "q"']
\end{tikzcd}$$

Commutativity of the triangle holds by Theorem 7.1.5. For the square, notice that the components of κ are simply the quotient algebras as constructed in the proof of Lemma 7.1.3, and q is an algebra morphism by construction. □

Remark 7.1.8. The monad morphism $q\colon T \Rightarrow T^{\mathcal{E}}$ induces a functor

$$\mathcal{T}^{\mathcal{E}}\text{-Alg} \to \mathcal{T}\text{-Alg}.$$

By Lemma 7.1.6 (2), the comparison functor $(\mathcal{T}, \mathcal{E})$-Alg $\to \mathcal{T}^{\mathcal{E}}$-Alg followed by the functor $\mathcal{T}^{\mathcal{E}}$-Alg $\to \mathcal{T}$-Alg coincides with the inclusion $(\mathcal{T}, \mathcal{E})$-Alg $\to \mathcal{T}$-Alg.

The above construction yields a monad $\mathcal{T}^{\mathcal{E}}$ given a set of operations and equations. Intuitively, any monad which is isomorphic to $\mathcal{T}^{\mathcal{E}}$ is presented by these same operations and equations; this is captured by the following definition.

Definition 7.1.9. Let Σ be an endofunctor on $\mathcal{C}$, Σ^* the free monad for Σ, and $\mathcal{T}^{\mathcal{E}}$ the quotient monad of Σ^* with respect to some Σ^*-equations $\mathcal{E}$. A monad $\mathcal{K} = (K, \theta, \nu)$ is *presented by* Σ *and* $\mathcal{E}$ if there is a monad isomorphism $i\colon (\Sigma^*)^{\mathcal{E}} \Rightarrow K$.

Example 7.1.10. The *idempotent semiring monad* is defined by the Set endofunctor that maps a set X to the set $\mathcal{P}_\omega(X^*)$ of finite languages over X and a function $f\colon X \to Y$ to $\mathcal{P}_\omega(f^*)(L) = \bigcup\{f(x_1)\cdots f(x_n) \mid x_1 \cdots x_n \in L\}$. The unit $\eta_X : X \to \mathcal{P}_\omega(X^*)$ and the multiplication $\mu_X : \mathcal{P}_\omega(\mathcal{P}_\omega(X^*)^*) \to \mathcal{P}_\omega(X^*)$ are given by

$$\begin{aligned}
\eta_X(x) &= \{x\}, \\
\mu_X(\mathcal{L}) &= \textstyle\bigcup_{L_1 \cdots L_n \in \mathcal{L}} \{w_1 \cdots w_n \mid w_i \in L_i\}.
\end{aligned}$$

Consider the functor Σ and equations $\mathcal{E}$ for the free monad Σ^*, where

$$\Sigma X = X \times X + X \times X + 1 + 1$$

models two binary operators (to represent addition $+$ and multiplication $\cdot$) and two constants (to represent 0 and 1). The equations $\mathcal{E}$ for Σ^* are given by the idempotent semiring axioms. We obtain a quotient monad $(\Sigma^*)^{\mathcal{E}}$, and by Theorem 7.1.5 a monad morphism:

$$q \colon \Sigma^* \Rightarrow (\Sigma^*)^{\mathcal{E}} .$$

Since we have chosen $\mathcal{E}$ to be the idempotent semiring axioms, we have a monad isomorphism $(\Sigma^*)^{\mathcal{E}} \cong \mathcal{P}_\omega(\mathsf{Id}^*)$ (using these equations, every term is equivalent to a sum of products of variables). Thus, the monad $\mathcal{P}_\omega(\mathsf{Id}^*)$ is presented by the (semiring) signature Σ and the axioms for idempotent semirings.

7.2 Quotients of distributive laws

In the previous section, we saw how equations give rise to quotients of algebras, and we gave a construction of the resulting quotient monad. Next, we investigate conditions under which distributive laws and equations give rise to quotients of distributive laws.

7.2.1 Distributive laws over plain behaviour functors

Let $\lambda \colon TB \Rightarrow BT$ be a distributive law of a monad $\mathcal{T} = (T, \eta, \mu)$ over a (plain) behaviour functor B (Section 3.5). Given equations $\mathcal{E} = (E, l, r)$ for $\mathcal{T}$ we provide a condition on λ and $\mathcal{E}$ that ensures that we get a distributive law $\lambda^{\mathcal{E}} : T^{\mathcal{E}} B \Rightarrow BT^{\mathcal{E}}$ of the quotient monad over B. We use the notion of morphisms of distributive laws from [PW02, Wat02].

Definition 7.2.1. Let $\mathcal{T} = (T, \eta, \mu)$ and $\mathcal{K} = (K, \theta, \nu)$ be monads, and let $\lambda \colon TB \Rightarrow BT$ and $\kappa \colon KB \Rightarrow BK$ be distributive laws of $\mathcal{T}$ and $\mathcal{K}$ over B. A natural transformation $\tau \colon T \Rightarrow K$ is a *morphism of distributive laws* from λ to κ if τ is a monad morphism and the following square commutes:

$$\begin{tikzcd} TB \arrow[r, Rightarrow, "\tau B"] \arrow[d, Rightarrow, "\lambda"'] & KB \arrow[d, Rightarrow, "\kappa"] \\ BT \arrow[r, Rightarrow, "B\tau"'] & BK \end{tikzcd} \tag{7.3}$$

There are generalizations of the above definition that allow natural transformations between behaviour functors [Wat02]. For our purposes, we do not need to change the behaviour type.

Definition 7.2.2. We say that $\lambda\colon TB \Rightarrow BT$ *preserves (equations in)* $\mathcal{E}$ if for every object X in $\mathcal{C}$:

$$\begin{tikzcd} EBX \arrow[r, shift left, "l_{BX}"] \arrow[r, shift right, "r_{BX}"'] & TBX \arrow[r, "\lambda_X"] & BTX \arrow[r, "Bq_X"] & BT^{\mathcal{E}}X \end{tikzcd} \tag{7.4}$$

commutes.

In Set, preservation of equations can be conveniently formulated in terms of relation lifting (Section 3.2.1).

Lemma 7.2.3. *Suppose* $B\colon \mathsf{Set} \to \mathsf{Set}$ *preserves weak pullbacks. Denote by* $\equiv_X$ *the congruence* $\mathsf{ker}(q_X)$ *on* TX *generated by the equations. Then* λ *preserves* $\mathcal{E}$ *if and only if for every set* X *and every* $b \in EBX$*:*

$$(\lambda_X(l_{BX}(b)), \lambda_X(r_{BX}(b))) \in \mathsf{Rel}(B)(\equiv_X)\,. \tag{7.5}$$

Proof. By Lemma 3.2.4:

- $\mathsf{Rel}(B)$ preserves diagonal relations, i.e., $\mathsf{Rel}(B)(\Delta_X) = \Delta_{BX}$, and
- $\mathsf{Rel}(B)$ preserves inverse images, since B preserves weak pullbacks.

Hence $\mathsf{Rel}(B)$ preserves kernel relations (cf. [Jac12, Lemma 3.2.5(i)]):

$$\begin{array}{lll} \mathsf{Rel}(B)(\equiv_X) & = \mathsf{Rel}(B)(\mathsf{ker}(q_X)) & \text{definition } \equiv_X \\ & = \mathsf{Rel}(B)((q_X \times q_X)^{-1}(\Delta_X)) & \\ & = (Bq_X \times Bq_X)^{-1}(\mathsf{Rel}(B)(\Delta_X)) & \mathsf{Rel}(B) \text{ pres. inverse images} \\ & = (Bq_X \times Bq_X)^{-1}(\Delta_{BX}) & \mathsf{Rel}(B) \text{ pres. diagonals} \\ & = \mathsf{ker}(Bq_X) & \end{array}$$

Thus, the condition from the statement of the lemma is satisfied if and only if for every X and every $b \in EBX$ we have

$$(\lambda_X(l_{BX}(b)), \lambda_X(r_{BX}(b))) \in \mathsf{ker}(Bq_X)$$

which coincides with preservation of equations. □

We now come to the main result of this chapter. It shows how to obtain a distributive law for the quotient monad under the assumption of preservation of equations. Preservation of equations can be proved by explicit calculations, as shown in several examples below.

Theorem 7.2.4. *If* $\lambda\colon TB \Rightarrow BT$ *preserves equations* $\mathcal{E}$ *then there is a (unique) distributive law* $\lambda^{\mathcal{E}}\colon T^{\mathcal{E}}B \Rightarrow BT^{\mathcal{E}}$ *such that* $q\colon T \Rightarrow T^{\mathcal{E}}$ *is a morphism of distributive laws from* λ *to* $\lambda^{\mathcal{E}}$*.*

Proof. Suppose λ preserves equations $\mathcal{E}$. We first prove that the top rows of the following diagram commute:

$$\begin{tikzcd}
TETBX \arrow[r, shift left, "l^\sharp_{TBX}"] \arrow[r, shift right, "r^\sharp_{TBX}"'] & TTBX \arrow[r, "\mu_{BX}"] & TBX \arrow[r, "\lambda_X"] & BTX \arrow[r, "Bq_X"] & BT^{\mathcal{E}}X \\
ETBX \arrow[u, "\eta_{ETBX}"] \arrow[ru, bend right, shift left, "l_{TBX}"] \arrow[ru, bend right, shift right, "r_{TBX}"']
\end{tikzcd} \tag{7.6}$$

In order to do so, we prove that

1. $Bq_X \circ \lambda_X$ is an algebra morphism from (TBX, μ_{BX}), and
2. the bottom two paths, i.e., from $ETBX$ to $BT^{\mathcal{E}}$, commute.

Commutativity of the top rows then follows from the fact that homomorphic extensions are unique.

For the first item, consider the following diagram:

$$\begin{tikzcd}
TTBX \arrow[r, "T\lambda_X"] \arrow[dd, "\mu_{BX}"'] & TBTX \arrow[r, "TBq_X"] \arrow[d, "\lambda_{TX}"] & TBT^{\mathcal{E}}X \arrow[d, "\lambda_{T^{\mathcal{E}}X}"] \\
 & BTTX \arrow[r, "BTq_X"] \arrow[d, "B\mu_X"] & BTT^{\mathcal{E}}X \arrow[d, "B(\mu_X)_{\mathcal{E}}"] \\
TBX \arrow[r, "\lambda_X"'] & BTX \arrow[r, "Bq_X"'] & BT^{\mathcal{E}}X
\end{tikzcd}$$

The rectangle (on the left) is the multiplication law for λ, which holds since λ is a distributive law of $\mathcal{T}$ over B (Section 3.5.1). The upper right square commutes by naturality, the lower by the fact that q_X is an algebra morphism.

For the second item, we need to prove that the top two rows in the following diagram commute:

$$\begin{tikzcd}
ETBX \arrow[r, shift left, "l_{TBX}"] \arrow[r, shift right, "r_{TBX}"'] \arrow[d, "E\lambda_X"'] \arrow[dr, phantom, "(\text{nat. } l, r)"] & TTBX \arrow[r, "\mu_{BX}"] \arrow[d, "T\lambda_X"] \arrow[dr, phantom, "(\text{mult. } \lambda)"] & TBX \arrow[r, "\lambda_X"] & BTX \arrow[r, "Bq_X"] \arrow[dr, phantom, "(q \text{ monad morphism})"] & BT^{\mathcal{E}}X \\
EBTX \arrow[r, shift left, "l_{BTX}"] \arrow[r, shift right, "r_{BTX}"'] & TBTX \arrow[r, "\lambda_{TX}"'] & BTTX \arrow[r, "Bq_{TX}"'] \arrow[ur, "B\mu_X"] & BT^{\mathcal{E}}TX \arrow[r, "BT^{\mathcal{E}}q_X"'] & BT^{\mathcal{E}}T^{\mathcal{E}}X \arrow[u, "B\mu^{\mathcal{E}}_X"']
\end{tikzcd}$$

The two squares on the left (for l, r respectively) commute by naturality of l and r. The two other shapes commute by the multiplication law of λ and the fact that q is a monad morphism (Corollary 7.1.7). The crucial point is that the two paths from $EBTX$ to $BT^{\mathcal{E}}TX$ commute by the assumption that λ preserves $\mathcal{E}$ (instantiated to the object TX). It follows that the top rows commute, as desired.

Thus, we have shown that (7.6) commutes. By the universal property of the coequalizer q_{BX} we obtain $\lambda^{\mathcal{E}}_X$:

$$\begin{tikzcd}
TETBX \arrow[r, shift left, "l^{\sharp}_{TBX}"] \arrow[r, shift right, "r^{\sharp}_{TBX}"'] & TTBX \arrow[r, "\mu_{BX}"] & TBX \arrow[r, "q_{BX}"] \arrow[d, "\lambda_X"] & T^{\mathcal{E}}BX \arrow[d, dashed, "\lambda^{\mathcal{E}}_X"] \\
 & & BTX \arrow[r, "Bq_X"] & BT^{\mathcal{E}}X
\end{tikzcd} \tag{7.7}$$

Naturality of $\lambda^{\mathcal{E}}$ follows from (7.7), naturality of λ and q, and the fact that q is componentwise epic in the underlying category $\mathcal{C}$ (Theorem 7.1.5). Due to the commutativity of the square in (7.7), q is a morphism of distributive laws from λ to $\lambda^{\mathcal{E}}$ once we show that $\lambda^{\mathcal{E}}$ is, in fact, a distributive law of monad over functor (Section 3.5.1).

The unit law for $\lambda^{\mathcal{E}}$ holds due to the unit law for λ, (7.7) and the fact that $\eta^{\mathcal{E}} = q \circ \eta$ (Theorem 7.1.5):

$$\begin{tikzcd}
BX \arrow[r, "\eta_{BX}"] \arrow[rr, bend left=50, "\eta^{\mathcal{E}}_{BX}"] \arrow[d, equal] & TBX \arrow[r, "q_{BX}"] \arrow[d, "\lambda_X"] \arrow[r, phantom, shift right=6ex, "(7.7)"] & T^{\mathcal{E}}BX \arrow[d, "\lambda^{\mathcal{E}}_X"] \\
BX \arrow[r, "B\eta_X"'] \arrow[rr, bend right=50, "B\eta^{\mathcal{E}}_X"'] & BTX \arrow[r, "Bq_X"'] & BT^{\mathcal{E}}X
\end{tikzcd} \tag{7.8}$$

Multiplication law for $\lambda^{\mathcal{E}}$:

$$\begin{tikzcd}
TBX \arrow[rr, "\lambda_X"] \arrow[dddd, bend right=40, "q_{BX}"'] & & BTX \arrow[dddd, bend left=40, "Bq_X"] \\
TTBX \arrow[u, "\mu_{BX}"] \arrow[r, "T\lambda_X"] \arrow[d, "q_{TBX}"] \arrow[urr, phantom, "(\text{mult. } \lambda)"] & TBTX \arrow[r, "\lambda_{TX}"] \arrow[d, "q_{BTX}"] & BTTX \arrow[u, "B\mu_X"'] \arrow[d, "Bq_{TX}"] \\
T^{\mathcal{E}}TBX \arrow[r, "T^{\mathcal{E}}\lambda_X"] \arrow[d, "T^{\mathcal{E}}q_{BX}"] \arrow[ur, phantom, "(\text{nat. } q)"] & T^{\mathcal{E}}BTX \arrow[r, "\lambda^{\mathcal{E}}_{TX}"] \arrow[d, "T^{\mathcal{E}}Bq_X"] \arrow[ur, phantom, "(7.7)"] & BT^{\mathcal{E}}TX \arrow[d, "BT^{\mathcal{E}}q_X"] \\
T^{\mathcal{E}}T^{\mathcal{E}}BX \arrow[r, "T^{\mathcal{E}}\lambda^{\mathcal{E}}_X"] \arrow[d, "\mu^{\mathcal{E}}_{BX}"] \arrow[ur, phantom, "(7.7)"] & T^{\mathcal{E}}BT^{\mathcal{E}}X \arrow[r, "\lambda^{\mathcal{E}}_{T^{\mathcal{E}}X}"] \arrow[ur, phantom, "(\text{nat. } \lambda^{\mathcal{E}})"] & BT^{\mathcal{E}}T^{\mathcal{E}}X \arrow[d, "B\mu^{\mathcal{E}}_X"'] \\
T^{\mathcal{E}}BX \arrow[rr, "\lambda^{\mathcal{E}}_X"] & & BT^{\mathcal{E}}X
\end{tikzcd} \tag{7.9}$$

The small upper-left square commutes by naturality of q. The small lower-left square commutes by applying $T^{\mathcal{E}}$ to (7.7). The outer crescents commute since q is a monad morphism, and the outermost part does due to (7.7). Finally, we use that by naturality of q, $T^{\mathcal{E}}q_{BX} \circ q_{TBX} = q_{T^{\mathcal{E}}BX} \circ Tq_{BX}$, which by Theorem 7.1.5 is an epi, and hence can be right-cancelled to yield commutativity of the lower rectangle as desired. □

Remark 7.2.5. Every distributive law uniquely corresponds to a functor lifting on $\mathcal{T}$-algebras. The distributive law $\lambda^{\mathcal{E}}$ in Theorem 7.2.4 exists if and only if the lifting restricts to $\mathcal{T}^{\mathcal{E}}$-algebras. A similar statement for the case when B is a monad is made in [MM07, Corollary 3.4.2].

As a corollary we obtain the analogue of Theorem 7.2.4 for monads presented by operations and equations.

Corollary 7.2.6. *Suppose $\mathcal{K} = (K, \theta, \nu)$ is a monad that is presented by operations Σ and equations $\mathcal{E}$ with a monad isomorphism $i\colon T^{\mathcal{E}} \Rightarrow K$, and suppose we have a distributive law $\lambda\colon \Sigma^* B \Rightarrow B\Sigma^*$ of Σ^* over B that preserves $\mathcal{E}$. Then there exists a unique distributive law $\kappa\colon KB \Rightarrow BK$ of $\mathcal{K}$ over B such that $i \circ q\colon \lambda \Rightarrow \kappa$ is a morphism of distributive laws.*

Proof. By Theorem 7.2.4 we obtain a distributive law $\lambda^{\mathcal{E}}$ of $\mathcal{T}^{\mathcal{E}}$ over B. The distributive law $\kappa\colon KB \Rightarrow BK$ is defined as $\kappa = Bi \circ \lambda^{\mathcal{E}} \circ i^{-1}$. The proof proceeds by checking that κ indeed satisfies the defining axioms of a distributive law, which is an easy but tedious exercise. □

Theorem 7.2.4 states that if λ preserves the equations $\mathcal{E}$, then we can *present* $\lambda^{\mathcal{E}}$ *as "λ modulo equations"*. We illustrate this with an example.

Example 7.2.7 (Stream calculus)**.** Behavioural differential equations are used to define streams and stream operations (Section 3.1.1). We define the following system of behavioural differential equations:

$$\begin{array}{ll} (\sigma \times \tau)_0 = \sigma_0 \cdot \tau_0 & (\sigma \times \tau)' = (\sigma' \times [\tau_0]) + ((\sigma' \times (\mathtt{X} \times \tau')) + ([\sigma_0] \times \tau')) \\ \mathtt{X}_0 = 0 & \mathtt{X}' = [1] \end{array}$$

where the sum $+$ and the constants $[r] = (r, 0, 0, \ldots)$ for all $r \in \mathbb{R}$, are as defined in Section 3.1.1. The operation $\times$ is the convolution product, defined differently here than in Section 3.1.1; we explain this choice at the end of the example.

Since we are defining two binary operations ($+$ and $\times$), one constant stream $\mathtt{X}$ and $\mathbb{R}$ many streams $[r]$, the signature under consideration is $\Sigma X = X \times X + X \times X + 1 + \mathbb{R}$. The differential equations can be modelled as a natural transformation $\rho\colon \Sigma(\mathbb{R} \times \mathsf{Id}) \Rightarrow \mathbb{R} \times \Sigma^*$, where Σ^* is the free monad for Σ. On a component X, ρ is given by:

$$\begin{array}{rcl} \rho_X^{[r]} & = & (r, [0]) \\ \rho_X^{\mathtt{X}} & = & (0, [1]) \\ \rho_X^{+}((a, x), (b, y)) & = & (a + b, x + y) \\ \rho_X^{\times}((a, x), (b, y)) & = & (a \cdot b, (x \times [b]) + ((x \times (\mathtt{X} \times y)) + ([a] \times y))) \end{array}$$

This differs from Example 3.5.5, where we considered GSOS specifications, which are of a slightly different form, involving the copointed functor $(\mathbb{R} \times \mathsf{Id}) \times \mathsf{Id}$ on the left-hand side. Similar to what is described for GSOS specifications in Section 3.5.2, the above natural transformation ρ induces a distributive law $\rho^\dagger \colon \Sigma^*(\mathbb{R} \times \mathsf{Id}) \Rightarrow (\mathbb{R} \times \mathsf{Id})\Sigma^*$.

Let $\mathcal{E}$ be given by the following axioms where v, u, w are variables and $a, b \in \mathbb{R}$ (see Example 7.1.1 for an explanation of how this corresponds to a functor with two natural transformations):

$$\begin{array}{lll}
(v + u) + w = v + (u + w) & [0] + v = v & v + u = u + v \\
(v \times u) \times w = v \times (u \times w) & [1] \times v = v & v \times u = u \times v \\
v \times (u + w) = (v \times u) + (v \times w) & [0] \times v = [0] & \\
[a + b] = [a] + [b] & [a \cdot b] = [a] \times [b] &
\end{array}$$

$\mathcal{E}$ consists of the *commutative* semiring axioms together with axioms stating the inclusion of the underlying semiring of the reals. We would like to apply Theorem 7.2.4 to obtain a distributive law for the quotient monad arising from Σ^* and $\mathcal{E}$. To this end, we show that $\rho^\dagger$ preserves $\mathcal{E}$. Let $(a, x), (b, y), (c, z) \in \mathbb{R} \times X$ for some set X. First note that (r_1, t_1) $\mathsf{Rel}(\mathbb{R} \times \mathsf{Id})(\equiv_X)$ (r_2, t_2) iff $r_1 = r_2$ and $t_1 \equiv_X t_2$. It is straightforward to check preservation of the axioms that only concern addition, as well as of $[1] \times v = v$, $[0] \times v = [0]$ and $v \times u = u \times v$. We show that $[a \cdot b] = [a] \times [b]$ is preserved:

$$\begin{array}{lcl}
\rho^\dagger_X([a] \times [b]) & = & (a \cdot b, [0] \times [b] + [0] \times \mathtt{X} \times [0] + [a] \times [0]) \\
 & \mathsf{Rel}(\mathbb{R} \times \mathsf{Id})(\equiv_X) & (a \cdot b, [0]) \\
 & = & \rho^\dagger_X([a \cdot b])
\end{array}$$

We check that $\rho^\dagger$ preserves the distribution axiom:

$$\begin{array}{ll}
\multicolumn{2}{l}{\rho^\dagger_X((a, x) \times ((b, y) + (c, z)))} \\
= & (a \cdot (b + c), (x \times [b + c]) + (x \times X \times (y + z)) + [a] \times (y + z)) \\
\mathsf{Rel}(\mathbb{R} \times \mathsf{Id})(\equiv_X) & (a \cdot (b + c), (x \times [b + c]) + (x \times X \times y) + (x \times X \times z) + \\
 & \qquad ([a] \times y) + ([a] \times z)) \\
\mathsf{Rel}(\mathbb{R} \times \mathsf{Id})(\equiv_X) & ((a \cdot c) + (b \cdot c), (x \times [b]) + (x \times X \times y) + ([a] \times y) + \\
 & \qquad (x \times [c]) + (x \times X \times z) + ([a] \times z)) \\
= & \rho^\dagger_X(((a, x) \times (b, y)) + ((a, x) \times (c, z)))
\end{array}$$

Note that we used $[a + b] = [a] + [b]$. Similarly, preservation of $\times$-associativity can be verified, and it uses the axiom $[a \cdot b] = [a] \times [b]$. We have thus shown that $\rho^\dagger$ preserves $\mathcal{E}$, and by Theorem 7.2.4 we obtain a distributive law of the quotient monad over $\mathbb{R} \times \mathsf{Id}$.

The derivative of the convolution product is usually given differently than we defined it above. However, with the usual definition (Section 3.1.1), we did not manage to show that the commutativity of $\times$ is preserved although all other axioms remain preserved. However, the convolution product (interpreted in the final coalgebra) *is* commutative. This suggests that, even if a given set of equations holds in (the algebra induced by the distributive law on) the final coalgebra, these equations are not necessarily preserved (cf. Example 7.2.9 below).

In the above example, we did not have a specific monad in mind; we simply considered a free monad and a set of equations. In Example 7.2.11 below, we give an example for the idempotent semiring monad.

Remark 7.2.8. The concrete proof method for preservation of equations bears a close resemblance to *bisimulation up to congruence* as presented in Chapters 2,4 and 5, since one must show that for every pair in the (image of the) equations, its derivatives are related by the least *congruence* $\equiv_X$ instead of just the equivalence relation induced by the equations.

Example 7.2.9. In this example we show that it is not always possible to show that a given λ preserves a given equation that holds in the final coalgebra. Again, we consider stream systems, i.e., coalgebras for the functor $BX = \mathbb{R} \times X$. We define the constant stream of zeros by three different constants n_1, n_2 and n_3 by the following behavioural differential equations:

$$\begin{array}{rclcrcl} n_1(0) & = & 0 & \quad & n_1' & = & n_1 \\ n_2(0) & = & 0 & & n_2' & = & n_3 \\ n_3(0) & = & 0 & & n_3' & = & n_3 \end{array}$$

The corresponding signature functor is $\Sigma X = 1 + 1 + 1$, and the above specification gives rise to a distributive law $\lambda \colon \Sigma^* B \Rightarrow B\Sigma^*$. Now consider the equation $n_1 = n_2$; this clearly holds when interpreted in the final coalgebra. However, this equation is not preserved by λ. To see this, notice that $\lambda(n_1) = (0, n_1)$ and $\lambda(n_2) = (0, n_3)$, but $n_1 \not\equiv_X n_3$, so $\lambda(n_1)$ and $\lambda(n_2)$ are not related by $\mathsf{Rel}(B)(\equiv_X)$.

7.2.2 Distributive laws over copointed functors

We now show that the main result of this chapter also holds for distributive laws over copointed functors. This extends our method to deal with operations specified in the abstract GSOS format (Section 3.5.2).

Proposition 7.2.10. *Theorem 7.2.4 and Corollary 7.2.6 hold as well for any distributive law of a monad over a copointed functor.*

Proof. Let (B, ϵ) be a copointed functor and $\lambda \colon TB \Rightarrow BT$ a distributive law of $\mathcal{T}$ over (B, ϵ). Suppose λ preserves equations $\mathcal{E}$. Then by Theorem 7.2.4 there is a distributive law $\lambda^{\mathcal{E}}$ of $\mathcal{T}^{\mathcal{E}}$ over B such that $q \colon T \Rightarrow T^{\mathcal{E}}$ is a morphism of distributive laws. In order to show that $\lambda^{\mathcal{E}}$ is a distributive law of $\mathcal{T}^{\mathcal{E}}$ over (B, ϵ), we only need to prove that $\lambda^{\mathcal{E}}$ satisfies the additional axiom, i.e., that the right-hand crescent in the following diagram commutes:

$$\begin{tikzcd}
TBX \arrow[r, "q_{BX}"] \arrow[d, "\lambda_X"] \arrow[dd, bend right=60, "T\epsilon_X"'] & T^{\mathcal{E}}BX \arrow[d, "\lambda^{\mathcal{E}}_X"'] \arrow[dd, bend left=60, "T^{\mathcal{E}}\epsilon_X"] \\
BTX \arrow[r, "Bq_X"] \arrow[d, "\epsilon_{TX}"] & BT^{\mathcal{E}}X \arrow[d, "\epsilon_{T^{\mathcal{E}}X}"'] \\
TX \arrow[r, "q_X"'] & T^{\mathcal{E}}X
\end{tikzcd}$$

The outermost part commutes by naturality of q, the upper square commutes since q is a morphism of distributive laws, the lower square commutes commute by naturality of ϵ, and the left crescent commutes by the fact that λ is a distributive law of $\mathcal{T}$ over (B, ϵ). Consequently we have $\epsilon_{T^{\mathcal{E}}X} \circ \lambda^{\mathcal{E}}_X \circ q_{BX} = T^{\mathcal{E}}\epsilon_X \circ q_{BX}$, and since q_{BX} is an epi (Theorem 7.1.5) we obtain $\epsilon_{T^{\mathcal{E}}X} \circ \lambda^{\mathcal{E}}_X = T^{\mathcal{E}}\epsilon_X$ as desired. □

Example 7.2.11 (Context-free languages)**.** A context free grammar (in Greibach normal form) consists of a finite set A of terminal symbols, a (finite) set X of non-terminal symbols, and a map $\langle o, t\rangle\colon X \to 2 \times \mathcal{P}_\omega(X^*)^A$, i.e., it is a coalgebra for the behaviour functor $B = 2 \times \mathsf{Id}^A$ composed with the idempotent semiring monad $\mathcal{P}_\omega(\mathsf{Id}^*)$ from Example 7.1.10. Intuitively, $o(x) = 1$ means that the variable x can generate the empty word, whereas $w \in t(x)(a)$ if and only if x can generate aw (see [WBR13, Win14]).

It is a rather difficult task to describe concretely a distributive law of the monad $\mathcal{P}_\omega(\mathsf{Id}^*)$ over $B \times \mathsf{Id}$ defining the sum $+$ and sequential composition $\cdot$ of context-free grammars (and it is impossible to use B rather than $B \times \mathsf{Id}$, see [Win14]). Instead, we use Example 7.1.10, which presents the monad $\mathcal{P}_\omega(\mathsf{Id}^*)$ by the operations and axioms of idempotent semirings. We proceed by defining a distributive law of the free monad Σ^* generated by the signature functor $\Sigma X = 1 + 1 + (X \times X) + (X \times X)$ (to be interpreted as the constants $0, 1$ and the binary operators $+$, $\cdot$) over the copointed functor $(B \times \mathsf{Id}, \pi_2)$, and show that it preserves the semiring axioms. This distributive law arises from the abstract GSOS specification $\rho\colon \Sigma(B \times \mathsf{Id}) \Rightarrow B\Sigma^*$ whose components are given by:

$$\begin{array}{rcl}
\rho^0_X & = & (0, a \mapsto 0) \\
\rho^1_X & = & (1, a \mapsto 0) \\
\rho^+_X((x, o, f), (y, p, g)) & = & (\mathsf{max}\{o, p\}, a \mapsto f(a) + g(a)) \\
\rho^{\cdot}_X((x, o, f), (y, p, g)) & = & \left(\mathsf{min}\{o, p\}, a \mapsto \begin{cases} f(a) \cdot y & \text{if } p = 0 \\ f(a) \cdot y + g(a) & \text{if } p = 1 \end{cases}\right)
\end{array}$$

We proceed to show that the induced distributive law $\rho^\dagger$ preserves the defining equations of idempotent semirings. We only treat the case of distributivity, i.e., $u \cdot (v + w) = u \cdot v + u \cdot w$. To this end, let X be arbitrary and suppose that

$(o, d, x), (p, e, y), (q, f, z) \in BX \times X$. Notice that either $o = 0$ or $o = 1$; we treat both cases separately:

$$\begin{array}{cl}
\multicolumn{2}{l}{\rho^\dagger((0, d, x) \cdot ((p, e, y) + (q, f, z)))} \\
= & (0, a \mapsto d(a) \cdot (y + z), x \cdot (y + z)) \\
\mathsf{Rel}(B)(\equiv_X) & (0, a \mapsto d(a) \cdot y + d(a) \cdot z, x \cdot y + x \cdot z) \\
= & \rho^\dagger((0, d, x) \cdot (p, e, y) + (0, d, x) \cdot (q, f, z))
\end{array}$$

$$\begin{array}{cl}
\multicolumn{2}{l}{\rho^\dagger((1, d, x) \cdot ((p, e, y) + (q, f, z)))} \\
= & (p + q, a \mapsto d(a) \cdot (y + z) + (e(a) + f(a)), x \cdot (y + z)) \\
\mathsf{Rel}(B)(\equiv_X) & (p + q, a \mapsto (d(a) \cdot y + d(a) \cdot z) + (e(a) + f(a)), x \cdot y + x \cdot z) \\
\mathsf{Rel}(B)(\equiv_X) & (p + q, a \mapsto (d(a) \cdot y + e(a)) + (d(a) \cdot z + f(a)), x \cdot y + x \cdot z) \\
= & \rho^\dagger((1, d, x) \cdot (p, e, y) + (1, d, x) \cdot (q, f, z)) \, .
\end{array}$$

In a similar way, one can show that $\rho^\dagger$ preserves the other idempotent semiring equations. Thus, from Proposition 7.2.10 and Corollary 7.2.6 we obtain a distributive law κ of $\mathcal{P}_\omega(\mathsf{Id}^*)$ over $B \times \mathsf{Id}$ such that $i \circ q \colon \rho^\dagger \Rightarrow \kappa$ is a morphism of distributive laws, i.e., κ is presented by $\rho^\dagger$ (which is in turn determined by ρ) and the equations of idempotent semirings.

7.2.3 Distributive laws over comonads

A further type of distributive law, which generalizes all of the above, is that of a distributive law of a monad over a comonad. These arise from GSOS laws as well as from *coGSOS* laws, which allow to model operational rules which involve look-ahead in the premises. We refer to [Kli11] for technical details and an example of a coGSOS format on streams. In this subsection, we prove for future reference that when constructing the quotient distributive law as above for a distributive law over a comonad, the axioms are preserved, i.e., the quotient is again a distributive law over the comonad.

Proposition 7.2.12. *Theorem 7.2.4 and Corollary 7.2.6 hold as well for any distributive law of a monad over a comonad.*

Proof. Let (D, ϵ, δ) be a comonad and $\lambda \colon TD \Rightarrow DT$ a distributive law of the monad (T, η, μ) over the comonad (D, ϵ, δ). Suppose λ preserves equations $\mathcal{E}$. By Proposition 7.2.10 there is a distributive law $\lambda^{\mathcal{E}}$ of $\mathcal{T}^{\mathcal{E}}$ over the copointed functor (D, ϵ). To show that $\lambda^{\mathcal{E}}$ is a distributive law over the comonad (D, ϵ, δ), we need to check that the corresponding axiom holds.

$$\begin{tikzcd}
TD \arrow[rr, "\lambda"] \arrow[d, "T\delta"] \arrow[ddd, bend right=60, "q_D"'] & & DT \arrow[d, "\delta_T"] \arrow[ddd, bend left=60, "Dq"] \\
TDD \arrow[r, "\lambda_D"] \arrow[d, "q_{DD}"] & DTD \arrow[r, "D\lambda"] \arrow[d, "Dq_D"] & DDT \arrow[d, "DDq"] \\
T^{\mathcal{E}}DD \arrow[r, "\lambda^{\mathcal{E}}_D"] & DT^{\mathcal{E}}D \arrow[r, "D\lambda^{\mathcal{E}}"] & DDT^{\mathcal{E}} \\
T^{\mathcal{E}}D \arrow[rr, "\lambda^{\mathcal{E}}"] \arrow[u, "T^{\mathcal{E}}\delta"] & & DT^{\mathcal{E}} \arrow[u, "\delta_{T^{\mathcal{E}}}"']
\end{tikzcd}$$

The outermost part and the right-hand square both commute by the fact that q is a morphism of distributive laws. The outer crescents commute by naturality of q and δ. The upper rectangle commutes by the assumption that λ is a distributive law over the comonad. Checking that the lower rectangle commutes, which is what we need to prove, is now an easy diagram chase, using that q_D is epic (Theorem 7.1.5). □

7.3 Quotients of bialgebras

We show how initial and final λ-bialgebras for a distributive law relate to initial and final bialgebras for a quotiented distributive law as constructed in the previous section. We study this in the general setting of morphisms of distributive laws, and to this end we assume:

- monads $\mathcal{T} = (T, \eta, \mu)$ and $\mathcal{K} = (K, \theta, \nu)$;
- distributive laws $\lambda\colon TB \Rightarrow BT$ and $\kappa\colon KB \Rightarrow BK$ (both of monad over functor);
- a morphism of distributive laws $\tau\colon T \Rightarrow K$ from λ to κ.

Morphisms of distributive laws are defined to be monad morphisms, and hence respect the algebraic structure. The next proposition shows that, as one might expect, they also respect the coalgebraic structure, and hence morphisms of distributive laws induce morphisms between bialgebras.

Proposition 7.3.1. *Let $\widehat{T}\colon TB\text{-coalg} \to B\text{-coalg}$ and $\widehat{K}\colon KT\text{-coalg} \to K\text{-coalg}$ be liftings induced by λ and κ as in Equation* (3.14) *of Section 3.5. For all $\delta\colon X \to BTX$, τ_X is a B-coalgebra morphism from $\widehat{T}(X,\delta)$ to $\widehat{K}(X, B\tau_X \circ \delta)$.*

Proof. The following diagram commutes:

$$\begin{tikzcd}
TX \arrow[r, "\tau_X"] \arrow[d, "T\delta"'] \arrow[dr, phantom, "(\text{nat. } \tau)"] & KX \arrow[d, "K\delta"] & \\
TBTX \arrow[r, "\tau_{BTX}"] \arrow[d, "\lambda_{TX}"'] \arrow[dr, phantom, "(\text{morph. of distr. laws})"] & KBTX \arrow[r, "KB\tau_X"] \arrow[d, "\kappa_{TX}"] \arrow[dr, phantom, "(\text{nat. } \kappa)"] & KBKX \arrow[d, "\kappa_{KX}"] \\
BTTX \arrow[r, "B\tau_{TX}"] \arrow[d, "B\mu_X"'] & BKTX \arrow[r, "BK\tau_X"] \arrow[d, phantom, "(\tau \text{ monad morphism})"] & BKKX \arrow[d, "B\nu_X"] \\
BTX \arrow[rr, "B\tau_X"'] & & BKX
\end{tikzcd}$$

Commutativity of the outside is the desired result. □

If τ arises from a set of preserved equations $\mathcal{E}$ as in Section 7.2 (with $\kappa = \lambda^{\mathcal{E}}$), then Proposition 7.3.1 states that, for any coalgebra $\delta\colon X \to BTX$, the coalgebra $\widehat{K}(X, B\tau_X \circ \delta)$ is a quotient of the coalgebra $\widehat{T}(X, \delta)$, and in particular, the congruence $\equiv_X$ is included in behavioural equivalence on $\widehat{T}(X, \delta)$.

Example 7.3.2. Recall from Example 7.2.11 that the abstract GSOS specification for context-free grammars induces a morphism $i \circ q\colon \Sigma^* \Rightarrow \mathcal{P}_\omega(X^*)$ of distributive laws, where Σ^* is the free monad for the signature $\Sigma X = X \times X + X \times X + 1 + 1$ representing a binary choice $+$, a binary composition $\cdot$, and constants 0 and 1. These distributive laws induce liftings $\widehat{\Sigma^*}$ and $\widehat{\mathcal{P}_\omega(\mathsf{Id}^*)}$.

By Proposition 7.3.1 we have the following commutative diagram for any coalgebra of the form $\delta\colon X \to 2 \times (\Sigma^* X)^A$:

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[dr, "\delta"'] & \Sigma^* X \arrow[r, "(i\circ q)_X"] \arrow[d, "\widehat{\Sigma^*}(\delta)"] & \mathcal{P}_\omega(X^*) \arrow[r] \arrow[d, "\widehat{\mathcal{P}_\omega(\mathsf{Id}^*)}(Bi_X \circ Bq_X \circ \delta)"] & \mathcal{P}(A^*) \arrow[d, "\zeta"] \\
& 2 \times (\Sigma^* X)^A \arrow[r, "\mathsf{id} \times ((i\circ q)_X)^A"'] & 2 \times \mathcal{P}_\omega(X^*)^A \arrow[r] & 2 \times \mathcal{P}(A^*)^A
\end{tikzcd} \tag{7.10}$$

where ζ is the final coalgebra for $BX = 2 \times X^A$.

This gives the expected correspondence between two of the three different coalgebraic approaches to context-free languages introduced in [WBR13] (the third approach is about fixed-point expressions and is outside the scope of this chapter). These two approaches are:

1. A context-free grammar is defined as a coalgebra $X \to 2 \times (\mathcal{P}_\omega(X^*))^A$ and inductively extended to a coalgebra $\mathcal{P}_\omega(X^*) \to 2 \times (\mathcal{P}_\omega(X^*))^A$, and the language semantics arises by finality. This extension coincides with our lifting $\widehat{\mathcal{P}_\omega(\mathsf{Id}^*)}$.

2. A context-free grammar is defined more syntactically (viewed as a system of behavioural differential equations in [WBR13]) as a coalgebra $X \to 2 \times (\Sigma^* X)^A$, which is inductively extended to a coalgebra $\Sigma^* X \to 2 \times (\Sigma^* X)^A$ to obtain its language semantics. This extension coincides with our lifting $\widehat{\Sigma^*}$.

The situation in diagram (7.10) yields the correspondence between these two approaches.

Similarly, if B has a final coalgebra (Z, ζ), then the algebra on ζ induced by λ (Lemma 3.5.1) factors through the algebra on ζ induced by κ.

Proposition 7.3.3. *Let $\alpha \colon TZ \to Z$ and $\alpha' \colon KZ \to Z$ be the algebras induced by λ and κ respectively on the final B-coalgebra (Z, ζ). Then $\alpha = \alpha' \circ \tau_Z$.*

Proof. Consider the following diagram:

$$\begin{tikzcd}
TZ \arrow[r, "\tau_Z"] \arrow[d, "T\zeta"'] & KZ \arrow[r, "\alpha'"] \arrow[d, "K\zeta"] & Z \arrow[dd, "\zeta"] & TZ \arrow[l, "\alpha"'] \arrow[d, "T\zeta"] \\
TBZ \arrow[r, "\tau_{BZ}"] \arrow[d, "\lambda_Z"'] & KBZ \arrow[d, "\kappa_Z"] & & TBZ \arrow[d, "\lambda_Z"] \\
BTZ \arrow[r, "B\tau_Z"'] & BKZ \arrow[r, "B\kappa"'] & BZ & BTZ \arrow[l, "B\alpha"]
\end{tikzcd}$$

The upper left square commutes by naturality of τ, whereas the lower left square commutes since τ is a morphism of distributive laws. The two rectangles commute by definition of α and α'. Thus $\alpha' \circ \tau_Z$ and α are both coalgebra homomorphisms from $(TZ, \lambda_Z \circ T\zeta)$ to (Z, ζ) and consequently $\alpha' \circ \tau_Z = \alpha$ by finality. □

Example 7.3.4. Continuing Example 7.3.2, it follows from Proposition 7.3.3 that the algebra $\alpha \colon \Sigma^*(\mathcal{P}(A^*)) \to \mathcal{P}(A^*)$ induced by the distributive law for the free monad for Σ can be decomposed as $i \circ q \circ \alpha'$, where α' is the algebra on $\mathcal{P}(A^*)$ induced by the distributive law for $\mathcal{P}_\omega(\mathsf{Id}^*)$. It can be shown by induction that α is the algebra on languages given by union and concatenation product.

Now $\alpha' \colon \mathcal{P}_\omega(\mathcal{P}(A^*)^*) \to \mathcal{P}(A^*)$ can be given by selecting a representative term and applying α, and it follows that

$$\alpha'(\mathcal{L}) = \bigcup_{L_1 \cdots L_n \in \mathcal{L}} \{w_1 \cdots w_n \mid w_i \in L_i\} .$$

We thus retrieved this algebra α' induced by the distributive law for $\mathcal{P}_\omega(\mathsf{Id}^*)$ from the algebra $\alpha \colon \Sigma^*(\mathcal{P}(A^*)) \to \mathcal{P}(A^*)$ on terms.

7.4 Discussion and related work

We presented a preservation condition that is sufficient for the existence of a distributive law $\lambda^{\mathcal{E}}$ for a monad with equations, given a distributive law λ for the underlying monad. This condition consists of checking that the equations are preserved by λ. We demonstrated the method by constructing distributive laws for stream calculus over commutative semirings, and for context-free grammars which use the monad of idempotent semirings. The reader is invited to compare the complexity of checking that λ preserves the equations with describing and verifying the distributive law requirements directly.

Morphisms of distributive laws are used in [Wat02] as a general approach for studying translations between operational semantics. In the current chapter, we investigated in detail the case of quotients of distributive laws. Distributive laws for monad quotients and equations are also studied in [LPW04, MM07]. The setting and motivation of [MM07] is different as they study distributive laws of one monad over another with the aim to compose these monads. We study distributive laws of a monad over a plain functor, a copointed functor or a comonad. The approach in [LPW04] (in particular Theorem 31) differs from ours in that the desired distributive law is contingent on two given distributive laws and the existence of the coequalizer (in the category of monads) which encodes equations. We have given a more direct analysis which includes a concrete proof principle.

We have focused on adding equations which already hold in the final bialgebra, whereas in Chapter 6 we introduced an approach for adding equations to a distributive law via structural congruence. The results of these chapters can possibly be combined to give a more general account of equations and structural congruences for different monads.

In the case of GSOS on labelled transition systems, proving equations to hold at the level of a specification was considered in [ACI12], based on the notion of *rule-matching bisimulation*, a refinement of De Simone's *FH-bisimulation*. Rule-matching bisimulations are based on the syntactic notion of *ruloids*, while our technique is based on preservation of equations at the level of distributive laws. It is currently not clear what the precise relation between these two approaches is; one difference is that preserving equations naturally incorporates reasoning up to congruence. Further, we do not know how, and to what extent, the decidability result of [ACI12], which is based on identifying a finite set of ruloids, is reflected at the more abstract level of the current chapter.