5
Monads, Comonads and Distributive Laws

Monads and comonads – the duals of monads – form one of the basic notions in category theory, like adjunction. A monad is a special kind of endofunctor with some additional structure (unit and multiplication), a bit like for a monoid. Various computationally relevant functors are actually monads: lift, list, powerset, multiset, distribution. Associated with a (co)monad two categories are of interest.

- The *Kleisli* category captures the computations associated with a (co)monad. Coalgebras can be described within such Kleisli categories, namely as endomaps, and the Kleisli structure can be used for sequential composition of coalgebras. A prominent application involves final coalgebra inside such Kleisli categories. It gives a systematic description of so-called trace semantics; see Section 5.3.
- The *Eilenberg–Moore* category contains (co)algebras for a (co)monad as objects. They are the mathematical structures associated with the (co)monad. Where (co)algebras of a *functor* describe only operations, (co)algebras of a *(co)monad* additionally capture constraints, in the form of equations or other assertions. Many standard mathematical structures, such as monoids, vector spaces and complete lattices, are algebras of a monad. A systematic, uniform description of such structures is convenient, since it gives many results at once, such as the existence of limit and colimits.

Most of the material in this chapter is standard (basic) category theory. It is presented here with a special focus on the context of coalgebras as dynamical state-based systems. Section 5.1 introduces the basic definitions, together with the main examples. Kleisli categories will be discussed in Section 5.2, with an emphasis on the lifting of functors to such categories. These liftings play an important role in Section 5.3, on trace semantics. Section 5.4 describes Eilenberg–Moore categories, together with their basic properties. Section 5.5

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

concludes this chapter with bialgebras, combining both algebras (structure) and coalgebras (behaviour), for the operational semantics of programming languages.

The topic of algebras and coalgebras in combination with assertions is postponed until Chapter 6. Assertions require the notion of invariant, in the coalgebraic case. Once this concept is in place, (co)algebraic specifications can be introduced, as descriptions of (co)algebras satisfying certain logical properties.

5.1 Monads and Comonads: Definition and Examples

Monads are special endofunctors with some additional structure. Our prime example is the powerset functor $\mathcal{P}$ which comes equipped with a singleton map $\{-\}\colon X \to \mathcal{P}(X)$ and a (big) union $\bigcup\colon \mathcal{P}^2(X) \to \mathcal{P}(X)$. These operations are natural in X and satisfy some basic equations. This is axiomatised in the notion of monad; see Definition 5.1.1 below. It turns out that the 'collection' functors (list, powerset, multiset, distribution) from Figure 4.1 all have such monad structure.

As we shall see, monads have two distinct roles.

1. **Monads capturing types of computations**. In this sense monads are similar to the endofunctors F whose coalgebras $X \to F(X)$ we have been studying so far: a distinction is made between values/states in X and computations in $F(X)$, returning values in X. But the additional structure that monads have gives rise to the basic operation of sequential composition. We have already seen such composition, for instance in Exercise 1.1.2, without identifying the underlying monad structure. This composition is captured in what is called the Kleisli category associated with a monad. It will be the main focus of the next section.
2. **Monads capturing algebraic theories**. Monads also form an abstraction that describes the essentials of an algebraic theory, given by operations and equations between them. Here one associates a different category with a monad, namely its category of so-called Eilenberg–Moore algebras. These categories will be introduced in Section 5.4, but the aspect of monads as algebraic theories will be postponed to Section 6.7.

A combination of these two roles is achieved in the theory of algebraic effects [387, 388].

Historically, the algebraic view on monads preceded the computational view. This other, computational view, using Kleisli categories, has been introduced

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

by Moggi [357]. It has been widely adopted in functional programming – notably in the programming language Haskell – in order to deal with special kinds of computational effects (see e.g. [284], and for combinations with initial algebras/final coalgebras, see [370, 44]). In fact, in this second computational approach to monads an alternative formulation of the notion of monad has emerged, in terms of so-called Kleisli extensions. Here we stick to the standard formulation in the next definition and formulate the Kleisli version separately (in Proposition 5.2.3 in the next section).

From the computational perspective monads are thus used for structuring the *outputs* of computations. At the end of this section we briefly discuss comonads, which can be used to structure *inputs* of computations. As an aside, a combined structuring can be achieved via so-called arrows, generalising both monads and comonads. They will not be discussed here, but the interested reader is referred to [229, 372, 219, 255].

This section will introduce the notion of (co)monad and focus mostly on examples – of which there are plenty, contributing to the relevance of the notion. The next section will define the associated concept of a Kleisli category.

We are now ready to see define a monad. It is an endofunctor, typically written as T, with additional structure given by two natural transformations that play the role of ‘singleton’ and ‘flattening’. This will be illustrated in the subsequent examples.

Definition 5.1.1 A **monad** on an arbitrary category $\mathbb{C}$ consists of an endofunctor $T\colon \mathbb{C} \to \mathbb{C}$ together with two natural transformations: a **unit** $\eta\colon \mathrm{id}_{\mathbb{C}} \Rightarrow T$ and **multiplication** $\mu\colon T^2 \Rightarrow T$. These are required to make the following diagrams commute, for $X \in \mathbb{C}$:

$$\begin{tikzcd}
T(X) \arrow[r, "\eta_{T(X)}"] \arrow[dr, equal] & T^2(X) \arrow[d, "\mu_X"] & T(X) \arrow[l, "T(\eta_X)"'] \arrow[dl, equal] \\
& T(X) &
\end{tikzcd}
\qquad
\begin{tikzcd}
T^3(X) \arrow[r, "\mu_{T(X)}"] \arrow[d, "T(\mu_X)"'] & T^2(X) \arrow[d, "\mu_X"] \\
T^2 \arrow[r, "\mu_X"'] & T(X)
\end{tikzcd}$$

Often we simply write T for a monad (T, η, μ) since we standardly use η and μ for the unit and multiplication involved.

Before seeing examples of monads we immediately illustrate their usefulness in the theory of coalgebras: they introduce (sequential) composition of coalgebras. This composition will be presented in slightly more general form later on, via Kleisli categories.

Lemma 5.1.2 *For a monad T, coalgebras $X \to T(X)$ form a monoid – where X is a fixed object.*

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Proof For two coalgebras $c, d \colon X \to T(X)$ we define a 'c then d' coalgebra $d \circledcirc c \colon X \to T(X)$ as composite:

$$d \circledcirc c \overset{\text{def}}{=} \Big(X \xrightarrow{c} T(X) \xrightarrow{T(d)} T^2(X) \xrightarrow{\mu_X} T(X)\Big).$$

The associated neutral element 'skip' is the unit $\eta_X \colon X \to T(X)$ of the monad. The fact that $(\eta, \circledcirc)$ forms a monoid follows almost immediately from the monad equations. ❑

The notion of monad, like most of the concepts in category theory, is best understood via concrete examples. In order to see the common structure it may help to fill in some of the missing verifications of the monad requirements in the series of examples below.

Example 5.1.3 1. As mentioned in the beginning, the powerset functor $\mathcal{P}$ is a monad with unit and multiplication given by singleton and union:

$$\begin{array}{ccc} X \xrightarrow{\eta_X} \mathcal{P}(X) & \qquad & \mathcal{P}(\mathcal{P}(X)) \xrightarrow{\mu_X} \mathcal{P}(X) \\ x \longmapsto \{x\} & \qquad & A \longmapsto \bigcup A = \{x \in X \mid \exists U \in A.\, x \in U\}. \end{array}$$

The first two of the monad equations are easy: for $V \in \mathcal{P}(X)$,

$$\begin{aligned} (\mu_X \circ \eta_{\mathcal{P}(X)})(V) &= \bigcup\{V\} = V \\ (\mu_X \circ \mathcal{P}(\eta_X))(V) &= \bigcup\{\{x\} \mid x \in V\} = V. \end{aligned}$$

The μ-equation requires more care. For $\mathcal{A} \in \mathcal{P}^3(X) = \mathcal{P}(\mathcal{P}(\mathcal{P}(X)))$ one has

$$\begin{aligned} (\mu_X \circ \mathcal{P}(\mu_X))(\mathcal{A}) &= \bigcup\{\bigcup B \mid B \in \mathcal{A}\} \\ &= \{x \in X \mid \exists B \in \mathcal{A}.\, \exists U \in B.\, x \in U\} \\ &= \{x \in X \mid \exists U \in \bigcup \mathcal{A}.\, x \in U\} \\ &= \bigcup \bigcup \mathcal{A} \\ &= (\mu_X \circ \mu_{\mathcal{P}(X)})(\mathcal{A}). \end{aligned}$$

Also the non-empty powerset $\mathcal{P}^{\neq\emptyset}$ is a monad, and so are their finitary versions $\mathcal{P}_{\text{fin}}$ and $\mathcal{P}^{\neq\emptyset}_{\text{fin}}$ (taking only *finite* subsets).

We recall that coalgebras $X \to \mathcal{P}(X)$ of this powerset monad $\mathcal{P}$ – or of non-empty/finite variations thereof – are unlabelled transition systems, as special cases of non-deterministic automata; see Section 2.2.4. Composition $\circledcirc$ as in Lemma 5.1.2 corresponds to relational composition: for coalgebras $c, d \colon X \to \mathcal{P}(X)$ we have

$$(d \circledcirc c)(x) = \bigcup\{d(x') \mid x' \in c(x)\} = \{x'' \mid \exists x'.\, x \xrightarrow{c} x' \text{ and } x' \xrightarrow{d} x''\}.$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

2. The non-empty powerset takes subsets with *at least* one element. The 'lift' or 'maybe' functor $\mathcal{L}(X) = 1 + X$ adds a base point to a set X; it may be understood as a 'collection' functor that takes subsets of X with *at most* one element (also known as subsingletons). Lift $\mathcal{L} = 1 + (-)\colon \mathbf{Sets} \to \mathbf{Sets}$ is a monad with unit and multiplication:

$$X \xrightarrow{\eta = \kappa_2} 1 + X \qquad\qquad 1 + (1 + X) \xrightarrow{\mu = [\kappa_1, \mathrm{id}]} 1 + X.$$

This lift monad may be defined on categories other than **Sets**; see Exercise 5.1.1 (or [303]). Coalgebras $X \to \mathcal{L}(X)$ of the lift monad are partial (endo)functions on X; their sequential composition $\circledcirc$ from Lemma 5.1.2 is the usual composition of partial functions.

3. Recall the list functor $(-)^\star\colon \mathbf{Sets} \to \mathbf{Sets}$ that sends a set X to the set $X^\star = \{\langle x_1, \ldots, x_n\rangle \mid x_i \in X\}$ of finite sequences of elements of X. It forms a monad via:

$$\begin{array}{ccc} X \xrightarrow{\;\eta\;} X^\star & \qquad\qquad & X^{\star\star} \xrightarrow{\qquad\mu\qquad} X^\star \\ x \longmapsto \langle x\rangle & & \langle \vec{x_1}, \ldots, \vec{x_n}\rangle \longmapsto \vec{x_1} \cdots \vec{x_n}. \end{array}$$

The multiplication μ flattens a list of lists to a single list, by removing inner brackets.

4. The distribution functor $\mathcal{D}$ from Definition 4.1.5 forms a monad with singleton 'Dirac' distribution as unit, and matrix multiplication as (monad) multiplication.

$$\begin{array}{ccc} X \xrightarrow{\qquad\eta\qquad} \mathcal{D}(X) & \qquad & \mathcal{D}(\mathcal{D}(X)) \xrightarrow{\qquad\mu\qquad} \mathcal{D}(X) \\ x \longmapsto 1|x\rangle = \lambda y.\begin{cases} 1 & \text{if } y = x \\ 0 & \text{if } y \neq x \end{cases} & & \Psi \longmapsto \lambda y.\displaystyle\sum_{\varphi \in \mathcal{D}(X)} \Psi(\varphi)\cdot\varphi(y). \end{array}$$

This unit was already described as a Dirac distribution in Exercise 4.1.4. The multiplication applied to a multiset of multisets $\Psi = \sum_i r_i|\varphi_i\rangle$ yields a distribution $\mu(\Psi) \in \mathcal{D}(X)$ that assigns to $y \in X$ the probability $\mu(\Psi)(y) = \sum_i r_i \cdot \varphi_i(y)$. For instance, if we have distributions

$$\varphi = \tfrac{1}{2}|x\rangle + \tfrac{1}{2}|y\rangle \quad\text{and}\quad \psi = \tfrac{1}{3}|y\rangle + \tfrac{2}{3}|z\rangle,$$

then

$$\begin{aligned} \mu(\tfrac{3}{4}|\varphi\rangle + \tfrac{1}{4}|\psi\rangle) &= (\tfrac{3}{4}\cdot\tfrac{1}{2})|x\rangle + (\tfrac{3}{4}\cdot\tfrac{1}{2})|y\rangle + (\tfrac{1}{4}\cdot\tfrac{1}{3})|y\rangle + (\tfrac{1}{4}\cdot\tfrac{2}{3})|z\rangle \\ &= \tfrac{3}{8}|x\rangle + \tfrac{11}{24}|y\rangle + \tfrac{1}{6}|z\rangle. \end{aligned}$$

As discussed in Section 4.1, coalgebras of the distribution monad are Markov chains. Their composition, occurring already in Exercise 4.1.5,

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

corresponds to standard composition of such chains – usually given by matrix multiplication; see Exercise 5.1.3.

It is not hard to see that subdistributions – with sum of probabilities at most 1, instead of equal to 1 – also form a monad, written as $\mathcal{D}_{\leq 1}$.

5. If M is a monoid, say with multiplicative structure $(1, \cdot)$, then the functor $M \times (-) \colon \textbf{Sets} \to \textbf{Sets}$ is a monad. The multiplication map $\eta \colon X \to M \times X$ is $\eta(x) = (1, x)$ and multiplication $\mu \colon M \times (M \times X) \to M \times X$ is simply $\mu(m_1, (m_2, x)) = (m_1 \cdot m_2, x)$. The monad laws follow directly from the monoid laws.

   There are close connections between monoids and monads; see for instance the adjunction in Exercise 5.2.17. On a more abstract level one can describe monads $\mathbb{C}$ as 'monoids in the category of endofunctors' on $\mathbb{C}$; see [344, VII.3] for more information.

6. The above first few examples of monads all involve collections (subsets, lists, distributions) of some sort. The remaining examples are also relevant in computing but are of a different kind. We start with the state monad. It involves a fixed set S, elements of which are seen as states that are passed around in a computation. The state monad $\mathcal{S} \colon \textbf{Sets} \to \textbf{Sets}$ is given by

$$\mathcal{S}(X) = (S \times X)^S.$$

   It comes equipped with unit and multiplication operations:

$$\begin{tikzcd}
X \arrow[r, "\eta"] & (S \times X)^S & \big(S \times (S \times X)^S\big)^S \arrow[r, "\mu"] & (S \times X)^S \\
x \arrow[r, mapsto] & \lambda s \in S.\, \langle s, x \rangle & \langle h_1, h_2 \rangle \arrow[r, mapsto] & \lambda s \in S.\, h_2(s)(h_1(s)).
\end{tikzcd}$$

7. Our next example, the continuation monad, is also motivated by programming semantics. It starts from a fixed set C and takes the 'double dual' of a set, where C is used as dualising object (see [300] for a more general setting). This is the pure form of the monad, which we shall describe first; some variations will be discussed subsequently. To start, we define a functor $\mathcal{C} \colon \textbf{Sets} \to \textbf{Sets}$ by

$$\mathcal{C}(X) = C^{(C^X)} \qquad \text{and} \qquad \mathcal{C}(X \xrightarrow{f} Y) = \lambda h \in C^{(C^X)}.\, \lambda g \in C^Y.\, h(g \circ f).$$

   Earlier, in Exercise 2.2.7, we have seen the neighbourhood functor as special case $\mathcal{N}(X) = 2^{(2^X)}$ for $C = 2$.

   This functor $\mathcal{C}$ forms a monad via

$$\begin{tikzcd}
X \arrow[r, "\eta"] & C^{(C^X)} & C^{\left(C^{\left(C^{(C^X)}\right)}\right)} \arrow[r, "\mu"] & C^{(C^X)} \\
x \arrow[r, mapsto] & \lambda g \in C^X.\, g(x) & H \arrow[r, mapsto] & \lambda g \in C^X.\, H(\lambda k \in C^{(C^X)}.\, k(g)).
\end{tikzcd}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

It requires a bit of elementary bookkeeping to check that these η and μ are natural and satisfy the three monad equations.

Coalgebras $X \to C^{(C^X)}$ of the continuation monad capture an indirect way of computing a result in C. The computation involves an explicit argument function $X \to C$ that is usually called a continuation. This may be useful to get a better handle on intermediate values and argument evaluation; see e.g. [460] for more information.

The monad $\mathcal{C}$ describes what may be called the 'pure' version of the continuation monad. There are many variations which restrict the kind of functions involved (and form submonads). This will be illustrated via a several examples.

- For an arbitrary set X an ultrafilter on X is a subset $\mathcal{F} \subseteq \mathcal{P}(X)$ which is a filter ($\mathcal{F}$ is closed under finite intersections and is upclosed) satisfying $\emptyset \notin \mathcal{F}$ and for each $U \in \mathcal{P}(X)$ either $U \in \mathcal{F}$ or $\neg U \in \mathcal{F}$. Such an ultrafilter can be identified with a morphism of Boolean algebras $\mathcal{P}(X) \to \{0,1\}$. The set of ultrafilters on X may thus be defined as a hom-set in the category **BA** of Boolean algebras:

$$\begin{aligned} \mathcal{UF}(X) &= \mathbf{BA}(\mathcal{P}(X), \{0,1\}) \\ &= \mathbf{BA}(\{0,1\}^X, \{0,1\}) \\ &= \{f \in \{0,1\}^{(\{0,1\}^X)} \mid f \text{ is a morphism of Boolean algebras}\}. \end{aligned}$$

  This is a subset of $\mathcal{C}(X) = C^{(C^X)}$, with set $C = \{0,1\}$. In this case $\mathcal{UF}$ is still a monad (see e.g. [280] for more information).

- Next we take the unit interval $[0,1] \subseteq \mathbb{R}$ as constant C. It is a complete lattice and hence certainly a dcpo. Also, for each set X, the function space $[0,1]^X$ is a dcpo. In [292] the following monad is defined:

$$\mathcal{G}(X) = \{f \in [0,1]^{([0,1]^X)} \mid f \text{ is continuous and sublinear}\}.$$

  Sublinearity means that $f(r \cdot g) = r \cdot f(g)$ and $f(g_1 + g_2) \leq f(g_1) + f(g_2)$, if $g_1(x) + g_2(x) \leq 1$ for all $x \in X$. This monad $\mathcal{G}$ is used for a semantics and logic of a 'probabilistic-nondeterministic' programming language in [292].

- A variation on the previous point is the expectation monad given by:

$$\mathcal{E}(X) = \{f \in [0,1]^{([0,1]^X)} \mid f \text{ is a map of effect modules}\}.$$

  These (effect module map) requirements are slightly different and amount to $f(r \cdot g) = r \cdot f(g)$, $f(\lambda x.\, 1) = 1$ and $f(g_1 + g_2) = f(g_1) + f(g_2)$, if $g_1(x) + g_2(x) \leq 1$ for all $x \in X$. More information can be found in [257, 254, 259, 258].

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

8. In [264] a monad $\mathcal{J} \colon \mathbf{Sets} \to \mathbf{Sets}$ is introduced for the semantics of Java programs that combines several of the monads mentioned earlier. We recall that statements (and expressions) in Java:

   - can fail to terminate, typically because of an infinite ('for' or 'while') loop; this is modelled via the lift monad $\mathcal{L} = 1 + (-)$
   - can terminate normally, producing a successor state (for statements), and a successor state together with a result (for an expression) or
   - can terminate exceptionally, yielding an exception, say of type E, together with a successor state.

   When we put these ingredients together we obtain a monad:

   $$\mathcal{J}(X) = \big(1 + (S \times X) + (S \times E)\big)^{S}.$$

   The unit $\eta \colon X \to \mathcal{J}(X)$ is given by $\eta(x) = \lambda s \in S.\,\kappa_2\langle s, x\rangle$, as for the state monad. The second coprojection κ_2, used for the middle option, corresponds to normal/successful termination. The multiplication operation $\mu \colon \mathcal{J}^2(X) \to \mathcal{J}(X)$ is

   $$\mu(h)(s) = \begin{cases} \kappa_1 * & \text{if } h(s) = \kappa_1 * \\ k(s') & \text{if } h(s) = \kappa_2(s', k) \text{ where } k \in \mathcal{J}(X) \\ \kappa_3(s, e) & \text{if } h(s) = \kappa_3(s, e). \end{cases}$$

   Thus, in the first case (non-termination) and the third case (abrupt termination) the outcome of h is returned, and no subsequent evaluation takes place. This is what happens in Java. But exceptions in Java have more substructure, which is ignored here. A more complete formalisation occurs in [240].

   Producing new monads from existing ones, as we have done here, may be described more systematically in terms of so-called monad transformers; see [336, 65, 233].

9. Let I, O be two arbitrary but fixed sets, used as sets of 'inputs' and 'outputs'. Consider for each set X the initial algebra $F^*(X)$ of the functor

   $$Y \longmapsto X + F(Y), \qquad \text{where} \qquad F(Y) = Y^I + (O \times Y),$$

   with initial algebra map

   $$X + F^*(X)^I + (O \times F^*(X)) \xrightarrow{\quad\cong\quad} F^*(X).$$

   The mapping $X \mapsto F^*(X)$ is then a monad. Later on, with Proposition 5.1.8, we shall recognise this as an instance of a more general construction of free monads on a functor. The above special case is called the I/O monad; see [357, 65]. See also [171] for similar store/stack/tape monads.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

10. In point (5.1.3) we have seen the state monad $X \mapsto (S \times X)^S$, for a fixed set of states. There are associated predicate transformer monads on **Sets** (see [253]), which come in two flavours, namely after Dijkstra [439] and after Hoare [364, 440]. The Dijkstra monad $\mathfrak{D}$ on **Sets** is given by

$$\mathfrak{D}(X) = \mathcal{P}(S)^{\mathcal{P}(S\times X)} \qquad \text{and} \qquad \mathfrak{D}(X \xrightarrow{f} Y) = \lambda h.\, \lambda Q.\, h((\mathrm{id}_S \times f)^{-1}(Q)).$$

Such functions $\mathcal{P}(S \times X) \to \mathcal{P}(S)$ in $\mathfrak{D}(X)$ are predicate transformers that send a postcondition predicate $Q \subseteq S \times X$ on the state S and the output type X, to a precondition $P \subseteq S$. The associated unit and multiplication are

$$\begin{array}{ll} X \xrightarrow{\ \eta\ } \mathcal{P}(S)^{\mathcal{P}(S\times X)} & \qquad \mathcal{P}(S)^{\mathcal{P}(S\times \mathcal{P}(S)^{\mathcal{P}(S\times X)})} \xrightarrow{\quad \mu \quad} \mathcal{P}(S)^{\mathcal{P}(S\times X)} \\ x \mapsto \lambda P.\, \{s \mid \langle s, x\rangle \in P\} & \qquad\qquad\qquad \Phi \longmapsto \lambda P.\, \Phi(\{\langle s, h\rangle \mid s \in h(P)\}). \end{array}$$

Notice that this unit $\eta(x)\colon \mathcal{P}(S \times X) \to \mathcal{P}(S)$ is the inverse image $(\eta^{\mathcal{S}}(x))^{-1}$ of the unit of the state monad $\mathcal{S}$ from point (5.1.3). In fact, there is a 'map of monads' from state to Dijkstra; see Exercise 5.1.10.

The Hoare monad $\mathcal{H}$ on **Sets** involves dependent sums and products:

$$\mathcal{H}(X) = \coprod_{Pre\subseteq S} \ \coprod_{Post\subseteq S\times X\times S} \ \prod_{s\in Pre} (s \times \mathrm{id} \times \mathrm{id})^{-1}(Post).$$

An element of $\mathcal{H}(X)$ is thus a 'Hoare triple' given by (1) a precondition predicate $Pre \subseteq S$ on the state space S; (2) a postcondition relation $Post \subseteq S \times X \times S$ on an initial state, an output element in X and a final state; (3) a dependent function h which sends a state s satisfying the precondition Pre to a pair of elements $h(s) = (x, s') \in X \times S$ such that $Post(s, x, s')$ – i.e. such that $h(s) \in (s \times \mathrm{id} \times \mathrm{id})^{-1}(Post)$.

For a function $f\colon X \to Y$ we obtain $\mathcal{H}(f)\colon \mathcal{H}(X) \to \mathcal{H}(Y)$ given by

$$\mathcal{H}(f)(P, Q, h) = \langle\, P,\, \{(s, f(x), s') \mid (s, x, s') \in Q\},\, (f \times \mathrm{id}) \circ h\,\rangle.$$

It is not hard to see that this triple is well defined.

The unit $\eta\colon X \to \mathcal{H}(X)$ of this Hoare monad is given by

$$\eta(x) = \langle\, \top,\, \{(s, x, s) \mid s \in S\},\, \lambda s \in S.\, \langle x, s\rangle\,\rangle.$$

Its precondition is the truth predicate $\top \subseteq S$; its postcondition relates the input value x with all identical initial and final states; and its operation is essentially the unit $\eta^{\mathcal{S}}(x)$ of the state monad from point (5.1.3).

The multiplication operation $\mu\colon \mathcal{H}^2(X) \to \mathcal{H}(X)$ is basically the same as multiplication for the state monad, except that the relevant pre- and post-conditions have to be take into a account. Thus μ acts on a triple $\langle P, Q, h\rangle \in \mathcal{H}^2(X)$, where $P \subseteq S$, $Q \subseteq S \times \mathcal{H}(X) \times S$ and

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$h(s) \in (s \times \mathrm{id} \times \mathrm{id})^{-1}(Q)$, for each $s \in P$. We will produce a triple $\mu(P, Q, h) \in \mathcal{H}(X)$, where

$$\begin{aligned} \mu(P, Q, h) = \langle\, & \{s \mid P(s) \wedge \forall g, s'.\, Q(s, g, s') \Rightarrow (\pi_1 g)(s')\}, \\ & \{(s, x, s'') \mid \exists g, s'.\, Q(s, g, s') \wedge (\pi_2 g)(s', x, s'')\}, \\ & \lambda s.\, \mathsf{let}\ (g, s') = h(s)\ \mathsf{in}\ (\pi_3 g)(s') \,\rangle. \end{aligned}$$

To see that μ is well defined, write $\mu(P, Q, h) = (P', Q', h')$ and assume $s \in P'$. Then $s \in P$ and $Q(s, g, s') \Rightarrow (\pi_1 g)(s')$. The first conjunct yields $h(s) \in \{(g, s') \mid (s, g, s') \in Q\}$. So write $h(s) = (g, s')$, where $(s, g, s') \in Q$. Let's write $g \in \mathcal{H}(X)$ as $g = (U, V, k)$. Then $U(s') = (\pi_1 g)(s')$ holds by assumption. Hence $k(s') \in \{(x, s'') \mid (s', x, s'') \in V\}$. Now we write $k(s') = (x, s'')$, where $V(s', x, s'') = (\pi_2 g)(s', x, s'')$. Hence we are done.

We leave the verifications that $\mathcal{H}$ is a monad to the interested reader. Programs represented as maps $X \to \mathcal{H}(Y)$ automatically incorporate appropriate pre- and post-conditions via the dependent types involved. This is the essence of the programming language Ynot; see [364].

One may expect that the multiset functor $\mathcal{M}_M$ from Definition 4.1.1 also forms a monad (like distribution above). This is indeed the case if one assumes that the multiplicities in M used for counting elements not only form an (additive) commutative monoid, but also carry a multiplication operation. In short, multiplicities must form a semiring, that is, a 'ring without additive inverses', sometimes also called a 'rig'. For convenience, we define it explicitly.

Definition 5.1.4 A **semiring** is a set S carrying:

- a commutative monoid structure, written additively as $(0, +)$
- another monoid structure, written multiplicatively as $(1, \cdot)$

in such a way that multiplication distributes over addition:

$$\begin{aligned} 0 \cdot z &= 0 & z \cdot 0 &= 0 \\ (x + y) \cdot z &= x \cdot z + y \cdot y & z \cdot (x + y) &= z \cdot x + z \cdot y. \end{aligned}$$

The semiring is called commutative if its multiplication operation $\cdot$ is commutative.

A morphism of semirings $f\colon S \to R$ is a function between the underlying sets which is both additively and a multiplicatively a homomorphism of monoids. We write **SRng** for the category of semirings and their homomorphisms. The full subcategory of commutative semirings is then written as **CSRng** $\hookrightarrow$ **SRng**.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Recall that a ring is a semiring with additive inverses, so the additive monoid involved is an (Abelian) group. The natural numbers $\mathbb{N}$ form a semiring, but not a ring. Similarly, the non-negative natural numbers $\mathbb{R}_{\geq 0} = \{r \in \mathbb{R} \mid r \geq 0\}$ form a semiring but not a ring.

Lemma 5.1.5 *If S is a semiring, then the multiset functor $\mathcal{M}_S :$ **Sets** $\rightarrow$ **Sets**, taking multiplicities in S, is a monad, with unit and multiplication as for the distribution monad:*

$$\eta(x) = 1|x\rangle \qquad \textit{and} \qquad \mu\big(\textstyle\sum_i s_i|\varphi_i\rangle\big)(x) = \textstyle\sum_i s_i \cdot \varphi_i(x),$$

where $\varphi_i \in \mathcal{M}_S(X) = \{\psi \colon X \rightarrow S \mid \mathrm{supp}(\psi)$ is finite$\}$. ❑

The proof involves some routine calculations and is left to the reader. Bascially, the multiplication μ is given by matrix multiplication; see Exercise 5.1.3 below.

We turn to some more categorical aspects of monads. The following basic result (see e.g. [344, VI 1]) shows that adjunctions form a source of monads.

Lemma 5.1.6 *For each adjunction $F \dashv G$ the endofunctor GF is a monad, as depicted in*

$$\begin{tikzcd} \mathbb{D} \arrow[d, "G"] \\ \mathbb{C} \arrow[u, bend left=60, "F"] \arrow[u, phantom, "\dashv"] \arrow[loop right, "T = GF \text{ monad}"'] \end{tikzcd}$$

Proof Write the unit and counit of the adjunction $F \dashv G$ as $\eta \colon \mathrm{id}_{\mathbb{C}} \Rightarrow GF$ and $\varepsilon \colon FG \Rightarrow \mathrm{id}_{\mathbb{D}}$. They satisfy the triangular identities $G(\varepsilon_Y) \circ \eta_{GY} = \mathrm{id}_{GY}$ and $\varepsilon_{FX} \circ F(\eta_X) = \mathrm{id}_{FX}$ described in Exercise 2.5.7. We write $T = GF \colon \mathbb{C} \rightarrow \mathbb{C}$. The unit η of the adjunction forms the unit $\eta \colon \mathrm{id}_{\mathbb{C}} \Rightarrow T$ for the monad T. The multiplication $\mu \colon T^2 \Rightarrow T$ is defined with component at $X \in \mathbb{C}$:

$$TT(X) = GFGF(X) \xrightarrow{\mu_X \overset{\text{def}}{=} G(\varepsilon_{FX})} GF(X) = T(X). \tag{5.1}$$

It is easy to see that this is a natural transformation satisfying the monad requirements from Definition 5.1.1. For instance,

$$\begin{aligned} \mu_X \circ T(\eta_X) &= G(\varepsilon_{FX}) \circ GF(\eta_X) \\ &= G(\varepsilon_{FX} \circ F(\eta_X)) \\ &= G(\mathrm{id}_{FX}) \\ &= \mathrm{id}_{TX}. \end{aligned}$$ ❑

Of course, in category theory one does not introduce a notion without saying what the associated morphism is.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Definition 5.1.7 Let $T, S : \mathbb{C} \to \mathbb{C}$ be two monads on the same category. A **map of monads** $\sigma \colon T \Rightarrow S$ is a natural transformation that commutes with the respective units and multiplications, as in

$$\begin{tikzcd}
X \arrow[r, equal] \arrow[d, "\eta_X"'] & X \arrow[d, "\eta_X"] \\
T(X) \arrow[r, "\sigma_X"'] & S(X)
\end{tikzcd}
\qquad
\begin{tikzcd}
T^2(X) \arrow[r, "\sigma_{TX}"] \arrow[d, "\mu_X"'] & S(T(X)) \arrow[r, "S(\sigma_X)"] & S^2(X) \arrow[d, "\mu_X"] \\
T(X) \arrow[rr, "\sigma_X"'] & & S(X)
\end{tikzcd}$$

In this way one obtains a category $\mathbf{Mnd}(\mathbb{C})$ of monads on $\mathbb{C}$ with maps between them.

Maps of monads arise naturally; see for instance Exercises 5.1.7 and 5.1.8 describing the functoriality of some monad constructions from Example 5.1.3. Also, the earlier examples (4.5) of natural transformations between collection functors are all maps of monads.

We also use maps of monads in the following result. It describes how every endofunctor can be turned into a monad, in a free manner, assuming that certain initial algebras exist.

Proposition 5.1.8 *Let $F \colon \mathbb{C} \to \mathbb{C}$ be an endofunctor on a category $\mathbb{C}$ with coproducts $+$. Assume for each object X, the functor $X + F(-) \colon \mathbb{C} \to \mathbb{C}$ has an initial algebra, written as*

$$X + F(F^*(X)) \xrightarrow[\cong]{\alpha_X} F^*(X).$$

The mapping $X \mapsto F^*(X)$ forms a monad, that is the free one on the functor F, via a universal natural transformation $\theta \colon F \Rightarrow F^*$.

Proof We first show that $X \mapsto F^*(X)$ is functorial. For a map $f \colon X \to Y$ in $\mathbb{C}$ write $F^*(f) \colon F^*(X) \to F^*(Y)$ for the unique algebra homomorphism obtained by initiality in

$$\begin{tikzcd}
X + F(F^*(X)) \arrow[r, dashed, "\mathrm{id} + F(F^*(f))"] \arrow[dd, "\alpha_X"', "\cong"] & X + F(F^*(Y)) \arrow[d, "f + \mathrm{id}"] \\
& Y + F(F^*(Y)) \arrow[d, "\alpha_Y", "\cong"'] \\
F^*(X) \arrow[r, dashed, "F^*(f)"'] & F^*(Y)
\end{tikzcd} \tag{5.2}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

It is easy to see that F^* preserves identities and composition. We get a unit natural transformation $\eta\colon \mathrm{id} \Rightarrow F^*$ with components

$$\eta_X \overset{\text{def}}{=} \Big(X \xrightarrow{\kappa_1} X + F(F^*(X)) \xrightarrow[\cong]{\alpha_X} F^*(X)\Big).$$

It is natural since for $f\colon X \to Y$,

$$\begin{aligned} F^*(f) \circ \eta_X &= \alpha_Y \circ (f + \mathrm{id}) \circ (\mathrm{id} + F(F^*(f))) \circ \alpha_X^{-1} \circ \alpha_X \circ \kappa_1 \\ &= \alpha_Y \circ \kappa_1 \circ f \;=\; \eta_Y \circ f. \end{aligned}$$

The multiplication map $\mu\colon F^*F^* \Rightarrow F^*$ arises as a unique map in

$$\begin{tikzcd} F^*(X) + F(F^*F^*(X)) \arrow[r, dashed, "\mathrm{id} + F(\mu_X)"] \arrow[d, "\alpha_{F^*(X)}"', "\cong"] & F^*(X) + F(F^*(X)) \arrow[d, "{[\mathrm{id}, \alpha_X \circ \kappa_2]}"] \\ F^*F^*(X) \arrow[r, dashed, "\mu_X"'] & F^*(X) \end{tikzcd}$$

The first of the monad equations is easy:

$$\begin{aligned} \mu_X \circ \eta_{F^*(X)} &= \mu_X \circ \alpha_{F^*(X)} \circ \kappa_1 \\ &= [\mathrm{id}, \alpha_X \circ \kappa_2] \circ (\mathrm{id} + F(\mu_X)) \circ \kappa_1 \\ &= [\mathrm{id}, \alpha_X \circ \kappa_2] \circ \kappa_1 \\ &= \mathrm{id}. \end{aligned}$$

The other two equations can be obtained via (the uniqueness part of) initiality – which we leave to the reader. We continue with the universal property. There is a natural transformation $\theta\colon F \Rightarrow F^*$ with components

$$\theta_X \overset{\text{def}}{=} \Big(F(X) \xrightarrow{F(\eta_X)} F(F^*(X)) \xrightarrow{\kappa_2} X + F(F^*(X)) \xrightarrow[\cong]{\alpha_X} F^*(X)\Big).$$

If $T = (T, \eta^T, \mu^T)$ is an arbitrary monad with a natural transformation $\sigma\colon F \Rightarrow T$, then there is a unique map of monads $\overline{\sigma}\colon F^* \Rightarrow T$ with $\overline{\sigma} \circ \theta = \sigma$. This $\overline{\sigma}$ is obtained by initiality in

$$\begin{tikzcd} X + F(F^*(X)) \arrow[r, dashed, "\mathrm{id} + F(\overline{\sigma}_X)"] \arrow[d, "\alpha_X"', "\cong"] & X + F(T(X)) \arrow[d, "{[\eta^T_X, \mu^T_X \circ \sigma_{T(X)}]}"] \\ F^*(X) \arrow[r, dashed, "\overline{\sigma}_X"'] & T(X) \end{tikzcd}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Then indeed

$$\begin{aligned}
\overline{\sigma}_X \circ \theta_X &= \overline{\sigma}_X \circ \alpha_X \circ \kappa_2 \circ F(\eta_X) \\
&= [\eta^T_X, \mu^T_X \circ \sigma_{T(X)}] \circ (\mathrm{id} + F(\overline{\sigma}_X)) \circ \kappa_2 \circ F(\eta_X) \\
&= \mu^T_X \circ \sigma_{T(X)} \circ F(\overline{\sigma}_X)) \circ F(\eta_X) \\
&= \mu^T_X \circ T(\overline{\sigma}_X) \circ T(\eta_X) \circ \sigma_X \\
&= \mu^T_X \circ T(\overline{\sigma}_X \circ \alpha_X \circ \kappa_1) \circ \sigma_X \\
&= \mu^T_X \circ T([\eta^T_X, \mu^T_X \circ \sigma_{T(X)}] \circ (\mathrm{id} + F(\overline{\sigma}_X)) \circ \kappa_1) \circ \sigma_X \\
&= \mu^T_X \circ T(\eta^T_X) \circ \sigma_X \\
&= \sigma_X.
\end{aligned}$$

Uniqueness of $\overline{\sigma}$ is left to the interested reader. ❑

Free monads will be used to give a systematic description of terms in algebraic specification; see Sections 6.6 and 6.7. There we shall see further properties of such free monads (on **Sets**), such as: F^* is finitary (or weak pullback preserving) if F is; see Lemma 6.6.4 and Exercise 6.7.4.

5.1.1 Comonads

A comonad is the dual of a monad, in the sense that a comonad on category $\mathbb{C}$ is a monad on the opposite category $\mathbb{C}^{\mathrm{op}}$. Basically, we can now use what we have already learned about monads. However, it is convenient to make a few things explicit.

Definition 5.1.9 A **comonad** on a category $\mathbb{C}$ is given by a functor $S : \mathbb{C} \to \mathbb{C}$ together with two natural transformations, namely a **counit** $\varepsilon\colon S \Rightarrow \mathrm{id}$ and a **comultiplication** $\delta\colon S \Rightarrow S^2$ making for each object $X \in \mathbb{C}$ the following diagrams commute:

$$\begin{tikzcd}
 & S(X) \arrow[dl, equal] \arrow[d, "\delta_X"] \arrow[dr, equal] & \\
S(X) & S^2(X) \arrow[l, "S(\varepsilon_X)"] \arrow[r, "\varepsilon_{S(X)}"'] & S(X)
\end{tikzcd}
\qquad
\begin{tikzcd}
S(X) \arrow[r, "\delta_X"] \arrow[d, "\delta_X"'] & S^2(X) \arrow[d, "S(\delta_X)"] \\
S^2(X) \arrow[r, "\delta_{S(X)}"'] & S^3(X)
\end{tikzcd}$$

A **map of comonads** is a natural transformation $\sigma\colon S \Rightarrow S'$ that commutes appropriately with the counits ε and comultiplications δ.

In Lemma 5.1.6 we have seen that an adjunction $F \dashv G$ gives rise to a monad GF. It is not hard to see that the (other) composite FG is a comonad. Still, comonads are not so common as monads. We conclude by listing some examples. They can all be understood as providing structure to the context of

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

computations (as in [454]). Further examples can be obtained as cofree comonads on a functor, via final coalgebras (see Exercise 5.1.15). In Section 6.8 we will see how to obtain comonads from coalgebraic specifications.

Example 5.1.10 Probably the clearest example of a comonad is given by streams, i.e. by the functor $X \mapsto X^{\mathbb{N}}$ sending a set X to the set $X^{\mathbb{N}}$ of infinite sequences of its elements:

$$\begin{tikzcd}
X & X^{\mathbb{N}} \arrow[l, "\varepsilon"'] \arrow[r, "\delta"] & (X^{\mathbb{N}})^{\mathbb{N}} \\
\alpha(0) & \alpha \arrow[l, mapsto] \arrow[r, mapsto] & \lambda n.\, \lambda m.\, \alpha(n+m).
\end{tikzcd}$$

The counit ε thus selects the head of the stream. The comultiplication δ maps a stream to a stream of streams, where the ith stream of $\delta(\alpha)$ is the substream $\alpha(i), \alpha(i+1), \alpha(i+2), \ldots$ of α starting at i. It is not hard to see that this forms a comonad.

One way of understanding this stream comonad $X^{\mathbb{N}}$ is as providing infinitely many copies of X as input for a computation. This is also the idea of the exponential ! in linear logic [155] – which is standardly modelled via a comonad; see e.g. [424, 64]. There are some variations on the stream comonad, with additional structure keeping track of a position in the sequence. We shall describe them in concrete form, for both discrete time and real time inputs. Exercise 5.1.11 describes some of these comonads in more abstract form.

The diagram below presents the discrete time comonads, together with comonad maps between them:

$$\begin{tikzcd}
X^{\star} \times X & X^{\mathbb{N}} \times \mathbb{N} \arrow[l, "\text{no future}"'] \arrow[r, "\text{no past}"] & X^{\mathbb{N}} \\
(\langle \alpha(0), \ldots, \alpha(n-1) \rangle, \alpha(n)) & (\alpha, n) \arrow[l, mapsto] \arrow[r, mapsto] & \lambda m.\, \alpha(n+m).
\end{tikzcd} \tag{5.3}$$

The comonad structure for $X^{\mathbb{N}} \times \mathbb{N}$ in the middle is

$$\begin{tikzcd}
X & X^{\mathbb{N}} \times \mathbb{N} \arrow[l, "\varepsilon"'] \arrow[r, "\delta"] & (X^{\mathbb{N}} \times \mathbb{N})^{\mathbb{N}} \times \mathbb{N} \\
\alpha(n) & (\alpha, n) \arrow[l, mapsto] \arrow[r, mapsto] & (\lambda m.\, (\alpha, m), n).
\end{tikzcd}$$

The intuition for a pair $(\alpha, n) \in X^{\mathbb{N}} \times \mathbb{N}$ is that the number n represents the present stage in the stream $\alpha = \langle \alpha(0), \alpha(1), \ldots, \alpha(n-1), \alpha(n), \alpha(n+1), \ldots \rangle$, where everything before n is past input, and everything after n is future input.

The comonad structure for $X^{\star} \times X$ on the left in (5.3) is

$$\begin{tikzcd}
X & X^{\star} \times X \arrow[l, "\varepsilon"'] \arrow[r, "\delta"] & (X^{\star} \times X)^{\star} \times X \\
x & (\alpha, x) \arrow[l, mapsto] \arrow[r, mapsto] & (\langle (\langle\rangle, x_1), \ldots, (\langle x_1, \ldots, x_{n-1} \rangle, x_n) \rangle,\ (\alpha, x)),
\end{tikzcd}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

where $\alpha = \langle x_1, \ldots, x_n \rangle$. The comultiplication δ thus turns a pair (α, x) into another pair $(\beta, (\alpha, x))$, where β is a list of pairs $(\alpha_{<i}, x_i)$, in which x_i is the ith element in α and $\alpha_{<i}$ is the sublist of α of elements up to i.

The two arrows in the diagram (5.3) are homomorphisms of comonads, commuting with the relevant comonad/context structure.

The real-time analogue uses the (semiring of) non-negative real numbers $\mathbb{R}_{\geq 0} = \{r \in \mathbb{R} \mid r \geq 0\}$ to represent time. One obtains a diagram like (5.3):

$$\begin{tikzcd}
\Big(\coprod_{t \in \mathbb{R}_{\geq 0}} X^{[0,t)}\Big) \times X & X^{\mathbb{R}_{\geq 0}} \times \mathbb{R}_{\geq 0} \arrow[l] \arrow[r] & X^{\mathbb{R}_{\geq 0}}.
\end{tikzcd} \tag{5.4}$$

The leftmost comonad may also be described as: $\coprod_{t \in \mathbb{R}_{\geq 0}} X^{[0,t]}$.

The notion of ‘arrow’ developed in [229] forms a generalisation of both monads and comonads; see also [394, 255, 337].

Exercises

5.1.1 Let $\mathbb{C}$ be a category with coproducts $+$. Fix an arbitrary object $A \in \mathbb{C}$ and prove that the functor $X \mapsto A + X$ forms a monad on $\mathbb{C}$.

5.1.2 Describe sequential composition $\circledcirc$ for Java programs, modelled as coalgebras $X \to \mathcal{J}(X)$, using the Java monad $\mathcal{J}$ described in Example 5.1.3.8.

5.1.3 Let S be a semiring and X a finite set, say $X = \{x_1, \ldots, x_n\}$.

1. Check that coalgebras $c \colon X \to \mathcal{M}_S(X)$ can be identified with $n \times n$ matrices, with entries in S; write M_c for the matrix corresponding to c.
2. Prove that $M_{d \circledcirc c} = M_c M_d$, where the left-hand side uses sequential composition $\circledcirc$ for coalgebras (from Lemma 5.1.2) and the right-hand side involves matrix composition.
3. Check that the same works for finite Markov chains, as coalgebras $X \to \mathcal{D}(X)$, where X is finite.

5.1.4 Fix a set C and consider the contravariant functor $X \mapsto C^X$. Prove that there is an adjunction

$$\begin{tikzcd}
\mathbf{Sets}^{\mathrm{op}} \arrow[r, bend left, "C^{(-)}"] \arrow[r, phantom, "\top"] & \mathbf{Sets} \arrow[l, bend left, "C^{(-)}"]
\end{tikzcd}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Check that the monad induced by this adjunction on **Sets**, as in Lemma 5.1.6, is the continuation monad from Example 5.1.3.7.

5.1.5 Fix a set C and consider the continuation monad $\mathcal{C}(X) = C^{(C^X)}$ from Example 5.1.3.7. Assume the set C carries an operation $f\colon C^n \to C$, for some $n \in \mathbb{N}$. Prove that it gives rise to a natural transformation:

$$(\mathcal{C}(X))^n \Longrightarrow \mathcal{C}(X).$$

5.1.6 Recall from Section 2.2.6 that Turing machines can be modelled as coalgebras $X \to T(n \cdot X)^n$, where n is the number of (control) states.

1. ([249]) Show that the mapping $X \mapsto T(n \cdot X)^n$ is a monad if T is a monad.
2. Describe the two-step composite $c \circledcirc c$ for the Turing-machine-as-coalgebra example monad c described in (2.27) and (2.28).

5.1.7 Show that sending a monoid M to the monad $M \times (-)$, as in Example 5.1.3.5, yields a functor **Mon** → **Mnd**(**Sets**). (This functor has a right adjoint; see Exercise 5.2.17.)

5.1.8 Prove that mapping a semiring S to the corresponding multiset monad $\mathcal{M}_S$ yields a functor **SRng** → **Mnd**(**Sets**).

5.1.9 Prove that the support natural transformation $\mathcal{D} \Rightarrow \mathcal{P}$ from (4.5) is a map of monads.

5.1.10 Prove that there is a map of monads $\mathcal{S} \Rightarrow \mathfrak{D}$, from the state monad $\mathcal{S}$ to the Dijkstra monad $\mathfrak{D}$ (see Examples 5.1.3.6 and 5.1.3.10), given by inverse image

$$\mathcal{S}(X) = (S \times X)^S \xrightarrow{\;h \,\mapsto\, h^{-1}\;} \mathcal{P}(S)^{\mathcal{P}(S \times X)} = \mathfrak{D}(X).$$

It sends a program to the corresponding weakest precondition operation.

5.1.11 Show that for each monoid M and set E the following functors are comonads on **Sets**:

1. $X \mapsto X^M$
2. $X \mapsto X^E \times E$.

Show also that there is a map of comonads $(-)^M \times M \Rightarrow (-)^M$, as used twice in Example 5.1.10.

5.1.12 Prove that the arrows $X^{\mathbb{N}} \times \mathbb{N} \to X^{\star} \times X$ in Diagram (5.3) form a map of comonads.

5.1.13 Define the comonad structure on $\left(\coprod_{t \in \mathbb{R}_{\geq 0}} (-)^{[0,t)}\right) \times X$ and the comonad map $(-)^{\mathbb{R}_{\geq 0}} \times \mathbb{R}_{\geq 0} \Rightarrow \left(\coprod_{t \in \mathbb{R}_{\geq 0}} (-)^{[0,t)}\right) \times X$ in Diagram (5.4).

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

5.1.14 Let T be a monad on a category $\mathbb{C}$, and let $\mathbb{A}, \mathbb{B}$ be arbitrary categories. Prove that pre- and post-composition with T yields monads $(-) \circ T \colon \mathbb{A}^{\mathbb{C}} \to \mathbb{A}^{\mathbb{C}}$ and $T \circ (-) \colon \mathbb{C}^{\mathbb{B}} \to \mathbb{C}^{\mathbb{B}}$ on categories of functors $\mathbb{C} \to \mathbb{A}$ and $\mathbb{B} \to \mathbb{C}$.

5.1.15 Prove the dual of Proposition 5.1.8: the cofree comonad on a functor $F \colon \mathbb{C} \to \mathbb{C}$ can be constructed as the mapping $F^{\infty} \colon \mathbb{C} \to \mathbb{C}$ that sends $X \in \mathbb{C}$ to the carrier of the final coalgebra of the functor $X \times F(-)$, assuming this final coalgebra exists. Check that:

1. This comonad arises from the cofree coalgebra construction in Proposition 2.5.3.
2. The stream comonad $X \mapsto X^{\mathbb{N}}$ is the cofree comonad on the identity functor.

5.2 Kleisli Categories and Distributive Laws

This section shows how to associate with a monad T a category of 'computations of type T'. This category is called the Kleisli category of T and is written as $\mathcal{K}\ell(T)$. Composition of morphisms in Kleisli categories generalises composition $\circledcirc$ of coalgebras from Lemma 5.1.2. Dually, for comonads S there is a also a Kleisli category, for which we use the same notation: $\mathcal{K}\ell(S)$. In general, it will be clear from the context whether this is a Kleisli category of a monad or of a comonad.

The basics of the theory of Kleisli categories are described in this section. Included is the notion of distributive law that will be used to lift functors to Kleisli categories. Such liftings play a crucial role in the description of trace semantics for coalgebras in the next section.

Definition 5.2.1 Let T be a monad on a category $\mathbb{C}$. The **Kleisli** category $\mathcal{K}\ell(T)$ of T has the same objects as $\mathbb{C}$, but morphisms $X \to Y$ in $\mathcal{K}\ell(T)$ are maps $X \to T(Y)$ in $\mathbb{C}$.

The identity map $X \to X$ in $\mathcal{K}\ell(T)$ is the unit $\eta_X \colon X \to T(X)$. Composition of $f \colon X \to Y$ and $g \colon Y \to Z$ in $\mathcal{K}\ell(T)$, that is, of $f \colon X \to T(Y)$ and $g \colon Y \to T(Z)$ in $\mathbb{C}$, is written as $\circledcirc$ and defined as in Lemma 5.1.2:

$$g \circledcirc f \stackrel{\text{def}}{=} \Big(X \xrightarrow{\;f\;} T(Y) \xrightarrow{\;T(g)\;} T^2(Z) \xrightarrow{\;\mu_Z\;} T(Z)\Big).$$

Having a separate notation for composition in Kleisli categories may help to distinguish composition in $\mathbb{C}$ from composition in $\mathcal{K}\ell(T)$. Kleisli maps $X \to Y$ may be seen as generalised coalgebras $X \to T(Y)$, with different domain and

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

codomain. The composition monoid described in Lemma 5.1.2 is the monoid of endomorphisms $X \to X$ in the Kleisli category $\mathcal{K}\ell(T)$.

The Kleisli construction captures many well-known categories. For instance, the Kleisli category $\mathcal{K}\ell(\mathcal{P})$ of the powerset monad is the category **SetsRel** of sets with relations between them – using the correspondence (2.16). The Kleisli category $\mathcal{K}\ell(\mathcal{L})$ of the lift monad is the category of sets and partial functions. The Kleisli category $\mathcal{K}\ell(\mathcal{D})$ of the distribution monad is the category of sets with Markov chains or 'stochastic relations' between them.

There are some general observations about Kleisli categories.

Proposition 5.2.2 *Let T be a monad on an arbitrary category $\mathbb{C}$.*

1. *There is an adjunction*

$$\begin{tikzcd} \mathcal{K}\ell(T) \arrow[d, "U"] \\ \mathbb{C} \arrow[u, bend left=60, "J"] \arrow[u, phantom, "\dashv" near start, shift right=3] \end{tikzcd} \quad \textit{where} \quad \begin{cases} U(Y) = T(Y) \\ U(f) = \mu \circ T(f) \end{cases} \quad \textit{and} \quad \begin{cases} J(X) = X \\ J(f) = \eta \circ f. \end{cases}$$

2. *If the monad T comes as $T = HL$ from an adjunction $L \dashv H$, as in Lemma 5.1.6, then there is a 'comparison' functor K in*

$$\begin{tikzcd} \mathcal{K}\ell(T) \arrow[rr, "K"] \arrow[dr, "U"] & & \mathbb{D} \arrow[dl, bend left, "H"] \\ & \mathbb{C} \arrow[ul, bend left, "J"] \arrow[ur, "L"] \arrow[loop below, "T = HL"] & \end{tikzcd} \quad \textit{where} \quad \begin{cases} K(Y) = L(Y) \\ K(f) = \varepsilon \circ L(f). \end{cases}$$

This functor comparison functor K satisfies $K \circ J = L$ and $H \circ K = U$.

3. *The Kleisli category inherits coproducts and coequalisers from $\mathbb{C}$ – if any – and $J \colon \mathbb{C} \to \mathcal{K}\ell(T)$ preserves them.*

Proof 1. It is not hard to see that J and U as defined above are functors, using some care with respect to the order of compositions. For instance

$$\begin{aligned} U(g \circledcirc f) &= \mu \circ T(\mu \circ T(g) \circ f) \\ &= \mu \circ \mu \circ T^2(g) \circ T(f) && \text{via the } \mu\text{-law from Definition 5.1.1} \\ &= \mu \circ T(g) \circ \mu \circ T(f) && \text{by naturality of } \mu \\ &= U(g) \circ U(f) \\ J(g \circ f) &= \eta \circ g \circ f \\ &= T(g) \circ \eta \circ f \\ &= \mu \circ T(\eta \circ g) \circ \eta \circ f \\ &= (\eta \circ g) \circledcirc (\eta \circ f) \\ &= J(g) \circledcirc J(f). \end{aligned}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

The adjunction $J \dashv U$ follows directly by unravelling the structure:

$$\begin{array}{cl} J(X) = X \xrightarrow{\;f\;} Y & \text{in } \mathcal{K}\ell(T) \\ \hline\hline X \xrightarrow[f]{} T(Y) = U(Y) & \text{in } \mathbb{C}. \end{array}$$

2. Let $\eta\colon \mathrm{id} \Rightarrow HL = T$ and $\varepsilon\colon LH \Rightarrow \mathrm{id}$ be the unit and counit of the adjunction $L \dashv H$. For a morphism $f\colon X \to Y$ in $\mathcal{K}\ell(T)$, that is for $f\colon X \to HLY$ in $\mathbb{C}$, the transpose $K(f) = \varepsilon_{LY} \circ L(f)\colon LX \to LY$ yields a map in $\mathbb{D}$. This K preserves identities and composition, since

$$\begin{array}{rcll} K(\mathrm{id}_X) & = & K(\eta_X) & \\ & = & \varepsilon_{LX} \circ L(\eta_X) & \\ & = & \mathrm{id}_{LX} & \text{by the triangular identities} \\ K(g \circledcirc f) & = & K(\mu \circ T(g) \circ f) & \\ & = & \varepsilon_{LZ} \circ L(H(\varepsilon_{LZ}) \circ HL(g) \circ f) & \text{by (5.1)} \\ & = & \varepsilon_{LZ} \circ L(g) \circ \varepsilon_{LY} \circ L(f) & \text{by naturality} \\ & = & K(g) \circ K(f). & \end{array}$$

3. We shall do the (simple) case of binary coproducts $+$ in $\mathbb{C}$. They also form coproducts in $\mathcal{K}\ell(T)$ with coprojections $J(\kappa_i) = \eta \circ \kappa_i\colon X_i \to X_1 + X_2$. For two maps $f_i\colon X_i \to Y$ in $\mathcal{K}\ell(T)$, that is for $f_i\colon X_i \to T(Y)$ in $\mathbb{C}$, their cotuple $[f_1, f_2]\colon X_1 + X_2 \to T(Y)$ in $\mathbb{C}$ forms the cotuple in $\mathcal{K}\ell(T)$. Alternatively, one can use the bijective correspondences

$$\begin{array}{c} X + Y \longrightarrow Z \\ \hline\hline X + Y \longrightarrow T(Z) \\ \hline\hline \begin{array}{c} X \longrightarrow T(Z) \\ \hline\hline X \longrightarrow Z \end{array} \qquad \begin{array}{c} Y \longrightarrow T(Z) \\ \hline\hline Y \longrightarrow Z \end{array} \end{array}$$

where the arrows at the top and bottom are in $\mathcal{K}\ell(T)$. ❑

This Kleisli construction inspires an alternative formulation of the notion of monad that is popular in the (functional) programming community. It avoids functoriality and naturality and speaks only about a type constructor with operations satisfying some equations.

Proposition 5.2.3 *Let $\mathbb{C}$ be an arbitrary category with*

- *a mapping $\mathbb{C} \ni X \longmapsto T(X) \in \mathbb{C}$, on objects only*
- *a map $\eta_X\colon X \to T(X)$, for each $X \in \mathbb{C}$.*

The following are then equivalent:

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

1. *The mapping $X \mapsto T(X)$ is functorial, η is a natural transformation, and there is a natural transformation $\mu\colon T^2 \Rightarrow T$ making (T, η, μ) a monad.*
2. *There is a 'Kleisli extension' operation $(-)^\$$ sending*

$$X \xrightarrow{f} T(Y) \qquad \textit{to} \qquad T(X) \xrightarrow{f^\$} T(Y) \tag{5.5}$$

in such a way that the following equations hold:

$$f^\$ \circ \eta = f \qquad\qquad \eta^\$ = \mathrm{id} \qquad\qquad g^\$ \circ f^\$ = (g^\$ \circ f)^\$.$$

Proof First assume we have a monad (T, η, μ). Then define extension $(-)^\$$ as

$$f^\$ \stackrel{\text{def}}{=} \Big(T(X) \xrightarrow{T(f)} T^2(Y) \xrightarrow{\mu_Y} T(Y)\Big).$$

It is not hard to check that it satisfies the above three equations.

Conversely, assume a Kleisli extension operation $(-)^\$$ as in (5.5). We first turn the mapping $X \mapsto T(X)$ into a functor by defining for a map $f\colon X \to Y$ a new map $T(f) = (\eta_Y \circ f)^\$\colon T(X) \to T(Y)$. Then $T(\mathrm{id}) = \mathrm{id}$ obviously holds, and

$$\begin{aligned} T(g) \circ T(f) &= (\eta \circ g)^\$ \circ (\eta \circ f)^\$ \\ &= \big((\eta \circ g)^\$ \circ \eta \circ f\big)^\$ \\ &= (\eta \circ g \circ f)^\$ \\ &= T(g \circ f). \end{aligned}$$

This also makes the maps $\eta_X\colon X \to T(X)$ natural: for $f\colon X \to Y$,

$$T(f) \circ \eta_X = (\eta_Y \circ f)^\$ \circ \eta_X = \eta_Y \circ f.$$

The multiplication $\mu_X\colon T^2(X) \to T(X)$ can then be defined as $\mu_X = (\mathrm{id}_{TX})^\$$. Remaining details are left to the interested reader. ❑

With this Kleisli extension $(-)^\$$ one can define composition $g \odot f$ in a Kleisli category as $g^\$ \circ f$.

For the record we mention that the **Kleisli category** category $\mathcal{K}\ell(S)$ of a comonad S has the same objects as $\mathbb{C}$ and its morphisms $X \to Y$ are maps $S(X) \to Y$ in $\mathbb{C}$. The obvious functor $\mathcal{K}\ell(S) \to \mathbb{C}$ from the Kleisli category of a comonad has a right adjoint. Further, Kleisli categories of comonads inherit limits (products and equalisers) from the underlying category.

We come to the next big topic of this section.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Definition 5.2.4 Let $T: \mathbb{C} \to \mathbb{C}$ be a monad and $F: \mathbb{C} \to \mathbb{C}$ an ordinary endofunctor, both on the same category $\mathbb{C}$. A **distributive law** or also a **$\mathcal{K}\ell$-law** of F over T is a natural transformation $\lambda: FT \Rightarrow TF$ that is compatible with T's monad structure:

$$\begin{tikzcd}
F(X) \arrow[r, equal] \arrow[d, "F(\eta_X)"'] & F(X) \arrow[d, "\eta_{F(X)}"] \\
FT(X) \arrow[r, "\lambda_X"'] & TF(X)
\end{tikzcd}
\qquad
\begin{tikzcd}
FT^2(X) \arrow[r, "\lambda_{T(X)}"] \arrow[d, "F(\mu_X)"'] & TFT(X) \arrow[r, "T(\lambda_X)"] & T^2F(X) \arrow[d, "\mu_{F(X)}"] \\
FT(X) \arrow[rr, "\lambda_X"'] & & TF(X)
\end{tikzcd} \tag{5.6}$$

Distributive laws $FT \Rightarrow TF$ as described here correspond to liftings to Kleisli categories $\mathcal{K}\ell(T)$. That is why we also refer to these distributive laws as $\mathcal{K}\ell$-laws. Such liftings play an important role in the next section. Later on, we shall also see distributive laws $TG \Rightarrow GT$ that correspond to liftings to Eilenberg–Moore categories $\mathcal{EM}(T)$. We shall call such distributive laws $\mathcal{EM}$-laws.

Proposition 5.2.5 *Let $T: \mathbb{C} \to \mathbb{C}$ be a monad and $F: \mathbb{C} \to \mathbb{C}$ a functor. Then there is a bijective correspondence between*

1. *a distributive $\mathcal{K}\ell$-law $\lambda: FT \Rightarrow TF$*
2. *a lifting of $F: \mathbb{C} \to \mathbb{C}$ to a functor $\mathcal{K}\ell(F): \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ in a commuting diagram:*

$$\begin{tikzcd}
\mathcal{K}\ell(T) \arrow[r, "\mathcal{K}\ell(F)"] & \mathcal{K}\ell(T) \\
\mathbb{C} \arrow[u, "J"] \arrow[r, "F"'] & \mathbb{C} \arrow[u, "J"']
\end{tikzcd} \tag{5.7}$$

The notation $\mathcal{K}\ell(F)$ for this lifting is convenient, but a bit dangerous: not only does it leave the dependence on λ implicit – so $\mathcal{K}\ell_\lambda(F)$ would be better – but it may also give the wrong impression that the Kleisli construction as such is functorial. Bearing that caveat in mind, we continue to write $\mathcal{K}\ell(F)$ for the lifting.

Proof Assuming a distributive law $\lambda: FT \Rightarrow TF$ we can define the functor $\mathcal{K}\ell(F): \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ as

$$\mathcal{K}\ell(F)(X) = F(X) \qquad \mathcal{K}\ell(F)(X \xrightarrow{f} Y) = (F(X) \xrightarrow{F(f)} FT(Y) \xrightarrow{\lambda_X} TF(Y)).$$

The above two requirements in (5.6) for λ precisely say that $\mathcal{K}\ell(F)$ is a functor. For instance,

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$\begin{aligned}
\mathcal{K}\ell(F)(g \circledcirc f) &= \lambda \circ F(\mu \circ T(g) \circ f) \\
&= \mu \circ T(\lambda) \circ \lambda \circ FT(g) \circ F(f) \\
&= \mu \circ T(\lambda) \circ TF(g) \circ \lambda \circ F(f) \\
&= \mu \circ T(\mathcal{K}\ell(F)(g)) \circ \mathcal{K}\ell(f) \\
&= \mathcal{K}\ell(F)(g) \circledcirc \mathcal{K}\ell(F)(f).
\end{aligned}$$

Conversely, assume there is a functor $G\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ in a commuting square as in (5.7). Then, on objects, $G(X) = F(X)$. Further, for a map $f\colon X \to T(Y)$ in $\mathbb{C}$ we get $G(f)\colon F(X) \to TF(Y)$ in $\mathbb{C}$. This suggests how to define a distributive law: the identity map $\mathrm{id}_{TX}\colon T(X) \to T(X)$ in $\mathbb{C}$ forms a map $T(X) \to X$ in $\mathcal{K}\ell(T)$, so that we can define $\lambda_X = G(\mathrm{id}_{TX})\colon FT(X) \to TF(X)$ in $\mathbb{C}$. We check that these λs form a natural transformation, again using commutation of the diagram: for an arbitrary $f\colon X \to Y$ in $\mathbb{C}$ we have $G(\eta \circ f) = \eta \circ F(f)$. Then we get in $\mathbb{C}$:

$$\begin{aligned}
TF(f) \circ \lambda_X &= TF(f) \circ G(\mathrm{id}_{TX}) \\
&= \mu \circ T(\eta \circ F(f)) \circ G(\mathrm{id}_{TX}) \\
&= G(\eta \circ f) \circledcirc G(\mathrm{id}_{TX}) \\
&= G((\eta \circ f) \circledcirc \mathrm{id}_{TX}) \\
&= G(\mu \circ T(\eta \circ f) \circ \mathrm{id}_{TX}) \\
&= GT(f) \\
&= G(\mu \circ T(\mathrm{id}_{TY}) \circ \eta \circ Tf) \\
&= G(\mathrm{id}_{TY} \circledcirc (\eta \circ T(f))) \\
&= G(\mathrm{id}_{TY}) \circledcirc G(\eta \circ T(f)) \\
&= \mu \circ T(\lambda_Y) \circ \eta \circ FT(f) \\
&= \mu \circ \eta \circ \lambda_Y \circ FT(f) \\
&= \lambda_Y \circ FT(f) \\
&= \lambda_Y \circ FT(f).
\end{aligned}$$

Similarly one verifies that λ interacts appropriately with η and μ. ❑

The commuting diagram (5.7) in this result yields an identity as isomorphism in diagram (2.31) in Theorem 2.5.9, needed to lift an adjunction to categories of algebras. Thus we obtain the following result as immediate consequence. At this stage it may look like a formality, but it turns out to be very useful in the next section.

Corollary 5.2.6 *Let F be an endofunctor and T a monad on a category $\mathbb{C}$, together with a distributive $\mathcal{K}\ell$-law $FT \Rightarrow TF$ of F over T. The adjunction $J \dashv U$ between the Kleisli category $\mathcal{K}\ell(T)$ and $\mathbb{C}$ lifts to an adjunction* $\mathbf{Alg}(J) \dashv \mathbf{Alg}(U)$ *by Theorem 2.5.9:*

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$\begin{tikzcd}
\mathbf{Alg}(F) \arrow[rr, bend left=10, "\mathbf{Alg}(J)"] \arrow[rr, phantom, "\perp"] \arrow[d] & & \mathbf{Alg}(\mathcal{K}\ell(F)) \arrow[ll, bend left=10, "\mathbf{Alg}(U)"] \arrow[d] \\
\mathbb{C} \arrow[rr, bend left=10, "J"] \arrow[rr, phantom, "\perp"] \arrow[loop left, "F"] & & \mathcal{K}\ell(T) \arrow[ll, bend left=10, "U"] \arrow[loop right, "\mathcal{K}\ell(F)"]
\end{tikzcd}$$

□

Explicitly, the functor $\mathbf{Alg}(J)\colon \mathbf{Alg}(F) \to \mathbf{Alg}(\mathcal{K}\ell(F))$ at the top of this diagram maps an algebra $a\colon F(X) \to X$ to the composite:

$$\mathcal{K}\ell(F)(X) = \Big(F(X) \xrightarrow{\ a\ } X \xrightarrow{\ \eta\ } T(X)\Big).$$

It is an algebra $\mathcal{K}\ell(F)(X) \to X$ in the Kleisli category $\mathcal{K}\ell(T)$. Conversely, the functor $\mathbf{Alg}(U)\colon \mathbf{Alg}(\mathcal{K}\ell(F)) \to \mathbf{Alg}(F)$ sends an algebra $b\colon F(Y) \to T(Y)$ to the F-algebra:

$$FT(Y) \xrightarrow{\ \lambda\ } TF(Y) \xrightarrow{\ T(b)\ } T^2(Y) \xrightarrow{\ \mu\ } T(Y).$$

We now concentrate on distributive $\mathcal{K}\ell$-laws where the monad involved is powerset $\mathcal{P}$. The next result from [244] derives a distributive law with respect to the powerset monad from preservation of weak pullbacks. It is described as ∇, following [457], because of its role in Moss' coalgebraic modal logic, with its ∇ operator; see Section 6.5.2. This ∇ should not be confused with the codiagonal notation $\nabla = [\mathrm{id}, \mathrm{id}]\colon X + X \to X$.

Lemma 5.2.7 *For each weak-pullback-preserving functor $F\colon$ **Sets** $\to$ **Sets** there is a distributive $\mathcal{K}\ell$-law $\nabla\colon F\mathcal{P} \Rightarrow \mathcal{P}F$ obtained by applying relation lifting to the reverse inhabitation relation $\ni \subseteq \mathcal{P}(X) \times X$.*

This ∇ may be used to describe relation lifting: if, for an arbitrary relation $R \subseteq X \times Y$ we write $\widehat{R}\colon X \to \mathcal{P}(Y)$ for the corresponding function, as in (2.16), then the following triangle commutes:

$$\begin{tikzcd}
 & F\mathcal{P}(Y) \arrow[d, "\nabla"] \\
F(X) \arrow[ur, "F(\widehat{R})"] \arrow[r, "\widehat{\mathrm{Rel}(F)(R)}"'] & \mathcal{P}F(Y)
\end{tikzcd} \tag{5.8}$$

At this stage we use relation lifting as described in Definition 4.4.1, for arbitrary endofunctors on **Sets**, using the standard logical factorisation system of injections and surjections.

Proof By applying relation lifting $\mathrm{Rel}(F)$ to $\ni_X \subseteq \mathcal{P}(X) \times X$ we obtain a relation $\mathrm{Rel}(F)(\ni_X) \subseteq F(\mathcal{P}(X)) \times F(X)$. It can be reformulated, via (2.16),

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

as

$$\begin{tikzcd}
F(\mathcal{P}(X)) \arrow[r, "\nabla_X"] & \mathcal{P}(F(X)) \\
u \arrow[r, mapsto] & \{v \in F(X) \mid (u, v) \in \mathrm{Rel}(F)(\ni_X)\}.
\end{tikzcd} \tag{5.9}$$

In order to show that the ∇s behave appropriately we need the basic properties of relation lifting from Proposition 4.4.3 – which all apply because epis are split in **Sets** – together with basic connections between inhabitation and the unit $\eta = \{-\}$ and multiplication $\mu = \bigcup$ of the powerset monad (see Example 5.1.3.1).

a. $(\mathcal{P}(f) \times \mathrm{id})^{-1}(\ni_Y) = \coprod_{\mathrm{id}\times f}(\ni_X)$
b. $(\eta_X \times \mathrm{id})^{-1}(\ni_X) = \mathrm{Eq}(X)$
c. $(\mu_X \times \mathrm{id})^{-1}(\ni_X) = \ni_X \circ \ni_{\mathcal{P}(X)}$.

The proof of these three properties is left as an exercise for the reader.

For naturality of ∇ we use that relation lifting preserves inverse and direct images. For $f\colon X \to Y$ and $u \in F(\mathcal{P}(X))$,

$$\begin{aligned}
(\nabla_Y \circ F\mathcal{P}(f))(u) &= \{w \in F(Y) \mid (F\mathcal{P}(f)(u), w) \in \mathrm{Rel}(F)(\ni_Y)\} \\
&= \{w \mid (u, w) \in (F(\mathcal{P}(f)) \times \mathrm{id})^{-1}(\mathrm{Rel}(F)(\ni_Y))\} \\
&= \{w \mid (u, w) \in \mathrm{Rel}(F)((\mathcal{P}(f) \times \mathrm{id})^{-1}(\ni_Y))\} \\
&\overset{(a)}{=} \{w \mid (u, w) \in \mathrm{Rel}(F)(\textstyle\coprod_{\mathrm{id}\times f}(\ni_X))\} \\
&= \{w \mid (u, w) \in \textstyle\coprod_{F(\mathrm{id})\times F(f)}(\mathrm{Rel}(F)(\ni_X))\} \\
&= \{w \mid \exists (u', v).\, u' = u \wedge F(f)(v) = w \wedge (u', v) \in \mathrm{Rel}(F)(\ni_X)\} \\
&= \{F(f)(v) \mid (u, v) \in \mathrm{Rel}(F)(\ni_X)\} \\
&= (\mathcal{P}F(f) \circ \nabla_X)(u).
\end{aligned}$$

This ∇ also interacts appropriately with the unit $\eta = \{-\}$ and multiplication $\mu = \bigcup$ of the powerset monad, because relation lifting preserves equality and composition, and because of the above points (b) and (c):

$$\begin{aligned}
&(\nabla_X \circ F(\eta_X))(u) \\
&= \{v \mid (F(\eta)(u), v) \in \mathrm{Rel}(F)(\ni_X)\} \\
&= \{v \mid (u, v) \in (F(\eta) \times F(\mathrm{id}))^{-1}(\mathrm{Rel}(F)(\ni_X))\} \\
&= \{v \mid (u, v) \in \mathrm{Rel}(F)((\eta \times \mathrm{id})^{-1}(\ni_X))\} \\
&\overset{(b)}{=} \{v \mid (u, v) \in \mathrm{Rel}(F)(\mathrm{Eq}(X))\} \\
&= \{v \mid (u, v) \in \mathrm{Eq}(F(X))\} \\
&= \{u\} \\
&= \eta_{F(X)}(u)
\end{aligned}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$\begin{aligned}
&(\mu_{F(X)} \circ \mathcal{P}(\nabla_X) \circ \nabla_{\mathcal{P}(X)})(u) \\
&\quad = \bigcup\{\nabla_X(v) \mid (u, v) \in \mathrm{Rel}(F)(\ni_{\mathcal{P}(X)})\} \\
&\quad = \{w \mid \exists v.\, (v, w) \in \mathrm{Rel}(F)(\ni_X) \wedge (u, v) \in \mathrm{Rel}(F)(\ni_{\mathcal{P}(X)})\} \\
&\quad = \{w \mid (u, w) \in \mathrm{Rel}(F)(\ni_X) \circ \mathrm{Rel}(F)(\ni_{\mathcal{P}(X)})\} \\
&\quad = \{w \mid (u, w) \in \mathrm{Rel}(F)(\ni_X \circ \ni_{\mathcal{P}(X)})\} \\
&\quad \overset{(c)}{=} \{w \mid (u, w) \in \mathrm{Rel}(F)((\mu_X \times \mathrm{id})^{-1}(\ni_X))\} \\
&\quad = \{w \mid (u, w) \in (F(\mu_X) \times F(\mathrm{id}))^{-1}(\mathrm{Rel}(F)(\ni_X))\} \\
&\quad = \{w \mid (F(\mu_X)(u), w) \in \mathrm{Rel}(F)(\ni_X)\} \\
&\quad = (\nabla_X \circ F(\mu_X))(u).
\end{aligned}$$

Commutation of the triangle (5.8) follows from an easy calculation. For $u \in F(X)$ and $v \in F(Y)$ we have:

$$\begin{aligned}
v \in (\nabla \circ F(\widehat{R}))(u) &\Longleftrightarrow (F(\widehat{R})(u), v) \in \mathrm{Rel}(F)(\ni) \\
&\Longleftrightarrow (u, v) \in (F(\widehat{R}) \times \mathrm{id})^{-1}(\mathrm{Rel}(F)(\ni)) \\
&\Longleftrightarrow (u, v) \in \mathrm{Rel}(F)((\widehat{R} \times \mathrm{id})^{-1}(\ni)) \\
&\Longleftrightarrow (u, v) \in \mathrm{Rel}(F)(R) \\
&\Longleftrightarrow v \in \widehat{\mathrm{Rel}(F)(R)}(u). \qquad \square
\end{aligned}$$

For a weak-pullback-preserving functor on **Sets** there are thus *two* liftings to two categories of relations: once with relations as morphisms and once as objects. Since they are easily confused, we describe them explicitly.

Corollary 5.2.8 *A weak-pullback-preserving functor F :* **Sets** $\to$ **Sets** *has two liftings:*

1. *a 'computational' one, to* **SetsRel** $\to$ **SetsRel** *via Proposition 5.2.5, using the distributive $\mathcal{K}\ell$-law $\nabla : F\mathcal{P} \Rightarrow \mathcal{P}F$ from the previous lemma and the fact that the category* **SetsRel** *of sets and relations* as morphisms *is the Kleisli category $\mathcal{K}\ell(\mathcal{P})$ of the powerset monad*
2. *a 'logical' one, to a relator* **Rel** $\to$ **Rel** *via Theorem 4.4.6, where* **Rel** $=$ Rel(**Sets**) *is the category with relations* as objects *and morphisms preserving relation inclusion.*

The action $(X \xrightarrow{R} \mathcal{P}(Y)) \longmapsto (F(X) \xrightarrow{\nabla \circ F(R)} \mathcal{P}(F(Y)))$ of the functor in point 1 on morphisms is the same as the action $(R \rightarrowtail X \times Y) \longmapsto (\mathrm{Rel}(F)(R) \rightarrowtail F(X) \times F(Y))$ of the functor in point 2 on objects, via the triangle (5.8). $\square$

We close this section by extending the notion of strength from functors to monads, together with the associated property of commutativity for monads. Commutativity allows us to define some more distributive laws and to define a tensor on Kleisli categories.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

In Exercise 2.5.4 we have already briefly seen strength for functors on **Sets**. In general, for functors $F\colon \mathbb{C} \to \mathbb{C}$ on a category $\mathbb{C}$ with finite products, a functor is called **strong** if it comes with a **strength** natural transformation st with components

$$F(X) \times Y \xrightarrow{\;\mathrm{st}_{X,Y}\;} F(X \times Y)$$

that commute appropriately with trivial projections and associativity isomorphisms. Here are the diagrams, starting with naturality:

$$\begin{tikzcd}
F(X) \times Y \arrow[r, "\mathrm{st}"] \arrow[d, "F(f) \times g"'] & F(X \times Y) \arrow[d, "F(f \times g)"] \\
F(Z) \times W \arrow[r, "\mathrm{st}"] & F(Z \times W)
\end{tikzcd}
\qquad
\begin{tikzcd}
F(X) \times 1 \arrow[r, "\mathrm{st}"] \arrow[dr, "\cong", "\pi_1"'] & F(X \times 1) \arrow[d, "\cong"', "F(\pi_1)"] \\
& F(X)
\end{tikzcd}$$

$$\begin{tikzcd}
(F(X) \times Y) \times Z \arrow[r, "\mathrm{st} \times \mathrm{id}"] \arrow[d, "\cong"'] & F(X \times Y) \times Z \arrow[r, "\mathrm{st}"] & F((X \times Y) \times Z) \arrow[d, "\cong"] \\
F(X) \times (Y \times Z) \arrow[rr, "\mathrm{st}"'] & & F(X \times (Y \times Z))
\end{tikzcd} \tag{5.10}$$

As an aside: cartesian products are not really needed in order to define strength; monoidal (tensor) products suffice, since we do not really need projections or diagonals.

A monad T is called **strong** if it has a strength $\mathrm{st}\colon T(X) \times Y \to T(X \times Y)$, as above, that commutes with the unit and multiplication of the monad:

$$\begin{tikzcd}
X \times Y \arrow[r, equal] \arrow[d, "\eta \times \mathrm{id}"'] & X \times Y \arrow[d, "\eta"] \\
T(X) \times Y \arrow[r, "\mathrm{st}"'] & T(X \times Y)
\end{tikzcd}
\qquad
\begin{tikzcd}
T^2(X) \times Y \arrow[r, "\mathrm{st}"] \arrow[d, "\mu \times \mathrm{id}"'] & T(T(X) \times Y) \arrow[r, "T(\mathrm{st})"] & T^2(X \times Y) \arrow[d, "\mu"] \\
T(X) \times Y \arrow[rr, "\mathrm{st}"'] & & T(X \times Y)
\end{tikzcd} \tag{5.11}$$

Given such a strength $\mathrm{st}\colon F(X) \times Y \to F(X \times Y)$ one can also form a ‘swapped’ strength in the second argument, written as

$$\mathrm{st}' \overset{\text{def}}{=} \Big(X \times F(Y) \xrightarrow[\cong]{\;\mathrm{swap}\;} F(Y) \times X \overset{\mathrm{st}}{\to} F(Y \times X) \xrightarrow[\cong]{\;F(\mathrm{swap})\;} F(X \times Y) \Big) \tag{5.12}$$

where, of course, $\mathrm{swap} = \langle \pi_2, \pi_1 \rangle$.

Given both st and st′ maps for a monad T, there are two ways of going $T(X) \times T(Y) \to T(X \times Y)$, namely via first st and then st′ or the other way around, in

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$\begin{tikzcd}
& T(X \times T(Y)) \arrow[r, "T(\mathrm{st}')"] & T^2(X \times Y) \arrow[dr, "\mu"] & \\
T(X) \times T(Y) \arrow[ur, "\mathrm{st}"] \arrow[dr, "\mathrm{st}'"'] & & & T(X \times Y) \\
& T(T(X) \times Y) \arrow[r, "T(\mathrm{st})"'] & T^2(X \times Y) \arrow[ur, "\mu"'] &
\end{tikzcd} \tag{5.13}$$

The monad T is called **commutative** if this diagram commutes. We wrap up the foregoing in a definition.

Definition 5.2.9 Let $\mathbb{C}$ be a category with finite products.

1. A functor $F\colon \mathbb{C} \to \mathbb{C}$ is called **strong** if there is strength natural transformation st satisfying (5.10).
2. A monad $T\colon \mathbb{C} \to \mathbb{C}$ is **strong** if it comes equipped with a strength map satisfying both (5.10) and (5.11).
3. A strong monad is called **commutative** if its strength makes diagram (5.13) commute. In that case we sometimes write $\mathrm{dst}\colon T(X) \times T(Y) \to T(X \times Y)$ for the resulting 'double strength' map.
4. If we have two strong monads T and S, then a map of monads $\sigma\colon T \Rightarrow S$ is called a **strong map** if it commutes with strengths, as in

$$\begin{tikzcd}
T(X) \times Y \arrow[r, "\sigma_X \times \mathrm{id}"] \arrow[d, "\mathrm{st}^T"'] & S(X) \times Y \arrow[d, "\mathrm{st}^S"] \\
T(X \times Y) \arrow[r, "\sigma_{X \times Y}"'] & S(X \times Y)
\end{tikzcd} \tag{5.14}$$

This σ then also commutes with st′ and dst.

We shall write $\mathbf{CMnd}(\mathbb{C}) \hookrightarrow \mathbf{StMnd}(\mathbb{C}) \hookrightarrow \mathbf{Mnd}(\mathbb{C})$ for the subcategories of commutative and strong monads, with strong monad maps between them.

Commutativity of monads can also be formulated by saying that the monad is monoidal (see e.g. [299]), but this requires the machinery of monoidal categories which is out of scope. Alternative formulations using exponents occur in Exercise 5.2.16.

In Exercise 2.5.4 we have seen that every endofunctor on **Sets** is strong, via a uniform strength. The same holds for monads – and also for comonads; see Exercise 5.2.14.

Lemma 5.2.10 $\mathbf{StMnd}(\mathbf{Sets}) = \mathbf{Mnd}(\mathbf{Sets})$. *That is, each monad on* **Sets**, *and each monad map between them, is automatically strong.*

Proof Recall from Exercise 2.5.4 the definition of $\mathrm{st}\colon T(X) \times Y \to T(X \times Y)$:

$$\mathrm{st}(u, y) = T(\lambda x \in X.\,(x, y))(u).$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

It is not hard to see that it commutes with T's unit η and multiplication μ, using their naturality. For instance,

$$\begin{aligned}(\mathrm{st}\circ(\eta\times\mathrm{id}))(z,y) &= T(\lambda x\in X.\,(x,y))(\eta(z)) \\ &= \big(T(\lambda x\in X.\,(x,y))\circ\eta\big)(z) \\ &= \big(\eta\circ(\lambda x\in X.\,(x,y))\big)(z) \\ &= \eta(z,y).\end{aligned}$$

And for a map of monads $\sigma\colon T \Rightarrow S$ the relevant diagram (5.14) automatically commutes, by naturality:

$$\begin{aligned}(\mathrm{st}^T\circ(\sigma\times\mathrm{id}))(u,y) = \mathrm{st}^T(\sigma(u),y) &= T(\lambda x\in X.\,(x,y))(\sigma(u)) \\ &= \sigma(S(\lambda x\in X.\,(x,y))(u)) \\ &= (\sigma\circ\mathrm{st}^S)(u,y).\end{aligned} \qquad \square$$

The next result shows that commutativity of monads is a reasonable notion.

Lemma 5.2.11 *1. A monoid M is commutative if and only if the associated monad $M\times(-)\colon$ **Sets** $\to$ **Sets** is commutative.*

*2. A semiring S is commutative if and only if the multiset monad $\mathcal{M}_S\colon$ **Sets** $\to$ **Sets** is commutative.*

Proof By the previous lemma both monads are strong. For the monad $M\times(-)$ the strength map $\mathrm{st}\colon (M\times X)\times Y \to M\times(X\times Y)$ is associativity: $\mathrm{st}((m,x),y) = (m,(x,y))$. The swapped strength map $\mathrm{st}'\colon X\times(M\times Y) \to M\times(X\times Y)$ is similar: $\mathrm{st}'(x,(m,y)) = (m,(x,y))$. The upper and lower path in (5.13) are

$$\begin{tikzcd}(M\times X)\times(M\times Y) \arrow[rr, bend left, "{((m,x),(k,y))\mapsto(m\cdot k,(x,y))}"] \arrow[rr, bend right, "{((m,x),(k,y))\mapsto(k\cdot m,(x,y))}"'] & & M\times(X\times Y)\end{tikzcd}$$

Hence they are equal if and only if M is commutative.

The strength $\mathrm{st}\colon \mathcal{M}_S(X)\times Y \to \mathcal{M}_S(X\times Y)$ and the swapped strength st' for the multiset monad are given by

$$\mathrm{st}(\textstyle\sum_i s_i|x_i\rangle, y) = \sum_i s_i|x_i,y\rangle \qquad \mathrm{st}'(x,\sum_j t_j|y_j\rangle) = \sum_j t_j|x,y_j\rangle.$$

The resulting two paths in (5.13) are

$$\begin{tikzcd}\mathcal{M}_S(X)\times\mathcal{M}_S(Y) \arrow[rr, bend left, "{(\sum_i s_i|x_i\rangle,\sum_j t_j|y_j\rangle)\mapsto\sum_{i,j} s_i\cdot t_j|x_i,y_j\rangle}"] \arrow[rr, bend right, "{(\sum_i s_i|x_i\rangle,\sum_j t_j|y_j\rangle)\mapsto\sum_{i,j} t_j\cdot s_i|x_i,y_j\rangle}"'] & & \mathcal{M}_S(X\times Y)\end{tikzcd}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Thus, commutativity of the (multiplication of the) semiring S implies commutativity of the monad $\mathcal{M}_S$. For the converse, it suffices to choose $X = Y = 1$ and use that $\mathcal{M}_S(1) \cong S$. ❑

Commutativity of monads is not only a reasonable but also a useful notion because it gives us distributive laws, and thus liftings to Kleisli categories. The following result comes from [206] and defines a distributive law of a simple polynomial functor – with identity, constants, products × and coproducts $\coprod$ only (see Definition 2.2.1) – over an arbitrary commutative monad.

Lemma 5.2.12 *Let $T\colon$ **Sets** $\to$ **Sets** be a commutative monad, and $F\colon$ **Sets** $\to$ **Sets** a simple polynomial functor. Then there is a distributive $\mathcal{K}\ell$-law $\lambda\colon FT \Rightarrow TF$.*

Proof The construction of the distributive law λ proceeds by induction on the structure of the functor F.

- If F is the identity functor, then λ is the identity natural transformation $T \Rightarrow T$.
- If F is a constant functor, say $X \mapsto A$, then λ is the unit map $\eta_A\colon A \to T(A)$.
- If $F = F_1 \times F_2$, we use induction and assume distributive laws $\lambda_i\colon F_iT \Rightarrow TF_i$ for $i \in \{1, 2\}$. Then we can form a new distributive law, using that T is commutative:

$$F_1T(X) \times F_2T(X) \xrightarrow{\lambda_1 \times \lambda_2} TF_1(X) \times TF_2(X) \xrightarrow{\text{dst}} T(F_1(X) \times F_2(X)).$$

- If F is a coproduct $\coprod_{i\in I} F_i$ then we may assume laws $\lambda_i\colon F_iT \Rightarrow TF_i$ for $i \in I$, and define

$$\textstyle\coprod_{i\in I} F_iT(X) \xrightarrow{[T(\kappa_i) \circ \lambda_i]_{i\in I}} T(\coprod_{i\in I} F_i(X)).$$

It is straightforward to check that these λs are natural and compatible with the monad structure. ❑

We conclude by remarking that the Kleisli category of a commutative monad carries a tensor ⊗. Formally, this is expressed in terms of a symmetric monoidal structure (see [344]). Here we describe it more concretely.

Proposition 5.2.13 *Let $\mathbb{C}$ be a category with finite products and $T\colon \mathbb{C} \to \mathbb{C}$ a commutative monoid. The cartesian product then becomes a functor $\times\colon \mathcal{K}\ell(T) \times \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$.*

Proof For $f\colon X \to Y$ and $g\colon Z \to W$ in $\mathcal{K}\ell(T)$ we have to define a map $X \times Z \to Y \times W$. Using double strength we take

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$X \times Z \xrightarrow{f \times g} T(Y) \times T(W) \xrightarrow{\text{dst}} T(Y \times W).$$

Via the properties of Exercise 5.2.15 it easy to see that identities and composition are preserved. □

Exercises

5.2.1 Show that a map of monads $\sigma \colon T \Rightarrow S$ induces a functor $\mathcal{K}\ell(T) \to \mathcal{K}\ell(S)$.

5.2.2 Let $T_1, T_2 \colon \mathbb{C} \to \mathbb{C}$ be two monads, with unit and multiplication η_i, μ_i and Kleisli extension $(-)^{\$_i}$. Prove that a natural transformation $\sigma \colon T_1 \Rightarrow T_2$ is a map of monads (i.e. commutes with the η_i, μ_i) if and only if σ commutes with the η_i (that is, $\sigma \circ \eta_1 = \eta_2$) and σ commutes with Kleisli extension: for each map $f \colon X \to T_1(Y)$ the following diagram commutes:

$$\begin{tikzcd}
T_1(Y) \arrow[r, "\sigma_Y"] & T_2(Y) \\
T_1(X) \arrow[u, "f^{\$_1}"] \arrow[r, "\sigma_X"'] & T_2(X) \arrow[u, "(\sigma_Y \circ f)^{\$_2}"']
\end{tikzcd}$$

5.2.3 Prove the three properties (a)–(c) in the proof of Lemma 5.2.7.

5.2.4 Prove in detail that if a category $\mathbb{C}$ has coequalisers, then so has the Kleisli category $\mathcal{K}\ell(T)$ for a monad T on $\mathbb{C}$ – as claimed in Proposition 5.2.2.

5.2.5 Let T be a monad on a category $\mathbb{C}$ with an initial object $0 \in \mathbb{C}$. Prove that the following two statements are equivalent.

1. The object $T(0) \in \mathbb{C}$ is final.
2. The object $0 \in \mathcal{K}\ell(T)$ is a zero object; i.e. 0 is both initial and final in $\mathcal{K}\ell(T)$.

5.2.6 Prove that there is a bijective correspondence

$$\frac{X \longrightarrow Y \qquad \text{in } \mathcal{K}\ell(\mathcal{P}) = \textbf{SetsRel}}{\mathcal{P}(Y) \longrightarrow \mathcal{P}(X) \qquad \text{in } \textbf{CL}_\wedge}$$

where $\textbf{CL}_\wedge$ is the category of complete lattices and arbitrary meet preserving functions. (This is the heart of the correspondence between (non-deterministic) state transformers $X \to Y$ and predicate transformers $\mathcal{P}(Y) \to \mathcal{P}(X)$ that forms the basis for the weakest precondition condition calculus in program verification; see [122, 341, 254].)

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

5.2.7 Recall that non-deterministic automata can be turned into deterministic ones by changing the state space. If we consider the state transition map only, this can be described as transforming a coalgebra $X \to \mathcal{P}(X)^A$ into a coalgebra $\mathcal{P}(X) \to \mathcal{P}(X)^A$.

1. Describe this transformation concretely.
2. Describe this transformation via (pointwise) Kleisli extension.
3. Show that it leads to a functor $\mathbf{CoAlg}(\mathcal{P}(-)^A) \to \mathbf{CoAlg}((-)^A)$.

5.2.8 In Definition 5.2.4 we have seen a distributive $\mathcal{K}\ell$-law $FT \Rightarrow TF$ of a functor F over a monad T. There is also a notion of distributive law when F is a monad too (going back to [61]). In that case the natural transformation should commute with all units and multiplications.

1. Write down the relevant diagrams.
2. Let $ST \Rightarrow TS$ be such a distributive law, where S and T are monads, and prove that the composite TS is then also a monad.
3. Prove that the units yield maps of monads $S \Rightarrow TS$ and $T \Rightarrow TS$.
4. Check that the lifted functor $\mathcal{K}\ell(S) \colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ is again a monad.
5. Show that the Kleisli category $\mathcal{K}\ell(\mathcal{K}\ell(S))$ of this lifted monad is the same as the Kleisli category $\mathcal{K}\ell(TS)$ of the composite monad.

We consider only the last two points. As unit $\overline{\eta}_X \colon X \to \mathcal{K}\ell(S)(X)$ in $\mathcal{K}\ell(T)$ we take $\overline{\eta}_X = \eta^T_{SX} \circ \eta^S_X \colon X \to TS(X)$. As multiplication $\overline{\mu}_X \colon \mathcal{K}\ell(S)^2(X) \to \mathcal{K}\ell(S)(X)$ we take $\overline{\mu}_X = \eta^T_{SX} \circ \mu^S_X \colon S^2(X) \to TS(X)$. Hence they are obtained via the inclusion functor $F \colon \mathbb{C} \to \mathcal{K}\ell(T)$, so all relevant monad equations hold because S is a monad.

5.2.9 Let $T \colon \mathbb{C} \to \mathbb{C}$ be an arbitrary monad on a category $\mathbb{C}$ with coproducts. Fix an object $A \in \mathbb{C}$, recall from Exercise 5.1.1 that $A + (-)$ is a monad on $\mathbb{C}$ and prove that the maps

$$A + T(X) \xrightarrow{\lambda_X = [T(\kappa_1) \circ \eta_A, T(\kappa_2)]} T(A + X)$$

form a $\mathcal{K}\ell$-law of monads (as defined in the previous exercise). Conclude that $T(A + (-))$ is also a monad on $\mathbb{C}$.

In the special case where A is the singleton (final) set 1, and T is the distribution on **Sets**, check that the monad $\mathcal{D}(1 + (-))$ is the subdistribution $\mathcal{D}_{\leq 1}$.

5.2.10 Consider the mapping $T \mapsto \mathcal{E}(T) \overset{\text{def}}{=} T(A + (-))$ from the previous exercise.

1. Prove that it gives rise to a functor $\mathcal{E} \colon \mathbf{Mnd}(\mathbb{C}) \to \mathbf{Mnd}(\mathbb{C})$.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

2. Show that $\mathcal{E}$ is even a monad, on the category $\mathbf{Mnd}(\mathbb{C})$ of monads on $\mathbb{C}$.

(This mapping $\mathcal{E}$ is often called the exception monad transformer. It is investigated more systematically in [335] together with associated operations for handling exceptions in programs. The proof that $\mathcal{E}$ is a monad requires a certain level of categorical fearlessness but is in essence not difficult.)

5.2.11 Let T be a strong monad. Check that strength gives for each object Y a $\mathcal{K}\ell$-law of the functor $(-) \times Y$ over T.

5.2.12 ([233, proposition 1]) Let F be an endofunctor and $T = (T, \eta, \mu)$ be a monad, both on the same category. Show that the maps $\lambda_X = \eta_{FT(X)} \circ F(\mu_X)\colon FTT(X) \to FT(X) \to TFT(X)$ form a distributive $\mathcal{K}\ell$-law of the functor FT over the monad T.

5.2.13 Show that the powerset $\mathcal{P}$ and distribution $\mathcal{D}$ monad are commutative, with double strengths:

$$\begin{tikzcd}[row sep=small] \mathcal{P}(X) \times \mathcal{P}(Y) \arrow[r, "\mathrm{dst}"] & \mathcal{P}(X \times Y) & \mathcal{D}(X) \times \mathcal{D}(Y) \arrow[r, "\mathrm{dst}"] & \mathcal{D}(X \times Y) \\ (U, V) \arrow[r, mapsto] & U \times V & (\varphi, \psi) \arrow[r, mapsto] & \lambda(x, y).\, \varphi(x) \cdot \psi(x). \end{tikzcd}$$

Check that the list monad $(-)^\star$ is *not* commutative.

5.2.14 Prove the following analogue of Lemma 5.2.10 for comonads: for each comonad $S : \mathbf{Sets} \to \mathbf{Sets}$, the associated strength map st makes the following diagrams commute:

$$\begin{tikzcd} S(X) \times Y \arrow[r, "\mathrm{st}"] \arrow[d, "\varepsilon \times \mathrm{id}"'] & S(X \times Y) \arrow[d, "\varepsilon"] \\ X \times Y \arrow[r, equal] & X \times Y \end{tikzcd}$$

$$\begin{tikzcd} S(X) \times Y \arrow[rr, "\mathrm{st}"] \arrow[d, "\delta \times \mathrm{id}"'] & & S(X \times Y) \arrow[d, "\delta"] \\ S^2(X) \times Y \arrow[r, "\mathrm{st}"'] & S(S(X) \times Y) \arrow[r, "S(\mathrm{st})"'] & S^2(X \times Y) \end{tikzcd}$$

5.2.15 Show that the double strength map dst of a commutative monad T is a natural transformation and makes the following diagrams commute:

$$\begin{tikzcd} X \times Y \arrow[r, equal] \arrow[d, "\eta \times \eta"'] & X \times Y \arrow[d, "\eta"] \\ T(X) \times T(Y) \arrow[r, "\mathrm{dst}"'] & T(X \times Y) \end{tikzcd}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$\begin{tikzcd}
T^2(X)\times T^2(Y) \arrow[r, "\mathrm{dst}"] \arrow[d, "\mu\times\mu"'] & T(T(X)\times T(Y)) \arrow[r, "T(\mathrm{dst})"] & T^2(X\times Y) \arrow[d, "\mu"] \\
T(X)\times T(Y) \arrow[rr, "\mathrm{dst}"'] & & T(X\times Y)
\end{tikzcd}$$

5.2.16 This exercise involves an alternative formulation of strength st, both for functors and for monads, using exponents. Let $\mathbb{C}$ be a cartesian closed category with an endofunctor $F\colon \mathbb{C}\to\mathbb{C}$.

1. Prove that F is strong if and only if there are maps $r_{X,Y}\colon F(X^Y)\to F(X)^Y$ which are natural in X and Y and make the following three diagrams commute:

$$\begin{tikzcd}
F(X_1^{Y_2}) \arrow[r, "r"] \arrow[d, "F(f^g)"'] & F(X_1)^{Y_2} \arrow[d, "F(f)^g"] \\
F(X_2^{Y_1}) \arrow[r, "r"'] & F(X_2)^{Y_1}
\end{tikzcd}$$

where $f^g = \Lambda(f\circ \mathrm{ev}\circ(\mathrm{id}\times g))$, and

$$\begin{tikzcd}
F(X) \arrow[r, equal] \arrow[d, "F(\Lambda(\pi_1))"', "\cong"] & F(X) \arrow[d, "\Lambda(\pi_1)", "\cong"'] \\
F(X^1) \arrow[r, "r"'] & F(X)^1
\end{tikzcd}
\qquad
\begin{tikzcd}
F((X^Y)^Z) \arrow[r, "r"] \arrow[d, "\cong"'] & F(X^Y)^Z \arrow[r, "r^Z"] & (F(X)^Y)^Z \arrow[d, "\cong"] \\
F(X^{Z\times Y}) \arrow[rr, "r"'] & & F(X)^{Z\times Y}
\end{tikzcd}$$

where the isomorphisms on the right are $\Lambda(\mathrm{ev}\circ(\mathrm{ev}\times\mathrm{id})\circ\alpha)$, with associativity isomorphism $\alpha = \langle(\mathrm{id}\times\pi_1), \pi_2\circ\pi_2\rangle\colon A\times(B\times C)\xrightarrow{\cong}(A\times B)\times C$.

2. Prove also that a monad $T\colon\mathbb{C}\to\mathbb{C}$ is strong if and only if there are such $r\colon T(X^Y)\to T(X)^Y$ satisfying

$$\begin{tikzcd}
X^Y \arrow[r, equal] \arrow[d, "\eta"'] & X^Y \arrow[d, "\eta^Y"] \\
T(X^Y) \arrow[r, "r"'] & T(X)^Y
\end{tikzcd}
\qquad
\begin{tikzcd}
T^2(X^Y) \arrow[r, "T(r)"] \arrow[d, "\mu"'] & T(T(X)^Y) \arrow[r, "r"] & T^2(X)^Y \arrow[d, "\mu^Y"] \\
T(X^Y) \arrow[rr, "r"'] & & T(X)^Y
\end{tikzcd}$$

5.2.17 Show that there are adjunctions between (commutative) monoids and (commutative) monads:

$$\begin{tikzcd}
\mathbf{Mon} \arrow[d, bend right=40, "M\mapsto M\times(-)"'] \arrow[d, phantom, "\dashv"] \\
\mathbf{Mnd}(\mathbf{Sets}) \arrow[u, bend right=40, "T\mapsto T(1)"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{CMon} \arrow[d, bend right=40, "M\mapsto M\times(-)"'] \arrow[d, phantom, "\dashv"] \\
\mathbf{CMnd}(\mathbf{Sets}) \arrow[u, bend right=40, "T\mapsto T(1)"']
\end{tikzcd}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

The adjunction on the left comes from [466] where it is described in the more general setting of monoidal categories. See also [111] for more such adjunctions, also involving the multiset monad. *Hint*: Define a monoid structure on $T(1)$ via strength, as $T(1) \times T(1) \to T(1 \times T(1)) \xrightarrow{\cong} T^2(1) \to T(1)$.

5.3 Trace Semantics via Finality in Kleisli Categories

This section describes how finite traces of observables of computations can be described via finality, not in the familiar category **Sets** but in the Kleisli category $\mathcal{K}\ell(T)$ of a monad T on **Sets**. We shall be considering coalgebras of the form $X \to TF(X)$, where T is our monad capturing the kind of computation involved (non-deterministic, probabilistic, etc.) and F is a functor that describes the branching structure. The main examples that will be elaborated here involve non-deterministic and probabilistic automata and context-free grammars. The use of Kleisli categories for trace semantics goes back to [395, 244]. (The latter reference [244] deals not with *finite* trace semantics – as in this section – but with the infinite version. It involves only weak finality; i.e. it lacks uniqueness of the trace map.)

We begin with an illustration that uses the powerset monad $\mathcal{P}$ on **Sets**. Recall that its Kleisli category $\mathcal{K}\ell(\mathcal{P})$ is the category **SetsRel** of sets and relations between them, since a relation $R \subseteq X \times Y$ may equivalently be described as a characteristic function $R\colon X \to \mathcal{P}(Y)$. We shall often switch back-and-forth between these two notations. For convenience we recall that the identity map $X \to X$ in **SetsRel** is the equality relation $\mathrm{Eq}(X) \subseteq X \times X$. And (Kleisli) composition of $R\colon X \to Y$ and $S : Y \to Z$ is the usual relational composition: $S \circledcirc R = \{(x,z) \in X \times Z \mid \exists y \in Y.\, R(x,y) \wedge S(y,z)\}$. There is an obvious 'graph' functor **Sets** → **SetsRel** as already described in Example 1.4.4.4. It maps a set to itself, and a function $f\colon X \to Y$ to its graph relation $\mathrm{Graph}(f) = \{(x,y) \in X \times Y \mid f(x) = y\}$. This graph functor is the standard map $J\colon \mathbb{C} \to \mathcal{K}\ell(T)$ associated with a Kleisli category; see Proposition 5.2.2.

Let us now fix a set L of 'labels' and write F for the 'list' functor **Sets** → **Sets** given by $X \mapsto 1 + (L \times X)$. We shall consider transition systems of the form $c\colon X \to \mathcal{P}F(X) = \mathcal{P}(1 + (L \times X))$. As noticed in Exercise 2.2.3, such transition systems correspond to non-deterministic automata $X \to \mathcal{P}(X)^L \times 2$. Here we prefer the formulation with the powerset on the outside, because it allows us to describe such automata as maps $X \to F(X)$ in the Kleisli category $\mathcal{K}\ell(\mathcal{P}) =$ **SetsRel**.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

These transition systems $X \to \mathcal{P}(1 + (L \times X))$ describe besides usual transitions $x \xrightarrow{a} x'$, as shorthand for $(a, x') \in c(x)$, also terminating transitions $x \to *$ for $* \in c(x)$. A *trace* for such a coalgebra starting in a state $x \in X$ consists of a list of elements $\langle a_1, \ldots, a_n\rangle \in L^\star$ for which there is a list of states $x_0, \ldots, x_n \in X$ where:

- $x_0 = x$
- $x_i \xrightarrow{a_i} x_{i+1}$ for all $i < n$
- $x_n \to *$.

Thus, traces are sequences of labels appearing in terminating sequences of computations.

One can then define a function $\mathrm{trace}_c \colon X \to \mathcal{P}(L^\star)$ that maps a state to the set of traces starting in that state. We shall show how to define this function by finality. It turns out not to be a coincidence that the set $L^\star$ of lists is the initial algebra of the functor F.

First we need to move from sets to relations as morphisms. We shall lift the functor $F\colon \mathbf{Sets} \to \mathbf{Sets}$ to $\mathcal{K}\ell(F)\colon \mathbf{SetsRel} \to \mathbf{SetsRel}$. Essentially, this lifting to Kleisli categories happens as in Proposition 5.2.5 –using the distributive law from Lemma 5.2.7 –but at this moment we wish to proceed more concretely. The lifting $\mathcal{K}\ell(F)\colon \mathbf{SetsRel} \to \mathbf{SetsRel}$ maps an object (or set) X to $F(X)$, and a morphism $R\colon X \to Y$ in **SetsRel** to the morphism $\mathrm{Rel}(F)(R)\colon F(X) \to F(Y)$ obtained by relation lifting; see Corollary 5.2.8. Explicitly, according to Definition 3.1.1, the lifted relation $\mathrm{Rel}(F)(R) \subseteq F(X) \times F(Y)$ is described by

$$\mathrm{Rel}(F)(R) \;=\; \{(*, *)\} \cup \{\langle (a, x), (a, x')\rangle \mid a \in L \text{ and } R(x, x')\}.$$

We now have three important observations.

1. The first one is simple: a transition system, or $\mathcal{P}F$-coalgebra, $c\colon X \to \mathcal{P}F(X)$ as considered above, is an $\mathcal{K}\ell(F)$-coalgebra $X \to \mathcal{K}\ell(F)(X)$ in the category **SetsRel**, for the lifted functor $\mathcal{K}\ell(F)\colon \mathbf{SetsRel} \to \mathbf{SetsRel}$.
2. Further, the above trace map $\mathrm{trace}_c\colon X \to \mathcal{P}(L^\star)$, considered as a homomorphism $X \to L^\star$ in **SetsRel**, is a homomorphism of $\mathcal{K}\ell(F)$-coalgebras in **SetsRel**:

$$\begin{tikzcd}
F(X) \arrow[r, "\mathrm{Rel}(F)(\mathrm{trace}_c)"] & F(L^\star) \\
X \arrow[u, "c"] \arrow[r, "\mathrm{trace}_c"'] & L^\star \arrow[u, "\cong", "\mathrm{Graph}(\alpha^{-1})"']
\end{tikzcd} \tag{5.15}$$

where $\alpha = [\mathsf{nil}, \mathsf{cons}]\colon F(L^\star) \xrightarrow{\cong} L^\star$ is the initial algebra map, and the graph functor $\mathrm{Graph}(-)\colon \mathbf{Sets} \to \mathbf{SetsRel}$ is the canonical map $J\colon \mathbf{Sets} \to$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$\mathcal{K}\ell(\mathcal{P})$ described in Proposition 5.2.2. Commutation of the diagram (5.15) amounts to

$$\sigma \in \mathrm{trace}_c(x) \iff \exists u \in c(x).\, \langle u, \alpha^{-1}(\sigma)\rangle \in \mathrm{Rel}(F)(\mathrm{trace}_c).$$

This may be split in two cases, depending on whether the list $\sigma \in L^\star$ is empty or not:

$$\begin{aligned}
\mathsf{nil} \in \mathrm{trace}_c(x) &\iff \exists u \in c(x).\, \langle u, *\rangle \in \mathrm{Rel}(F)(\mathrm{trace}_c) \\
&\iff * \in c(x) \\
&\iff x \to * \\
\mathsf{cons}(a, \sigma) \in \mathrm{trace}_c(x) &\iff \exists u \in c(x).\, \langle u, (a, \sigma)\rangle \in \mathrm{Rel}(F)(\mathrm{trace}_c) \\
&\iff \exists x' \in X.\, (a, x') \in c(x) \wedge \langle x', \sigma\rangle \in \mathrm{trace}_c \\
&\iff \exists x' \in X.\, x \xrightarrow{a} x' \wedge \sigma \in \mathrm{trace}_c(x').
\end{aligned}$$

This shows that the trace map trace_c in (5.15) indeed makes the diagram commute.

3. Finally, this trace map trace_c is obtained by coinduction since the graph of the initial F-algebra $\mathrm{Graph}(\alpha^{-1})\colon L^\star \xrightarrow{\cong} F(L^\star)$ is a final $\mathcal{K}\ell(F)$-coalgebra in **SetsRel**.

We have already seen that there is a homomorphism trace_c for an arbitrary $\mathcal{K}\ell(F)$-coalgebra c in $\mathbf{SetsRel} = \mathcal{K}\ell(\mathcal{P})$. Here we check that it is unique: if a relation $S \subseteq X \times L^\star$ also satisfies $\mathrm{Graph}(\alpha^{-1}) \circledcirc S = \mathrm{Rel}(F)(S) \circledcirc c$, then $S = \mathrm{trace}_c$. Recall that we write $\circledcirc$ for composition in the Kleisli category $\mathcal{K}\ell(\mathcal{P})$, which we identify with relational composition. Uniqueness holds, since for $x \in X$ one obtains $\sigma \in S(x) \Leftrightarrow \sigma \in \mathrm{trace}_c(x)$ by induction on $\sigma \in L^\star$:

$$\begin{aligned}
\mathsf{nil} \in S(x) &\iff \exists \tau \in L^\star.\, * = \alpha^{-1}(\tau) \wedge \tau \in S(x) \\
&\iff \exists \tau \in L^\star.\, * \in \mathrm{Graph}(\alpha^{-1})(\tau) \wedge \tau \in S(x) \\
&\iff (x, *) \in \mathrm{Graph}(\alpha^{-1}) \circledcirc S \\
&\iff (x, *) \in \mathrm{Rel}(F)(S) \circledcirc c \\
&\iff \exists u \in c(x).\, * \in \mathrm{Rel}(F)(S)(u) \\
&\iff * \in c(x) \\
&\iff \mathsf{nil} \in \mathrm{trace}_c(x) \\
\mathsf{cons}(a, \sigma) \in S(x) &\iff \exists \tau \in L^\star.\, (a, \sigma) = \alpha^{-1}(\tau) \wedge \tau \in S(x) \\
&\iff \exists \tau \in L^\star.\, (a, \sigma) \in \mathrm{Graph}(\alpha^{-1})(\tau) \wedge \tau \in S(x) \\
&\iff (x, (a, \sigma)) \in \mathrm{Graph}(\alpha^{-1}) \circledcirc S \\
&\iff (x, (a, \sigma)) \in \mathrm{Rel}(F)(S) \circledcirc c \\
&\iff \exists u \in c(x).\, (a, \sigma) \in \mathrm{Rel}(F)(S)(u) \\
&\iff \exists x' \in X.\, \sigma \in S(x') \wedge (a, x') \in c(x) \\
&\overset{\text{(IH)}}{\iff} \exists x' \in X.\, \sigma \in \mathrm{trace}_c(x') \wedge x \xrightarrow{a} x' \\
&\iff \mathsf{cons}(a, \sigma) \in \mathrm{trace}_c(x).
\end{aligned}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

In the next result we shall describe this situation for the powerset monad in greater generality, following [203]. Later on we shall describe the situation in still more general form, for other monads, as in [206]. But we prefer to stick to the powerset monad first, because it involves a very common situation and because it has a snappy proof that avoids domain-theoretic machinery.

Theorem 5.3.1 *Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a weak-pullback-preserving functor with an initial algebra $\alpha\colon F(A) \xrightarrow{\cong} A$. Lemma 5.2.7 then yields a distributive $\mathcal{K}\ell$-law $F\mathcal{P} \Rightarrow \mathcal{P}F$ and thus a lifting of F to an endofunctor $\mathcal{K}\ell(F)$ on the Kleisli category $\mathcal{K}\ell(\mathcal{P}) = \mathbf{SetsRel}$.*

The map $\mathrm{Graph}(\alpha^{-1})\colon A \xrightarrow{\cong} \mathcal{K}\ell(F)(A)$ is then a final coalgebra in $\mathbf{SetsRel}$, for the lifted functor $\mathcal{K}\ell(F)\colon \mathbf{SetsRel} \to \mathbf{SetsRel}$.

Proof Our first observation is that reversal of relations, sending $R \subseteq X \times Y$ to $R^{\dagger} \subseteq Y \times X$, makes the (Kleisli) category $\mathbf{SetsRel} = \mathcal{K}\ell(\mathcal{P})$ self-dual: $\mathbf{SetsRel} \cong \mathbf{SetsRel}^{\mathrm{op}}$. When we combine this duality with the lifting of Corollary 5.2.6 we obtain the following situation:

$$\begin{tikzcd}
\mathbf{Alg}(F) \arrow[r, bend left, "\mathbf{Alg}(J)"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathbf{Alg}(\mathcal{K}\ell(F)) \arrow[l, bend left, "\mathbf{Alg}(U)"] \arrow[d] \arrow[r, phantom, "\cong"] & \mathbf{Alg}(\mathcal{K}\ell(F)^{\mathrm{op}}) = \mathbf{CoAlg}(\mathcal{K}\ell(F))^{\mathrm{op}} \arrow[d] \\
\mathbf{Sets} \arrow[r, bend left, "J"] \arrow[r, phantom, "\perp"] \arrow[loop below, "F"] & \mathbf{SetsRel} \arrow[l, bend left, "U"] \arrow[loop below, "\mathcal{K}\ell(F)"] \arrow[r, equal, "\sim"] & \mathbf{SetsRel}^{\mathrm{op}} \arrow[loop below, "\mathcal{K}\ell(F)^{\mathrm{op}}"]
\end{tikzcd}$$

First we consider the bottom row. By composing the adjunction $J \dashv U$ from Proposition 5.2.2 with the isomorphism $\mathbf{SetsRel} \cong \mathbf{SetsRel}^{\mathrm{op}}$ we obtain an adjunction between the categories $\mathbf{Sets}$ and $\mathbf{SetsRel}^{\mathrm{op}}$, say $J^{\dagger} \dashv U^{\dagger}$, where $J^{\dagger}\colon \mathbf{Sets} \to \mathbf{SetsRel}^{\mathrm{op}}$ is given by $J^{\dagger}(X) = X$ and $J^{\dagger}(f\colon X \to Y) = \mathrm{Graph}(f)^{\dagger} = \{(y, x) \mid f(x) = y\} \subseteq Y \times X$. In the reverse direction we have $U^{\dagger}(Y) = \mathcal{P}(Y)$ and $U^{\dagger}(R\colon X \to Y) = \lambda U \in \mathcal{P}(Y).\, \{x \in X \mid \exists y \in U.\, R(x, y)\}$.

The resulting lifted adjunction in the top row, between the categories of algebra $\mathbf{Alg}(F)$ and $\mathbf{Alg}(\mathcal{K}\ell(F)^{\mathrm{op}}) = \mathbf{CoAlg}(\mathcal{K}\ell(F))^{\mathrm{op}}$, say $\mathbf{Alg}(J^{\dagger}) \dashv \mathbf{Alg}(U^{\dagger})$, allows us to finish the argument, since left adjoints preserve initial objects. Hence $\mathbf{Alg}(J^{\dagger})$ sends the initial F-algebra $F(A) \xrightarrow{\cong} A$ to the initial object in $\mathbf{Alg}(\mathcal{K}\ell(F)^{\mathrm{op}})$, i.e. to the final object in $\mathbf{CoAlg}(\mathcal{K}\ell(F))$, i.e. to the final coalgebra of the lifted functor $\mathcal{K}\ell(F)\colon \mathbf{SetsRel} \to \mathbf{SetsRel}$. ❑

This theorem captures traces for transition systems as described in the beginning of this section. The main application of [203] is the parsed language associated with a context-free grammar; see Example 5.3.2.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Example 5.3.2 Recall from Section 2.2.5 that a context-free grammar (CFG) is described coalgebraically as a function $g\colon X \to \mathcal{P}((X + L)^{\star})$ where X is the state space of nonterminals, and L is the alphabet of terminals (or tokens). Such a CFG is thus a coalgebra in **SetsRel** of the lifting of the functor $F(X) = (X + L)^{\star}$ on **Sets**. Let us write its initial F-algebra as

$$(L^{\triangle} + L)^{\star} \xrightarrow[\cong]{\quad\alpha\quad} L^{\triangle}\,.$$

Notice that $F(X) = (X + L)^{\star} = \coprod_{n\in\mathbb{N}}(X + L)^{n} = \coprod_{\sigma\in(1+L)^{\star}} X^{\|\sigma\|}$ where $\|\sigma\| \in \mathbb{N}$ is the number of $* \in 1$ occurring in $\sigma \in (1{+}L)^{\star}$. Hence an F-algebra $F(X) \to X$ involves an n-ary operation $X^{n} \to X$ for each $\sigma \in (1 + L)^{\star}$ with $n = \|\sigma\|$. An example of such a σ is $\langle \mathsf{if}, *, \mathsf{then}, *, \mathsf{else}, *, \mathsf{fi}\rangle$. It may be understood as a ternary operation $X^{3} \to X$. The initial algebra $L^{\triangle}$ then consists of terms built with such operations. Hence it contains all structured or parsed words over L (according to the grammar g).

Theorem 5.3.1 gives for each CFG $g\colon X \to \mathcal{P}((X + L)^{\star})$ a unique homomorphism $\mathrm{trace}_{g}\colon X \to \mathcal{P}(L^{\triangle})$. It maps a nonterminal $x \in X$ to the collection of parsed words that can be produced from x. Coalgebraic trace semantics was first used to describe the language of a context-free grammar in [203]; see also [464]. The situation is elaborated further in Exercise 5.3.2.

At this stage we wish to generalise the result of Theorem 5.3.1 to monads other than powerset $\mathcal{P}$. The theorem itself says that an initial algebra $F(A) \xrightarrow{\cong} A$ yields a final coalgebra $A \xrightarrow{\cong} F(A) = \mathcal{K}\ell(F)(A)$ of the lifting $\mathcal{K}\ell(F)$ of the functor F to a Kleisli category. One aspect that we ignored is that the initial algebra $F(A) \xrightarrow{\cong} A$ also yields an initial algebra in the Kleisli category, since the canonical functor $J\colon \mathbf{Sets} \to \mathcal{K}\ell(\mathcal{P})$ is a left adjoint (see Proposition 5.2.2) and thus preserves colimits of ω-chains (Exercise 4.6.3). Thus A carries both an initial algebra and a final coalgebra structure in the Kleisli category. Such coincidence has been a topic of extensive study; see notably [433, 144, 145, 132]. Here we concentrate on the essentials and refer to these original sources for further information.

There is one more concept that we need. A category $\mathbb{C}$ is called **dcpo-enriched** if each hom-set $\mathbb{C}(X, Y)$ of map $X \to Y$ in $\mathbb{C}$ forms a directed complete partial order (dcpo) and the dcpo structure is preserved under composition: both pre- and post-composition $h \circ (-)$ and $(-) \circ g$ are maps in the category **Dcpo**. Again, the Kleisli category $\mathcal{K}\ell(\mathcal{P})$ is an example: for a collection of maps $f_i\colon X \to \mathcal{P}(Y)$ we can form the pointwise join $\bigvee_i f_i\colon X \to \mathcal{P}(Y)$, namely $(\bigvee_i f_i)(x) = \bigcup_i f_i(x)$. It is not hard to see that $h \circ (\bigvee_i f_i) = \bigvee_i (h \circ f_i)$ and similarly $(\bigvee_i f_i) \circ g = \bigvee_i (f_i \circ g)$.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

In such a dcpo-enriched category the hom-sets $\mathbb{C}(X, Y)$ carry a partial order $\leq$ for which directed joins exist. An arbitrary map $f\colon X \to Y$ is called an **embedding** if there is a 'projection' map $f^p\colon Y \to X$ in the other direction with

$$f^p \circ f = \mathrm{id}_X \quad \text{and} \quad f \circ f^p \leq \mathrm{id}_Y.$$

This map f^p, if it exists, is uniquely determined. As a consequence, $(g \circ f)^p = f^p \circ g^p$. The pair (f, f^p) is sometimes called an embedding-projection pair. More formally, we have an adjunction (Galois connection) with an isomorphism as unit (also known as coreflection).

The heart of the matter is the following limit-colimit coincidence result from [433].

Proposition 5.3.3 *Let $\mathbb{C}$ be a dcpo-enriched category. Assume an ω-chain*

$$X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} X_2 \xrightarrow{f_2} \cdots$$

with colimit $A \in \mathbb{C}$. If the maps f_i are embeddings, then the colimit A is also a limit in $\mathbb{C}$, namely the limit of the ω-chain of associated projections $f_i^p\colon X_{i+1} \to X_i$. This will be proven via the following intermediate results.

1. *The coprojection maps $\kappa_n\colon X_n \to A$ associated with the colimit A are embeddings, and their projections $\pi_n = \kappa_n^p\colon A \to X_n$ form a cone,* i.e. *satisfy $f_n^p \circ \pi_{n+1} = \pi_n$.*
2. *These (co)projections satisfy $\bigvee_{n\in\mathbb{N}} \kappa_n \circ \pi_n = \mathrm{id}_A$.*
3. *The projections $\pi_n\colon A \to X_n$ are jointly monic: if $h, h'\colon Y \to A$ satisfy $\pi_n \circ h = \pi_n \circ h'$ for all $n \in \mathbb{N}$, then $h = h'$.*
4. *For a cone $g_n\colon Y \to X_n$, where $f_n^p \circ g_{n+1} = g_n$, there is a unique mediating map $g\colon Y \to A$, given by $g = \bigvee_n \kappa_n \circ g_n$.*

Proof 1. For each $n \in \mathbb{N}$ we first show that the object X_n forms a cone: for $m \in \mathbb{N}$ there is a map $f_{mn}\colon X_m \to X_n$, namely:

$$f_{mn} = \begin{cases} f_{n-1} \circ \cdots \circ f_m \ :\ X_m \to X_{m+1} \to \cdots \to X_n & \text{if } m \leq n \\ f_n^p \circ \cdots \circ f_{m-1}^p \ :\ X_m \to X_{m-1} \to \cdots \to X_n & \text{if } m > n. \end{cases}$$

These maps f_{mn} commute with the maps $f_i\colon X_i \to X_{i+1}$ in the chain: $f_{(m+1)n} \circ f_m = f_{mn}$, and thus form a cone. Since A is the colimit, there is a unique map $\pi_n\colon A \to X_n$ with $\pi_n \circ \kappa_m = f_{mn}$. In particular, for $m = n$ we get $\pi_n \circ \kappa_n = f_{nn} = \mathrm{id}$. We postpone the proof that $\kappa_n \circ \pi_n \leq \mathrm{id}_A$ for a moment.

2. The maps $\kappa_n \circ \pi_n\colon A \to A$ form an ascending chain, since $\kappa_n \circ \pi_n = \kappa_{n+1} \circ f_n \circ f_n^p \circ \pi_{n+1} \leq \kappa_{n+1} \circ \pi_{n+1}$. Hence their join exists, which we abbreviate as $f = \bigvee_n \kappa_n \circ \pi_n\colon A \to A$. Then for each $i \in \mathbb{N}$,

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$\begin{aligned} f \circ \kappa_i &= \textstyle\bigvee_{n\in\mathbb{N}} \kappa_n \circ \pi_n \circ \kappa_i \\ &= \textstyle\bigvee_{n\geq i} \kappa_n \circ \pi_n \circ \kappa_i \\ &= \textstyle\bigvee_{n\geq i} \kappa_n \circ \pi_n \circ \kappa_n \circ f_{in} \\ &= \textstyle\bigvee_{n\geq i} \kappa_n \circ f_{in} \\ &= \textstyle\bigvee_{n\geq i} \kappa_i \\ &= \kappa_i. \end{aligned}$$

Hence $f\colon A \to A$ must be the identity, by uniqueness of mediating maps out of colimits.

At this stage we see

$$\kappa_i \circ \pi_i \leq \textstyle\bigvee_{n\in\mathbb{N}} \kappa_n \circ \pi_n = \mathrm{id}_A.$$

Now we can conclude that $\kappa_i\colon X_i \to A$ is an embedding, with associated projection $\kappa_i^p = \pi_i$. Further, these projections π_i form a cone for the ω-chain $f_n^p\colon X_{n+1} \to X_n$ since $f_n^p \circ \pi_{n+1} = f_n^p \circ \kappa_{n+1}^p = (\kappa_{n+1} \circ f_n)^p = \kappa_n^p = \pi_n$.

3. Assume $h, h'\colon Y \to A$ satisfy $\pi_n \circ h = \pi_n \circ h'$, for all $n \in \mathbb{N}$. Then by (2),

$$\begin{aligned} h = \mathrm{id}_A \circ h = \left(\textstyle\bigvee_n \kappa_n \circ \pi_n\right) \circ h &= \textstyle\bigvee_n \kappa_n \circ \pi_n \circ h \\ &= \textstyle\bigvee_n \kappa_n \circ \pi_n \circ h' \\ &= \left(\textstyle\bigvee_n \kappa_n \circ \pi_n\right) \circ h' = h'. \end{aligned}$$

4. Assume $g_n\colon Y \to X_n$ satisfying $f_n^p \circ g_{n+1} = g_n$. The maps $\kappa_n \circ g_n\colon Y \to A$ then form an ascending chain, since $\kappa_n \circ g_n = \kappa_{n+1} \circ f_n \circ f_n^p \circ g_{n+1} \leq \kappa_{n+1} \circ g_{n+1}$. Hence their join $g = \bigvee_n \kappa_n \circ g_n\colon Y \to A$ exists. It is a mediating map since

$$\begin{aligned} \pi_i \circ g &= \textstyle\bigvee_{n\in\mathbb{N}} \pi_i \circ \kappa_n \circ g_n \\ &= \textstyle\bigvee_{n>i} \pi_i \circ \kappa_n \circ g_n \\ &= \textstyle\bigvee_{n>i} f_{ni} \circ \pi_n \circ \kappa_n \circ g_n \\ &= \textstyle\bigvee_{n>i} f_{ni} \circ g_n \\ &= \textstyle\bigvee_{n>i} g_i \\ &= g_i. \end{aligned}$$

Moreover, this g is unique by the previous point. □

We now wish to obtain a fixed point $F(A) \xrightarrow{\cong} A$ which is an initial algebra and its inverse $A \xrightarrow{\cong} F(A)$ is a final coalgebra. The idea is to combine the previous result about limit-colimit coincidence with the chain-based approach from Proposition 4.6.1, where one uses a colimit of $F^n(0)$ to obtain the initial algebra and the limit of $F^n(1)$ for the final coalgebra. But if we are in a situation where we have a zero object 0 – which is both initial and final – then we obtain the same chain with coinciding limit and colimit.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Our next assumption is thus that the category at hand has a zero object, commonly written as 0. The singleton monoid is an example of a zero object in the category of monoids; see Exercise 2.1.6. Interestingly, the empty set 0 is also a zero object in the Kleisli category $\mathcal{K}\ell(\mathcal{P}) = \mathbf{SetsRel}$ of the powerset monad: for each set X there is a unique arrow $0 \to \mathcal{P}(X)$ in **Sets**, because 0 is initial in **Sets**, and there is a unique arrow $X \to \mathcal{P}(0)$, since $\mathcal{P}(0) = 1$ is final in **Sets**.

Given such a zero object 0 in a category $\mathbb{C}$, there is for each pair of objects $X, Y \in \mathbb{C}$ an arrow $X \to Y$, which we write as

$$\perp_{X,Y} = (X \xrightarrow{\;!\;} 0 \xrightarrow{\;!\;} Y), \tag{5.16}$$

using that 0 is both initial and final. Such a map $\perp_{X,Y}$ is sometimes called a zero map; it is typical in an algebraic setting, for instance in a category of groups or of vector or Hilbert spaces. The subscripts X, Y are often omitted. Notice that $\perp \circ f = \perp = g \circ \perp$, for all maps f, g.

We use Propositions 5.3.3 and 4.6.1 to generalise Theorem 5.3.1 to more general monads, as in [206].

Theorem 5.3.4 *Assume we have the following data on a category* $\mathbb{C}$.

- *Both an initial and a final object* $0, 1 \in \mathbb{C}$.
- *A monad* $T : \mathbb{C} \to \mathbb{C}$ *for which*
  - *the Kleisli category* $\mathcal{K}\ell(T)$ *is dcpo-enriched*
  - *the map* $T(0) \to 1$ *is an isomorphism, so that* $0 \in \mathcal{K}\ell(T)$ *is a zero object (see Exercise 5.2.5)*
  - *the resulting zero maps (5.16) in* $\mathcal{K}\ell(T)$ *are least elements in the hom-set dcpos.*
- *A functor* $F : \mathbb{C} \to \mathbb{C}$ *for which*
  - *there is a distributive* $\mathcal{K}\ell$*-law* $FT \Rightarrow TF$*, or equivalently by Proposition 5.2.5, a lifting* $\mathcal{K}\ell(F) : \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$*;*
  - *this lifting* $\mathcal{K}\ell(F) : \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ *is 'locally monotone':* $f \leq g$ *implies* $\mathcal{K}\ell(T)(f) \leq \mathcal{K}\ell(T)(g)$*;*
  - *there is an initial algebra* $\alpha : F(A) \xrightarrow{\cong} A$ *in* $\mathbb{C}$*, obtained as colimit of the chain* $F^n(0)$ *as in (4.12).*

The map $J(\alpha^{-1}) : A \xrightarrow{\;\cong\;} F(A) = \mathcal{K}\ell(F)(A)$ *is then a final coalgebra in the Kleisli category* $\mathcal{K}\ell(T)$*, for the lifted functor* $\mathcal{K}\ell(F) : \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Proof By assumption, the initial algebra structure $\alpha\colon F(A) \xrightarrow{\cong} A$ is obtained on the colimit A of the ω-chain $F^n(0)$, as in (4.12). The functor $J\colon \mathbb{C} \to \mathcal{K}\ell(T)$ is a left adjoint, and thus preserves this colimit; see Exercise 4.6.3. Hence in the Kleisli category $\mathcal{K}\ell(T)$ we obtain $A = J(A)$ as colimit of the chain $J(F^n(0)) = \mathcal{K}\ell(F)^n(J(0)) = \mathcal{K}\ell(F)^n(0) = F^n(0)$. This ω-chain in $\mathcal{K}\ell(T)$ thus starts from the zero object 0 and can be described explicitly as

$$\begin{tikzcd}
0 \arrow[r, bend right, "\perp"'] & F(0) \arrow[l, bend right, "\perp"'] \arrow[r, bend right, "\mathcal{K}\ell(F)(\perp)"'] & F^2(0) \arrow[l, bend right, "\mathcal{K}\ell(F)(\perp)"'] \arrow[r, bend right, "\mathcal{K}\ell(F)^2(\perp)"'] & \cdots \arrow[l, bend right, "\mathcal{K}\ell(F)^2(\perp)"']
\end{tikzcd} \tag{5.17}$$

The two maps written as $\perp$ are the unique map $0 \to F(0)$ obtained by initiality and the unique map $F(0) \to 0$ obtained by finality, as in (5.16). The maps $\mathcal{K}\ell(F)^n(\perp)\colon F^n(0) \to F^{n+1}(0)$ are embeddings, since 0 is zero object (on the left below) and $\perp$ is least (on the right):

$$\begin{tikzcd}
0 \arrow[r, "\perp"] \arrow[dr, equal] & F(0) \arrow[d, "\perp"] \\
& 0
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
F(0) \arrow[r, "\perp"] \arrow[dr, bend left, "\perp"] \arrow[dr, bend right, "\mathrm{id}"'] \arrow[dr, phantom, "\geq"] & 0 \arrow[d, "\perp"] \\
& F(0)
\end{tikzcd}$$

After applying $\mathcal{K}\ell(F)^n$ we obtain a chain of embeddings, by local monotonicity.

Thus, Proposition 5.3.3 applies and the colimit $A \in \mathcal{K}\ell(T)$ of the ω-chain of embeddings $\mathcal{K}\ell(F)^n(\perp)\colon F^n(0) \to F^{n+1}(0)$ in (5.17) is also the limit of the chain of associated projections $\mathcal{K}\ell(F)^n(\perp)\colon F^{n+1}(0) \to F^n(0)$. Hence it carries a final coalgebra structure $J(\alpha^{-1})\colon A \xrightarrow{\cong} F(A)$.

The finality of $J(\alpha^{-1})\colon A \xrightarrow{\cong} F(A)$ in $\mathcal{K}\ell(T)$ can also be proven directly, using the points (1)–(4) listed in Proposition 5.3.3. For a coalgebra $c\colon X \to TF(X)$ in $\mathbb{C}$, that is, for a coalgebra $X \to \mathcal{K}\ell(F)(X)$ in $\mathcal{K}\ell(T)$ we obtain a map $\mathrm{trace}_c\colon X \to A$ in $\mathcal{K}\ell(T)$ as join:

$$\mathrm{trace}_c = \bigvee_{n\in\mathbb{N}} \kappa_n \circledcirc c_n \text{ where } \begin{cases} c_0 = \perp\colon X \to 0 = F^0(0) \\ c_{n+1} = \mathcal{K}\ell(F)(c_n) \circledcirc c\colon X \to F(X) \to F^{n+1}(0) \end{cases}$$

and where the maps $\kappa_n\colon F^n(0) \to A$ form the colimit cone, so that $J(\alpha) \circledcirc \mathcal{K}\ell(F)(\kappa_n) = \kappa_{n+1}$. This gives $\mathcal{K}\ell(F)(\pi_n) \circledcirc J(\alpha^{-1}) = \pi_{n+1}$ by taking the associated projections $(-)^p$. This trace map satisfies $\pi_n \circledcirc \mathrm{trace}_c = c_n$, by construction; see Proposition 5.3.3.4. We then get a commuting diagram in $\mathcal{K}\ell(T)$:

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$\begin{tikzcd}
F(X) \arrow[r, "\mathcal{K}\ell(F)(\mathrm{trace}_c)"] & F(A) \\
X \arrow[u, "c"] \arrow[r, "\mathrm{trace}_c"'] & A \arrow[u, "\cong"', "J(\alpha^{-1})"]
\end{tikzcd}$$

Commutation is obtained by using that the projections $\pi_n = \kappa_n^p : A \to F^n(0)$ are jointly monic (see Proposition 5.3.3.3):

$$\begin{aligned}
\pi_n \circledcirc J(\alpha) \circledcirc \mathcal{K}\ell(F)(\mathrm{trace}_c) \circledcirc c &= \mathcal{K}\ell(F)(\pi_{n-1}) \circledcirc \mathcal{K}\ell(F)(\mathrm{trace}_c) \circledcirc c \\
&= \mathcal{K}\ell(F)(\pi_{n-1} \circledcirc \mathrm{trace}_c) \circledcirc c \\
&= \mathcal{K}\ell(F)(c_{n-1}) \circledcirc c \\
&= c_n \\
&= \pi_n \circledcirc \mathrm{trace}_c.
\end{aligned}$$

In a similar way one obtains uniqueness: assume a map $g : X \to A$ in $\mathcal{K}\ell(T)$ satisfies $\mathcal{K}\ell(F)(g) \circledcirc c = J(\alpha^{-1}) \circledcirc g$. The equation $g = \mathrm{trace}_c$ follows if $\pi_n \circledcirc g = \pi_n \circledcirc \mathrm{trace}_c$ holds for each n. This is shown by induction on n, using that $\pi_n \circledcirc \mathrm{trace}_c = c_n$. The case $n = 0$ is trivial, and

$$\begin{aligned}
\pi_{n+1} \circledcirc g &= \pi_{n+1} \circledcirc J(\alpha) \circledcirc \mathcal{K}\ell(F)(g) \circledcirc c \\
&= \mathcal{K}\ell(F)(\pi_n) \circledcirc \mathcal{K}\ell(F)(g) \circledcirc c \\
&= \mathcal{K}\ell(F)(\pi_n \circledcirc g) \circledcirc c \\
&\overset{\text{(IH)}}{=} \mathcal{K}\ell(F)(c_n) \circledcirc c \\
&= c_{n+1}.
\end{aligned}$$

❑

This result requires that the lifted functor $\mathcal{K}\ell(F)$ is only locally monotone, instead of locally continuous. The latter is a more common assumption in this setting; it leads to a slightly simpler proof (see Exercise 5.3.4).

Example 5.3.5 The powerset monad $\mathcal{P}$ satisfies the assumptions of Theorem 5.3.4. Another example is the lift monad $\mathcal{L} = 1 + (-)$ on **Sets**, where maps $f, g : X \to \mathcal{L}(Y)$ are ordered via the so-called flat order: $f \leq g$ iff $f(x) = y \in Y$ implies $g(x) = y$. A more interesting example is obtained via the infinite subdistribution monad $\mathcal{D}_{\leq 1}^{\infty}$ on **Sets**. It involves distributions $\varphi \in \mathcal{D}_{\leq 1}^{\infty}(X)$ given by functions $\varphi : X \to [0, 1]$ satisfying $\sum_x \varphi(x) \leq 1$. Notice that there is no finiteness requirement for the support (but supports have at countably many elements, as in Exercise 4.1.8). Clearly, $\mathcal{D}_{\leq 1}^{\infty}(0) \cong 1$. The resulting zero maps $\perp : X \to \mathcal{D}_{\leq 1}^{\infty}(Y)$ in the Kleisli category are given by $\perp(x)(y) = 0$. Subdistributions $\varphi, \psi \in \mathcal{D}_{\leq 1}^{\infty}(X)$ can be ordered pointwise: $\varphi \leq \psi$ iff $\varphi(x) \leq \psi(x)$ for all $x \in X$. This yields a dcpo, with $\perp$ as least element.

As functor $F : \mathbf{Sets} \to \mathbf{Sets}$ we take $F(X) = 1 + (L \times X)$, as before, with initial algebra $[\mathsf{nil}, \mathsf{cons}] : F(L^\star) \xrightarrow{\cong} L^\star$ given by lists of labels. There is a distributive

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

law $\lambda\colon F(\mathcal{D}^{\infty}_{\leq 1}(X)) \to \mathcal{D}^{\infty}_{\leq 1}(F(X))$ by Lemma 5.2.12, since the monad $\mathcal{D}^{\infty}_{\leq 1}$ is commutative. Explicitly, this λ is given by

$$\lambda(*)(u) = \begin{cases} 1 & \text{if } u = * \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad \lambda(a,\varphi)(u) = \begin{cases} \varphi(x) & \text{if } u = (a,x) \\ 0 & \text{otherwise.} \end{cases}$$

The resulting lifting $\mathcal{K}\ell(F)\colon \mathcal{K}\ell(\mathcal{D}^{\infty}_{\leq 1}) \to \mathcal{K}\ell(\mathcal{D}^{\infty}_{\leq 1})$ sends a map $f\colon X \to \mathcal{D}^{\infty}_{\leq 1}(Y)$ to the map $\mathcal{K}\ell(F)(\backslash, f)\colon 1 + L \times X \to \mathcal{D}^{\infty}_{\leq 1}(1 + L \times Y)$ given by

$$\begin{aligned} \mathcal{K}\ell(\,f)(*)(u) &= \begin{cases} 1 & \text{if } u = * \\ 0 & \text{otherwise} \end{cases} \\ \mathcal{K}\ell(F)(\,f)(a,x)(u) &= \begin{cases} f(x)(y) & \text{if } u = (a,y) \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Theorem 5.3.4 now says that we have a final coalgebra $\eta \circ [\mathsf{nil}, \mathsf{cons}]^{-1}\colon L^{\star} \xrightarrow{\cong} F(L^{\star})$ in $\mathcal{K}\ell(\mathcal{D}^{\infty}_{\leq 1})$. This will be illustrated via an example.

Assume our state space X consists of an urn containing black, white and red balls. Thus X can be described as the set of multisets $\mathcal{M}_{\mathbb{N}}(3) = \{\varphi\colon 3 \to \mathbb{N} \mid \mathrm{supp}(\varphi) \text{ is finite}\} \cong \mathbb{N}^3$, where $3 = \{\mathsf{B}, \mathsf{W}, \mathsf{R}\}$ is the three-element set of balls. A typical state $\varphi \in \mathcal{M}_{\mathbb{N}}(3)$ has the multiset form $\varphi = n_B|\mathsf{B}\rangle + n_W|\mathsf{W}\rangle + n_R|\mathsf{R}\rangle$, where $n_B = \varphi(\mathsf{B}), n_W = \varphi(\mathsf{W}), n_R = \varphi(\mathsf{R})$ describe the number of balls of each colour. We write $\|\varphi\| = \varphi(\mathsf{B}) + \varphi(\mathsf{W}) + \varphi(\mathsf{R})$ for the total number of balls, and $\varphi - \mathsf{B} \in \mathcal{M}_{\mathbb{N}}(3)$ for the multiset with one black ball removed (if any). Formally,

$$(\varphi - \mathsf{B})(\mathsf{B}) = \begin{cases} \varphi(\mathsf{B}) - 1 & \text{if } \varphi(\mathsf{B}) > 0 \\ 0 & \text{otherwise} \end{cases} \qquad \begin{aligned} (\varphi - \mathsf{B})(\mathsf{W}) &= \varphi(\mathsf{W}) \\ (\varphi - \mathsf{B})(\mathsf{R}) &= \varphi(\mathsf{R}). \end{aligned}$$

The multisets $\varphi - \mathsf{W}$ and $\varphi - \mathsf{R}$ are defined similarly.

Taking out a ball constitutes a transition. The probability that you take a certain colour is determined by the relative multiplicities: the balls are taken blindly; they cannot be chosen. If a black or white ball is taken out, a next ball may be picked, but the process stops with a red ball. All this is formalised via a coalgebra:

$$\begin{tikzcd} X \arrow[r, "c"] & \mathcal{D}^{\infty}_{\leq 1}(1 + 3 \times X) \\ \varphi \arrow[r, mapsto] & \frac{\varphi(\mathsf{B})}{\|\varphi\|}\big|\mathsf{B}, \varphi - \mathsf{B}\big\rangle + \frac{\varphi(\mathsf{W})}{\|\varphi\|}\big|\mathsf{W}, \varphi - \mathsf{W}\big\rangle + \frac{\varphi(\mathsf{R})}{\|\varphi\|}\big| * \big\rangle \end{tikzcd}$$

where $\varphi \in X = \mathcal{M}_{\mathbb{N}}(3)$ is a multiset of coloured balls, representing the contents of the urn. Notice that in order to describe this coalgebra we could have used the monad $\mathcal{D}$ of ordinary distributions instead of $\mathcal{D}^{\infty}_{\leq 1}$. However, a finite system may have infinitely many traces, which can be modelled only via $\mathcal{D}^{\infty}_{\leq 1}$.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Theorem 5.3.4 now gives a coalgebra map $\mathrm{trace}_c \colon X \to \mathcal{D}^{\infty}_{\leq 1}(3^\star)$ in the Kleisli category $\mathcal{K}\ell(\mathcal{D}^{\infty}_{\leq 1})$. For a filled urn $\varphi \in X$, this map $\mathrm{trace}_c(\varphi)$ gives a probability distribution over sequences of (consecutive) balls, described as elements of the set $3^\star = \{\mathsf{B}, \mathsf{W}, \mathsf{R}\}^\star$. For a specific sequence σ of (white and black) balls $\mathrm{trace}_c(\varphi)(\sigma)$ gives the probability of consecutively taking the balls in σ, until a red ball appears. The finality diagram in $\mathcal{K}\ell(\mathcal{D}^{\infty}_{\leq 1})$ precisely captures how to compute these probabilities. This diagram in $\mathcal{K}\ell(\mathcal{D}^{\infty}_{\leq 1})$ is

$$\begin{tikzcd}
1 + 3 \times X \arrow[r, "\mathcal{K}\ell(F)(\mathrm{trace}_c)"] & 1 + 3 \times 3^\star \\
X \arrow[u, "c"] \arrow[r, "\mathrm{trace}_c"'] & 3^\star \arrow[u, "\cong", "J([\mathsf{nil}, \mathsf{cons}]^{-1})"']
\end{tikzcd}$$

Commutation says:

$$\begin{aligned}
\mathrm{trace}_c(\varphi)(\sigma) &= (\mathcal{K}\ell(F)(\mathrm{trace}_c) \circledcirc J(\alpha) \circledcirc c)(\varphi)(\sigma) \\
&= \sum_{u \in 1+3\times X} \; \sum_{v \in 1+3\times 3^\star} c(\varphi)(u) \cdot \mathcal{K}\ell(F)(\mathrm{trace}_c)(u)(v) \cdot J(\alpha)(v)(\sigma) \\
&= \begin{cases} c(\varphi)(*) & \text{if } \sigma = \mathsf{nil} \\ c(\varphi)(C, \varphi - C) \cdot \mathrm{trace}_c(\varphi - C)(\sigma') & \text{if } \sigma = \mathsf{cons}(C, \sigma') \end{cases} \\
&= \begin{cases} \frac{\varphi(\mathsf{R})}{\|\varphi\|} & \text{if } \sigma = \mathsf{nil} \\ \frac{\varphi(\mathsf{B})}{\|\varphi\|} \cdot \mathrm{trace}_c(\varphi - \mathsf{B})(\sigma') & \text{if } \sigma = \mathsf{cons}(\mathsf{B}, \sigma') \\ \frac{\varphi(\mathsf{W})}{\|\varphi\|} \cdot \mathrm{trace}_c(\varphi - \mathsf{W})(\sigma') & \text{if } \sigma = \mathsf{cons}(\mathsf{W}, \sigma') \\ 0 & \text{if } \sigma = \mathsf{cons}(\mathsf{R}, \sigma'). \end{cases}
\end{aligned}$$

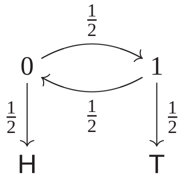
Thus, for instance, if we start with a state/urn $\varphi = 3|\mathsf{B}\rangle + 2|\mathsf{W}\rangle + 1|\mathsf{R}\rangle$, then the probability of a trace $\langle \mathsf{B}, \mathsf{W}, \mathsf{B}, \mathsf{B}\rangle$ can be calculated as

$$\begin{aligned}
&\mathrm{trace}_c(3|\mathsf{B}\rangle + 2|\mathsf{W}\rangle + 1|\mathsf{R}\rangle)(\langle \mathsf{B}, \mathsf{W}, \mathsf{B}, \mathsf{B}\rangle) \\
&= \tfrac{3}{6} \cdot \mathrm{trace}_c(2|\mathsf{B}\rangle + 2|\mathsf{W}\rangle + 1|\mathsf{R}\rangle)(\langle \mathsf{W}, \mathsf{B}, \mathsf{B}\rangle) \\
&= \tfrac{3}{6} \cdot \tfrac{2}{5} \cdot \mathrm{trace}_c(2|\mathsf{B}\rangle + 1|\mathsf{W}\rangle + 1|\mathsf{R}\rangle)(\langle \mathsf{B}, \mathsf{B}\rangle) \\
&= \tfrac{3}{6} \cdot \tfrac{2}{5} \cdot \tfrac{2}{4} \cdot \mathrm{trace}_c(1|\mathsf{B}\rangle + 1|\mathsf{W}\rangle + 1|\mathsf{R}\rangle)(\langle \mathsf{B}\rangle) \\
&= \tfrac{3}{6} \cdot \tfrac{2}{5} \cdot \tfrac{2}{4} \cdot \tfrac{1}{3} \cdot \mathrm{trace}_c(1|\mathsf{W}\rangle + 1|\mathsf{R}\rangle)(\langle\rangle) \\
&= \tfrac{3}{6} \cdot \tfrac{2}{5} \cdot \tfrac{2}{4} \cdot \tfrac{1}{3} \cdot \tfrac{1}{2} \\
&= \tfrac{1}{60}.
\end{aligned}$$

Example 5.3.6 Our next illustration again uses the subdistribution monad $\mathcal{D}^{\infty}_{\leq 1}$. It shows how an unfair coin can be simulated by two fair ones. This is based on [310, 311]. Consider the coalgebra $c \colon 2 \to \mathcal{D}^{\infty}_{\leq 1}(\{\mathsf{H}, \mathsf{T}\} + 2)$, with

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

two-element state space $2 = \{0, 1\}$, described in the following diagram:



Explicitly, this coalgebra c can be described on the state space 2 via convex sums:

$$c(0) = \tfrac{1}{2}|\mathsf{H}\rangle + \tfrac{1}{2}|1\rangle \qquad \text{and} \qquad c(1) = \tfrac{1}{2}|\mathsf{T}\rangle + \tfrac{1}{2}|0\rangle.$$

The coalgebra is of the form $2 \to \mathcal{D}^{\infty}_{\leq 1}(F(2))$ for the functor $F(X) = \{\mathsf{H}, \mathsf{T}\} + X$. The initial F-algebra is $\mathbb{N} \times \{\mathsf{H}, \mathsf{T}\}$, with structure map:

$$\{\mathsf{H}, \mathsf{T}\} + \mathbb{N} \times \{\mathsf{H}, \mathsf{T}\} \xrightarrow[\cong]{\alpha} \mathbb{N} \times \{\mathsf{H}, \mathsf{T}\} \quad \text{given by} \quad \begin{cases} \alpha(\mathsf{A}) = (0, \mathsf{A}) \\ \alpha(n, \mathsf{A}) = (n + 1, \mathsf{A}), \end{cases}$$

where $\mathsf{A} \in \{\mathsf{H}, \mathsf{T}\}$.

The theory described above now yields a trace map by finality in the Kleisli category. It is of the form

$$2 \xrightarrow{\;\text{trace}_c\;} \mathcal{D}^{\infty}_{\leq 1}(\mathbb{N} \times \{\mathsf{H}, \mathsf{T}\}),$$

given by the infinite convex sums

$$\begin{aligned} \text{trace}_c(0) &= \tfrac{1}{2}|0, \mathsf{H}\rangle + \tfrac{1}{4}|1, \mathsf{T}\rangle + \tfrac{1}{8}|2, \mathsf{H}\rangle + \tfrac{1}{16}|3, \mathsf{T}\rangle + \cdots \\ &= \sum_{n \in \mathbb{N}} \frac{1}{2^{2n+1}}|2n, \mathsf{H}\rangle + \frac{1}{2^{2n+2}}|2n + 1, \mathsf{T}\rangle \\ \text{trace}_c(1) &= \tfrac{1}{2}|0, \mathsf{T}\rangle + \tfrac{1}{4}|1, \mathsf{H}\rangle + \tfrac{1}{8}|2, \mathsf{T}\rangle + \tfrac{1}{16}|3, \mathsf{H}\rangle + \cdots \\ &= \sum_{n \in \mathbb{N}} \frac{1}{2^{2n+1}}|2n, \mathsf{T}\rangle + \frac{1}{2^{2n+2}}|2n + 1, \mathsf{H}\rangle. \end{aligned}$$

A summand like $\frac{1}{8}|2, \mathsf{H}\rangle$ in $\text{trace}_c(0)$ represents the probability of $\frac{1}{8}$ of getting outcome H via two transitions, starting in state 0.

We can add all the summands with the same coin together, by applying the second projection $\pi_2 \colon \mathbb{N} \times \{\mathsf{H}, \mathsf{T}\} \to \{\mathsf{H}, \mathsf{T}\}$ to these sums. The resulting composite map

$$\text{unfair-coin} = \Big(2 \xrightarrow{\;\text{trace}_c\;} \mathcal{D}^{\infty}_{\leq 1}(\mathbb{N} \times \{\mathsf{H}, \mathsf{T}\}) \xrightarrow{\;\mathcal{D}^{\infty}_{\leq 1}(\pi_2)\;} \mathcal{D}^{\infty}_{\leq 1}(\{\mathsf{H}, \mathsf{T}\})\Big)$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

is given by

$$\begin{aligned}
\text{unfair-coin}(0) &= \Bigg(\sum_{n\in\mathbb{N}} \frac{1}{2^{2n+1}}\Bigg)|\mathsf{H}\rangle + \Bigg(\sum_{n\in\mathbb{N}} \frac{1}{2^{2n+2}}\Bigg)|\mathsf{T}\rangle = \tfrac{2}{3}|\mathsf{H}\rangle + \tfrac{1}{3}|\mathsf{T}\rangle \\
\text{unfair-coin}(1) &= \Bigg(\sum_{n\in\mathbb{N}} \frac{1}{2^{2n+1}}\Bigg)|\mathsf{T}\rangle + \Bigg(\sum_{n\in\mathbb{N}} \frac{1}{2^{2n+2}}\Bigg)|\mathsf{H}\rangle = \tfrac{2}{3}|\mathsf{T}\rangle + \tfrac{1}{3}|\mathsf{H}\rangle.
\end{aligned}$$

For these last steps we use the familiar equation $\sum_n a^n = \frac{1}{1-a}$ if $|a| < 1$, for instance in

$$\textstyle\sum_n \frac{1}{2^{2n+1}} = \frac{1}{2}\sum_n \left(\frac{1}{2}\right)^{2n} = \frac{1}{2}\sum_n \left(\frac{1}{4}\right)^{n} = \frac{1}{2}\cdot\frac{1}{1-\frac{1}{4}} = \frac{1}{2}\cdot\frac{4}{3} = \frac{2}{3}.$$

The initial algebra $\mathbb{N}\times\{\mathsf{H},\mathsf{T}\}$ can be described more abstractly as a copower $\mathbb{N}\cdot\{\mathsf{H},\mathsf{T}\}$. The second projection π_2 used above then becomes a codiagonal $\nabla\colon \mathbb{N}\cdot\{\mathsf{H},\mathsf{T}\} \to \{\mathsf{H},\mathsf{T}\}$. This description is used in [248], where a 'monoidal trace operator' in a Kleisli category is constructed from coalgebraic traces. This monoidal trace can thus be used to compute the outcome of the unfair coin directly. Alternatively, as in [311] one can use recursive coalgebras; see [93, 21].

Finally, we briefly return to the transition system example that we used in the beginning of this section.

Remark 5.3.7 For the special case of transition systems $X \to \mathcal{P}F(X) = \mathcal{P}(1+(L\times X))$ involving the powerset monad, there are two alternative ways to obtain its trace semantics: one via determinisation and one via logical interpretation. We briefly review these constructions.

1. As already noted –for instance in Exercise 2.2.3 – a transition system of the form $X \to \mathcal{P}(1 + (L \times X))$ can also be viewed as a non-deterministic automaton $\langle\delta,\varepsilon\rangle\colon X \to \mathcal{P}(X)^L \times 2$. As such it may be 'determinised' into an automaton $\langle\delta',\varepsilon'\rangle\colon \mathcal{P}(X) \to \mathcal{P}(X)^L \times 2$ via the standard definitions:

   $$\delta'(U)(a) = \bigcup\{\delta(x)(a) \mid x \in U\} \qquad \varepsilon'(U) = 1 \iff \exists x \in U.\, \varepsilon(x) = 1.$$

   See also [427] and Exercise 5.2.7. Since $\mathcal{P}(L^\star)$ is the final deterministic automaton (see Corollary 2.3.6.2), we obtain a unique coalgebra homomorphism $h\colon \mathcal{P}(X) \to \mathcal{P}(L^\star)$. The trace map $X \to \mathcal{P}(L^\star)$ is then $h \circ \{-\}$. This approach is elaborated in Example 5.4.12.1, and in [429, 272].
2. The logically oriented approach uses the fact that the set $\mathcal{P}(X)$ of predicates on the state space of our non-deterministic automaton $\langle\delta,\varepsilon\rangle\colon X \to \mathcal{P}(X)^L \times 2$ carries elementary logical structure, in the form of an algebra for the functor

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$F = 1 + (L \times -)$. Indeed, there is an algebra $[\varepsilon, \neg \bigcirc \neg]\colon F(\mathcal{P}(X)) \to \mathcal{P}(X)$ given by $\varepsilon\colon 1 \to \mathcal{P}(X)$ as a subset of final states and by a labelled (strong) nexttime $\neg\bigcirc\neg\colon L \times \mathcal{P}(X) \to \mathcal{P}(X)$ operator. It maps a pair $(a, P) \in L \times \mathcal{P}(X)$ to the predicate $\neg \bigcirc \neg(a, P)$, commonly written as $\neg \bigcirc_a \neg(P)$, given by

$$\neg \bigcirc_a \neg(P) \;=\; \{x \in X \mid \exists x' \in X.\, x \xrightarrow{a} x' \wedge P(x')\}.$$

By initiality of $L^\star$ we then obtain an algebra homomorphism $\ell\colon L^\star \to \mathcal{P}(X)$. The adjunction $\mathbf{Sets}^{\mathrm{op}} \leftrightarrows \mathbf{Sets}$ from Exercise 2.5.2 now yields the trace map $X \to \mathcal{P}(L^\star)$. This second approach fits in the ‘testing’ framework of [377] and uses coalgebraic modal logic; see Section 6.5 for a more general account.

The trace semantics that we have described in this section induces a new type of equivalence on states (of suitable coalgebras), namely trace equivalence, given by the relation $\{(x, y) \mid \mathrm{trace}(x) = \mathrm{trace}(y)\}$. Exercise 5.3.5 below shows that bisimilarity implies trace equivalence, but not the other way around.

Exercises

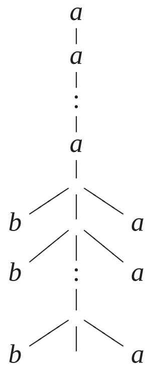
5.3.1 Suppose we have a monad $T\colon \mathbb{C} \to \mathbb{C}$ that satisfies $T(0) \cong 0$ and $T(1) \cong 1$. Use Exercise 5.2.9 to transform T into a monad $T' = T(1 + (-))$ that satisfies $T'(0) \cong 1$, as needed in Theorem 5.3.4. (This transformation trick applies for instance to the ordinary distribution monad $\mathcal{D}$; a non-trivial dcpo structure on these discrete distributions in $\mathcal{D}(X)$ is described in [106].)

5.3.2 Consider the alphabet $\{a, b\}$ with two non-terminals V, W and productions

$$V \longrightarrow a \cdot V \qquad V \longrightarrow W \qquad W \longrightarrow b \cdot W \cdot a \qquad W \longrightarrow \langle\rangle.$$

1. Describe this CFG as a coalgebra $X \to \mathcal{P}(FX)$ with state space $X = \{V, W\}$ and functor $F = ((-) + \{a, b\})^\star$.
2. Check that the unparsed language generated by the symbol V is the set $\{a^n b^m a^m \mid n, m \in \mathbb{N}\}$.
3. Consider the parsed language map $\mathrm{trace}_g\colon X \to \{a, b\}^\triangle$ from Example 5.3.2 and show that a term $t \in \mathrm{trace}_g(V)$ can be drawn as a tree:

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006



5.3.3 1. Define a function $L^{\triangle} \to L^{\star}$ that maps parsed words to 'flat' words by initiality. Prove that this function is a split epi.

2. Prove that the assignment $L \mapsto L^{\triangle}$ is functorial and that the mapping $L^{\triangle} \twoheadrightarrow L^{\star}$ from (1) forms a natural transformation.

3. Let $L^{\wedge}$ be the final coalgebra of the functor $X \mapsto (X + L)^{\star}$ from Example 5.3.2. It consists of both the finite and infinite parsed words. Define a map $L^{\triangle} \to L^{\wedge}$ and show that it is injective.

(This gives the following fundamental span of languages $L^{\star} \twoheadleftarrow L^{\triangle} \rightarrowtail L^{\wedge}$. The three operations involved $L \mapsto L^{\star}, L^{\triangle}, L^{\wedge}$ are all monads and the mappings between them preserve the monad structure; see [203].)

5.3.4 Consider the situation as described in Theorem 5.3.4, but with the stronger assumption that the lifted functor $\mathcal{K}\ell(F)\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ is *locally continuous*. This means that directed joins in hom-sets are preserved: $\mathcal{K}\ell(F)(\bigvee_i f_i) = \bigvee_i \mathcal{K}\ell(F)(f_i)$. Check that under this assumption the proof of Theorem 5.3.4 can be simplified in the following manner. For an initial F-algebra $\alpha\colon F(A) \xrightarrow{\cong} A$ and a coalgebra $c\colon X \to \mathcal{K}\ell(F)(X)$, consider the following two operators on Kleisli hom-sets:

$$\begin{array}{ccc} \mathcal{K}\ell(T)(A,A) \xrightarrow{\Phi} \mathcal{K}\ell(T)(A,A) & \qquad & \mathcal{K}\ell(T)(X,A) \xrightarrow{\Psi} \mathcal{K}\ell(T)(X,A) \\ g \longmapsto \mathcal{K}\ell(F)(g) \circledcirc J(\alpha) \circledcirc J(\alpha^{-1}) & & h \longmapsto \mathcal{K}\ell(F)(h) \circledcirc J(\alpha) \circledcirc c. \end{array}$$

1. Prove that both Φ and Ψ are continuous.
2. Prove that the least fixed point $g = \mathrm{fix}(\Phi) = \bigvee_{n\in\mathbb{N}} \Phi^n(\bot)$ is a map of algebras $J(\alpha) \to J(\alpha)$. Conclude $g = \mathrm{id}$ from Proposition 5.2.6.
3. Check that the least fixed point $h = \mathrm{fix}(\Psi) = \bigvee_{n\in\mathbb{N}} \Psi^n(\bot)$ is a map of coalgebras $c \to J(\alpha^{-1})$.
4. Prove that this h is the unique such map. *Hint*: Use that if h' is also such a coalgebra map, then $\Phi^n(\bot) \circledcirc h' = \Psi^n(\bot)$.

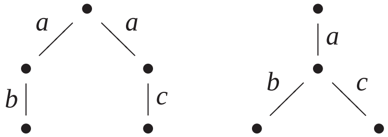
Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

5.3.5 Let $c\colon X \to \mathcal{P}F(X)$ and $d\colon Y \to \mathcal{P}F(Y)$ be two coalgebras, where $F\colon \mathbf{Sets} \to \mathbf{Sets}$ preserves weak pullbacks and has an initial algebra $F(A) \xrightarrow{\cong} A$ (as in Theorem 5.3.1).

1. Let $f\colon X \to Y$ be a homomorphisms (of $\mathcal{P}F$-coalgebras). Prove that $\mathrm{trace}_d \circledcirc f = \mathrm{trace}_c\colon X \to A$ in $\mathcal{K}\ell(\mathcal{P})$.
2. Conclude that bisimilarity is included in trace equivalence:

$$x \mathrel{{}_{c}\underline{\leftrightarrow}_{d}} y \Longrightarrow \mathrm{trace}_c(x) = \mathrm{trace}_d(y).$$

3. Check that the reverse implication ($\Longleftarrow$) in (2) does not hold, for instance via the following two pictures:



5.4 Eilenberg–Moore Categories and Distributive Laws

This section introduces the category $\mathcal{EM}(T)$, called the Eilenberg–Moore category, for a monad or comonad T. Its objects are algebras or coalgebras of T, but with some additional requirements. These requirements make them different from coalgebras of a functor.

Definition 5.4.1 Given a monad $T = (T, \eta, \mu)$ on a category $\mathbb{C}$ (see Definition 5.1.1), one defines an **algebra** of this monad as an algebra $\alpha\colon T(X) \to X$ of the functor T satisfying two additional requirements, expressed via the diagrams

$$\begin{tikzcd} X \arrow[r, "\eta_X"] \arrow[dr, equal] & T(X) \arrow[d, "\alpha"] \\ & X \end{tikzcd} \qquad \begin{tikzcd} T^2(X) \arrow[r, "\mu_X"] \arrow[d, "T(\alpha)"'] & T(X) \arrow[d, "\alpha"] \\ T(X) \arrow[r, "\alpha"'] & X \end{tikzcd} \tag{5.18}$$

We write $\mathcal{EM}(T) = \mathcal{EM}(T, \eta, \mu)$ for the category with such monad-algebras as objects. The morphisms are the same as homomorphisms of functor-algebras: a morphism from $\alpha\colon T(X) \to X$ to $\beta\colon T(Y) \to Y$ is a map $f\colon X \to Y$ in $\mathbb{C}$ with $\beta \circ T(f) = f \circ \alpha$.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Similarly, for a comonad $S = (S, \varepsilon, \delta)$ a **coalgebra** is a coalgebra $\alpha\colon X \to S(X)$ of the functor S satisfying

$$\begin{tikzcd} S(X) \arrow[r, "\varepsilon_X"] & X \\ X \arrow[u, "\alpha"] \arrow[ur, equal] & \end{tikzcd} \qquad \begin{tikzcd} S(X) \arrow[r, "\delta_X"] & S^2(X) \\ X \arrow[u, "\alpha"] \arrow[r, "\alpha"'] & S(X) \arrow[u, "S(\alpha)"'] \end{tikzcd} \tag{5.19}$$

We write $\mathcal{EM}(S) = \mathcal{EM}(S, \varepsilon, \delta)$ for this category of comonad coalgebras, with homomorphisms given by ordinary coalgebra homomorphisms.

There is ample room for confusion at this stage. First of all, the same notation $\mathcal{EM}(T)$ is used for both a category of algebras, if T is a monad, and for a category of coalgebras, if T is a comonad. The context should make clear which option is intended. What might help a bit is that we generally use the letter T for a monad and S for a comonad. Notice, by the way, that for Kleisli categories we have the same overloading of notation.

Next, ‘algebra’ or ‘coalgebra’ may be used both for a functor – in which case it is just a map of a certain form – and for a (co)monad; in the latter case such a map should satisfy additional equations, as described in (5.18) and (5.19) above. In order to emphasise what is intended we sometimes explicitly speak of a *functor algebra* or of a *monad algebra*/*Eilenberg–Moore algebra* – and similarly, we can have a *functor coalgebra* and a *comonad coalgebra*/*Eilenberg–Moore coalgebra*.

Recall that we use the notation **Alg**(F) and **CoAlg**(F) for the categories of algebras and coalgebras of a functor F. They are related to categories $\mathcal{EM}(-)$ of Eilenberg–Moore algebras and coalgebras via two full and faithful (horizontal) functors in commuting triangles:

$$\begin{tikzcd} \mathcal{EM}(T, \eta, \mu) \arrow[rr] \arrow[dr] & & \mathbf{Alg}(T) \arrow[dl] \\ & \mathbb{C} & \end{tikzcd} \qquad \begin{tikzcd} \mathcal{EM}(S, \varepsilon, \delta) \arrow[rr] \arrow[dr] & & \mathbf{CoAlg}(S) \arrow[dl] \\ & \mathbb{C} & \end{tikzcd}$$

Here, T is a monad and S is a comonad. In writing **Alg**(T) and **CoAlg**(S) we consider them as ordinary functors, forgetting about their (co)unit and (co)multiplication.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Lemma 5.4.2 *Let T be a monad and S a comonad on a category $\mathbb{C}$. The obvious forgetful functors have left and right adjoints*

$$\begin{tikzcd}
\mathcal{EM}(T) \arrow[d, ""{name=U}] \\
\mathbb{C} \arrow[u, bend left=60, "\mathcal{F}"] \arrow[u, phantom, "\dashv"]
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathcal{EM}(S) \arrow[d] \\
\mathbb{C} \arrow[u, bend right=60, "\mathcal{G}"'] \arrow[u, phantom, "\dashv"]
\end{tikzcd}$$

given by multiplication and comultiplication:

$$\mathcal{F}(X) = \left(\begin{array}{c} T^2(X) \\ \downarrow\mu_X \\ T(X) \end{array}\right) \qquad \textit{and} \qquad \mathcal{G}(Y) = \left(\begin{array}{c} S^2(Y) \\ \uparrow\delta_Y \\ S(Y) \end{array}\right).$$

These adjunctions give rise to a comonad on the category $\mathcal{EM}(T)$ of algebras, and to a monad on the category $\mathcal{EM}(S)$ of coalgebras; see Exercise 5.4.8.

Also the Kleisli categories can be embedded full and faithfully in Eilenberg–Moore categories, via commuting triangles:

$$\begin{tikzcd}
\mathcal{K}\ell(T) \arrow[rr] \arrow[dr] & & \mathcal{EM}(T) \arrow[dl] \\
& \mathbb{C} &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathcal{K}\ell(S) \arrow[rr] \arrow[dr] & & \mathcal{EM}(S) \arrow[dl] \\
& \mathbb{C} &
\end{tikzcd}$$

Recall that maps in Kleisli categories are used to represent computations. Through the full and faithful embeddings into Eilenberg–Moore categories these computations can also be studied, between free algebras or cofree coalgebras, in richer universes, with more categorical structure; see Lemma 5.4.5 and Proposition 5.4.6 below.

Proof We shall sketch the proof for the comonad case. The monad case is dual. For an arbitrary comonad coalgebra $\alpha\colon X \to S(X)$ we have to produce a bijective correspondence:

$$\begin{array}{c}
\left(\begin{array}{c} S(X) \\ \alpha\uparrow \\ X \end{array}\right) \xrightarrow{\qquad f \qquad} \left(\begin{array}{c} S^2(Y) \\ \uparrow\delta_Y \\ S(Y) \end{array}\right) \\
\hline\hline
X \xrightarrow[\qquad g \qquad]{} Y
\end{array}.$$

It is given as follows.

- For a map of coalgebras $f\colon X \to S(Y)$ take $\overline{f} = \varepsilon \circ f\colon X \to Y$.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

- For an ordinary map $g\colon X \to Y$ in $\mathbb{C}$ take $\overline{g} = S(g) \circ \alpha\colon X \to S(Y)$. It is a map of coalgebras since

$$\begin{aligned} S(\overline{g}) \circ \alpha &= S^2(g) \circ S(\alpha) \circ \alpha \\ &= S^2(g) \circ \delta \circ \alpha && \text{see (5.19)} \\ &= \delta \circ S(g) \circ \alpha && \text{by naturality of } \delta \\ &= \delta \circ \overline{g}. \end{aligned}$$

These transformations are each other's inverse – i.e. $\overline{\overline{f}} = f$ and $\overline{\overline{g}} = g$ – via the coalgebra laws (5.19).

The functor $\mathcal{K}\ell(S) \to \mathcal{EM}(S)$ is on objects $Y \mapsto \mathcal{G}(Y) = (\delta_Y\colon S(Y) \to S^2(Y))$. A Kleisli map $f\colon Y \to Z$, where $f\colon S(Y) \to Z$ in $\mathbb{C}$ yields $S(f) \circ \delta\colon S(Y) \to S(Z)$. We skip the proof that this is a map of coalgebras and show that each coalgebra map $g\colon \mathcal{G}(Y) \to \mathcal{G}(Z)$ is of this form. Take $f = \varepsilon \circ g\colon S(Y) \to Z$. Then

$$S(f) \circ \delta = S(\varepsilon) \circ S(g) \circ \delta = S(\varepsilon) \circ \delta \circ g = g. \qquad ❑$$

We turn to examples of Eilenberg–Moore categories. It may happen that quite a few details have to be checked to verify that a certain category is (isomorphic to) an Eilenberg–Moore category. We sketch only the essentials of the various constructions. In Theorem 6.7.11 we describe a general way to obtain Eilenberg–Moore categories via specifications (algebras or coalgebras with assertions). In general, one calls a category **algebraic** or **monadic** if it is isomorphic to a category of Eilenberg–Moore algebras.

Example 5.4.3 1. For a monoid $M = (M, 1, \cdot)$ there is an associated monad $M \times (-)\colon$ **Sets** $\to$ **Sets**; see Example 5.1.3.5 and Exercise 5.1.7. Recall that the unit is $\eta(x) = (1, x)$ and the multiplication $\mu(m, k, x) = (m \cdot k, x)$. An Eilenberg–Moore algebra $\alpha\colon M \times X \to X$ is given by a set X with this operation α. It corresponds to a **monoid action**, since the equations (5.18) amount to

$$\alpha(1, x) = x \qquad \alpha(m, \alpha(k, x)) = \alpha(m \cdot k, x).$$

Usually such an action is written as a dot, as in a scalar multiplication:

$$1 \bullet x = x \qquad m \bullet (k \bullet x) = (m \cdot k) \bullet x,$$

where $m \bullet x = \alpha(m, x)$. It is easy to see that homomorphisms of algebras, that is, maps in $\mathcal{EM}(M \times -)$, correspond to scalar-product-preserving functions f, satisfying $f(m \bullet x) = m \bullet f(x)$.

2. For a semiring S we have the multiset monad $\mathcal{M}_S$; see Lemma 5.1.5. An algebra $\alpha\colon \mathcal{M}_S(X) \to X$ forms a clever encoding of both a scalar

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

multiplication $\bullet$ and a commutative monoid structure $(0, +)$ on X. This structure can be extracted from α as follows:

$$0 = \alpha(0) \qquad x + y = \alpha(1|x\rangle + 1|y\rangle) \qquad s \bullet x = \alpha(s|x\rangle).$$

The element $0 \in \mathcal{M}_S(X)$ on the right in the first equation is the empty multiset, and $1|x\rangle = \eta(x) \in \mathcal{M}_S(X)$ is the singleton multiset.

These operations turn X into a **module** over the semiring S. As illustration, we check the zero law for the monoid structure. It involves translating the required equality into one of the algebra laws (5.18):

$$\begin{aligned} x + 0 = \alpha(1|x\rangle + 1|0\rangle) &\overset{(5.18)}{=} \alpha(1|\alpha(1|x\rangle)\rangle + 1|\alpha(0)\rangle) \\ &= \alpha(\mathcal{M}_S(\alpha)(1|1|x\rangle\rangle + 1|0\rangle)) \\ &= \alpha(\mathcal{M}_S(\alpha)(1|1|x\rangle\rangle + 1|0|0\rangle\rangle)) \\ &\overset{(5.18)}{=} \alpha(\mu(1|1|x\rangle\rangle + 1|0|0\rangle\rangle)) \\ &= \alpha((1 \cdot 1)|x\rangle + (1 \cdot 0)|0\rangle) = \alpha(1|x\rangle) = x. \end{aligned}$$

Conversely, if Y is a module over S, then we can define an algebra structure $\beta \colon \mathcal{M}_S(Y) \to Y$ by

$$\beta(s_1|y_1\rangle + \cdots + s_n|y_n\rangle) = s_1 \bullet y_1 + \cdots + s_n \bullet y_n.$$

Notice that the sum on the left is a formal sum (multiset) in $\mathcal{M}_S(Y)$, whereas the sum on the right is an actual sum, in the module Y. It is not hard to see that algebra maps correspond to module maps, preserving scalar multiplication and sums. Thus we have

$$\mathcal{EM}(\mathcal{M}_S) \cong \mathbf{Mod}_S,$$

where $\mathbf{Mod}_S$ is the category of modules over the semiring S.

Some special cases are worth mentioning:

$\mathbf{Mod}_{\mathbb{N}} \cong \mathbf{CMon}$	the category of commutative monoids
$\mathbf{Mod}_{\mathbb{Z}} \cong \mathbf{Ab}$	the category of commutative/Abelian groups
$\mathbf{Mod}_K \cong \mathbf{Vect}_K$	the category of vector spaces over a field K.

We see that monads and their algebras give a uniform description of these mathematical structures.

The distribution monad $\mathcal{D}$ is similar to the multiset monad, but its algebras are quite different. They are **convex sets**, where each formal *convex* combination $\sum_i r_i|x_i\rangle$, with $r_i \in [0, 1]$ satisfying $\sum_i r_i = 1$, has an actual sum; see [438, 441, 247].

3. Algebras $\alpha \colon \mathcal{P}(X) \to X$ of the powerset monad encode complete lattice structure on the set X. For an arbitrary subset $U \subseteq X$ one defines

$$\bigvee U = \alpha(U) \in X.$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

This is the join with respect to an order on X defined as $x \leq y$ iff $y = \bigvee\{x, y\} = \alpha(\{x, y\})$. This is a partial order. Reflexivity $x \leq x$ holds because $\alpha(\{x, x\}) = \alpha(\{x\}) = x$, since the singleton map is the unit of the powerset monad; see Example 5.1.3.1. Antisymmetry is trivial, and for transitivity one proceeds as follows. If $x \leq y$ and $y \leq z$, that is $\alpha(\{x, y\}) = y$ and $\alpha(\{y, z\}) = z$, then $x \leq z$ since

$$\begin{array}{rcl}
\alpha(\{x, z\}) & = & \alpha(\{\alpha(\{x\}), \alpha(\{y, z\})) \\
& = & \alpha(\mathcal{P}(\alpha)(\{\{x\}, \{y, z\}\})) \\
& \overset{(5.18)}{=} & \alpha(\bigcup\{\{x\}, \{y, z\}\}) \\
& = & \alpha(\{x, y, z\}) \\
& \overset{(5.18)}{=} & \alpha(\bigcup\{\{x, y\}, \{z\}\}) \\
& = & \alpha(\mathcal{P}(\alpha)(\{\{x, y\}, \{z\}\})) \\
& = & \alpha(\{\alpha(\{x, y\}), \alpha(\{z\})) \\
& = & \alpha(\{y, z\}) \\
& = & z.
\end{array}$$

Similarly one can prove $\forall x \in U.\, x \leq y$ iff $\bigvee U \leq y$.

Conversely, each complete lattice L forms a $\mathcal{P}$-algebra $\mathcal{P}(L) \to L$ via $U \mapsto \bigvee U$. Algebra maps correspond to $\bigvee$-preserving (also known as linear) maps. In this way we obtain isomorphisms

$$\mathcal{EM}(\mathcal{P}) \cong \mathbf{CL} \qquad \text{and} \qquad \mathcal{EM}(\mathcal{P}_{\text{fin}}) \cong \mathbf{JSL},$$

where **CL** is the category of complete lattices and linear maps, and **JSL** is the category of join semi-lattices and join-preserving maps. In such join semilattices one has only finite joins $(\bot, \vee)$. The situation for meet semilattices is described in Exercise 5.4.6.

4. On the category **PoSets** one can define the ideal monad Idl. An ideal in a poset $X = (X, \leq)$ is a directed downset $I \subseteq X$; thus, I is non-empty and satisfies:

- if $x \leq y \in I$, then $x \in I$
- if $y, y' \in I$, then there is an element $x \in I$ with $y \leq x$ and $y' \leq x$.

We write $\mathrm{Idl}(X)$ for the poset of ideals in X, ordered by inclusion. The mapping $X \mapsto \mathrm{Idl}(X)$ forms a monad on **PoSets**, with unit $\eta\colon X \to \mathrm{Idl}(X)$ given by $\eta(x) = \downarrow x = \{y \in X \mid y \leq x\}$. Union is multiplication. As in the previous point, algebras $\mathrm{Idl}(X) \to X$ for this monad are supremum maps. Since they are defined for directed (down)sets only, they turn the set X into a directed complete partial order (dcpo). Thus,

$$\mathcal{EM}(\mathrm{Idl}) \cong \mathbf{Dcpo}, \qquad \text{the category of dcpos and continuous maps.}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

5. Most of our examples of comonad coalgebras appear in the next chapter, as coalgebras satisfying certain assertions. We shall consider one example here. A coalgebra $X \to X^{\mathbb{N}}$ of the stream comonad from Example 5.1.10 can be identified with an endomap $f: X \to X$. The coalgebra is then given by $c(x) = \langle x, f(x), f^2(x), f^3(x), \ldots\rangle$. The reason is that such a coalgebra can be identified with a map of monoids $c': \mathbb{N} \to X^X$. Hence it is determined by the function $f = c'(1): X \to X$.

   More generally, for a monoid M we know that the functor $M \times (-)$ is a monad, by Example 5.1.3.5, and that $(-)^M$ is a comonad, by Exercise 5.1.11. In that case it can be shown that there is a bijective correspondence

$$\begin{array}{c} \text{coalgebras } X \longrightarrow X^M \\ \hline\hline \text{algebras } M \times X \longrightarrow X \\ \hline\hline \text{monoid maps } M \longrightarrow X^X \end{array}$$

We continue with some basic categorical facts. Eilenberg–Moore categories inherit limits (for algebras) and colimits (for coalgebras) from the underlying categories. This is the same as for functor (co)algebras.

Lemma 5.4.4 *The Eilenberg–Moore category $\mathcal{EM}(T)$ for a monad $T: \mathbb{C} \to \mathbb{C}$ has the same kind of limits as the underlying category $\mathbb{C}$. These limits are constructed elementwise, as in*

$$\left(\begin{array}{c} T(X) \\ \alpha\big\downarrow \\ X \end{array}\right) \times \left(\begin{array}{c} T(Y) \\ \big\downarrow\beta \\ Y \end{array}\right) = \left(\begin{array}{c} T(X \times Y) \\ \big\downarrow \langle \alpha\circ T(\pi_1), \beta\circ T(\pi_2)\rangle \\ X \times Y \end{array}\right).$$

The same holds for colimits in $\mathcal{EM}(S)$, for a comonad S. □

The following useful result due to Linton depends on some special properties of the category **Sets**. For a proof we refer to [56, §9.3, prop. 4]. It means that a category $\mathcal{EM}(T)$ of algebras for a monad T on **Sets** always has both limits and colimits.

Lemma 5.4.5 *For a monad T on* **Sets**, *the category $\mathcal{EM}(T)$ of algebras is cocomplete (and complete).* □

There are some more pleasant properties of such Eilenberg–Moore categories of monads on **Sets**. For instance, they always carry a logical factorisation system (see Exercise 5.4.3), given by injective and surjective algebra homomorphisms.

The structure of categories of Eilenberg–Moore coalgebras is investigated systematically in [282, 283].

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

In [111] it is shown that Eilenberg–Moore $\mathcal{EM}(T)$ and Kleisli categories $\mathcal{K}\ell(T)$ have biproducts if and only if the monad T is 'additive', i.e. maps coproducts to products, as in Exercise 2.1.10 for the (finite) powerset monad $\mathcal{P}$ and in Exercise 4.1.2 for the multiset monads $\mathcal{M}_S$.

We mention a related result, also without proof, for monads on **Sets**. It describes monoidal closed structure (see [344]) in categories of algebras, as in Proposition 5.2.13 for Kleisli categories. It explains why categories of abelian groups or vector spaces have tensors ⊗ and associated function spaces. The result can be generalised to other categories, provided suitable coequalisers of algebras exist.

Proposition 5.4.6 (From [302, 301]) *Let T be a commutative monad on* **Sets**. *The associated Eilenberg–Moore category $\mathcal{EM}(T)$ is then symmetric monoidal closed. The free functor $F\colon$* **Sets** $\rightarrow$ *$\mathcal{EM}(T)$ sends cartesian products* $(1,\times)$ *to monoidal products* $(I,\otimes)$, *in the sense that $F(1)\cong I$ is the tensor unit, and $F(X\times Y)\cong F(X)\otimes F(Y)$.* ❑

Let $T(X)\xrightarrow{\alpha} X$, $T(Y)\xrightarrow{\beta} Y$ and $T(Z)\xrightarrow{\gamma} Z$ be algebras. The tensors ⊗ for algebras are constructed in such a way that there is a bijective correspondence:

$$\begin{array}{c}\text{homomorphisms of algebras }\ \alpha\otimes\beta \longrightarrow \gamma\\ \hline\hline \text{bihomomorphisms }\ X\times Y \longrightarrow Z\end{array}\,.$$

Intuitively, a bihomomorphism $f\colon X\times Y\rightarrow Z$ is a homomorphism in each coordinate separately: $f(x,-)\colon Y\rightarrow Z$ and $f(-,y)\colon X\rightarrow Z$ are algebra maps, for each $x\in X$ and $y\in Y$. More formally this is described via a commuting diagram, involving the 'double strength' map dst of a commutative monad (see Definition 5.2.9):

$$\begin{tikzcd}
T(X)\times T(Y) \arrow[r, "\mathrm{dst}"] \arrow[d, "\alpha\times\beta"'] & T(X\times Y) \arrow[r, "T(f)"] & T(Z) \arrow[d, "\gamma"] \\
X\times Y \arrow[rr, "f"'] & & Z
\end{tikzcd}$$

Under suitable circumstances one can construct for a functor F the free monad or the cofree comonad on F. The Eilenberg–Moore categories of these (co)free extensions turn out to be the same as the categories of (co)algebras of the original functor F. This is the content of the next result.

Proposition 5.4.7 *Let $F\colon\mathbb{C}\rightarrow\mathbb{C}$ be an arbitrary functor. Recall the free monad $F^{*}\colon\mathbb{C}\rightarrow\mathbb{C}$ on F from Proposition 5.1.8, and the cofree comonad the*

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

cofree comonad $F^\infty \colon \mathbb{C} \to \mathbb{C}$ on F from Exercise 5.1.15, described via the initial algebras and final coalgebras:

$$X + F(F^*(X)) \xrightarrow[\cong]{\alpha_X} F^*(X) \qquad \textit{and} \qquad F^\infty(X) \xrightarrow[\cong]{\zeta_X} X \times F(F^\infty(X)).$$

Assuming these constructions exist in $\mathbb{C}$, there are isomorphisms of categories of monad (co)algebras and functor (co)algebras:

$$\mathcal{EM}(F^*) \cong \mathbf{Alg}(F) \qquad \textit{and} \qquad \mathcal{EM}(F^\infty) \cong \mathbf{CoAlg}(F).$$

Proof We do the proof in the coalgebra case only. We first list the relevant structure. The counit $\varepsilon_X \colon F^\infty(X) \to X$ is given by $\varepsilon_X = \pi_1 \circ \zeta_X$. The comultiplication $\delta_X \colon F^\infty(X) \to F^\infty F^\infty(X)$ is obtained by finality in

$$\begin{tikzcd}
F^\infty(X) \times F(F^\infty(X)) \arrow[r, dashed, "\mathrm{id} \times F(\delta_X)"] & F^\infty(X) \times F(F^\infty F^\infty(X)) \\
F^\infty(X) \arrow[u, "\langle \mathrm{id}, \pi_2 \circ \zeta_X \rangle"] \arrow[r, dashed, "\delta_X"'] & F^\infty F^\infty(X) \arrow[u, "\cong"', "\zeta_{F^\infty(X)}"]
\end{tikzcd}$$

There is a universal natural transformation $\theta \colon F^\infty \Rightarrow F$ by

$$\theta_X = \Big(F^\infty(X) \xrightarrow[\cong]{\zeta_X} X \times F(F^\infty(X)) \xrightarrow{\pi_2} F(F^\infty(X)) \xrightarrow{F(\varepsilon_X)} F(X) \Big).$$

We now define two functors:

$$\begin{tikzcd}
\mathcal{EM}(F^\infty) \arrow[r, shift left, "L"] & \mathbf{CoAlg}(F). \arrow[l, shift left, "K"]
\end{tikzcd}$$

The functor L is easy: it takes a comonad coalgebra $\beta \colon X \to F^\infty(X)$ and sends it to $\theta_X \circ \beta \colon X \to F(X)$. Naturality of θ makes L a functor.

In the other direction, the functor K sends a functor coalgebra $c \colon X \to F(X)$ to the map $K(c)$ defined by finality in

$$\begin{tikzcd}
X \times F(X) \arrow[r, dashed, "\mathrm{id} \times F(K(c))"] & X \times F(F^\infty(X)) \\
X \arrow[u, "\langle \mathrm{id}, c \rangle"] \arrow[r, dashed, "K(c)"'] & F^\infty(X) \arrow[u, "\cong"', "\zeta_X"]
\end{tikzcd}$$

Then

$$\begin{aligned}
\varepsilon_X \circ K(c) \;=\; \pi_1 \circ \zeta_X \circ K(c) &\;=\; \pi_1 \circ (\mathrm{id} \times F(K(c))) \circ \langle \mathrm{id}, c \rangle \\
&\;=\; \pi_1 \circ \langle \mathrm{id}, c \rangle \;=\; \mathrm{id}.
\end{aligned}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Using uniqueness of maps to final coalgebras one can prove that $\delta \circ K(c) = F^{\infty}(K(c)) \circ K(c)$. Hence $K(c)\colon X \to F^{\infty}(X)$ is a comonad coalgebra. Also by uniqueness one can show that maps of functor coalgebras are also maps of monad coalgebras.

We have $LK = \mathrm{id}$ since:

$$\begin{aligned} LK(c) \;=\; \theta_X \circ K(c) &= F(\varepsilon_X) \circ \pi_2 \circ \zeta_X \circ K(c) \\ &= F(\varepsilon_X) \circ \pi_2 \circ (\mathrm{id} \times F(K(c))) \circ \langle \mathrm{id}, c\rangle \\ &= F(\varepsilon_X) \circ F(K(c)) \circ \pi_2 \circ \langle \mathrm{id}, c\rangle \\ &= F(\mathrm{id}) \circ c \\ &= c. \end{aligned}$$

We get $KL(\beta) = \beta$ because $\beta\colon X \to F^{\infty}(X)$ is a map of coalgebras in the diagram defining $KL(\beta)$, namely:

$$\begin{tikzcd}
X \times F(X) \arrow[r, "\mathrm{id} \times F(\beta)"] & X \times F(F^{\infty}(X)) \\
X \arrow[u, "{\langle \mathrm{id}, L(\beta)\rangle}"] \arrow[r, "\beta"'] & F^{\infty}(X) \arrow[u, "\cong"', "\zeta_X"]
\end{tikzcd}$$

This diagram commutes since

$$\begin{aligned} (\mathrm{id} \times F(\beta)) \circ \langle \mathrm{id}, L(\beta)\rangle &= \langle \mathrm{id}, F(\beta) \circ \theta \circ \beta\rangle \\ &= \langle \mathrm{id}, \theta \circ F^{*}(\beta) \circ \beta\rangle \\ &= \langle \mathrm{id}, \theta \circ \delta \circ \beta\rangle \\ &= \langle \mathrm{id}, F(\varepsilon) \circ \pi_2 \circ \zeta \circ \delta \circ \beta\rangle \\ &= \langle \mathrm{id}, F(\varepsilon) \circ \pi_2 \circ (\mathrm{id} \times F(\delta)) \circ \langle \mathrm{id}, \pi_2 \circ \zeta\rangle \circ \beta\rangle \\ &= \langle \varepsilon \circ \beta, F(\varepsilon) \circ F(\delta) \circ \pi_2 \circ \langle \mathrm{id}, \pi_2 \circ \zeta\rangle \circ \beta\rangle \\ &= \langle \pi_1 \circ \zeta \circ \beta, \pi_2 \circ \zeta \circ \beta\rangle \\ &= \zeta \circ \beta. \end{aligned}$$

❑

In Definition 5.2.4 we have seen distributive '$\mathcal{K}\ell$' laws $FT \Rightarrow TF$ that correspond to liftings $\mathcal{K}\ell(F)\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ of the functor $F\colon \mathbb{C} \to \mathbb{C}$ to Kleisli categories of a monad T. There is an analogous result for Eilenberg–Moore categories.

Definition 5.4.8 Let $T\colon \mathbb{C} \to \mathbb{C}$ be a monad and $G\colon \mathbb{C} \to \mathbb{C}$ an ordinary functor. A **distributive law** or an **$\mathcal{EM}$-law** of T over G is a natural

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

transformation $\rho\colon TG \Rightarrow GT$ that commutes appropriately with the unit and multiplication of the monad T, as in

$$\begin{tikzcd} G(X) \arrow[r, equal] \arrow[d, "\eta_{GX}"'] & G(X) \arrow[d, "G(\eta_X)"] \\ TG(X) \arrow[r, "\rho_X"'] & GT(X) \end{tikzcd} \qquad \begin{tikzcd} T^2G(X) \arrow[r, "T(\rho_X)"] \arrow[d, "\mu_{GX}"'] & TGT(X) \arrow[r, "\rho_{TX}"] & GT^2(X) \arrow[d, "G(\mu_X)"] \\ TG(X) \arrow[rr, "\rho_X"'] & & GT(X) \end{tikzcd} \tag{5.20}$$

We now show that these $\mathcal{EM}$-laws correspond to liftings to Eilenberg–Moore categories. We repeat the lifting result for Kleisli categories from Proposition 5.2.5 in order to get a clear picture of the whole situation (following [272]). A more abstract account in terms of 2-categories may be found in [333].

Proposition 5.4.9 (‘Laws and liftings’) *Let T be a monad and F, G be endofunctors on the same category $\mathbb{C}$, as above. There are bijective correspondences between $\mathcal{K}\ell/\mathcal{EM}$-laws and liftings of F to $\mathcal{K}\ell/\mathcal{EM}$-categories, in*

$$\frac{\mathcal{K}\ell\text{-law } FT \overset{\lambda}{\Longrightarrow} TF}{\begin{tikzcd} \mathcal{K}\ell(T) \arrow[r, "L"] \arrow[d] & \mathcal{K}\ell(T) \arrow[d] \\ \mathbb{C} \arrow[r, "F"] & \mathbb{C} \end{tikzcd}} \qquad \frac{\mathcal{EM}\text{-law } TG \overset{\rho}{\Longrightarrow} GT}{\begin{tikzcd} \mathcal{EM}(T) \arrow[r, "R"] \arrow[d] & \mathcal{EM}(T) \arrow[d] \\ \mathbb{C} \arrow[r, "G"] & \mathbb{C} \end{tikzcd}}$$

We standardly write $\mathcal{K}\ell(F) = L$ and $\mathcal{EM}(G) = R$ for these liftings, if confusion is unlikely. Thus we leave the laws λ and ρ implicit.

Proof The proof of first part about $\mathcal{K}\ell$-liftings has already been given in Proposition 5.2.5, so we concentrate on the second part. Thus, assume we have an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$. It gives rise to a functor $R\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$ by

$$\begin{pmatrix} T(X) \\ \downarrow\alpha \\ X \end{pmatrix} \longmapsto \begin{pmatrix} T(X) \\ \downarrow G(\alpha)\circ\rho \\ X \end{pmatrix} \qquad \text{and} \qquad f \longmapsto G(f).$$

Equations (5.20) guarantee that this yields a new T-algebra.

In the reverse direction, let $R\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$ be a lifting of G. Applying R to the multiplication μ_X yields an algebra $R(\mu_X)\colon TGT(X) \to GT(X)$. We then define $\rho_X = R(\mu_X) \circ TG(\eta_X)\colon TG(X) \to GT(X)$. Remaining details are left to the reader. ❑

The previous section illustrated how distributive $\mathcal{K}\ell$-laws are used to obtain final coalgebras in Kleisli categories. This requires non-trivial side-conditions, such as enrichment in dcpos; see Theorem 5.3.4. For $\mathcal{EM}$-laws the situation is much easier; see below. Instances of this result have been studied in [427]; see also [51].

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Proposition 5.4.10 *Let T be a monad and G an endofunctor on a category $\mathbb{C}$, with an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$ between them. If G has a final coalgebra $\zeta\colon Z \xrightarrow{\cong} G(Z)$ in $\mathbb{C}$, then Z carries an Eilenberg–Moore algebra structure obtained by finality, as in*

$$\begin{tikzcd}
GT(Z) \arrow[r, dashed, "G(\beta)"] & G(Z) \\
T(Z) \arrow[u, "\rho \circ T(\zeta)"] \arrow[r, dashed, "\beta"'] & Z \arrow[u, "\cong"', "\zeta"]
\end{tikzcd} \tag{5.21}$$

The map ζ then forms a map of algebras as below, which is the final coalgebra for the lifted functor $\mathcal{EM}(G)\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$:

$$\begin{pmatrix} T(Z) \\ \downarrow\beta \\ Z \end{pmatrix} \xrightarrow[\cong]{\zeta} \mathcal{EM}(G)\begin{pmatrix} T(Z) \\ \downarrow\beta \\ Z \end{pmatrix} = \begin{pmatrix} TG(Z) \\ \downarrow G(\beta)\circ\rho \\ G(Z) \end{pmatrix}.$$

Proof We leave it to the reader to verify that the map β defined in (5.21) is a T-algebra. By construction of β, the map ζ is a homomorphism of algebras $\beta \to \mathcal{EM}(G)(\beta)$. Suppose for an arbitrary algebra $\gamma\colon T(Y) \to Y$ we have a $\mathcal{EM}(G)$-coalgebra $c\colon \gamma \to \mathcal{EM}(G)(\gamma)$. Then $c\colon Y \to G(Y)$ satisfies $G(\gamma) \circ \rho \circ T(c) = c \circ \gamma$. By finality in **CoAlg**(G) there is a unique map $f\colon Y \to Z$ with $\zeta \circ f = G(f) \circ c$. This f is then the unique map $\gamma \to \beta$ in $\mathcal{EM}(T)$. ❑

In Section 5.5 we shall see that this result describes not only a final coalgebra in a category of algebras but a final bialgebra.

These $\mathcal{EM}$-laws are used for 'state extension'. This involves turning a coalgebra of the form $X \to GT(X)$ into a coalgebra in the category of Eilenberg–Moore algebras, with the free algebra $T(X)$ as state space. This state extension is functorial, in the following manner.

Lemma 5.4.11 *Given an $\mathcal{EM}$-law of the form $TG \Rightarrow GT$, the free algebra functor $\mathcal{F}\colon \mathbb{C} \to \mathcal{EM}(T)$ can be lifted, as in*

$$\begin{tikzcd}
\mathbf{CoAlg}(GT) \arrow[r, "\mathcal{F}_{\mathcal{EM}}"] \arrow[d] & \mathbf{CoAlg}(\mathcal{EM}(G)) \arrow[d] \\
\mathbb{C} \arrow[loop left, "G"] \arrow[loop below, "T"] \arrow[r, "\mathcal{F}"] & \mathcal{EM}(T) \arrow[loop right, "\mathcal{EM}(G)"]
\end{tikzcd}$$

The functor $\mathcal{F}_{\mathcal{EM}}\colon$ **CoAlg**$(GT) \to$ **CoAlg**$(\mathcal{EM}(G))$ gives an abstract description of what is called the generalised powerset construction in [429]. A similar functor exists for Kleisli categories; see Exercise 5.4.13 below.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Proof Let $\rho\colon TG \Rightarrow GT$ be an $\mathcal{EM}$-law. The functor $\mathcal{F}_{\mathcal{EM}}\colon \mathbf{CoAlg}(GT) \to \mathbf{CoAlg}(\mathcal{EM}(G))$ is defined by

$$\mathcal{F}_{\mathcal{EM}}\big(X \xrightarrow{c} GT(X)\big) = \big(T(X) \xrightarrow{T(c)} TGT(X) \xrightarrow{\rho_{T(X)}} GT^2(X) \xrightarrow{G(\mu)} GT(X)\big). \tag{5.22}$$

This $\mathcal{F}_{\mathcal{EM}}(c)$ is a well-defined coalgebra $\mu_X \to \mathcal{EM}(G)(\mu_X)$ on the free algebra $\mathcal{F}(X) = \mu_X$ since it is a map of algebras:

$$\begin{aligned}
\mathcal{EM}(G)(\mu_X) \circ T(\mathcal{F}_{\mathcal{EM}}(c)) &= G(\mu) \circ \rho \circ TG(\mu) \circ T(\rho) \circ T^2(c)\\
&= G(\mu) \circ GT(\mu) \circ \rho \circ T(\rho) \circ T^2(c)\\
&= G(\mu) \circ G(\mu) \circ \rho \circ T(\rho) \circ T^2(c)\\
&= G(\mu) \circ \rho \circ \mu \circ T^2(c)\\
&= G(\mu) \circ \rho \circ T(c) \circ \mu\\
&= \mathcal{F}_{\mathcal{EM}}(c) \circ \mu.
\end{aligned}$$

On morphisms one simply has $\mathcal{F}_{\mathcal{EM}}(f) = T(f)$.

Further, if $f\colon X \to Y$ is a map of GT-algebras, from $c\colon X \to GTX$ to $d\colon Y \to GTY$, then $\mathcal{F}_{\mathcal{EM}}(c)(f) = T(f)$ is obviously a map of algebras $\mu_X \to \mu_Y$ and also a map of $\mathcal{EM}(G)$-coalgebras:

$$\begin{aligned}
\mathcal{EM}(G)(\mathcal{F}_{\mathcal{EM}}(f)) \circ \mathcal{F}_{\mathcal{EM}}(c) &= GT(f) \circ G(\mu) \circ \rho \circ T(c)\\
&= G(\mu) \circ GT^2(f) \circ \rho \circ T(c)\\
&= G(\mu) \circ \rho \circ TGT(f) \circ T(c)\\
&= G(\mu) \circ \rho \circ T(d) \circ T(f)\\
&= \mathcal{F}_{\mathcal{EM}}(d) \circ \mathcal{F}_{\mathcal{EM}}(f).
\end{aligned}$$

❑

Example 5.4.12 1. A non-deterministic automaton can be described as a coalgebra $\langle \delta, \epsilon\rangle\colon X \to (\mathcal{P}X)^A \times 2$, which is of the form $X \to GT(X)$, where G is the functor $(-)^A \times 2$ and T is the powerset monad $\mathcal{P}$ on **Sets**. Since $2 = \{0, 1\}$ is the (carrier of the) free algebra $\mathcal{P}(1)$ on the singleton set $1 = \{*\}$, there is an $\mathcal{EM}$-law $TG \Rightarrow GT$ by Exercise 5.4.4. It is $\rho = \langle \rho_1, \rho_2\rangle\colon \mathcal{P}(2 \times X^A) \to \mathcal{P}(X)^A \times 2$, given by

$$\begin{cases} x \in \rho_1(U)(a) \iff \exists \langle b, h\rangle \in U.\, h(a) = x\\ \quad \rho_1(U) = 1 \iff \exists h \in X^A.\, \langle 1, h\rangle \in U. \end{cases}$$

Lemma 5.4.11 now yields a coalgebra $\mathcal{F}_{\mathcal{EM}}(\langle \delta, \epsilon\rangle) = \langle \delta_{\mathcal{EM}}, \epsilon_{\mathcal{EM}}\rangle\colon \mathcal{P}(X) \to \mathcal{P}(X)^A \times 2$ in the category $\mathcal{EM}(\mathcal{P}) \cong$ **CL** of complete lattices; see Example 5.4.3.3. This coalgebra is given by

$$\begin{cases} \delta_{\mathcal{EM}}(U)(a) = \bigcup_{x\in U} \delta(x)(a)\\ \epsilon_{\mathcal{EM}}(U) = 1 \Leftrightarrow \exists x \in U.\, \epsilon(x) = 1. \end{cases}$$

By Proposition 5.4.10 the final G-coalgebra $2^{A^\star} = \mathcal{P}(A^\star)$ of languages is also final for the lifted functor $\mathcal{EM}(G)$ on **CL** $\cong \mathcal{EM}(\mathcal{P})$. Hence we get a

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

map $\mathcal{P}(X) \to \mathcal{P}(A^\star)$ of $\mathcal{EM}(G)$-coalgebras by finality. Applying this map to the singleton set $\{x\} \in \mathcal{P}(X)$ yields the set of words that are accepted in the state $x \in X$. This yields the trace semantics for a non-deterministic automaton $X \to \mathcal{P}(X)^A \times 2$, via determinisation in the Eilenberg–Moore category. This was first described in [427, 429] and elaborated in terms of $\mathcal{EM}$-laws in [271, 272].

2. Let S be a semiring and $\mathcal{M}_S$ the associated multiset monad on **Sets**, with as Eilenberg–Moore category $\mathcal{EM}(\mathcal{M}_S) = \mathbf{Mod}_S$ the category of modules over S; see Example 5.4.3.2. Consider for a set A of labels the functor $G(X) = X^A \times S$ on **Sets**, with final coalgebra $S^{A^\star}$; see Proposition 2.3.5. We claim that this functor G can be lifted to $\mathcal{EM}(G)$ on $\mathbf{Mod}_S$ via the $\mathcal{EM}$-law $\mathcal{M}_S G \Rightarrow G\mathcal{M}_S$ of the form

$$\begin{array}{ccc} \mathcal{M}_S(X^A \times S) & \longrightarrow & \mathcal{M}_S(X)^A \times S \\ \sum_i s_i |\, f_i, t_i \rangle & \longmapsto & \langle \lambda a.\ \sum_i s_i |\, f_i(a) \rangle,\ \sum_i s_i \cdot t_i \rangle \end{array}$$

The final coalgebra $S^{A^\star}$ in **Sets** is then also final in the Eilenberg–Moore category $\mathbf{Mod}_S$, with obvious (pointwise) module structure; see Proposition 5.4.10. It is final for the lifted functor $(-)^A \times S$ on $\mathbf{Mod}_S$.

Trace semantics for a weighted automaton $c\colon X \to \mathcal{M}_S(X)^A \times S$ is now obtained as follows. The codomain $\mathcal{M}_S(X)^A \times S$ is a module over S. Hence we can freely extend the coalgebra c to a map $\overline{c}\colon \mathcal{M}_S(X) \to \mathcal{M}_S(X)^A \times S$ in $\mathbf{Mod}_S$. This $\overline{c}$ is a coalgebra of the lifted endofunctor $(-)^A \times S$ on $\mathbf{Mod}_S$. Hence there is unique map of coalgebras $\mathrm{beh}_{\overline{c}}\colon \mathcal{M}_S(X) \to S^{A^\star}$. By precomposition with the unit we get the trace map $X \to S^{A^\star}$.

3. Recall from Figure 4.2 that a ‘simple Segala’ system is a coalgebra of the form $X \to \mathcal{P}(A \times \mathcal{D}(X))$, combining non-deterministic and probabilistic computation. It can be turned into a non-deterministic transition system $\mathcal{D}(X) \to \mathcal{P}(A \times \mathcal{D}(X))$ with distributions as states. As before this is done via a $\mathcal{EM}$-law, namely of the form $\mathcal{DP}(A \times -) \Rightarrow \mathcal{P}(A \times -)\mathcal{D}$, of the monad $\mathcal{D}$ over the functor $\mathcal{P}(A \times -)$.

In [272] it is shown that such $\mathcal{EM}$-laws exist more generally, as soon as we have a map of monads. More concretely, each map of monads $\sigma\colon T \Rightarrow S$ induces an $\mathcal{EM}$-law

$$TS(A \times -) \xRightarrow{\ \rho\ } S(A \times T(-))$$

of the *monad* T over the *functor* $S(A \times -)$. The components of ρ are given by

$$\rho_X = \Big(TS(A \times X) \xrightarrow{\ \sigma\ } S^2(A \times X) \xrightarrow{S^2(\mathrm{id} \times \eta)} S^2(A \times TX) \xrightarrow{\ \mu\ } S(A \times TX) \Big).$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

Verifications are left to the interested reader. For simple Segala systems we can apply this general construction with $T = \mathcal{D}$ and $S = \mathcal{P}$, and the support map of monads $\mathcal{D} \Rightarrow \mathcal{P}$ from Exercise 5.1.9.

We conclude with another result about the free monads F^* on a functor F from Proposition 5.1.8.

Lemma 5.4.13 *For two endofunctors $F, G\colon \mathbb{C} \to \mathbb{C}$, there is a bijective correspondence*

$$\begin{array}{c} \textit{distributive } \mathcal{EM}\textit{-laws} \quad F^*G \overset{\rho}{\Longrightarrow} GF^* \\ \hline\hline \textit{natural transformations} \quad FG \underset{\tau}{\Longrightarrow} GF^* \end{array}$$

where F^* is the free monad from Proposition 5.1.8.

Proof The correspondence is given as follows.

- Starting from an $\mathcal{EM}$-law $\rho\colon F^*G \Rightarrow GF^*$ we take $\overline{\rho}_X = \rho_X \circ \theta_{GX}\colon FG(X) \to F^*G(X) \to GF^*(X)$, where $\theta\colon F \Rightarrow F^*$ is the universal map.
- Conversely, given a natural transformation $\tau\colon FG \Rightarrow GF^*$ we obtain a map $\overline{\tau}_X\colon F^*G(X) \to GF^*(X)$ by initiality in

$$\begin{tikzcd} G(X) + F(F^*G(X)) \arrow[r, dashed, "\mathrm{id} + F(\overline{\tau}_X)"] \arrow[d, "\alpha_{G(X)}"', "\cong"] & G(X) + F(GF^*(X)) \arrow[d, "{[G(\eta), G(\mu) \circ \tau]}"] \\ F^*G(X) \arrow[r, dashed, "\overline{\tau}_X"'] & GF^*(X) \end{tikzcd} \tag{5.23}$$

We leave it to the reader to verify that this map $\overline{\tau}$ is natural and commutes with the F^*'s unit and multiplication, as required in (5.20).

Proving that $\overline{\overline{\rho}} = \rho$ and $\overline{\overline{\tau}} = \tau$ is elementary. ❑

Exercises

5.4.1 Prove that the product of Eilenberg–Moore algebras, as described in Lemma 5.4.4, forms again an Eilenberg–Moore algebra. Formulate and prove a dual result for coalgebras.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

5.4.2 Let $\alpha\colon T(X) \to X$ be an algebra of a monad T. Prove that α is a coequaliser in the category $\mathcal{EM}(T)$ of algebras, in a diagram:

$$\begin{tikzcd}
\left(\begin{array}{c} T^3(X) \\ \downarrow{\scriptstyle \mu} \\ T^2(X) \end{array}\right) \arrow[r, shift left, "T(\alpha)"] \arrow[r, shift right, "\mu"'] & \left(\begin{array}{c} T^2(X) \\ \downarrow{\scriptstyle \mu} \\ T(X) \end{array}\right) \arrow[r, two heads, "\alpha"] & \left(\begin{array}{c} T(X) \\ \downarrow{\scriptstyle \alpha} \\ X \end{array}\right)
\end{tikzcd}.$$

5.4.3 Let T be a monad on a category $\mathbb{C}$ with a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$. Prove, as in Exercise 4.3.2, that if T preserves abstract epis, $\mathcal{EM}(T)$ also carries a logical factorisation system. Check that this is the case for monads on **Sets**, using the axiom of choice; see text before Lemma 2.1.7.

5.4.4 Let $T\colon \mathbb{C} \to \mathbb{C}$ be an arbitrary monad. Show that for the various functors $G\colon \mathbb{C} \to \mathbb{C}$ described below an $\mathcal{EM}$-law $TG \Rightarrow GT$ exists.

1. If G is the identity functor $\mathbb{C} \to \mathbb{C}$.
2. If $G = B$, the constant functor mapping $X \mapsto B$, where B carries an algebra $\beta\colon T(B) \to B$.
3. If $G = G_1 \times G_2$ and there are $\mathcal{EM}$-laws $\rho_i\colon TG_i \Rightarrow G_iT$.
4. If $G = H^A$, provided: $\mathbb{C}$ is cartesian closed, there is an $\mathcal{EM}$-law $TH \Rightarrow HT$, and T is a strong monad.
5. If $G = G_1 \circ G_2$ and there are $\mathcal{EM}$-laws $\rho_i\colon TG_i \Rightarrow G_iT$.
6. Assuming that T preserves coproducts we also have: if $G = \coprod_i G_i$ and there are $\mathcal{EM}$-laws $\rho_i\colon TG_i \Rightarrow G_iT$.

5.4.5 For each set X, the function space $[0,1]^X$ is the set of fuzzy predicates.

1. Check that $[0,1]^X$ is a convex set.
2. Define an algebra $\mathcal{D}([0,1]^X) \to [0,1]^X$ of the distribution monad $\mathcal{D}$.

5.4.6 Let **MSL** be the category of meet semilattices.

1. Prove that the forgetful functor **MSL** $\to$ **Sets** has a left adjoint $X \mapsto \mathcal{P}_{\text{fin}}(X)^{\text{op}}$, where $\mathcal{P}_{\text{fin}}(X)^{\text{op}}$ is the poset of finite subsets of X, with reverse inclusion $U \supseteq V$ as order.
2. Describe the unit and multiplication of the monad $\mathcal{P}_{\text{fin}}^{\text{op}}\colon$ **Sets** $\to$ **Sets** induced by this adjunction.
3. Prove **MSL** $\cong \mathcal{EM}(\mathcal{P}_{\text{fin}}^{\text{op}})$.

5.4.7 1. Prove in detail that taking ideals leads to a monad $\mathrm{Idl}\colon$ **PoSets** $\to$ **PoSets**, as claimed in Example 5.4.3.4, and also that $\mathcal{EM}(\mathrm{Idl}) \cong$ **Dcpo**.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

2. Define along the same lines a downset monad Dwn on **PoSets**, which sends a poset to the set of all its downclosed subsets, ordered by inclusion.
3. Prove that $\mathcal{EM}(\mathrm{Dwn}) \cong \mathbf{CL}$.

5.4.8 1. For an arbitrary monad T on a category $\mathbb{C}$, the free algebra adjunction $\mathcal{EM}(T) \leftrightarrows \mathbb{C}$ induces a comonad $\widehat{T}\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$. Describe the counit and comultiplication of this comonad in detail.
2. Write $\widehat{\mathrm{Idl}}\colon \mathbf{Dcpo} \to \mathbf{Dcpo}$ for the comonad induced in this way by $\mathbf{Dcpo} \leftrightarrows \mathbf{PoSets}$; see the previous exercise. Prove that the category $\mathcal{EM}(\widehat{\mathrm{Idl}})$ of Eilenberg–Moore coalgebras of this comonad is the category of *continuous* posets.
3. Write $\mathcal{M} = \mathcal{M}_{\mathbb{R}}$ for the multiset monad over the real numbers $\mathbb{R}$, with vector spaces over $\mathbb{R}$ as algebras; see Example 5.4.3.2. Prove that a coalgebra of the induced comonad $\widehat{\mathcal{M}}$ on a vector space corresponds to a basis for this space.

(For a more systematic study of coalgebras of the induced comonad $\widehat{T}$, see [251] and the references therein.)

5.4.9 Consider the distributive $\mathcal{K}\ell$-law $\nabla\colon (-)^{\star}\mathcal{P} \Rightarrow \mathcal{P}(-)^{\star}$ described in Lemma 5.2.7, for the list functor $(-)^{\star}$.

1. Prove that this ∇ is in fact a distributive law between two monads, as in Exercise 5.2.8.
2. As a result, there is a language monad $X \mapsto \mathcal{P}(X^{\star})$; describe its unit and multiplication in detail.
3. Verify that the Eilenberg–Moore algebras of this monad $\mathcal{P}((-)^{\star})$ are complete lattices with a monoid structure where multiplication preserves joins in both arguments separately. They are known as unital quantales (see [401]) or as Kleene algebras with arbitrary joins.

5.4.10 For a semiring S consider the category $\mathbf{Mod}_S = \mathcal{EM}(\mathcal{M}_S)$ of modules over S, as in Example 5.4.3.2.

1. Show that the (covariant) powerset functor can be lifted to modules as in

$$\begin{tikzcd}
\mathbf{Mod}_S \arrow[r, "\mathcal{P}"] \arrow[d] & \mathbf{Mod}_S \arrow[d] \\
\mathbf{Sets} \arrow[r, "\mathcal{P}"] & \mathbf{Sets}
\end{tikzcd}$$

and similarly for the finite powerset functor $\mathcal{P}_{\mathrm{fin}}$.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

2. Check that $\mathcal{P}\colon \mathbf{Mod}_S \to \mathbf{Mod}_S$ is also a monad –with unit and multiplication as on **Sets**.
3. Describe the $\mathcal{EM}$-law $\mathcal{M}_S\mathcal{P} \Rightarrow \mathcal{P}\mathcal{M}_S$ corresponding to this lifting $\mathcal{P}\colon \mathbf{Mod}_S \to \mathbf{Mod}_S$, as in Proposition 5.4.9.

5.4.11 Let $T\colon \mathbf{Sets} \to \mathbf{Sets}$ be a monad, and $\alpha\colon T(A) \to A$ an arbitrary (but fixed) Eilenberg–Moore algebra. Show that there is an adjunction

$$\begin{tikzcd}
\mathbf{Sets}^{\mathrm{op}} \arrow[r, bend left, "A^{(-)}"] \arrow[r, phantom, "\top"] & \mathcal{EM}(T). \arrow[l, bend left, "Hom(-{,}\,\alpha)"]
\end{tikzcd}$$

The canonical choice is to take $A = T(1)$, the free monad on the singleton set 1. Elaborate on what the resulting adjoint functors are in the cases $\mathbf{Sets}^{\mathrm{op}} \leftrightarrows \mathbf{JSL}$ and $\mathbf{Sets}^{\mathrm{op}} \leftrightarrows \mathbf{Mod}_S$, involving the finite powerset functor $\mathcal{P}_{\mathrm{fin}}$ and the multiset functor $\mathcal{M}_S$. This result plays a role in the abstract description of modalities; see [201, 221].

5.4.12 (From [51, 272]) Let $\lambda\colon FT \Rightarrow TF$ be a $\mathcal{K}\ell$-law and $\rho\colon TG \Rightarrow GT$ an $\mathcal{EM}$-law. Show that the standard adjunctions $\mathbb{C} \rightleftarrows \mathcal{K}\ell(T)$ from Proposition 5.2.2 and $\mathbb{C} \rightleftarrows \mathcal{EM}(T)$ from Lemma 5.4.2 lift to adjunctions between categories of, respectively, (functor) algebras and coalgebras, as described below:

$$\begin{tikzcd}
\mathbf{Alg}(F) \arrow[r, bend left, "\mathrm{Alg}(\mathcal{F})"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathbf{Alg}(\mathcal{K}\ell(F)) \arrow[l, bend left, "\mathrm{Alg}(\mathcal{U})"] \arrow[d] \\
\mathbb{C} \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] \arrow[loop below, "F"] & \mathcal{K}\ell(T) \arrow[l, bend left, "\mathcal{U}"] \arrow[loop below, "\mathcal{K}\ell(F)"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{CoAlg}(G) \arrow[r, bend left, "\mathrm{CoAlg}(\mathcal{F})"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathbf{CoAlg}(\mathcal{EM}(G)) \arrow[l, bend left, "\mathrm{CoAlg}(\mathcal{U})"] \arrow[d] \\
\mathbb{C} \arrow[r, bend left, "\mathcal{F}"] \arrow[r, phantom, "\perp"] \arrow[loop below, "G"] & \mathcal{EM}(T) \arrow[l, bend left, "\mathcal{U}"] \arrow[loop below, "\mathcal{EM}(G)"]
\end{tikzcd}$$

Hint: Use Theorem 2.5.9 and its dual.

5.4.13 Prove, in analogy with Lemma 5.4.11, that in presence of a $\mathcal{K}\ell$-law $FT \Rightarrow TF$ the free functor $\mathcal{F}\colon \mathbb{C} \to \mathcal{K}\ell(T)$ can be lifted, as in

$$\begin{tikzcd}
\mathbf{CoAlg}(TF) \arrow[r, "\mathcal{F}_{\mathcal{K}\ell}"] \arrow[d] & \mathbf{CoAlg}(\mathcal{K}\ell(F)) \arrow[d] \\
\mathbb{C} \arrow[r, "\mathcal{F}"] \arrow[loop left, "F"] \arrow[loop below, "T"] & \mathcal{K}\ell(T) \arrow[loop below, "\mathcal{K}\ell(F)"]
\end{tikzcd}$$

5.4.14 Recall that for two sets A, B, the deterministic automaton functor $G(X) = X^A \times B$ has final coalgebra $B^{A^\star}$; see Proposition 2.3.5. Let

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

T: **Sets** → **Sets** be an arbitrary monad, which is automatically strong by Lemma 5.2.10. Let $\beta\colon T(B) \to B$ be an algebra.

1. Give an explicit description of the $\mathcal{EM}$-law $T(X^A \times B) \to T(X)^A \times B$, following Exercise 5.4.4. The 'exponent' version r of strength, from Exercise 5.2.16, is convenient here.
2. Describe the resulting lifted functor $\mathcal{EM}(G)\colon \mathcal{EM}(T) \to \mathcal{EM}(T)$ explicitly.
3. Prove that the T-algebra (5.21) induced on the final G-coalgebra $B^{A^\star}$ is given by

$$T(B^{A^\star}) \xrightarrow{\;r\;} T(B)^{A^\star} \xrightarrow{\;\beta^{A^\star}\;} B^{A^\star}.$$

5.4.15 Let $\sigma\colon T \Rightarrow S$ be a map of monads.

1. Show that it induces a functor $(-)\circ\sigma\colon \mathcal{EM}(S) \to \mathcal{EM}(T)$ that commutes with the forgetful functors.
2. Assume that the category $\mathcal{EM}(S)$ has coequalisers. Consider the mapping that sends a T-algebra $\alpha\colon T(X) \to X$ to the following coequaliser in $\mathcal{EM}(S)$:

$$\begin{tikzcd} \begin{pmatrix} S^2T(X) \\ \downarrow\mu \\ ST(X) \end{pmatrix} \arrow[r, shift left, "\mu\circ S(\sigma)"] \arrow[r, shift right, "S(\alpha)"'] & \begin{pmatrix} S^2(X) \\ \downarrow\mu \\ S(X) \end{pmatrix} \arrow[r, two heads, "c"] & \begin{pmatrix} S(X_\sigma) \\ \downarrow\alpha_\sigma \\ X_\sigma \end{pmatrix} \end{tikzcd}.$$

Prove that $\alpha \mapsto \alpha_\sigma$ is the left adjoint to the functor $(-)\circ\sigma$ from (1).
3. Use Lemma 5.4.5 to conclude that if T, S are monads on **Sets**, such a left adjoint $(-)_\sigma$ always exists. As a result, for instance, the forgetful functor $\mathbf{Vect}_{\mathbb{R}} \to \mathbf{Ab}$ from vector spaces to abelian groups has a left adjoint.
4. Assume, as before, that the category $\mathcal{EM}(S)$ of S-algebras has coequalisers. Prove that the coproduct of two Eilenberg–Moore algebras $\alpha\colon S(X) \to X$ and $\beta\colon S(Y) \to Y$ is given by the following coequaliser in $\mathcal{EM}(S)$:

$$\begin{tikzcd} \begin{pmatrix} S^2(S(X)+S(Y)) \\ \downarrow\mu \\ S(S(X)+S(Y)) \end{pmatrix} \arrow[r, shift left, "\mu\circ S({[S(\kappa_1),S(\kappa_2)]})"] \arrow[r, shift right, "S(\alpha+\beta)"'] & \begin{pmatrix} S^2(X+Y) \\ \downarrow\mu \\ S(X+Y) \end{pmatrix} \arrow[r, two heads] & \begin{pmatrix} S(U) \\ \downarrow \\ U \end{pmatrix} \end{tikzcd}.$$

5.4.16 Use Proposition 5.4.13 to see that there is an $\mathcal{EM}$-law $F^*F \Rightarrow FF^*$. Check that it can be described explicitly as

$$F^*F(X) \xrightarrow[\cong]{\alpha^{-1}_{F(X)}} F(X) + F(F^*F(X)) \xrightarrow{[F(\eta_X),\, F(\mu_X \circ F^*(\theta_X))]} FF^*(X)$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

where $\theta \colon F \Rightarrow F^*$ is the universal map as in the proof of Proposition 5.1.8.

5.4.17 Assuming the relevant coproducts $+$ and free monads $(-)^*$ exist, define a natural isomorphism:

$$(F + G)^* \;\cong\; F^* + G^*,$$

where the sum $+$ on the right-hand side denotes the coproduct in the category of monads. (In [233, theorem 4] it is shown that the coproduct of monads $F^* + T$, where T is a monad, is the composite $T(FT)^*$.)

5.4.18 Let $\mathbb{C}$ be a category with a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$, and let $F \colon \mathbb{C} \to \mathbb{C}$ be a functor with free monad $F^* \colon \mathbb{C} \to \mathbb{C}$.

1. Deduce from Lemma 4.4.6 that there is a natural transformation $\mathrm{Rel}(F) \Rightarrow \mathrm{Rel}(F^*)$ and that the relation lifting functor $\mathrm{Rel}(F^*) \colon \mathrm{Rel}(\mathbb{C}) \to \mathrm{Rel}(\mathbb{C})$ is a monad; conclude that there is a map of monads $\mathrm{Rel}(F)^* \Rightarrow \mathrm{Rel}(F^*)$.
2. Use Exercise 4.5.1 and Proposition 5.4.7 to show that F-congruences are the same as F^*-congruences. More formally, show that there is an isomorphism between categories of algebras:

$$\begin{tikzcd}
\mathbf{Alg}(\mathrm{Rel}(F)) \arrow[dr] & \cong & \mathcal{EM}(\mathrm{Rel}(F^*)) \arrow[dl] \\
& \mathrm{Rel}(\mathbb{C}) \arrow[d] \arrow[loop left, "\mathrm{Rel}(F)"] \arrow[loop right, "\mathrm{Rel}(F^*)"] & \\
& \mathbb{C} \arrow[loop left, "F"] \arrow[loop right, "F^*"] &
\end{tikzcd}$$

5.4.19 This exercise describes an analogon of Proposition 5.2.2 for Eilenberg–Moore categories. Let T be a monad on an arbitrary category $\mathbb{C}$, and let functor $L \colon \mathbb{C} \to \mathbb{D}$ have a right adjoint H, so that $T = HL$ is the induced monad. Define a ‘comparison’ functor $K \colon \mathbb{D} \to \mathcal{EM}(T)$ in

$$\begin{tikzcd}
\mathbb{D} \arrow[rr, "K"] \arrow[dr, "H"'] & & \mathcal{EM}(T) \arrow[dl] \\
& \mathbb{C} &
\end{tikzcd}$$

1. Check that the functor $\mathcal{K}\ell(T) \to \mathcal{EM}(T)$ in Lemma 5.4.2 arises in this manner.
2. What is the dual statement for comonads?

5.4.20 (See [334, proposition 26]) Let $T \colon \mathbb{C} \to \mathbb{C}$ be an arbitrary strong monad on a cartesian (or monoidal) closed category $\mathbb{C}$. For an

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

object $C \in \mathbb{C}$ there is the continuation monad $X \mapsto C^{(C^X)}$ from Example 5.1.3.7. Prove that there is a bijective correspondence between monad maps $T \Rightarrow C^{(C^{(-)})}$ commuting with strength, and Eilenberg–Moore algebras $T(C) \to C$.

5.5 Bialgebras and Operational Semantics

We continue our investigation of distributive laws. The new perspective that will be explored is that an $\mathcal{EM}$-law $TG \Rightarrow GT$ induces algebraic structure on the final G-coalgebra, of the kind we have seen before on streams and on processes, namely in Sections 1.2 and 3.5.2. The connection between distributive laws and specification formats for algebraic operations was first made in [452] (see also [451]), with preliminary ideas stemming from [407]. This connection forms one of the main achievements of the discipline of coalgebras. Traditionally in process algebras these formats are described as 'structured operational semantics' (SOS) rules, following [383]. A subtle matter is which rule formats correspond precisely to which rules. This is investigated for instance in [58, 334]. An overview of this material may be found in [298].

We start this section with some categorical results, first on liftings of monads and comonads, and then on bialgebras with respect to a distributive law. A slightly smoother version of these results, involving comonads instead of functors, is described in Exercise 5.5.1. The liftings may also be described for 'copointed' functors; see e.g. [334]. Such a copointed functor F comes with a counit (or copoint) $\varepsilon\colon F \Rightarrow \mathrm{id}$. This notion sits in between ordinary functors and comonads.

Proposition 5.5.1 *Let T be a monad and F, G be two endofunctors, all on the same category $\mathbb{C}$. Distributive $\mathcal{K}\ell$- and $\mathcal{EM}$-laws give rise to liftings of the monad T to new monads* $\mathrm{Alg}(T)$ *and* $\mathrm{CoAlg}(T)$ *in*

$$\frac{\mathcal{K}\ell\text{-law } FT \overset{\lambda}{\Longrightarrow} TF}{\begin{tikzcd} \mathbf{Alg}(F) \arrow[r, "\mathrm{Alg}(T)"] \arrow[d] & \mathbf{Alg}(F) \arrow[d] \\ \mathbb{C} \arrow[r, "T"] & \mathbb{C} \end{tikzcd}} \qquad \frac{\mathcal{EM}\text{-law } TG \overset{\rho}{\Longrightarrow} GT}{\begin{tikzcd} \mathbf{CoAlg}(G) \arrow[r, "\mathrm{CoAlg}(T)"] \arrow[d] & \mathbf{CoAlg}(G) \arrow[d] \\ \mathbb{C} \arrow[r, "T"] & \mathbb{C} \end{tikzcd}}\,.$$

Explicitly, these liftings are given by

$$\begin{pmatrix} F(X) \\ \downarrow a \\ X \end{pmatrix} \xmapsto{\;\mathrm{Alg}(T)\;} \begin{pmatrix} FT(X) \\ \downarrow T(a)\circ\lambda \\ T(X) \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} G(X) \\ \uparrow c \\ X \end{pmatrix} \xmapsto{\;\mathrm{CoAlg}(T)\;} \begin{pmatrix} GT(X) \\ \uparrow \rho\circ T(c) \\ T(X) \end{pmatrix}.$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

On morphisms both functors are given by $f \mapsto T(f)$. The unit and multiplication of T are also unit and multiplication of $\mathrm{Alg}(T)$ *and* $\mathrm{CoAlg}(T)$*, in the appropriate categories.*

Notice that the rules in this proposition have only a single line, indicating a passage from top to bottom, and not a bidirectional correspondence as in Proposition 5.4.9.

Proof It is easy to see that $\mathrm{Alg}(T)$ and $\mathrm{CoAlg}(T)$ are functors. The functor $\mathrm{Alg}(T)$ is a monad, because the unit and multiplication η, μ of T also form maps of F-algebras:

$$\begin{pmatrix} F(X) \\ \downarrow a \\ X \end{pmatrix} \xrightarrow{\;\eta\;} \mathrm{Alg}(T)\begin{pmatrix} F(X) \\ \downarrow a \\ X \end{pmatrix} \xleftarrow{\;\mu\;} \mathrm{Alg}(T)^2\begin{pmatrix} F(X) \\ \downarrow a \\ X \end{pmatrix}.$$

This is checked via the properties of $\mathcal{K}\ell$-laws from (5.6):

$$\begin{aligned}
\mathrm{Alg}(T)(a) \circ F(\eta) &= T(a) \circ \lambda \circ F(\eta) \\
&= T(a) \circ T(\eta) \\
&= \mathrm{id} \\
\mathrm{Alg}(T)(a) \circ F(\mu) &= T(a) \circ \lambda \circ F(\mu) \\
&= T(a) \circ \mu \circ T(\lambda) \circ \lambda \\
&= \mu \circ T(T(a) \circ \lambda) \circ \lambda \\
&= \mu \circ \mathrm{Alg}(T)^2(a).
\end{aligned}$$

Similarly one shows that $\mathrm{CoAlg}(T)$ is a monad. ❑

In the remainder of this section we concentrate on $\mathcal{EM}$-laws. First we define bialgebras for such laws. They are combinations of algebras and coalgebras which interact appropriately via the distributive law. This turns out to be a useful notion. Intuitively one can think of the algebra as describing some programming language, and of the coalgebra as its operational semantics, in the form of transition steps that the various program constructs can make.

Definition 5.5.2 Let $\rho\colon TG \Rightarrow GT$ be an $\mathcal{EM}$-law, for a monad T and an endofunctor G. We write $\mathbf{BiAlg}(TG \overset{\rho}{\Rightarrow} GT)$ for the category of ρ-bialgebras. Its objects are algebra-coalgebra pairs $T(X) \overset{\alpha}{\to} X \overset{c}{\to} G(X)$ on the same carrier, where α is an Eilenberg–Moore algebra, and the following diagram commutes:

$$\begin{tikzcd}
T(X) \arrow[r, "\alpha"] \arrow[d, "T(c)"'] & X \arrow[r, "c"] & G(X) \\
TG(X) \arrow[rr, "\rho_X"'] & & GT(X) \arrow[u, "G(\alpha)"']
\end{tikzcd} \tag{5.24}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

A map of bialgebras, from $T(X) \xrightarrow{\alpha} X \xrightarrow{c} G(X)$ to $T(Y) \xrightarrow{\beta} Y \xrightarrow{d} G(Y)$ is a map $f\colon X \to Y$ in the underlying category which is a map of both algebras and coalgebras, as in

$$\begin{tikzcd}
T(X) \arrow[r, "T(f)"] \arrow[d, "\alpha"'] & T(Y) \arrow[d, "\beta"] \\
X \arrow[r, "f"] \arrow[d, "c"'] & Y \arrow[d, "d"] \\
G(X) \arrow[r, "G(f)"'] & G(Y)
\end{tikzcd}$$

Thus, by construction, there are two forgetful functors:

$$\begin{tikzcd}
\mathcal{EM}(T) & \mathbf{BiAlg}(TG \overset{\rho}{\Rightarrow} GT) \arrow[l] \arrow[r] & \mathbf{CoAlg}(G).
\end{tikzcd}$$

The next result comes from [452] and forms the basis for categorical operational semantics.

Theorem 5.5.3 *For an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$ consider the associated liftings $\mathcal{EM}(G)$ from Proposition 5.4.9 and* CoAlg(T) *from Proposition 5.5.1 in*

$$\begin{tikzcd}
\mathcal{EM}(T) \arrow[loop below, "\mathcal{EM}(G)"] & \mathbf{BiAlg}(TG \overset{\rho}{\Rightarrow} GT) \arrow[l] \arrow[r] & \mathbf{CoAlg}(G). \arrow[loop below, "\mathrm{CoAlg}(T)"]
\end{tikzcd}$$

Then there are isomorphism and adjoints as in

$$\begin{tikzcd}
\mathbf{CoAlg}(\mathcal{EM}(G)) \arrow[r, phantom, "\cong"] \arrow[d, shift right] & \mathbf{BiAlg}(TG \overset{\rho}{\Rightarrow} GT) \arrow[r, phantom, "\cong"] & \mathcal{EM}(\mathrm{CoAlg}(T)). \arrow[d, shift left] \\
\mathcal{EM}(T) \arrow[u, bend right, "\mathcal{G}"'] \arrow[u, phantom, "\dashv"] & & \mathbf{CoAlg}(G) \arrow[u, bend left, "\mathcal{F}"] \arrow[u, phantom, "\dashv"]
\end{tikzcd} \tag{5.25}$$

The left adjoint $\mathcal{F}$ is the lifting of the free algebra functor $\mathcal{F}\colon \mathbb{C} \to \mathcal{EM}(T)$ from Lemma 5.4.2. The right adjoint $\mathcal{G}$ exists, if we assume the existence of cofree G-coalgebras.

Proof Diagram (5.24) can be read at the same time as

- $c\colon X \to G(X)$ is a map of T-algebras:

$$\begin{pmatrix} T(X) \\ \downarrow{\scriptstyle\alpha} \\ X \end{pmatrix} \xrightarrow{\quad c \quad} \mathcal{EM}(G)\begin{pmatrix} T(X) \\ \downarrow{\scriptstyle\alpha} \\ X \end{pmatrix} = \begin{pmatrix} TG(X) \\ \downarrow{\scriptstyle G(\alpha)\circ\rho} \\ G(X) \end{pmatrix}.$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

- $\alpha \colon T(X) \to X$ is a map of G-coalgebras:

$$\begin{pmatrix} GT(X) \\ \uparrow{\scriptstyle \rho\circ T(c)} \\ T(X) \end{pmatrix} = \mathrm{CoAlg}(T)\begin{pmatrix} G(X) \\ \uparrow{\scriptstyle c} \\ X \end{pmatrix} \xrightarrow{\quad\alpha\quad} \begin{pmatrix} G(X) \\ \uparrow{\scriptstyle c} \\ X \end{pmatrix}.$$

This is the essence of the isomorphisms in (5.25).

We also need to check that $f \colon X \to Y$ is a map of bialgebras

$$\big(T(X) \overset{\alpha}{\to} X \overset{c}{\to} G(X)\big) \longrightarrow \big(T(Y) \overset{\beta}{\to} Y \overset{d}{\to} G(Y)\big)$$

if and only if

- f is both a map of T-algebras $\alpha \to \beta$ and a map of $\mathcal{EM}(G)$-coalgebras $c \to d$
- f is both a map of G-coalgebras $c \to d$ and a map of $\mathrm{CoAlg}(T)$-algebras $\alpha \to \beta$.

This is easy. We turn to the left adjoint $\mathcal{F} \colon \mathbf{CoAlg}(G) \to \mathbf{BiAlg}(TG \overset{\rho}{\Rightarrow} GT)$. Let $c \colon X \to G(X)$ be a coalgebra. We take the free algebra $\mu \colon T^2(X) \to T(X)$ and change c into a G-coalgebra $\mathbf{CoAlg}(T)(c) = \rho \circ T(c) \colon T(X) \to TG(X) \to GT(X)$. Together they form a bialgebra, since the next rectangle commutes:

$$\begin{tikzcd}
T^2(X) \arrow[r, "\mu"] \arrow[d, "T(\rho \circ T(c))"'] & T(X) \arrow[r, "\rho \circ T(c)"] & GT(X) \\
TGT(X) \arrow[rr, "\rho"'] & & GT^2(X) \arrow[u, "G(\mu)"']
\end{tikzcd}$$

Since

$$G(\mu) \circ \rho \circ T(\rho \circ T(c)) \overset{(5.20)}{=} \rho \circ \mu \circ T^2(c) = \rho \circ T(c) \circ \mu.$$

Moreover, we have an adjoint correspondence between f and h in

$$\begin{array}{c}
\begin{tikzcd}
T^2(X) \arrow[r, "T(f)"] \arrow[d, "\mu"'] & T(Y) \arrow[d, "\beta"] \\
T(X) \arrow[r, "f"] \arrow[d, "\rho\circ T(c)"'] & Y \arrow[d, "d"] \\
GT(X) \arrow[r, "G(f)"] & G(Y)
\end{tikzcd} \\
\hline\hline
\begin{tikzcd}
G(X) \arrow[r, "G(h)"] & G(Y) \\
X \arrow[u, "c"] \arrow[r, "h"] & Y \arrow[u, "d"']
\end{tikzcd}
\end{array}.$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

This the usual free algebra adjoint correspondence, given by $\overline{f} = f \circ \eta$ and $\overline{h} = \beta \circ T(h)$.

Next we assume the existence of cofree coalgebras, in the form of a right adjoint $\mathcal{G}\colon \mathbb{C} \to \mathbf{CoAlg}(G)$ to the forgetful functor. We show how it can be adapted to a functor $\mathcal{G}\colon \mathcal{EM}(T) \to \mathbf{BiAlg}(TG \overset{\rho}{\Rightarrow} GT)$. For an arbitrary object $Y \in \mathbb{C}$, we write the cofree coalgebra as $\zeta_Y\colon \mathcal{G}(Y) \to G(\mathcal{G}(Y))$, with counit $\varepsilon\colon \mathcal{G}(Y) \to Y$.

For an Eilenberg–Moore algebra $\beta\colon T(Y) \to Y$ consider the composite map $\beta \circ T(\varepsilon)\colon T(\mathcal{G}(Y)) \to T(Y) \to Y$. Its carrier can be equipped with a G-coalgebra $\rho \circ T(\zeta_Y)$ and yields a unique coalgebra map $\widehat{\beta}\colon T(\mathcal{G}(Y)) \to \mathcal{G}(Y)$ with $\varepsilon_Y \circ \widehat{\beta} = \beta \circ T(\varepsilon)$ in

$$\begin{tikzcd}
GT\mathcal{G}(Y) \arrow[r, "G(\widehat{\beta})"] & G\mathcal{G}(Y) \\
TG\mathcal{G}(Y) \arrow[u, "\rho"] & \\
T\mathcal{G}(Y) \arrow[u, "T(\zeta_Y)"] \arrow[r, "\widehat{\beta}"'] & \mathcal{G}(Y) \arrow[uu, "\zeta_Y"']
\end{tikzcd} \tag{5.26}$$

We leave it to the reader to check that $\widehat{\beta}$ is an Eilenberg–Moore algebra. By construction, the pair $(\widehat{\beta}, \zeta_Y)$ is a bialgebra. Further, for an arbitrary bialgebra (α, c) there is a bijective correspondence between f and h in

$$\begin{array}{c}
\begin{tikzcd}
T(X) \arrow[r, "T(f)"] \arrow[d, "\alpha"'] & T\mathcal{G}(Y) \arrow[d, "\widehat{\beta}"] \\
X \arrow[r, "f"] \arrow[d, "c"'] & \mathcal{G}(Y) \arrow[d, "\zeta_Y"] \\
G(X) \arrow[r, "G(f)"] & G\mathcal{G}(Y)
\end{tikzcd} \\
\hline\hline
\begin{tikzcd}
T(X) \arrow[r, "T(h)"] \arrow[d, "\alpha"'] & T(Y) \arrow[d, "\beta"] \\
X \arrow[r, "h"] & Y
\end{tikzcd}
\end{array}.$$

This correspondence arises as follows.

- Given a map of bialgebras $f\colon X \to \mathcal{G}(Y)$ we take $\overline{f} = \varepsilon \circ f\colon X \to Y$. It forms a map of algebras:

$$\beta \circ T(\overline{f}) = \beta \circ T(\varepsilon) \circ T(f) = \varepsilon \circ \widehat{\beta} \circ T(f) = \varepsilon \circ f \circ \alpha = \overline{f} \circ \alpha.$$

- In the other direction, let $h\colon X \to Y$ be a map of algebras. By cofreeness we obtain a unique map of coalgebras $\overline{h}\colon X \to \mathcal{G}(Y)$, with $\zeta_Y \circ \overline{h} = G(\overline{h}) \circ c$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

and $\varepsilon \circ \overline{h} = h$. This $\overline{h}$ is a map of bialgebras. The missing equation $\widehat{\beta} \circ T(\overline{h}) = \overline{h} \circ \alpha$ can be obtained as follows. Both sides of the latter equation form a map of coalgebras in

$$\begin{tikzcd}
GT(X) \arrow[r] & G\mathcal{G}(Y) \\
T(X) \arrow[u, "\rho \circ T(c)"] \arrow[r] & \mathcal{G}(Y) \arrow[u, "\zeta_Y"']
\end{tikzcd}$$

And both sides are equal when post-composed with the counit ε:

$$\varepsilon \circ \widehat{\beta} \circ T(\overline{h}) = \beta \circ T(\varepsilon) \circ T(\overline{h}) = \beta \circ T(h) = h \circ \alpha = \varepsilon \circ \overline{h} \circ \alpha.$$

By construction we have $\overline{\overline{h}} = \varepsilon \circ \overline{h} = h$. And $\overline{\overline{f}} = f$ holds by uniqueness. ❑

Corollary 5.5.4 *Let $\rho\colon TG \Rightarrow GT$ be an $\mathcal{EM}$-law for a monad T and an endofunctor G on a category $\mathbb{C}$.*

1. *If $\mathbb{C}$ has an initial object 0, then the category $\mathbf{BiAlg}(TG \overset{\rho}{\Rightarrow} GT)$ of bialgebras has an initial object, namely*

$$T^2(0) \xrightarrow{\mu} T(0) \xrightarrow{\rho \circ T(!)} GT(0).$$

It is given by the initial T-algebra $T^2(0) \to T(0)$ and the initial G-coalgebra $0 \to G(0)$.

2. *If the functor G has a final coalgebras $Z \underset{\cong}{\xrightarrow{\zeta}} G(Z)$, then there is also a final bialgebra*

$$T(Z) \xrightarrow{\beta} Z \underset{\cong}{\xrightarrow{\zeta}} G(Z),$$

where the Eilenberg–Moore algebra $\beta\colon T(Z) \to Z$ is obtained by finality from the coalgebra $\rho \circ T(\zeta)\colon T(Z) \to GT(Z)$, as in (5.21).

Proof The left adjoint $\mathcal{F}$ in (5.25) preserves initial objects, so applying it to the initial G-algebra $!\colon 0 \to G(0)$ yields the initial bialgebra.

Similarly, a final G-coalgebra $\zeta\colon Z \overset{\cong}{\Rightarrow} G(Z)$ is the cofree coalgebra $\mathcal{G}(1)$ at the final object $1 \in \mathbb{C}$. Hence the final T-algebra $!\colon T(1) \to 1$ yields an algebra $T(Z) \to Z$ as in (5.26) –or, equivalently, as in (5.21). ❑

Theorem 5.5.5 *Assume the presence of both the initial and final bialgebra as described in Corollary 5.5.4.*

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

1. *The algebraic semantics obtained by initiality and the coalgebraic semantics by finality coincide with the unique bialgebra map*

$$\begin{tikzcd}
T^2(0) \arrow[r, dashed] \arrow[d, "\mu"'] & T(Z) \arrow[d, "\beta"] \\
T(0) \arrow[r, dashed, "\mathrm{int} = \mathrm{beh}"] \arrow[d, "\rho \circ T(!)"'] & Z \arrow[d, "\zeta", "\cong"'] \\
GT(0) \arrow[r, dashed] & G(Z)
\end{tikzcd} \tag{5.27}$$

2. *As a result, bisimilarity/observational equivalence on $T(0)$ is a congruence.*

Proof 1. The algebra morphism $\mathrm{int}\colon T(0) \to Z$ is obtained by initiality of the algebra $\mu\colon T^2(0) \to T(0)$ in $\mathcal{EM}(T)$. We wish to show that it is also a coalgebra map, i.e. that it satisfies $\zeta \circ \mathrm{int} = G(\mathrm{int}) \circ \rho \circ T(!)$ in (5.27). Then, by uniqueness, $\mathrm{int} = \mathrm{beh}$. Define

$$\beta' = \Big(TG(Z) \xrightarrow[\cong]{T(\zeta^{-1})} T(Z) \xrightarrow{\beta} Z \xrightarrow[\cong]{\zeta} G(Z)\Big).$$

It is not hard to see that β' is an Eilenberg–Moore algebra. Hence we are done, by initiality, if both $\zeta \circ \mathrm{int}$ and $G(\mathrm{int}) \circ \rho \circ T(!)$ are algebra maps $\mu \to \beta'$ in a situation:

$$\begin{tikzcd}
T^2(0) \arrow[r, shift left] \arrow[r, shift right] \arrow[d, "\mu"'] & TG(Z) \arrow[d, "\beta'"] \\
T(0) \arrow[r, shift left, "\zeta \circ \mathrm{int}"] \arrow[r, shift right, "G(\mathrm{int}) \circ \rho \circ T(!)"'] & G(Z)
\end{tikzcd}$$

This is easy:

$$\begin{aligned}
&\beta' \circ T(\zeta \circ \mathrm{int}) \\
&\quad = \quad \zeta \circ \beta \circ T(\zeta^{-1}) \circ T(\zeta) \circ T(\mathrm{int}) \\
&\quad = \quad \zeta \circ \beta \circ T(\mathrm{int}) \\
&\quad = \quad \zeta \circ \mathrm{int} \circ \mu \\
&\beta' \circ T(G(\mathrm{int}) \circ \rho \circ T(!)) \\
&\quad = \quad \zeta \circ \beta \circ T(\zeta^{-1}) \circ TG(\mathrm{int}) \circ T(\rho) \circ T^2(!) \\
&\quad \overset{(5.21)}{=} \quad G(\beta) \circ \rho \circ T(\zeta) \circ T(\zeta^{-1}) \circ TG(\mathrm{int}) \circ T(\rho) \circ T^2(!) \\
&\quad = \quad G(\beta) \circ \rho \circ TG(\mathrm{int}) \circ T(\rho) \circ T^2(!) \\
&\quad = \quad G(\beta) \circ GT(\mathrm{int}) \circ \rho \circ T(\rho) \circ T^2(!) \\
&\quad = \quad G(\mathrm{int}) \circ G(\mu) \circ \rho \circ T(\rho) \circ T^2(!) \\
&\quad \overset{(5.20)}{=} \quad G(\mathrm{int}) \circ \rho \circ \mu \circ T^2(!) \\
&\quad = \quad G(\mathrm{int}) \circ \rho \circ T(!) \circ \mu.
\end{aligned}$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

2. Consider the following diagram, where the arrow e is an equaliser, describing the kernel of the (co)algebra map $\mathrm{int} = \mathrm{beh}\colon T(0) \to Z$ to the final coalgebra:

$$\begin{tikzcd}
T(\underline{\leftrightarrow}) \arrow[r, "{\langle T(\pi_1 \circ e), T(\pi_2 \circ e)\rangle}"] \arrow[d, dashed, "\gamma"'] & T^2(0) \times T^2(0) \arrow[r, shift left, "T(\mathrm{int}) \circ \pi_1"] \arrow[r, shift right, "T(\mathrm{int}) \circ \pi_2"'] \arrow[d, "\mu \times \mu"'] & T(Z) \arrow[d, "\beta"] \\
\underline{\leftrightarrow} \arrow[r, tail, "e"] & T(0) \times T(0) \arrow[r, shift left, "\mathrm{int} \circ \pi_1"] \arrow[r, shift right, "\mathrm{int} \circ \pi_2"'] & Z
\end{tikzcd}$$

The existence of the map γ follows from an easy diagram chase. It is an Eilenberg–Moore algebra and shows that bisimilarity $\underline{\leftrightarrow}$ on the initial algebra $T(0)$ is a congruence. □

In the remainder of this section we show how distributive laws can be used to define operations on final coalgebras, in the form of an Eilenberg–Moore algebra $T(Z) \to Z$ as above. Such an algebra exists as soon as we have an $\mathcal{EM}$-law $TG \Rightarrow GT$. In fact, this law forms a specification format for the algebra. Here we shall give not the most *general* format but the most *common* format. Such laws are called GSOS rules, because they correspond to the 'general structured operational semantics' specification format introduced in [74]. Here we do not go into the syntactic formulation of these rules, but only consider their categorical counterparts. For more information about the connection, see [58, 298].

Definition 5.5.6 Let T be a monad and G an endofunctor on a category $\mathbb{C}$. A **GSOS-law** is an $\mathcal{EM}$-law of the form

$$T(G \times \mathrm{id}) \overset{\sigma}{\Longrightarrow} (G \times \mathrm{id})T \qquad \text{satisfying} \qquad \pi_2 \circ \sigma = T(\pi_2).$$

For such a law we define a category $\mathbf{BiAlg}(T(G \times \mathrm{id}) \overset{\rho}{\Rightarrow} (G \times \mathrm{id})T)$ whose objects are bialgebras given by a pair of an Eilenberg–Moore algebra $\alpha\colon T(X) \to X$ and a coalgebra $c\colon X \to G(X)$ making the following diagram commute:

$$\begin{tikzcd}
T(X) \arrow[r, "\alpha"] \arrow[d, "T(\langle c, \mathrm{id}\rangle)"'] & X \arrow[r, "c"] & G(X) \\
T(G(X) \times X) \arrow[r, "\sigma_X"'] & GT(X) \times T(X) \arrow[r, "\pi_1"'] & GT(X) \arrow[u, "G(\alpha)"']
\end{tikzcd}$$

Morphisms of bialgebras are, as before, pairs of algebra maps and coalgebra maps.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

A GSOS-law is thus a distributive law of the monad T over the functor $G \times \mathrm{id} \colon \mathbb{C} \to \mathbb{C}$, given by $X \mapsto G(X) \times X$. The formulation used here (following [246]) generalises the original one from [452] for a free monad F^*.

Lemma 5.5.7 *For two endofunctors $F, G \colon \mathbb{C} \to \mathbb{C}$, there is a bijective correspondence:*

$$\frac{\textit{GSOS-laws}\;\; F^*(G \times \mathrm{id}) \overset{\sigma}{\Longrightarrow} (G \times \mathrm{id})F^*}{\overline{\textit{natural transformations}\;\; F(G \times \mathrm{id}) \underset{\tau}{\Longrightarrow} GF^*}}\,.$$

Thus this GSOS format involves maps $F(G(X) \times X) \to GF^*(X)$. These are more useful than the maps $FG(X) \to GF^*(X)$ from Lemma 5.4.13. The additional occurrence of the X on the input side increases the expressive power, as illustrated below. In view of this result we shall also call natural transformations $\tau \colon F(G \times \mathrm{id}) \Rightarrow GF^*$ GSOS-laws.

Proof We have the following correspondences:

$$\frac{\frac{\text{nat. transf.}\;\; F(G \times \mathrm{id}) \overset{\tau}{\Longrightarrow} GF^*}{\overline{\text{nat. transf.}\;\; F(G \times \mathrm{id}) \overset{\tau}{\Longrightarrow} (G \times \mathrm{id})F^* \;\text{ with }\; \pi_2 \circ \tau = \theta \circ F(\pi_2)}}}{\overline{\mathcal{EM}\text{-laws}\;\; F^*(G \times \mathrm{id}) \underset{\sigma}{\Longrightarrow} (G \times \mathrm{id})F^* \;\text{ with }\; \pi_2 \circ \sigma = F^*(\pi_2)}}$$

The first correspondence is easy and the second one is a minor adaptation of Lemma 5.4.13. ❏

For GSOS-laws we do not elaborate analogues of Theorem 5.5.3 and Corollary 5.5.4, but concentrate on the essentials for what is needed below.

Theorem 5.5.8 *Let $\sigma \colon T(G \times \mathrm{id}) \Rightarrow (G \times \mathrm{id})T$ be a GSOS-law. If the underlying category has an initial object 0, then the initial bialgebra exists with carrier $T(0)$ and algebra-coalgebra structure:*

$$T^2(0) \xrightarrow{\mu} T(0) \xrightarrow{T(!)} T(G(0) \times 0) \xrightarrow{\sigma_0} GT(0) \times T(0) \xrightarrow{\pi_1} GT(0).$$

If there is a final coalgebra $\zeta \colon Z \overset{\cong}{\to} G(Z)$, then there is also a final bialgebra with carrier Z, and structure maps

$$T(Z) \xrightarrow{\beta} Z \xrightarrow[\cong]{\zeta} G(Z),$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

where the Eilenberg–Moore algebra structure $\beta\colon T(Z) \to Z$ is obtained by finality in

$$\begin{tikzcd}
GT(Z) \arrow[r, dashed, "G(\beta)"] & G(Z) \\
T(Z) \arrow[u, "\pi_1 \circ \sigma \circ T(\langle \zeta, \mathrm{id}\rangle)"] \arrow[r, dashed, "\beta"'] & Z \arrow[u, "\zeta"', "\cong"]
\end{tikzcd} \tag{5.28}$$

The analogue of Theorem 5.5.5 holds in this situation: the map $T(0) \to Z$ obtained by initiality is the same as the map $T(0) \to Z$ by finality and is the unique map of bialgebras $T(0) \to Z$. And also: bisimilarity $\underline{\leftrightarrow}$ on $T(0)$ is a congruence.

Proof Let $T(X) \xrightarrow{\alpha} X \xrightarrow{c} G(X)$ be an arbitrary bialgebra. Consider $f = \alpha \circ T(!)\colon T(0) \to T(X) \to X$. This f is a map of bialgebras. Similarly, the unique coalgebra map $X \to Z$ is a map of bialgebras. ❑

Suppose we have a final coalgebra $\zeta\colon Z \to G(Z)$ and we wish to define function interpretations $f_i\colon Z^{\#i} \to Z$ on Z, for some arity $\#\colon I \to \mathbb{N}$. That is, we wish to define a (functor) algebra $F_\#(Z) \to Z$, where $F_\#$ is the arity functor $F_\#(X) = \coprod_{i\in I} X^{\#i}$ associated with #. Equivalently, by Proposition 5.4.7, we can define a monad algebra $F_\#^*(Z) \to Z$. The associated free monad $F_\#^*$ is characterised as the initial algebra

$$X + \coprod_{i\in I} F_\#^*(X)^{\#i} \xrightarrow[\cong]{\alpha_X} F_\#^*(X);$$

see Proposition 5.1.8. The unit of the free monad $F_\#^*$ is then $\eta = \alpha \circ \kappa_1\colon X \to F_\#^*(X)$.

Theorem 5.5.8 says that in order to define such an interpretation $F_\#^*(Z) \to Z$ it suffices to define a natural transformation $\tau\colon F_\#(G \times \mathrm{id}) \Rightarrow (G \times \mathrm{id})F_\#^*$. This means that for each $i \in I$ we need (natural) maps

$$(G(X) \times X)^{\#i} \cong G(X)^{\#i} \times X^{\#i} \xrightarrow{\tau_i} GF_\#^*(X),$$

one for each operation f_i. This τ_i describes the relevant behaviour associated with the operation f_i: it sends the appropriate number of behaviours and states $(u_k, x_k) \in G(X) \times X$, for k below the arity $\#i \in \mathbb{N}$ of f_i, to a behaviour over terms, in $GF_\#^*(X)$.

Now assume we have such τ_i as described above. Together they correspond to a cotuple natural transformation $\tau = [\tau_i]_{i\in I}\colon F_\#(G \times \mathrm{id}) \Rightarrow GF_\#^*$, which

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

corresponds by Lemma 5.5.7 to a GSOS-law $\overline{\tau}\colon F^{*}_{\#}(G \times \mathrm{id}) \Rightarrow (G \times \mathrm{id})F^{*}_{\#}$. By Theorem 5.5.8 we have a final bialgebra,

$$F^{*}_{\#}(Z) \xrightarrow{\;\beta\;} Z \xrightarrow[\cong]{\;\zeta\;} G(Z),$$

where the map β is obtained by finality as in (5.28). The interpretation $f_i\colon Z^n \to Z$ of each operation f_i with arity $\#i \in \mathbb{N}$ can then be described as the composite:

$$Z^{\#i} \xrightarrow{\;\eta^{\#i}\;} F^{*}_{\#}(Z)^{\#i} \xrightarrow{\;\kappa_i\;} \textstyle\coprod_{i\in I} F^{*}_{\#}(Z)^{\#i} \xrightarrow{\;\alpha\,\circ\,\kappa_2\;} F^{*}_{\#}(Z) \xrightarrow{\;\beta\;} Z.$$

We claim that these maps $f_i\colon Z^{\#i} \to Z$ are determined by the following equation:

$$\zeta \circ f_i \;=\; G(\beta \circ \mu) \circ \tau_i \circ \langle G(\eta) \circ \zeta, \eta\rangle^{\#i}. \tag{5.29}$$

It follows from a diagram chase:

$$\begin{tikzcd}
Z^{\#i} \arrow[r, "\eta^{\#i}"'] \arrow[rrr, bend left, "f_i"] \arrow[d, "\langle \zeta, \mathrm{id}\rangle^{\#i}"'] \arrow[dr, phantom, "\textcircled{1}"] & F^{*}_{\#}(Z)^{\#i} \arrow[r, "\alpha\,\circ\,\kappa_2\,\circ\,\kappa_i"] \arrow[d, "F^{*}_{\#}(\langle \zeta, \mathrm{id}\rangle)^{\#i}"] \arrow[dr, phantom, "\textcircled{2}"] & F^{*}_{\#}(Z) \arrow[r, "\beta"'] \arrow[d, "F^{*}_{\#}(\langle \zeta, \mathrm{id}\rangle)"] & Z \arrow[r, "\zeta", "\cong"'] \arrow[d, phantom, "\textcircled{3}"] & G(Z) \\
(G(Z) \times Z)^{\#i} \arrow[r, "\eta^{\#i}"] \arrow[dr, bend right, "(G(\eta) \times \eta)^{\#i}"'] \arrow[d, phantom, "\textcircled{4}"'] & F^{*}_{\#}(G(Z) \times Z)^{\#i} \arrow[r, "\alpha\,\circ\,\kappa_2\,\circ\,\kappa_i"'] \arrow[d, "\overline{\tau}^{\#i}"] & F^{*}_{\#}(G(Z) \times Z) \arrow[rr, "\overline{\tau}"] \arrow[d, phantom, "\textcircled{5}"] & & GF^{*}_{\#}(Z) \arrow[u, "G(\beta)"'] \\
& (GF^{*}_{\#}(Z) \times F^{*}_{\#}(Z))^{\#i} \arrow[rrr, "\tau_i"'] & & & GF^{*}_{\#}F^{*}_{\#}(Z) \arrow[u, "G(\mu)"']
\end{tikzcd}$$

We list why the various numbered subdiagrams commute:

① by naturality of η
② by definition of $F^{*}_{\#}$ on maps; see (5.2)
③ because the pair (β, ζ) is a bialgebra, by construction of β in (5.28)
④ and ⑤ by definition of $\overline{\tau}$ in (5.23), using that $\eta = \alpha \circ \kappa_1$.

Bisimilarity on the set $F^{*}_{\#}(0)$ of closed terms is then a congruence.

Example 5.5.9 We briefly review several ways of employing the above uniform approach to obtain algebraic operations on final coalgebras and coalgebraic structure on closed terms.

1. Recall from Section 1.2 the final coalgebra of finite and infinite sequences,

$$A^{\infty} \xrightarrow[\cong]{\;\mathsf{next}\;} 1 + (A \times A^{\infty}),$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

for the functor $G(X) = 1 + A \times X$ on **Sets**. Suppose we wish to define a sequential composition operation $\mathsf{comp}\colon A^\infty \times A^\infty \to A^\infty$ on such sequences. We expect the following rules for composition $\mathsf{comp}(s,t)$ of two sequences $s, t \in A^\infty$:

$$\frac{s \not\rightarrow \qquad t \not\rightarrow}{\mathsf{comp}(s,t) \not\rightarrow} \qquad \frac{s \not\rightarrow \qquad t \xrightarrow{a} t'}{\mathsf{comp}(s,t) \xrightarrow{a} \mathsf{comp}(s,t')} \qquad \frac{s \xrightarrow{a} s'}{\mathsf{comp}(s,t) \xrightarrow{a} \mathsf{comp}(s',t)}$$

where we recall from (1.5) that $s \not\rightarrow$ means $\mathsf{next}(s) = *$ and $s \xrightarrow{a} s'$ means $\mathsf{next}(s) = (a, s')$. Hence $\mathsf{comp}(s,t)$ is s if s is infinite; otherwise it is $s \cdot t$, where $\cdot$ is prefixing.

The arity functor for this single, binary composition operation is $F(X) = X \times X$. The associated free monad on F is written as F^*, as in Proposition 5.1.8, and determined as the initial algebra

$$X + F^*(X) \times F^*(X) \xrightarrow[\cong]{\alpha_X} F^*(X).$$

We now define a GSOS-law $\tau\colon F(G \times \mathrm{id}) \Rightarrow GF^*$ for sequential composition as follows:

$$\begin{array}{rcl}
\big((1 + A \times X) \times X\big) \times \big((1 + A \times X) \times X\big) & \xrightarrow{\quad\tau_X\quad} & 1 + A \times F^*(X). \\
(\langle *, x\rangle,\ \langle *, y\rangle) & \longmapsto & * \\
(\langle *, x\rangle,\ \langle\langle b, y'\rangle, y\rangle) & \longmapsto & \langle b, \alpha(\eta(x), \eta(y'))\rangle \\
(\langle\langle a, x'\rangle, x\rangle,\ \langle u, y\rangle) & \longmapsto & \langle a, \alpha(\eta(x'), \eta(y))\rangle.
\end{array}$$

The three cases of this GSOS-law clearly resemble the above three rules for comp.

As above we now get an algebra $\beta\colon F^*(A^\infty) \to A^\infty$, via (5.28). The actual function $\mathsf{comp}\colon A^\infty \times A^\infty \to A^\infty$ is the composite:

$$A^\infty \times A^\infty \xrightarrow{\eta \times \eta} F^*(A^\infty) \times F^*(A^\infty) \xrightarrow{\alpha \circ \kappa_2} F^*(A^\infty) \xrightarrow{\quad\beta\quad} A^\infty.$$

It satisfies, as in (5.29),

$$\mathsf{next} \circ \mathsf{comp} = G(\beta \circ \mu) \circ \tau \circ (\langle G(\eta) \circ \mathsf{next}, \eta\rangle \times \langle G(\eta) \circ \mathsf{next}, \eta\rangle).$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

It allows use to check that if $\mathsf{next}(s) = \mathsf{next}(t) = *$, then

$$\begin{aligned}
&\mathsf{next}(\mathsf{comp}(s,t)) \\
&= \big(G(\beta \circ \mu) \circ \tau \circ (\langle G(\eta) \circ \mathsf{next}, \eta\rangle \times \langle G(\eta) \circ \mathsf{next}, \eta\rangle)\big)(s,t) \\
&= \big(G(\beta \circ \mu) \circ \tau\big)(\langle G(\eta)(\mathsf{next}(s)), \eta(s)\rangle,\ \langle G(\eta)(\mathsf{next}(t)), \eta(t)\rangle) \\
&= \big(G(\beta \circ \mu) \circ \tau\big)(\langle G(\eta)(*), \eta(s)\rangle,\ \langle G(\eta)(*), \eta(t)\rangle) \\
&= G(\beta \circ \mu)(\tau(\langle *, \eta(s)\rangle,\ \langle *, \eta(t)\rangle)) \\
&= G(\beta \circ \mu)(*) \\
&= *.
\end{aligned}$$

In a similar manner one can prove

$$\mathsf{next}(\mathsf{comp}(s,t)) = \begin{cases} (b, \mathsf{comp}(s,t')) & \text{if } \mathsf{next}(s) = *,\ \mathsf{next}(t) = (b,t') \\ (a, \mathsf{comp}(s',t)) & \text{if } \mathsf{next}(s) = (a,s'). \end{cases}$$

Hence we have obtained a sequential composition function, described informally via the above three rules, and defined precisely via the GSOS-law/natural transformation τ.

2. In Section 3.5.2 we have seen a final coalgebra of processes

$$Z_A \xrightarrow[\cong]{\zeta_A} \mathcal{P}_{\mathrm{fin}}(Z_A)^A$$

for the functor $G(X) = \mathcal{P}_{\mathrm{fin}}(X)^A$ on **Sets**, where A is a (fixed) set of labels. A simple process language was defined via an arity functor

$$\Sigma_A(X) = 1 + (A \times X) + (X \times X)$$

with initial algebra $\xi_A \colon \Sigma_A(P_A) \xrightarrow{\cong} P_A$. Alternatively, we can describe this set P_A of process terms as initial monad algebra $\Sigma_A^*(0)$. Here we shall describe the situation in terms of bialgebras and distributive laws.

To start, there is an $\mathcal{EM}$-law of the form $\Sigma_A \mathcal{P}_{\mathrm{fin}}(-)^A \Rightarrow \mathcal{P}_{\mathrm{fin}}(-)^A \Sigma_A^*$, via Lemma 5.4.13, namely,

$$1 + (A \times \mathcal{P}_{\mathrm{fin}}(X)^A) + (\mathcal{P}_{\mathrm{fin}}(X)^A \times \mathcal{P}_{\mathrm{fin}}(X)^A) \xrightarrow{\tau_X} \mathcal{P}_{\mathrm{fin}}(\Sigma_A^*(X))^A$$

given by the following three clauses:

$$\begin{cases} * \longmapsto \lambda b \in A.\, \emptyset \\ (a,h) \longmapsto \lambda b \in A. \begin{cases} h(b) & \text{if } a = b \\ \emptyset \end{cases} \\ (h,k) \longmapsto \lambda b \in A.\, h(b) \cup k(b). \end{cases}$$

Corollary 5.5.4 now tells us that there is a unique map $P_A \to Z_A$ that is both a map of algebras and a map of coalgebras. This map that sends a process

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

term to its interpretation as a process is thus compositional. Moreover, by Theorem 5.5.5 bisimilarity on P_A is a congruence. Earlier in this book, in Propositions 3.5.2 and 3.5.3, we had to prove these results by hand.

3. In [246] it is described how regular languages and automata can be understood in terms of bialgebras and distributive laws. The arity functor for regular languages is:

$$\mathcal{R}(X) = \underbrace{1}_{\text{zero}} + \underbrace{1}_{\text{one}} + \underbrace{(X \times X)}_{\text{sum } +} + \underbrace{(X \times X)}_{\text{product } \cdot} + \underbrace{X}_{\text{star } (-)^*} .$$

For an arbitrary set A, the free monad $\mathcal{R}^*(A)$ captures the regular expressions with elements $a \in A$ as atomic expressions. There is the familiar interpretation $\mathcal{R}^*(A) \to \mathcal{P}(A^\star)$ of regular expressions as languages. This set of languages $\mathcal{P}(A^\star)$ is the final coalgebra of the deterministic automaton functor $G(X) = X^A \times 2$; see Corollary 2.3.6.2. This interpretation $\mathcal{R}^*(A) \to \mathcal{P}(A^\star)$ can then be described as a bialgebra for the GSOS-law:

$$\begin{array}{rcl}
\mathcal{R}(X^A \times 2 \times X) & \xrightarrow{\qquad\qquad\qquad\qquad} & \mathcal{R}^*(X)^A \times 2 \\
0 & \xmapsto{\qquad\text{zero}\qquad} & (\lambda a \in A.\, 0, 0) \\
1 & \xmapsto{\qquad\text{one}\qquad} & (\lambda a \in A.\, 0, 1) \\
\langle (h_1, b_1, x_1), (h_2, b_2, x_2) \rangle & \xmapsto{\text{plus}} & (\lambda a \in A.\, h_1(a) + h_2(a), b_1 \vee b_2) \\
\langle (h_1, b_1, x_1), (h_2, b_2, x_2) \rangle & \xmapsto{\text{product}} & (\lambda a \in A.\, h_1(a) \cdot x_2 + b_1 \cdot h_2(a), b_1 \wedge b_2) \\
(h, b, x) & \xmapsto{\qquad\text{star}\qquad} & (\lambda a \in A.\, h(a) \cdot x^*, 1).
\end{array}$$

More details can be found in [246].

We conclude this section with a strengthened form of coinduction for distributive $\mathcal{EM}$-laws. We shall refer to this generalised version as '$\mathcal{EM}$-coinduction'. An even stronger 'corecursive' version that builds on it is described in Exercise 5.5.4.

Proposition 5.5.10 *Assume we have*

- *a functor $G \colon \mathbb{C} \to \mathbb{C}$ with a final coalgebra $\zeta \colon Z \xrightarrow{\cong} G(Z)$*
- *a monad $T \colon \mathbb{C} \to \mathbb{C}$, on the same category, with an $\mathcal{EM}$-law $\rho \colon TG \Rightarrow GT$.*

Then the following '$\mathcal{EM}$-coinduction' principle holds: for each map $e \colon X \to GT(X)$ there is a unique map $s \colon X \to Z$ making the following diagram commute:

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

$$\begin{tikzcd}
GT(X) \arrow[r, "GT(s)"] & GT(Z) \arrow[d, "G(\beta)"] \\
 & G(Z) \\
X \arrow[uu, "e"] \arrow[r, "s"'] & Z \arrow[u, "\zeta"', "\cong"]
\end{tikzcd}$$

where $\beta\colon T(Z) \to Z$ is the induced Eilenberg–Moore algebra as in (5.21).

Proof There is a direct way to prove the result, and an indirect way via bialgebras. We shall present both proofs.

For the direct proof, first transform e into a G-coalgebra $\overline{e}$ as in

$$\overline{e} = \mathcal{F}_{\mathcal{EM}}(e) = \Big(T(X) \xrightarrow{T(e)} TGT(X) \xrightarrow{\rho_{T(X)}} GT^2(X) \xrightarrow{G(\mu_X)} GT(X)\Big),$$

where $\mathcal{F}_{\mathcal{EM}}\colon \mathbf{CoAlg}(GT) \to \mathbf{CoAlg}(\mathcal{EM}(G))$ is the state extension functor defined in (5.22). This coalgebra satisfies $\overline{e} \circ \eta = e$. By finality we get a unique coalgebra map $f\colon T(X) \to Z$ with $\zeta \circ f = G(f) \circ \overline{e}$. One can then show that $s = f \circ \eta\colon X \to Z$ is the required map.

$$\begin{aligned}
G(\beta) \circ GT(s) \circ e &= G(\beta) \circ GT(f) \circ GT(\eta) \circ e \\
&\overset{(*)}{=} G(f) \circ G(\mu) \circ GT(\eta) \circ e \\
&= G(f) \circ e \\
&= G(f) \circ \overline{e} \circ \eta \\
&= \zeta \circ f \circ \eta \\
&= \zeta \circ s
\end{aligned}$$

The marked equation holds because $\beta \circ T(f) = f \circ \mu$ by uniqueness of coalgebra maps $(\rho \circ T(\overline{e})) \to \zeta$.

The indirect proof uses the following bijective correspondence:

$$\begin{array}{c}
\text{maps } \; X \xrightarrow{c} GT(X) \\
\hline\hline
\text{bialgebras } \; T^2(X) \xrightarrow[\mu]{} T(X) \xrightarrow[d]{} GT(X)
\end{array}\; . \qquad (5.30)$$

This correspondence sends e to $\overline{e}$ as defined above. In the reverse direction d is mapped to $\overline{d} = d \circ \eta$.

Corollary 5.5.4 says that the pair (β, ζ) is a final bialgebra. Hence we get a unique map of bialgebras $(\mu, \overline{e}) \to (\beta, \zeta)$. It involves a map $f\colon T(X) \to Z$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

which is a map both of algebras and of coalgebras. Then $s = f \circ \eta$ is the desired map, since

$$\begin{aligned} G(\beta) \circ GT(s) \circ e &= G(\beta) \circ GT(f \circ \eta) \circ \overline{e} \circ \eta \\ &= G(f) \circ G(\mu) \circ GT(\eta) \circ \overline{e} \circ \eta \\ &= G(f) \circ \overline{e} \circ \eta \\ &= \zeta \circ f \circ \eta \\ &= \zeta \circ s. \end{aligned}$$ ❑

In this result we have used letters ‘e’ and ‘s’ for ‘equation’ and ‘solution’. Indeed, maps of the form $X \to GT(X)$ can be understood as abstract recursive equations, with maps $s\colon X \to Z$ as solutions in a final coalgebra. This view is elaborated in a series of papers (see e.g. [360, 11, 25, 26, 27]) formalising and extending in coalgebraic terms earlier work of especially Elgot, Bloom and Esik (see [75]). Of particular interest in this setting is the so-called free iterative monad F^∞ on a functor F, where $F^\infty(X)$ is the final coalgebra of the functor $X + F(-)$. The connections between this monad, distributive laws and $\mathcal{EM}$-coinduction are elaborated in [246].

Exercises

5.5.1 This exercise gives a more symmetric version of the situation described in the beginning of this section, dealing not with a monad and a functor but with a monad and a comonad. So let $T = (T, \eta, \mu)$ be a monad and $S = (S, \varepsilon, \delta)$ a comonad, on the same category $\mathbb{C}$. A distributive law of the monad T over the comonad S is a natural transformation $\rho\colon TS \Rightarrow ST$ which commutes both with the monad and with the comonad structure, as in

$$\begin{tikzcd} S \arrow[r, equal] \arrow[d, "\eta"'] & S \arrow[d, "S(\eta)"] \\ TS \arrow[r, "\rho"'] \arrow[d, "T(\varepsilon)"'] & ST \arrow[d, "\varepsilon"] \\ T \arrow[r, equal] & T \end{tikzcd} \qquad \begin{tikzcd} T^2S \arrow[r, "T(\rho)"] \arrow[d, "\mu"'] & TST \arrow[r, "\rho"] & ST^2 \arrow[d, "S(\mu)"] \\ TS \arrow[rr, "\rho"'] \arrow[d, "T(\delta)"'] & & ST \arrow[d, "\delta"] \\ TS^2 \arrow[r, "\rho"'] & STS \arrow[r, "S(\rho)"'] & S^2T \end{tikzcd}$$

1. Define a suitable category of bialgebras $\mathbf{BiAlg}(TS \overset{\rho}{\Rightarrow} ST)$, together with two forgetful functors:

$$\mathcal{EM}(T) \longleftarrow \mathbf{BiAlg}(TS \overset{\rho}{\Rightarrow} ST) \longrightarrow \mathcal{EM}(S).$$

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

2. Define liftings of the monad T to a monad $\mathcal{EM}(T)$ and of the comonad S to a comonad $\mathcal{EM}(S)$ in the following diagrams:

$$\begin{tikzcd}
\mathcal{EM}(S) \arrow[r, "\mathcal{EM}(T)"] \arrow[d] & \mathcal{EM}(S) \arrow[d] \\
\mathbb{C} \arrow[r, "T"'] & \mathbb{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{EM}(T) \arrow[r, "\mathcal{EM}(S)"] \arrow[d] & \mathcal{EM}(T) \arrow[d] \\
\mathbb{C} \arrow[r, "S"'] & \mathbb{C}
\end{tikzcd}$$

3. Strengthen the previous point by proving that there are bijective correspondences between distributive laws and liftings of (co)monads to (co)monads in

$$\begin{array}{c}
\text{distr. laws } \; TS \overset{\rho}{\Longrightarrow} ST \\
\hline\hline
\begin{tikzcd}
\mathcal{EM}(S) \arrow[r, "L"] \arrow[d] & \mathcal{EM}(S) \arrow[d] \\
\mathbb{C} \arrow[r, "T"] & \mathbb{C}
\end{tikzcd}
\end{array}
\qquad
\begin{array}{c}
\text{distr. laws } \; TS \overset{\rho}{\Longrightarrow} ST \\
\hline\hline
\begin{tikzcd}
\mathcal{EM}(T) \arrow[r, "R"] \arrow[d] & \mathcal{EM}(T) \arrow[d] \\
\mathbb{C} \arrow[r, "S"] & \mathbb{C}
\end{tikzcd}
\end{array}$$

4. Show that there are isomorphisms of categories

$$\mathcal{EM}(\mathcal{EM}(T)) \cong \mathbf{BiAlg}(TS \overset{\rho}{\Rightarrow} ST) \cong \mathcal{EM}(\mathcal{EM}(S)).$$

5. Prove that the (downwards pointing) forgetful functors have adjoints in

$$\begin{tikzcd}
& \mathbf{BiAlg}(TS \overset{\rho}{\Rightarrow} ST) \arrow[dl] \arrow[dr] & \\
\mathcal{EM}(T) \arrow[ur, bend right] \arrow[ur, phantom, "\dashv"] & & \mathcal{EM}(S) \arrow[ul, bend left] \arrow[ul, phantom, "\dashv"']
\end{tikzcd}$$

6. Conclude that if the category $\mathbb{C}$ has a final object 1 and a initial object 0, then, in the category of bialgebra one has both

$$\begin{cases}
\text{final object:} & TS(1) \xrightarrow{\rho} ST(1) \xrightarrow{S(!)} S(1) \xrightarrow{\delta} S^2(1) \\
\text{initial object:} & T^2(0) \xrightarrow{\mu} T(0) \xrightarrow{T(!)} TS(0) \xrightarrow{\rho} ST(0).
\end{cases}$$

7. Let $G\colon \mathbb{C} \to \mathbb{C}$ be an arbitrary functor, with cofree comonad G^∞. Show that each $\mathcal{EM}$-law $TG \Rightarrow GT$ gives rise to a distributive law $TG^\infty \Rightarrow G^\infty T$.

5.5.2 Recall the operation $\mathsf{merge}\colon A^\infty \times A^\infty \to A^\infty$ on sequences from Section 1.2. Define a GSOS-law corresponding to the three rules describing the behaviour of merge, as for sequential composition in Example 5.5.9.1.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006

5.5.3 Elaborate the details of the bijective correspondence (5.30). Show that it can be used to construct a functor $\mathbf{CoAlg}(GT) \to \mathbf{BiAlg}(TG \overset{\rho}{\Rightarrow} GT)$.

5.5.4 Let $G\colon \mathbb{C} \to \mathbb{C}$ be a functor with final coalgebra $\zeta\colon Z \xrightarrow{\cong} G(Z)$, and let $T\colon \mathbb{C} \to \mathbb{C}$ be a monad with an $\mathcal{EM}$-law $\rho\colon TG \Rightarrow GT$.

1. Recall from Exercise 5.2.9 that $T^Z = T(Z + (-))$ is also a monad. Transform ρ into an $\mathcal{EM}$-law $\rho^Z\colon T^Z G \Rightarrow GT^Z$. (This works for any G-coalgebra on Z, not necessarily the final one.)
2. Let $\beta\colon T(Z) \to Z$ be the Eilenberg–Moore algebra induced by ρ, as in (5.21). Check that the algebra induced by ρ^Z is $\beta \circ T([\mathrm{id}, \mathrm{id}])\colon T(Z + Z) \to Z$.
3. Deduce the following '$\mathcal{EM}$-corecursion' principle: for each map $e\colon X \to T(Z + X)$ there is a unique map $s\colon X \to Z$ in

$$\begin{tikzcd}
GT(Z+X) \arrow[r, "GT({[\mathrm{id}, s]})"] & GT(Z) \arrow[d, "G(\beta)"] \\
& G(Z) \\
X \arrow[uu, "e"] \arrow[r, "s"'] & Z \arrow[u, "\zeta"', "\cong"]
\end{tikzcd}$$

4. Conclude that this yields the dual of recursion, as described in Proposition 2.4.7, when T is the identity monad.
5. Is there also a strengthened form of '$\mathcal{EM}$-induction', dual to point (3), involving a comonad and a suitable distributive law?

5.5.5 Let A be a set and L a complete lattice, and consider the deterministic automaton functor $G(X) = X^A \times L$. What follows comes essentially from [58].

1. Use Exercise 5.4.4 to construct an $\mathcal{EM}$-law $\rho\colon \mathcal{P}G \Rightarrow G\mathcal{P}$, where $\mathcal{P}$ is the powerset monad.
2. Recall from Proposition 2.3.5 that the final G-coalgebra is $L^{A^\star}$. Describe the induced $\mathcal{P}$-algebra $\beta\colon \mathcal{P}(L^{A^\star}) \to L^{A^\star}$ according to (5.21).
3. Prove that via this $\mathcal{EM}$-law and the $\mathcal{EM}$-coinduction principle from Proposition 5.5.10 one can obtain trace semantics for non-deterministic automata, when L is the two-element lattice 2.

Downloaded from https://www.cambridge.org/core. Radboud University Nijmegen, on 31 Aug 2018 at 10:17:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/CBO9781316823187.006