6

Invariants and Assertions

The second important notion in the logic of coalgebras, besides bisimularity, is invariance. Whereas a bisimulation is a binary relation on state spaces that is closed under transitions, an invariant is a predicate, or unary relation if you like, on a state space which is closed under transitions. This means, once an invariant holds, it will continue to hold, no matter which state transition operations are applied. That is: coalgebras maintain their invariants.

Invariants are important in the description of systems, because they often express certain implicit assumptions, such as this integer value will always be non-zero (so that dividing by the integer is safe), or the contents of this tank will never be below a given minimum value. Thus, invariants are 'safety properties', which express that something bad will never happen.

This chapter will introduce a general notion of invariant for a coalgebra, via predicate lifting. Predicate lifting is the unary analogue of relation lifting. First it will be introduced for polynomial functors on sets, but later also for more general functors, using the categorical logic introduced in Section 4.3. Various properties of invariants are established, in particular their intimate relation to subcoalgebras. An important application of invariants lies in a generic temporal logic for coalgebras, involving henceforth $\Box$ and eventually $\Diamond$ operators on predicates (on a state space for a coalgebra), which will be introduced in Section 6.4. It uses $\Box P$ as the greatest invariant that is contained in the predicate P. Further, invariants play a role in the construction of equalisers and products for coalgebras.

The operator $\Box P$ involves closure of the predicate P under *all* operations of a coalgebra. In many situations one also likes to express closure under specific operations only. This can be done via the modal logic for coalgebras introduced in Section 6.5. The modal operators are themselves described via a functor as signature, and the meaning of the operators is given by a suitable natural transformation. In examples it is shown how to use such operations to describe coalgebras satisfying certain logical assertions.

The semantics of such assertions – in a set-theoretic context – is the topic of the second part of this chapter, starting in Section 6.6. First, the relatively familiar situation of algebras satisfying assertions is reviewed. In the style of Lawvere's functorial semantics it is shown that monad/functor algebras correspond to finite product preserving functors from a Kleisli category to **Sets**. Subsequently, Section 6.7 describes assertions as axioms for a monad (or functor). This leads to an important conceptual result, namely that functor algebras satisfying certain assertions correspond to Eilenberg–Moore algebras of an associated (quotient) monad. A dual result for coalgebras is described in Section 6.8: functor coalgebras satisfying assertions correspond to Eilenberg–Moore coalgebras of an associated (subset) comonad. This will be illustrated by actually determining such subset comonads in concrete cases. In order to obtain these results we use earlier work on congruences and invariants, in order to turn the given axioms into suitable least congruences and greatest invariants, so that quotient and subset constructions can be applied. The chapter (and book) closes with Section 6.9, illustrating coalgebras and assertions in the context of specification of classes, as in object-oriented programming languages.

6.1 Predicate Lifting

This section will introduce the technique of predicate lifting. It will be used in the next section in the definition of invariance for (co)algebras. Here we will first establish various elementary, useful properties of predicate lifting. Special attention will be paid to the left adjoint to predicate lifting, called predicate lowering.

Predicate lifting for a Kripke polynomial functor $F\colon \mathbf{Sets} \to \mathbf{Sets}$ is an operation $\mathrm{Pred}(F)$ which sends a predicate $P \subseteq X$ on a set X to a 'lifted' predicate $\mathrm{Pred}(F)(P) \subseteq F(X)$ on the result of applying the functor F to X. The idea is that P should hold on all occurrences of X inside $F(X)$, as suggested in

$$\begin{array}{rl} F(X) \;=\; & \boxed{\;\cdots\quad X \quad\cdots\quad A \quad\cdots\quad X \quad\cdots\;} \\[6pt] \mathrm{Pred}(F)(P) \;=\; & \qquad\quad\;\; \Big| \qquad\qquad\qquad\qquad\quad \Big| \\ & \qquad\quad\;\; P \qquad\qquad\qquad\qquad\quad P \end{array} \tag{6.1}$$

The formal definition proceeds by induction on the structure of the functor F, just as for relation lifting in Definition 3.1.1.

Definition 6.1.1 (Predicate lifting) Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a Kripke polynomial functor, and let X be an arbitrary set. The mapping $\mathrm{Pred}(F)$ which sends a predicate $P \subseteq X$ to a 'lifted' predicate $\mathrm{Pred}(F)(P) \subseteq F(X)$ is defined by induction on the structure of F, in accordance with the steps in Definition 2.2.1.

1. For the identity functor $\mathrm{id}\colon \mathbf{Sets} \to \mathbf{Sets}$ we have

$$\mathrm{Pred}(\mathrm{id})(P) = P.$$

2. For a constant functor $A\colon \mathbf{Sets} \to \mathbf{Sets}$, given by $Y \mapsto A$,

$$\mathrm{Pred}(A)(P) = \top_A = (A \subseteq A).$$

3. For a product functor,

$$\begin{aligned}&\mathrm{Pred}(F_1 \times F_2)(P)\\ &\;= \{(u, v) \in F_1(X) \times F_2(X) \mid \mathrm{Pred}(F_1)(P)(u) \wedge \mathrm{Pred}(F_2)(P)(v)\}.\end{aligned}$$

4. For a set-indexed coproduct,

$$\mathrm{Pred}(\textstyle\coprod_{i\in I} F_i)(P) = \bigcup_{j\in I}\{\kappa_j(u) \in \coprod_{i\in I} F_i(X) \mid \mathrm{Pred}(F_j)(P)(u)\}.$$

5. For an exponent functor,

$$\mathrm{Pred}(F^A)(P) = \{f \in F(X)^A \mid \forall a \in A.\, \mathrm{Pred}(F)(P)(f(a))\}.$$

6. For powerset functor,

$$\mathrm{Pred}(\mathcal{P}(F))(P) = \{U \subseteq X \mid \forall u \in U.\, \mathrm{Pred}(F)(P)(u)\}.$$

   This same formula will be used for a *finite* powerset $\mathcal{P}_{\mathrm{fin}}(F)$.

First we show that predicate and relation lifting are closely related.

Lemma 6.1.2 1. *Relation lifting* $\mathrm{Rel}(F)$ *and predicate lifting* $\mathrm{Pred}(F)$ *for a polynomial functor* $F\colon \mathbf{Sets} \to \mathbf{Sets}$ *are related in the following way:*

$$\mathrm{Rel}(F)(\textstyle\coprod_{\Delta_X}(P)) = \coprod_{\Delta_{F(X)}}(\mathrm{Pred}(F)(P)),$$

where $\Delta_X = \langle \mathrm{id}, \mathrm{id}\rangle\colon X \to X \times X$ *is the diagonal, and so* $\coprod_{\Delta_X}(P) = \{\Delta_X(x) \mid x \in P\} = \{(x, x) \mid x \in P\}$.

2. *Similarly,*

$$\mathrm{Pred}(F)(\textstyle\coprod_{\pi_i}(R)) = \coprod_{\pi_i}(\mathrm{Rel}(F)(R)),$$

where $\coprod_{\pi_1}(R) = \{x_1 \mid \exists x_2.\, R(x_1, x_2)\}$ *is the domain of the relation* R*, and* $\coprod_{\pi_2}(R) = \{x_2 \mid \exists x_1.\, R(x_1, x_2)\}$ *is its codomain.*

3. *As a result, predicate lifting can be expressed in terms of relation lifting:*

$$\mathrm{Pred}(F)(P) \;=\; \textstyle\coprod_{\pi_i} (\mathrm{Rel}(F)(\coprod_{\Delta}(P))),$$

for both $i = 1$ *and* $i = 2$.

Proof 1. and 2. By induction on the structure of F.

3. Since $\coprod_{\pi_i} \circ \coprod_{\Delta} = \coprod_{\pi_i \circ \Delta} = \coprod_{\mathrm{id}} = \mathrm{id}$, we use the previous point to get

$$\mathrm{Pred}(F)(P) \;=\; \textstyle\coprod_{\pi_i} \coprod_{\Delta} \mathrm{Pred}(F)(P) \;=\; \coprod_{\pi_i} \mathrm{Rel}(F)(\coprod_{\Delta} P). \qquad \square$$

Despite this last result, it is useful to study predicate lifting on its own, because it has some special properties that relation lifting does not enjoy – such as preservation of intersections (see the next result), and thus existence of a left adjoint; see Section 6.1.1.

Lemma 6.1.3 *Predicate lifting* Pred(F) *with respect to a Kripke polynomial functor* F : **Sets** → **Sets** *satisfies the following properties.*

1. *It preserves arbitrary intersections: for every collection of predicates* $(P_i \subseteq X)_{i \in I}$,

$$\mathrm{Pred}(F)(\textstyle\bigcap_{i\in I} P_i) \;=\; \displaystyle\bigcap_{i\in I} \mathrm{Pred}(F)(P_i).$$

A special case (intersection over $I = \emptyset$*) worth mentioning is preservation of truth:*

$$\mathrm{Pred}(F)(\top_X) \;=\; \top_{F(X)}.$$

Another consequence is that predicate lifting is monotone:

$$P \subseteq Q \implies \mathrm{Pred}(F)(P) \subseteq \mathrm{Pred}(F)(Q).$$

2. *It preserves inverse images: for a function* $f : X \to Y$ *and predicate* $Q \subseteq Y$,

$$\mathrm{Pred}(F)(f^{-1}(Q)) \;=\; F(f)^{-1}(\mathrm{Pred}(F)(Q)).$$

3. *Predicate lifting also preserves direct images: for* $f : X \to Y$ *and* $P \subseteq X$,

$$\mathrm{Pred}(F)(\textstyle\coprod_f(P)) \;=\; \coprod_{F(f)}(\mathrm{Pred}(F)(P)).$$

Proof 1. and 2. By induction on the structure of F.

3. This is easily seen via the link with relation lifting:

$$\begin{array}{lll}
& \mathrm{Pred}(F)(\coprod_f P) & \\
= & \coprod_{\pi_1} \mathrm{Rel}(F)(\coprod_{\Delta} \coprod_f P) & \text{by Lemma 6.1.2.3} \\
= & \coprod_{\pi_1} \mathrm{Rel}(F)(\coprod_{f\times f} \coprod_{\Delta} P) & \\
= & \coprod_{\pi_1} \coprod_{F(f)\times F(f)} \mathrm{Rel}(F)(\coprod_{\Delta} P) & \text{by Lemma 3.2.2.2} \\
= & \coprod_{F(f)} \coprod_{\pi_1} \mathrm{Rel}(F)(\coprod_{\Delta} P) & \\
= & \coprod_{F(f)}(\mathrm{Pred}(F)(P)) & \text{again by Lemma 6.1.2.3.}
\end{array} \qquad \square$$

Corollary 6.1.4 *Predicate lifting* Pred(F) *may be described as a natural transformation, both with the contra- and co-variant powerset functor: by Lemma 6.1.3.2 it forms a map*

$$\mathcal{P} \xRightarrow{\mathrm{Pred}(F)} \mathcal{P}F \qquad \text{for the contravariant} \qquad \mathbf{Sets}^{\mathrm{op}} \xrightarrow{\mathcal{P}} \mathbf{Sets} \tag{6.2}$$

and also, by Lemma 6.1.3.3,

$$\mathcal{P} \xRightarrow{\mathrm{Pred}(F)} \mathcal{P}F \qquad \text{for the covariant} \qquad \mathbf{Sets} \xrightarrow{\mathcal{P}} \mathbf{Sets}. \tag{6.3}$$

Predicate liftings are described as natural transformations (6.2) with respect to the contravariant powerset functor in [373], as starting point for temporal logics, as in Section 6.4.

Relation lifting for polynomial functors is functorial, on the category **Rel** = Rel(**Sets**) of relations; see Definition 3.2.3 and more generally Definition 4.3.3. Similarly, predicate lifting yields a functor on the category **Pred** = Pred(**Sets**) of predicates on sets. Explicitly, the category **Pred** has subsets $P \subseteq X$ as objects. A morphism $(P \subseteq X) \longrightarrow (Q \subseteq Y)$ consists of a function $f : X \to Y$ with $P(x) \implies Q(f(x))$ for all $x \in X$. This amounts to the existence of the necessarily unique dashed map in

$$\begin{tikzcd}
P \arrow[r, dashed] \arrow[d, tail] & Q \arrow[d, tail] \\
X \arrow[r, "f"'] & Y
\end{tikzcd}$$

Equivalently, $P \subseteq f^{-1}(Q)$, or $\coprod_f(P) \subseteq Q$, by the correspondence (2.15). There is a forgetful functor **Pred** → **Sets** which is so obvious that it does not get its own name.

Corollary 6.1.5 *Predicate lifting for a polynomial functor* F : **Sets** → **Sets** *yields a functor* Pred(F) *in a commuting square:*

$$\begin{tikzcd}
\mathbf{Pred} \arrow[r, "\mathrm{Pred}(F)"] \arrow[d] & \mathbf{Pred} \arrow[d] \\
\mathbf{Sets} \arrow[r, "F"] & \mathbf{Sets}
\end{tikzcd}$$

Proof Given a map $f : (P \subseteq X) \to (Q \subseteq Y)$ in the category **Pred** we get a map $F(f) : (\mathrm{Pred}(F)(P) \subseteq F(X)) \to (\mathrm{Pred}(F)(Q) \subseteq F(Y))$. This works as follows, using Lemma 6.1.3:

$$\begin{aligned} P \subseteq f^{-1}(Q) &\Longrightarrow \mathrm{Pred}(F)(P) \subseteq \mathrm{Pred}(F)(f^{-1}(Q)) \\ &\qquad \text{since predicate lifting is monotone} \\ &\Longrightarrow \mathrm{Pred}(F)(P) \subseteq F(f)^{-1}(\mathrm{Pred}(F)(Q)) \\ &\qquad \text{since predicate lifting commutes with substitution.} \end{aligned}$$ ❑

The next result is the analogue for predicate lifting of Lemma 3.3.1 for relation lifting. It relies on considering predicates as sets themselves and may be understood as: predicate lifting commutes with comprehension $\{-\}$; see Lemma 6.3.9. This is described in a more general logical setting in Section 6.1.2 and in [218]. But in the current set-theoretic context the connection between predicate lifting and functor application is extremely simple.

Lemma 6.1.6 *Let F be a Kripke polynomial functor. Applying predicate lifting* $\mathrm{Pred}(F)$ *to a predicate $m\colon P \rightarrowtail X$ is the same as applying the functor F itself to the domain P, in*

$$\begin{tikzcd} \mathrm{Pred}(F)(P) \arrow[rr, equal] \arrow[dr, tail] & & F(P) \arrow[dl, tail, "F(m)"] \\ & F(X) & \end{tikzcd}$$

Note that from the fact that Kripke polynomial functors preserve weak pullbacks (Proposition 4.2.6) we can already conclude that $F(m)$ is a mono; see Lemma 4.2.2.

Proof Formally, one proves for $z \in F(X)$,

$$z \in \mathrm{Pred}(F)(P) \Longleftrightarrow \exists! z' \in F(P).\, F(m)(z') = z.$$

This is obtained by induction on the structure of the polynomial functor F. ❑

6.1.1 Predicate Lowering as Liftings Left Adjoint

We continue this section with a Galois connection involving predicate lifting. In the next section we shall use predicate lifting for a nexttime operator $\bigcirc$ in a temporal logic of coalgebras. There is also a lasttime operator $\underleftarrow{\bigcirc}$ (see Section 6.4.1), for which we shall need this left adjoint (or lower Galois adjoint) to predicate lifting $\mathrm{Pred}(F)$. We shall write this left (or lower) adjoint as $\underleftarrow{\mathrm{Pred}}(F)$ and shall call it 'predicate lowering'. By Lemma 6.1.3.1, predicate lifting preserves arbitrary intersections, and thus has such a left adjoint for abstract reasons; see e.g. [348] or [280, I, theorem 4.2]. But the adjoint can also be defined concretely, by induction on the structure of the functor. This is what we shall do.

Proposition 6.1.7 (From [238, 243]) *Predicate lifting for a Kripke polynomial functor* $F\colon \mathbf{Sets} \to \mathbf{Sets}$ *forms a monotone function* $\mathrm{Pred}(F)\colon \mathcal{P}(X) \to \mathcal{P}(F(X))$ *between powerset posets. In the opposite direction there is also an operation* $\underleftarrow{\mathrm{Pred}}(F)\colon \mathcal{P}(F(X)) \to \mathcal{P}(X)$ *satisfying*

$$\underleftarrow{\mathrm{Pred}}(F)(Q) \subseteq P \iff Q \subseteq \mathrm{Pred}(F)(P).$$

Hence $\underleftarrow{\mathrm{Pred}}(F)$ *is the left adjoint of* $\mathrm{Pred}(F)$ *in a Galois connection* $\underleftarrow{\mathrm{Pred}}(F) \dashv \mathrm{Pred}(F)$.

Proof One can define $\underleftarrow{\mathrm{Pred}}(F)(Q) \subseteq X$ for $Q \subseteq F(X)$ by induction on the structure of the functor F.

1. For the identity functor $\mathrm{id}\colon \mathbf{Sets} \to \mathbf{Sets}$,

$$\underleftarrow{\mathrm{Pred}}(\mathrm{id})(Q) = Q.$$

2. For a constant functor A,

$$\underleftarrow{\mathrm{Pred}}(A)(Q) = \bot_A = (\emptyset \subseteq A).$$

3. For a product functor,

$$\begin{aligned} \underleftarrow{\mathrm{Pred}}(F_1 \times F_2)(Q) &= \underleftarrow{\mathrm{Pred}}(F_1)(\textstyle\coprod_{\pi_1}(Q)) \cup \underleftarrow{\mathrm{Pred}}(F_2)(\coprod_{\pi_2}(Q)) \\ &= \underleftarrow{\mathrm{Pred}}(F_1)(\{u \in F_1(X) \mid \exists v \in F_2(X).\, Q(u,v)\}) \\ &\qquad \cup \underleftarrow{\mathrm{Pred}}(F_2)(\{v \in F_2(X) \mid \exists u \in F_1(X).\, Q(u,v)\}). \end{aligned}$$

4. For a set-indexed coproduct functor,

$$\begin{aligned} \underleftarrow{\mathrm{Pred}}(\textstyle\coprod_{i\in I} F_i)(Q) &= \textstyle\bigcup_{i\in I} \underleftarrow{\mathrm{Pred}}(F_i)(\kappa_i^{-1}(Q)) \\ &= \textstyle\bigcup_{i\in I} \underleftarrow{\mathrm{Pred}}(F_i)(\{u \mid Q(\kappa_i(u))\}). \end{aligned}$$

5. For an exponent functor,

$$\underleftarrow{\mathrm{Pred}}(F^A)(Q) = \underleftarrow{\mathrm{Pred}}(F)(\{f(a) \mid a \in A \text{ and } f \in F(X)^A \text{ with } Q(f)\}).$$

6. For a powerset functor,

$$\underleftarrow{\mathrm{Pred}}(\mathcal{P}(F))(Q) = \underleftarrow{\mathrm{Pred}}(F)(\textstyle\bigcup Q).$$

This same formula will be used for a *finite* powerset $\mathcal{P}_{\mathrm{fin}}(F)$. ❑

Being a left adjoint means that functions $\underleftarrow{\mathrm{Pred}}(F)$ preserve certain ‘colimit’ structures.

Lemma 6.1.8 *Let F be a Kripke polynomial functor. Its opposite predicate lifting operations* $\underleftarrow{\mathrm{Pred}}(F)\colon \mathcal{P}(F(X)) \to \mathcal{P}(X)$ *preserve:*

1. *Unions $\bigcup$ of predicates*
2. *Direct images $\coprod$, in the sense that for $f\colon X \to Y$,*

$$\overleftarrow{\mathrm{Pred}}(F)(\textstyle\coprod_{F(f)}(Q)) \;=\; \textstyle\coprod_f \overleftarrow{\mathrm{Pred}}(F)(Q).$$

This means that $\overleftarrow{\mathrm{Pred}}(F)$ forms a natural transformation:

$$\mathcal{P}F \xRightarrow{\;\overleftarrow{\mathrm{Pred}}(F)\;} \mathcal{P} \qquad \textit{for the covariant} \qquad \mathbf{Sets} \xrightarrow{\;\mathcal{P}\;} \mathbf{Sets}$$

as in Corollary 6.1.4.

Proof 1. This is a general property of left (Galois) adjoints, as illustrated in the beginning of Section 2.5.

2. One can use a ‘composition of adjoints’ argument, or reason directly with the adjunctions:

$$\begin{array}{l}
\overleftarrow{\mathrm{Pred}}(F)(\coprod_{F(f)}(Q)) \;\subseteq\; P \\
\iff \coprod_{F(f)}(Q) \;\subseteq\; \mathrm{Pred}(F)(P) \\
\iff Q \;\subseteq\; F(f)^{-1}\mathrm{Pred}(F)(P) = \mathrm{Pred}(F)(f^{-1}(P)) \text{ by Lemma 6.1.3.2} \\
\iff \overleftarrow{\mathrm{Pred}}(F)(Q) \;\subseteq\; f^{-1}(Q) \\
\iff \coprod_f \overleftarrow{\mathrm{Pred}}(F)(Q) \;\subseteq\; P.
\end{array}$$

❑

This left adjoint to predicate lifting gives rise to a special kind of mapping, written as sts, from an arbitrary Kripke polynomial functor F to the powerset functor $\mathcal{P}$. The maps $\mathrm{sts}_X\colon F(X) \to \mathcal{P}(X)$ collect the states inside $F(X)$ – as suggested in Diagram (6.1). With this mapping each coalgebra can be turned into an unlabelled transition system.

Proposition 6.1.9 *For a Kripke polynomial functor F and an arbitrary set X, write $\mathrm{sts}_X\colon F(X) \to \mathcal{P}(X)$ for the following composite:*

$$\mathrm{sts}_X \stackrel{def}{=} \Big(F(X) \xrightarrow{\;\{-\}\;} \mathcal{P}(F(X)) \xrightarrow{\;\overleftarrow{\mathrm{Pred}}(F)\;} \mathcal{P}(X)\Big).$$

These sts *maps form a natural transformation $F \Rightarrow \mathcal{P}$: for each function $f\colon X \to Y$ the following diagram commutes:*

$$\begin{tikzcd}
F(X) \arrow[r, "\mathrm{sts}_X"] \arrow[d, "F(f)"'] & \mathcal{P}(X) \arrow[d, "\mathcal{P}(f) = \coprod_f"] \\
F(Y) \arrow[r, "\mathrm{sts}_Y"'] & \mathcal{P}(Y)
\end{tikzcd}$$

Proof We need only to prove naturality. For $u \in F(X)$:

$$\begin{aligned}
\textstyle\coprod_f (\mathrm{sts}_X(u)) &= \textstyle\coprod_f \underleftarrow{\mathrm{Pred}}(F)(\{u\}) \\
&= \underleftarrow{\mathrm{Pred}}(F)(\textstyle\coprod_{F(f)}\{u\}) \quad \text{by Lemma 6.1.8.2} \\
&= \underleftarrow{\mathrm{Pred}}(F)(\{F(f)(u)\}) \\
&= \mathrm{sts}_Y(F(f)(u)).
\end{aligned}$$ ❑

Example 6.1.10 Consider the deterministic automaton functor $(-)^A \times B$. The states contained in $(h, b) \in X^A \times B$ can be computed by following the inductive clauses in the proof of Proposition 6.1.7:

$$\begin{aligned}
\mathrm{sts}(h, b) &= \underleftarrow{\mathrm{Pred}}((-)^A \times B)(\{(h, b)\}) \\
&= \underleftarrow{\mathrm{Pred}}((-)^A)(h) \cup \underleftarrow{\mathrm{Pred}}(B)(b) \\
&= \underleftarrow{\mathrm{Pred}}(\mathrm{id})(\{h(a) \mid a \in A\}) \cup \emptyset \\
&= \{h(a) \mid a \in A\} \;\subseteq\; X.
\end{aligned}$$

By combining Propositions 6.1.9 and 2.5.5 we obtain a functor

$$\mathbf{CoAlg}(F) \xrightarrow{\;\mathrm{sts}\,\circ\,(-)\;} \mathbf{CoAlg}(\mathcal{P}) \tag{6.4}$$

from coalgebras of a polynomial functor to unlabelled transition systems. This translation will be further investigated in Section 6.4. As can be seen in Exercise 6.1.3, the translation removes much of the structure of the coalgebra. However, it makes the intuitive idea precise that states of a coalgebra can make transitions.

6.1.2 Predicate Lifting, Categorically

We have defined relation lifting concretely for polynomial functors, and more abstractly for functors on a category $\mathbb{C}$ carrying a logical factorisation system; see Definition 4.4.1. Here we shall now give a similarly abstract definition of predicate lifting. It involves the category $\mathrm{Pred}(\mathbb{C})$ of predicates with respect to such a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$ on $\mathbb{C}$; see Definition 4.3.3.

Definition 6.1.11 Let $F\colon \mathbb{C} \to \mathbb{C}$ be an endofunctor on a category $\mathbb{C}$ with a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$. For a predicate $(m\colon U \rightarrowtail X) \in \mathrm{Pred}(\mathbb{C})$, where $m \in \mathfrak{M}$, we define a new predicate $\mathrm{Pred}(F)(U) \rightarrowtail F(X)$ on $F(X)$ via factorisation in

$$\begin{tikzcd}
F(U) \arrow[rr, twoheadrightarrow, "\mathfrak{e}(F(m))"] \arrow[dr, "F(m)"'] & & \mathrm{Pred}(F)(U) \arrow[dl, rightarrowtail, "\mathfrak{m}(F(m))"] \\
& F(X) &
\end{tikzcd}$$

This yields a functor

$$
\begin{tikzcd}
\mathrm{Pred}(\mathbb{C}) \arrow[r, "\mathrm{Pred}(F)"] \arrow[d] & \mathrm{Pred}(\mathbb{C}) \arrow[d] \\
\mathbb{C} \arrow[r, "F"] & \mathbb{C}
\end{tikzcd}
$$

since for a map of predicates $f\colon (m\colon U \rightarrowtail X) \to (n\colon V \rightarrowtail X)$ the map $F(f)$ is a map $\mathrm{Pred}(F)(U) \to \mathrm{Pred}(F)(V)$ via diagonal-fill-in

$$
\begin{tikzcd}
F(U) \arrow[r, twoheadrightarrow, "\mathfrak{e}(F(m))"] \arrow[d] & \mathrm{Pred}(F)(U) \arrow[d, rightarrowtail, "\mathfrak{m}(F(m))"] \arrow[ddl, dashed] \\
F(V) \arrow[d] & F(X) \arrow[d, "F(f)"] \\
\mathrm{Pred}(F)(V) \arrow[r, rightarrowtail, "\mathfrak{m}(F(n))"'] & F(Y)
\end{tikzcd}
$$

where the vertical map $F(U) \to F(V)$ on the left is obtained by applying F to the map $U \to V$ that exists since f is a map of predicates (see in Definition 4.3.3).

Some of the properties we have seen earlier on in this section also hold for the abstract form of predicate lifting, under additional assumptions.

Proposition 6.1.12 *Let $F\colon \mathbb{C} \to \mathbb{C}$ be an endofunctor on a category $\mathbb{C}$ with a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$.*

1. *Predicate lifting is monotone and preserves truth $\top$.*
2. *If diagonals $\Delta = \langle \mathrm{id}, \mathrm{id}\rangle\colon X \to X \times X$ are in $\mathfrak{M}$ – that is, if internal and external equality coincide (see Remark 4.3.6) – then $\coprod_\Delta \mathrm{Pred}(F) = \mathrm{Rel}(F) \coprod_\Delta$, as in the square on the left below:*

$$
\begin{tikzcd}
\mathrm{Pred}(X) \arrow[r, "\coprod_\Delta"] \arrow[d, "\mathrm{Pred}(F)"'] & \mathrm{Rel}(X) \arrow[d, "\mathrm{Rel}(F)"] \\
\mathrm{Pred}(F(X)) \arrow[r, "\coprod_\Delta"'] & \mathrm{Rel}(F(X))
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathrm{Pred}(X) \arrow[d, "\mathrm{Pred}(F)"'] & \mathrm{Rel}(X) \arrow[l, "\coprod_{\pi_i}"'] \arrow[d, "\mathrm{Rel}(F)"] \\
\mathrm{Pred}(F(X)) & \mathrm{Rel}(F(X)) \arrow[l, "\coprod_{\pi_i}"]
\end{tikzcd}
$$

3. *If the functor F preserves abstract epis, then the rectangle on the right also commutes, for $i \in \{1, 2\}$.*
4. *If F preserves abstract epis, predicate lifting commutes with sums (direct images) $\coprod$, as in*

$$\mathrm{Pred}(F)(\textstyle\coprod_f(U)) = \coprod_{F(f)} \mathrm{Pred}(F)(U).$$

5. *If $\mathfrak{E} \subseteq$ SplitEpis and F preserves weak pullbacks, then predicate lifting commutes with inverse images:*

$$\mathrm{Pred}(F)(f^{-1}(V)) = F(f)^{-1}(\mathrm{Pred}(F)(V)).$$

Additionally, predicate lifting preserves meets $\wedge$ of predicates.

Proof 1. This is obvious.

2. We use some properties of images $\mathfrak{m}(-)$: for a predicate $(m\colon U \rightarrowtail X)$,

$$\begin{aligned}
\mathrm{Rel}(F)(\textstyle\coprod_\Delta(U)) &= \mathrm{Rel}(F)(\mathfrak{m}(\Delta \circ m)) \\
&= \mathrm{Rel}(F)(\Delta \circ m) && \text{since } \Delta, m \in \mathfrak{M} \\
&= \mathrm{Rel}(F)(\langle m, m\rangle) \\
&= \mathfrak{m}(\langle F(m), F(m)\rangle) \\
&= \mathfrak{m}(\Delta \circ F(m)) \\
&= \Delta \circ \mathfrak{m}(F(m)) && \text{by Exercise 4.3.1.3} \\
&= \textstyle\coprod_\Delta(\mathrm{Pred}(F)(P)).
\end{aligned}$$

3. We recall the basic constructions: for a relation $\langle r_1, r_2\rangle\colon R \rightarrowtail X \times X$,

$$\begin{tikzcd}
F(R) \arrow[rr, two heads] \arrow[dr, "{\langle F(r_1), F(r_2)\rangle}"'] & & \mathrm{Rel}(F)(R) \arrow[r, two heads] \arrow[dl, tail] & \coprod_{\pi_i} \mathrm{Rel}(F)(R) \arrow[d, tail] \\
& F(X) \times F(X) \arrow[rr, "\pi_i"'] & & F(X)
\end{tikzcd}
\qquad
\begin{tikzcd}
R \arrow[r, two heads, "e_i"] \arrow[d, tail, "{\langle r_1, r_2\rangle}"'] & \coprod_{\pi_i}(R) \arrow[d, tail] \\
X \times X \arrow[r, "\pi_i"'] & X
\end{tikzcd}$$

By assumption F preserves abstract epis. Thus $F(e_i) \in \mathfrak{E}$, which yields two dashed diagonals in the following rectangle – and thus the required equality of subobjects – since the south-east diagonal equals $F(r_i)$:

$$\begin{tikzcd}
F(R) \arrow[r, two heads] \arrow[d, two heads, "F(e_i)"'] & \mathrm{Rel}(F)(R) \arrow[r, two heads] & \coprod_{\pi_i} \mathrm{Rel}(F)(R) \arrow[dd, tail] \arrow[ddll, dashed, bend left=5] \\
F(\coprod_{\pi_i}(R)) \arrow[d, two heads] & & \\
\mathrm{Pred}(F)(\coprod_{\pi_i}(R)) \arrow[rr, tail] \arrow[uurr, dashed, bend left=5] & & F(X)
\end{tikzcd}$$

4. One first forms the sum by factorisation:

$$\begin{tikzcd}
U \arrow[r, two heads, "e"] \arrow[d, tail, "m"'] & \coprod_f(U) \arrow[d, tail] \\
X \arrow[r, "f"] & Y
\end{tikzcd}$$

The required equality follows since $F(e) \in \mathfrak{E}$, using

$$\begin{tikzcd}
F(U) \arrow[rr, twoheadrightarrow, "F(e)"] \arrow[d, twoheadrightarrow] & & F(\coprod_f(U)) \arrow[d, twoheadrightarrow] \\
\mathrm{Pred}(F)(U) \arrow[r, twoheadrightarrow] \arrow[d, rightarrowtail] & \coprod_{F(f)} \mathrm{Pred}(F)(U) \arrow[r, dashed, leftrightarrow, "\cong"] \arrow[d, rightarrowtail] & \mathrm{Pred}(F)(\coprod_f(U)) \arrow[dl, rightarrowtail] \\
F(X) \arrow[r, "F(f)"'] & F(Y) &
\end{tikzcd}$$

One sees that the south-east map $F(f \circ m) \colon F(U) \to F(Y)$ is factorised in two ways.

5. Preservation of inverse images is proven as in Proposition 4.4.3. Preservation of meets works as follows. Assume we have predicates $(m \colon U \rightarrowtail X)$ and $(n \colon V \rightarrowtail X)$ with meet given by

$$\begin{tikzcd}
P \arrow[r, rightarrowtail, "q"] \arrow[d, rightarrowtail, "p"'] \arrow[dr, rightarrowtail, "m \wedge n"] \arrow[dr, phantom, very near start, "\lrcorner"] & V \arrow[d, rightarrowtail, "n"] \\
U \arrow[r, rightarrowtail, "m"'] & X
\end{tikzcd}$$

We show that $\mathrm{Pred}(F)(P)$ is the pullback of $\mathrm{Pred}(F)(U)$ and $\mathrm{Pred}(F)(V)$, using that abstract epis split, via the maps s in

$$\begin{tikzcd}
A \arrow[drrr, bend left, "g"] \arrow[ddrr, bend right, "f"'] & & & \\
& F(P) \arrow[rr] \arrow[dr, twoheadrightarrow, "e_P"] \arrow[dd] & & F(V) \arrow[d, twoheadrightarrow, "e_V"'] \\
& & \mathrm{Pred}(F)(P) \arrow[r, "q'"] \arrow[d, "p'"'] \arrow[dr, rightarrowtail, "m_P"] & \mathrm{Pred}(F)(V) \arrow[u, bend right, "s_V"'] \arrow[d, rightarrowtail, "m_V"] \\
& F(V) \arrow[r, twoheadrightarrow, "e_U"] & \mathrm{Pred}(F)(U) \arrow[r, rightarrowtail, "m_U"'] \arrow[l, bend left, "s_U"] & F(X)
\end{tikzcd}$$

We assume that $m_U \circ f = m_V \circ g$. The two maps $f' = s_U \circ f \colon A \to F(U)$ and $g' = s_V \circ g \colon A \to F(V)$ satisfy $F(m) \circ f' = F(n) \circ g'$, since $e_U \circ s_U = \mathrm{id}$ (and similarly for V). Since F preserves weak pullbacks, there is a map $h \colon A \to F(P)$ with $F(p) \circ h = f'$ and $F(q) \circ h = g'$. We claim that $h' = e_P \circ h \colon A \to \mathrm{Pred}(F)(P)$ is the required mediating map. The equation $p' \circ h' = f$ holds, since m_U is monic:

$$\begin{aligned} m_U \circ p' \circ h' &= m_P \circ e_P \circ h \\ &= F(m \wedge n) \circ h \\ &= F(m) \circ F(p) \circ h \\ &= F(m) \circ f' \\ &= m_U \circ e_U \circ s_U \circ f \\ &= m_U \circ f. \end{aligned}$$

Analogously one shows $q' \circ h' = g$. Uniqueness of h' is obvious, since m_P is monic. ❑

Exercises

6.1.1 Show for a list functor $F^\star$ – using the description (2.17) – that for $P \subseteq X$,

$$\mathrm{Pred}(F^\star)(P) = \{\langle u_1, \ldots, u_n\rangle \in F(X)^\star \mid \forall i \le n.\, \mathrm{Pred}(F)(P)(u_i)\}.$$

And that for $Q \subseteq Y$,

$$\begin{aligned} &\overleftarrow{\mathrm{Pred}}(F^\star)(Q) \\ &= \bigcup_{n \in \mathbb{N}} \overleftarrow{\mathrm{Pred}}(F)(\{u_i \mid i \le n,\, \langle u_1, \ldots, u_n\rangle \in F(Y)^n,\ Q(\langle u_1, \ldots, u_n\rangle)\}). \end{aligned}$$

6.1.2 Use Lemmas 6.1.6 and 3.3.1 to check that relation lifting can also be expressed via predicate lifting. For a relation $\langle r_1, r_2\rangle \colon R \hookrightarrow X \times Y$,

$$\mathrm{Rel}(F)(R) = \coprod_{\langle F(r_1), F(r_2)\rangle} \mathrm{Pred}(F)(R).$$

6.1.3 Let $X \xrightarrow{\langle \delta, \varepsilon\rangle} X^A \times B$ be a deterministic automaton. Prove that the associated unlabelled transition system, according to (6.4), is described by:

$$x \to x' \iff \exists a \in A.\, \delta(x)(a) = x'.$$

6.1.4 Recall the multiset $\mathcal{M}_M$ and distribution $\mathcal{D}$ functors from Section 4.1. Use Definition 6.1.11, with the standard factorisation system on **Sets** of injections and surjections, to prove that the associated predicate liftings are

$$\begin{aligned} \mathrm{Pred}(\mathcal{M}_M)(P \subseteq X) &= \{\varphi \in \mathcal{M}_M(X) \mid \forall x.\, \varphi(x) \neq 0 \Rightarrow P(x)\} \\ \mathrm{Pred}(\mathcal{D})(P \subseteq X) &= \{\varphi \in \mathcal{D}(X) \mid \forall x.\, \varphi(x) \neq 0 \Rightarrow P(x)\}. \end{aligned}$$

Hint: Recall from Propositions 4.2.9 and 4.2.10 that the functors $\mathcal{M}_M$ and $\mathcal{D}$ preserve weak pullbacks, and thus injections, by Lemma 4.2.2.

6.1.5 (From [419, proposition 16]) Let $F \colon \mathbf{Sets} \to \mathbf{Sets}$ be an arbitrary functor. Prove that there is a bijective correspondence between natural transformations $\mathcal{P} \Rightarrow \mathcal{P}F$ as in (6.2) and subsets of $F(2)$ – where $2 = \{0,1\}$. *Hint*: Use that predicates $P \subseteq X$ can be identified with their characteristic functions $X \to 2$.

6.1.6 Prove functoriality of predicate lifting – as in Exercise 4.4.6 for relation lifting: a natural transformation $\sigma\colon F \Rightarrow G$ gives a lifting as in

$$\begin{tikzcd}
\mathrm{Pred}(\mathbb{C}) \arrow[rr, bend left, "\mathrm{Pred}(F)"] \arrow[rr, bend right, "\mathrm{Pred}(G)"'] \arrow[rr, phantom, "\Downarrow \mathrm{Pred}(\sigma)"] \arrow[d] & & \mathrm{Pred}(\mathbb{C}) \arrow[d] \\
\mathbb{C} \arrow[rr, bend left, "F"] \arrow[rr, bend right, "G"'] \arrow[rr, phantom, "\Downarrow \sigma"] & & \mathbb{C}
\end{tikzcd}$$

Prove also, for arbitrary functors $F, G\colon \mathbb{C} \to \mathbb{C}$, the existence of a natural transformation $\mathrm{Pred}(FG) \Rightarrow \mathrm{Pred}(F)\mathrm{Pred}(G)$.

6.2 Invariants

Invariants are predicates on the state space of a coalgebra with the special property that they are closed under transitions: once they are true, they remain true no matter which steps are taken (using the coalgebra). This section will introduce invariants via predicate lifting (from the previous section). It will first concentrate on invariants for coalgebras of polynomial functors, and later deal with more general functors. Invariants are closely related to subcoalgebras. Many of the results we describe for invariants occur in [411, section 6], but with subcoalgebra terminology, and thus with slightly different proofs.

We shall define the notion of invariant, both for coalgebras and for algebras, as (co)algebra of a predicate lifting functor $\mathrm{Pred}(F)$. In both cases an invariant is a predicate which is closed under the state transition operations. There does not seem to be an established (separate) terminology in algebra, so we simply use the phrase 'invariant' both for algebras and for coalgebras.

Definition 6.2.1 Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a Kripke polynomial functor, with predicate lifting functor $\mathrm{Pred}(F)\colon \mathbf{Pred} \to \mathbf{Pred}$ as in Corollary 6.1.5. Abstractly, an **invariant** is either a $\mathrm{Pred}(F)$-coalgebra $P \to \mathrm{Pred}(F)(P)$ or a $\mathrm{Pred}(F)$-algebra $\mathrm{Pred}(F)(P) \to P$, as in

$$\begin{tikzcd}
P \arrow[r, dashed] \arrow[d, tail] & \mathrm{Pred}(F)(P) \arrow[d, tail] \\
X \arrow[r] & F(X)
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathrm{Pred}(F)(P) \arrow[r, dashed] \arrow[d, tail] & P \arrow[d, tail] \\
F(X) \arrow[r] & X
\end{tikzcd}$$

More concretely, this means the following.

1. An **invariant** for a coalgebra $c\colon X \to F(X)$ is a predicate $P \subseteq X$ satisfying for all $x \in X$,

$$x \in P \Longrightarrow c(x) \in \mathrm{Pred}(F)(P).$$

Equivalently,

$$P \subseteq c^{-1}(\mathrm{Pred}(F)(P)) \qquad \text{or} \qquad \textstyle\coprod_c(P) \subseteq \mathrm{Pred}(F)(P).$$

2. An **invariant** for an algebra $a\colon F(X) \to X$ is a predicate $P \subseteq X$ satisfying for all $u \in F(X)$,

$$u \in \mathrm{Pred}(F)(P) \Longrightarrow a(u) \in P.$$

That is,

$$\mathrm{Pred}(F)(P) \subseteq a^{-1}(P) \qquad \text{or} \qquad \textstyle\coprod_a(\mathrm{Pred}(F)(P)) \subseteq P.$$

This section concentrates on invariants for coalgebras, but occasionally invariants for algebras are also considered. We first relate invariants to bisimulations. There are similar results for congruences; see Exercise 6.2.1.

Lemma 6.2.2 *Consider two coalgebras $c\colon X \to F(X)$ and $d\colon Y \to F(Y)$ of a Kripke polynomial functor F.*

1. *If $R \subseteq X \times Y$ is a bisimulation, then both its domain $\coprod_{\pi_1} R = \{x \mid \exists y.\, R(x,y)\}$ and codomain $\coprod_{\pi_2} R = \{y \mid \exists x.\, R(x,y)\}$ are invariants.*
2. *An invariant $P \subseteq X$ yields a bisimulation $\coprod_\Delta P = \{(x,x) \mid x \in P\} \subseteq X \times X$.*

Proof 1. If the relation R is a bisimulation, then the predicate $\coprod_{\pi_1} R \subseteq X$ is an invariant, since

$$\begin{array}{rcll} \coprod_c \coprod_{\pi_1} R & = & \coprod_{\pi_1} \coprod_{c\times d} R & \\ & \subseteq & \coprod_{\pi_1} \mathrm{Rel}(F)(R) & \text{because } R \text{ is a bisimulation} \\ & = & \mathrm{Pred}(F)(\coprod_{\pi_1} R) & \text{by Lemma 6.1.2.2.} \end{array}$$

Similarly, $\coprod_{\pi_2} R \subseteq Y$ is an invariant for the coalgebra d.

2. Suppose now that $P \subseteq X$ is an invariant. Then

$$\begin{array}{rcll} \coprod_{c\times c} \coprod_\Delta P & = & \coprod_\Delta \coprod_c P & \\ & \subseteq & \coprod_\Delta \mathrm{Pred}(F)(P) & \text{since } P \text{ is an invariant} \\ & = & \mathrm{Rel}(F)(\coprod_\Delta P) & \text{by Lemma 6.1.2.1.} \end{array}$$

❑

Example 6.2.3 We consider invariants for both deterministic and non-deterministic automata.

1. As is well known by now, a deterministic automaton $\langle\delta,\epsilon\rangle\colon X \to X^A \times B$ is a coalgebra for the functor $F = \mathrm{id}^A \times B$. Predicate lifting for this functor yields for a predicate $P \subseteq X$ a new predicate $\mathrm{Pred}(F)(P) \subseteq (X^A \times B)$, given by

$$\mathrm{Pred}(F)(P)(f, b) \iff \forall a \in A.\, P(f(a)).$$

A predicate $P \subseteq X$ is thus an invariant with respect to the coalgebra $\langle\delta,\epsilon\rangle\colon X \to X^A \times B$ if, for all $x \in X$,

$$\begin{aligned} P(x) &\implies \mathrm{Pred}(F)(P)((\delta(x), \epsilon(x))) \\ &\iff \forall a \in A.\, P(\delta(x)(a)) \\ &\iff \forall a \in A.\, \forall x' \in X.\, x \xrightarrow{a} x' \Rightarrow P(x'). \end{aligned}$$

Thus, once a state x is in an invariant P, all its – immediate and non-immediate – successor states are also in P. Once an invariant holds, it will continue to hold.

2. A non-deterministic automaton $\langle\delta,\epsilon\rangle\colon X \to \mathcal{P}(X)^A \times B$ is a coalgebra for the functor $F = \mathcal{P}(\mathrm{id})^A \times B$. Predicate lifting for this functor sends a predicate $P \subseteq X$ to the predicate $\mathrm{Pred}(F)(P) \subseteq (\mathcal{P}(X)^A \times B)$ given by:

$$\mathrm{Pred}(F)(P)(f, b) \iff \forall a \in A.\, \forall x' \in f(a).\, P(x')\,.$$

This $P \subseteq X$ is then an invariant for the automaton $\langle\delta,\epsilon\rangle\colon X \to \mathcal{P}(X)^A \times B$ if for all $x \in X$,

$$\begin{aligned} P(x) &\implies \mathrm{Pred}(F)(P)(\delta(x), \epsilon(x)) \\ &\iff \forall a \in A.\, \forall x' \in \delta(x)(a).\, P(x') \\ &\iff \forall a \in A.\, \forall x' \in X.\, x \xrightarrow{a} x' \Rightarrow P(x'). \end{aligned}$$

Proposition 6.2.4 *Let $X \xrightarrow{c} F(X)$ and $Y \xrightarrow{d} F(Y)$ be two coalgebras of a Kripke polynomial functor $F\colon$ **Sets** $\to$ **Sets**.*

1. *Invariants are closed under arbitrary unions and intersections: if predicates $P_i \subseteq X$ are invariants for $i \in I$, then their union $\bigcup_{i\in I} P_i$ and intersection $\bigcap_{i\in I} P_i$ are invariants.*

   In particular, falsity $\perp$ (union over $I = \emptyset$) and truth $\top$ (intersection over $I = \emptyset$) are invariants.

2. *Invariants are closed under direct and inverse images along homomorphisms: if $f\colon X \to Y$ is a homomorphism of coalgebras, and $P \subseteq X$ and $Q \subseteq Y$ are invariants, then so are $\coprod_f(P) \subseteq Y$ and $f^{-1}(Q) \subseteq X$.*

   In particular, the image $\mathrm{Im}(f) = \coprod_f(\top)$ of a coalgebra homomorphism is an invariant.

Proof 1. First we note that inverse images preserve both unions and intersections. Closure of invariants under unions then follows from monotonicity of predicate lifting: $P_i \subseteq c^{-1}(\mathrm{Pred}(F)(P_i)) \subseteq c^{-1}(\mathrm{Pred}(F)(\bigcup_{i\in I} P_i)$ for each $i \in I$, so that we may conclude $\bigcup_{i\in I} P_i \subseteq c^{-1}\mathrm{Pred}(F)(\bigcup_{i\in I} P_i)$. Similarly, closure under intersection follows because predicate lifting preserves intersections; see Lemma 6.1.3.1.

2. For preservation of direct images assume that $P \subseteq X$ is an invariant. Then

$$\begin{array}{rcll} \coprod_d \coprod_f P & = & \coprod_{F(f)} \coprod_c P & \text{because } f \text{ is a homomorphism} \\ & \subseteq & \coprod_{F(f)} \mathrm{Pred}(F)(P) & \text{since } P \text{ is an invariant} \\ & = & \mathrm{Pred}(F)(\coprod_f P) & \text{by Lemma 6.1.3.3.} \end{array}$$

Similarly, if $Q \subseteq Y$ is an invariant, then

$$\begin{array}{rcll} f^{-1}(Q) & \subseteq & f^{-1}d^{-1}(\mathrm{Pred}(F)(Q)) & \text{because } Q \text{ is an invariant} \\ & = & c^{-1}F(f)^{-1}(\mathrm{Pred}(F)(Q)) & \text{because } f \text{ is a homomorphism} \\ & = & c^{-1}(\mathrm{Pred}(F)(f^{-1}(Q)) & \text{by Lemma 6.1.3.2.} \end{array}$$ ❑

The next result readily follows from Lemma 6.1.6. It is the analogue of Theorem 3.3.2 and has important consequences.

Theorem 6.2.5 *Let $F\colon$ **Sets** $\to$ **Sets** be a Kripke polynomial functor.*

1. *A predicate $m\colon P \rightarrowtail X$ on the state space of a coalgebra $c\colon X \to F(X)$ is an invariant if and only if $P \rightarrowtail X$ is a* ***subcoalgebra****: there is a (necessarily unique) coalgebra structure $P \to F(P)$ making $m\colon P \rightarrowtail X$ a homomorphism of coalgebras:*

$$\begin{tikzcd} F(P) \arrow[r, tail, "F(m)"] & F(X) \\ P \arrow[u] \arrow[r, tail, "m"'] & X \arrow[u, "c"'] \end{tikzcd}$$

Uniqueness of this coalgebra $P \to F(P)$ follows because $F(m)$ is injective by Lemma 4.2.2.

2. *Similarly, a predicate $m\colon P \rightarrowtail X$ is an invariant for an algebra $a\colon F(X) \to X$ if P carries a (necessarily unique)* ***subalgebra*** *structure $F(P) \to P$ making $m\colon P \rightarrowtail X$ a homomorphism of algebras.* ❑

Earlier we have seen a generic binary induction principle in Theorem 3.1.4. At this stage we can prove the familiar unary induction principle for initial algebras.

Theorem 6.2.6 (Unary induction proof principle) *An invariant on an initial algebra is always true.*

Equivalently, the truth predicate is the only invariant on an initial algebra. The proof is a generalisation of the argument we have used in Example 2.4.4 to derive induction for the natural numbers from initiality.

Proof Let $m: P \rightarrowtail A$ be an invariant on the initial algebra $F(A) \xrightarrow{\cong} A$. This means by the previous theorem that P itself carries a subalgebra structure $F(P) \to P$, making the square below on the right commute. This subalgebra yields a homomorphism $f: A \to P$ by initiality, as on the left:

$$\begin{tikzcd}
F(A) \arrow[r, "F(f)"] \arrow[d, "\cong"'] & F(P) \arrow[r, tail, "F(m)"] \arrow[d] & F(A) \arrow[d, "\cong"] \\
A \arrow[r, "f"'] & P \arrow[r, tail, "m"'] & A
\end{tikzcd}$$

By uniqueness we then get $m \circ f = \mathrm{id}_A$, which tells that $t \in P$, for all $t \in A$. ❑

Example 6.2.7 Consider the binary trees from Example 2.4.5 as algebras of the functor $F(X) = 1 + (X \times A \times X)$, with initial algebra

$$1 + (\mathsf{BinTree}(A) \times A \times \mathsf{BinTree}(A)) \xrightarrow[\cong]{[\mathsf{nil}, \mathsf{node}]} \mathsf{BinTree}(A).$$

Predicate lifting $\mathrm{Pred}(F)(P) \subseteq F(X)$ of an arbitrary predicate $P \subseteq X$ is given by

$$\mathrm{Pred}(F)(P) = \{\kappa_1(*)\} \cup \{\kappa_2(x_1, a, x_2) \mid a \in A \wedge P(x_1) \wedge P(x_2)\}.$$

Therefore, a predicate $P \subseteq \mathsf{BinTree}(A)$ on the initial algebra is an invariant if both

$$\begin{cases} P(\mathsf{nil}) \\ P(x_1) \wedge P(x_2) \Rightarrow P(\mathsf{node}(x_1, a, x_2)). \end{cases}$$

The unary induction principle then says that such a P must hold for all binary trees $t \in \mathsf{BinTree}(A)$. This may be rephrased in rule form as

$$\frac{P(\mathsf{nil}) \qquad P(x_1) \wedge P(x_2) \Rightarrow P(\mathsf{node}(x_1, a, x_2))}{P(t)}.$$

6.2.1 Invariants, Categorically

The description in Definition 6.2.1 of invariants as (co)algebras of a predicate lifting functor $\mathrm{Pred}(F)$ generalises immediately from polynomial functors to arbitrary functors: if the underlying category $\mathbb{C}$ carries a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$, a predicate lifting functor $\mathrm{Pred}(F): \mathrm{Pred}(\mathbb{C}) \to \mathrm{Pred}(\mathbb{C})$ exists as in Definition 6.1.11. An invariant is then a predicate $(m: U \rightarrowtail X) \in \mathrm{Pred}(\mathbb{C})$

on the carrier $X \in \mathbb{C}$ of a coalgebra $c\colon X \to F(X)$, or of an algebra $a\colon F(X) \to X$, for which there are (dashed) maps in $\mathbb{C}$:

$$\begin{tikzcd} P \arrow[r, dashed] \arrow[d, tail] & \mathrm{Pred}(F)(P) \arrow[d, tail] \\ X \arrow[r] & F(X) \end{tikzcd} \qquad \begin{tikzcd} \mathrm{Pred}(F)(P) \arrow[r, dashed] \arrow[d, tail] & P \arrow[d, tail] \\ F(X) \arrow[r] & X \end{tikzcd}$$

This is the same as in Definition 6.2.1. Not all of the results that hold for invariants of (co)algebras of polynomial functors also hold in the abstract case. In particular, the tight connection between invariants of a coalgebra and subcoalgebras is lost – but it still holds in the algebraic case. We briefly discuss the main results.

Lemma 6.2.8 *Let $F\colon \mathbb{C} \to \mathbb{C}$ be an endofunctor on a category $\mathbb{C}$ with a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$. Assume predicates $m\colon U \rightarrowtail X$ and $n\colon V \rightarrowtail Y$, on the carriers of a coalgebra $c\colon X \to F(X)$ and an algebra $a\colon F(Y) \to Y$.*

1. *If $U \rightarrowtail X$ carries a subcoalgebra $c'\colon U \to F(U)$, then U is an invariant. The converse holds if abstract epis are split,* i.e. *if $\mathfrak{E} \subseteq SplitEpis$.*
2. *The predicate $V \rightarrowtail Y$ carries an subalgebra if and only if it is an invariant.*

Proof 1. A subcoalgebra $c'\colon U \to F(U)$ gives rise to a dashed map $e_U \circ c'\colon U \to \mathrm{Pred}(F)(U)$ by composition:

$$\begin{tikzcd} & F(U) \arrow[d, "e_U"] \arrow[dd, bend left=60, "F(m)"] \\ U \arrow[ur, "c'"] \arrow[r, dashed] \arrow[d, tail, "m"'] & \mathrm{Pred}(F)(U) \arrow[d, tail, "m_U"] \\ X \arrow[r, "c"] & F(X) \end{tikzcd}$$

If the map $e_U \in \mathfrak{E}$ is a split epi, say via $s\colon \mathrm{Pred}(F)(U) \to F(U)$, then an invariant yields a subcoalgebra, by post-composition with s.

2. For algebras an invariant $\mathrm{Pred}(F)(V) \to V$ gives rise to a subalgebra via pre–composition with $e_V\colon F(V) \twoheadrightarrow \mathrm{Pred}(F)(V)$. In the reverse direction, from a subalgebra $a'\colon F(U) \to U$ to an invariant, we use diagonal-fill-in:

$$\begin{tikzcd} F(V) \arrow[r, two heads, "e_V"] \arrow[dd, "a'"'] & \mathrm{Pred}(F)(V) \arrow[d, tail, "m_V"] \arrow[ddl, dashed] \\ & F(X) \arrow[d, "a"] \\ V \arrow[r, tail] & X \end{tikzcd}$$

□

Exercises

6.2.1 Let $F(X) \xrightarrow{a} X$ and $F(Y) \xrightarrow{b} Y$ be algebras of a Kripke polynomial functor F. Prove, in analogy with Lemma 6.2.2 that

1. If $R \subseteq X \times Y$ is a congruence, then both its domain $\coprod_{\pi_1} R \subseteq X$ and its codomain $\coprod_{\pi_2} R \subseteq Y$ are invariants.
2. If $P \subseteq X$ is an invariant, then $\coprod_{\Delta} P \subseteq X \times X$ is a congruence.

6.2.2 Use binary induction in Theorem 3.1.4, together with the previous exercise, to give an alternative proof of unary induction from Theorem 6.2.6.

6.2.3 Prove in the general context of Section 6.2.1 that for a coalgebra homomorphism f direct images $\coprod_f$, as in defined in Proposition 4.3.5, preserve invariants. Conclude that the image $\mathrm{Im}(f) = \coprod_f(\top) \rightarrowtail Y$ of a coalgebra homomorphism $f: X \to Y$ is an invariant.

6.2.4 The next result from [154] is the analogue of Exercise 3.2.7; it describes when a function is definable by coinduction. Let $Z \xrightarrow{\cong} F(Z)$ be final coalgebra of a finite Kripke polynomial functor F. Prove that an arbitrary function $f: X \to Z$ is defined by finality (i.e. is beh_c for some coalgebra $c: X \to F(X)$ on its domain X) if and only if its image $\mathrm{Im}(f) \subseteq Z$ is an invariant. *Hint*: Use the splitting of surjective functions from Lemma 2.1.7.

6.2.5 Let $S: \mathbb{C} \to \mathbb{C}$ be a comonad on a category $\mathbb{C}$ with a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$.

1. Use the Exercise 6.1.6 to derive that $\mathrm{Pred}(S)$ is a comonad on the category $\mathrm{Pred}(\mathbb{C})$ via
$$\mathrm{id} = \mathrm{Pred}(\mathrm{id}) \Longleftarrow \mathrm{Pred}(S) \Longrightarrow \mathrm{Pred}(S^2) \Rightarrow \mathrm{Pred}(S)^2.$$
2. Consider a pair of maps in a commuting square:
$$\begin{tikzcd} U \arrow[r, "\gamma'"] \arrow[d, tail] & \mathrm{Pred}(S)(U) \arrow[d, tail] \\ X \arrow[r, "\gamma"'] & S(X) \end{tikzcd}$$
Prove that if γ is an Eilenberg–Moore coalgebra for the comonad S, then the pair (γ, γ') is automatically an Eilenberg–Moore coalgebra for the comonad $\mathrm{Pred}(S)$ – and thus an invariant for γ.
3. Let $(X \xrightarrow{\gamma} S(X)) \xrightarrow{f} (Y \xrightarrow{\beta} S(Y))$ be a map of Eilenberg–Moore coalgebras. Prove, as in Exercise 6.2.3 for functor coalgebras, that

if $P \rightarrowtail X$ is an invariant, i.e. $\mathrm{Pred}(S)$-coalgebra, for the Eilenberg–Moore coalgebra γ, then so is $\coprod_f(P)$ for β.

6.2.6 Let $T\colon \mathbb{C} \to \mathbb{C}$ now be a monad on a category $\mathbb{C}$ with a logical factorisation system $(\mathfrak{M}, \mathfrak{E})$. Assume that T preserves abstract epis, i.e. $e \in \mathfrak{E} \Rightarrow T(e) \in \mathfrak{E}$.

1. Prove, via Exercise 4.4.6, that relation lifting $\mathrm{Rel}(T)\colon \mathrm{Rel}(\mathbb{C}) \to \mathrm{Rel}(\mathbb{C})$ is a monad. Describe its unit and multiplication explicitly.
2. Assume a commuting square:

$$\begin{tikzcd}
\mathrm{Rel}(T)(R) \arrow[r, "\gamma"] \arrow[d, tail] & R \arrow[d, tail] \\
T(X) \times T(Y) \arrow[r, "\alpha \times \beta"'] & X \times Y
\end{tikzcd}$$

Prove that if α and β are algebras for the monad T, then the above square is automatically an Eilenberg–Moore algebra for the monad $\mathrm{Rel}(T)$ – and thus a congruence for α, β.

6.3 Greatest Invariants and Limits of Coalgebras

In the first chapter – in Definition 1.3.2 to be precise – we introduced the predicate $\Box P$, describing 'henceforth P' for a predicate P on sequences. The meaning of $(\Box P)(x)$ is that P holds in state x and for all of its successor states. Here we shall extend this same idea to arbitrary coalgebras by defining the predicate $\Box P$ in terms of greatest invariants. These greatest invariants are useful in various constructions, most important in this section, in the construction of equalisers and products for coalgebras. In the next section it will be shown that they are important in a temporal logic for coalgebras.

Definition 6.3.1 Let $c\colon X \to F(X)$ be a coalgebra of a Kripke polynomial functor $F\colon \mathbf{Sets} \to \mathbf{Sets}$. For an arbitrary predicate $P \subseteq X$ on the state space of c, we define a new predicate $\Box P \subseteq X$, called **henceforth** P, as

$$(\Box P)(x) \text{ iff } \exists Q \subseteq X.\, Q \text{ is an invariant for } c \wedge Q \subseteq P \wedge Q(x),$$

that is,

$$\Box P = \bigcup\{Q \subseteq P \mid Q \text{ is an invariant}\}.$$

Since invariants are closed under union – by Proposition 6.2.4.1 – $\Box P$ is an invariant itself. Among all the invariants $Q \subseteq X$, it is the greatest one that is contained in P.

The definition of 'henceforth' resembles the definition of bisimilarity (see Definition 3.1.5). In fact, one could push the similarity by defining for an arbitrary relation R, another relation $\Box R$ as the greatest bisimilarity contained in R – so that bisimilarity $\leftrightarrow$ would appear as $\Box\top$. But there seems to be no clear use for this extra generality.

The next lemma lists some obvious properties of $\Box$. Some of these are already mentioned in Exercise 1.3.3 for sequences.

Lemma 6.3.2 *Consider the henceforth operator $\Box$ for a coalgebra $c\colon X \to F(X)$. The first three properties below express that $\Box$ is an interior operator. The fourth property says that its opens are invariants.*

1. $\Box P \subseteq P$
2. $\Box P \subseteq \Box\Box P$
3. $P \subseteq Q \Rightarrow \Box P \subseteq \Box Q$
4. *P is an invariant if and only if $P = \Box P$.*

Proof 1. This is obvious: if $\Box P(x)$, then $Q(x)$ for some invariant Q with $Q \subseteq P$. Hence $P(x)$.
2. If $\Box P(x)$, then we have an invariant Q, namely $\Box P$, with $Q(x)$ and $Q \subseteq \Box P$. Hence $\Box\Box P(x)$.
3. This is obvious.
4. The (if)-part is clear because we have already seen that $\Box P$ is an invariant. For the (only if)-part, by (1) we have only to prove $P \subseteq \Box P$, if P is an invariant. So assume $P(x)$, then we have an invariant Q, namely P, with $Q(x)$ and $Q \subseteq P$. Hence $\Box P(x)$. ❑

The following result gives an important structural property of greatest invariants. It may be understood abstractly as providing a form of comprehension for coalgebras, as elaborated in Section 6.3.1 below.

Proposition 6.3.3 *Consider a coalgebra $c\colon X \to F(X)$ of a Kripke polynomial functor F with an arbitrary predicate $P \subseteq X$. By Theorem 6.2.5.1 the greatest invariant $\Box P \subseteq P \subseteq X$ carries a subcoalgebra structure, say c_P, in*

$$\begin{tikzcd} F(\Box P) \arrow[r, hook, "F(m)"] & F(X) \\ \Box P \arrow[u, "c_P"] \arrow[r, hook, "m"'] & X \arrow[u, "c"'] \end{tikzcd}$$

This subcoalgebra has the following universal property: each coalgebra homomorphism $f\colon (Y \xrightarrow{d} F(Y)) \longrightarrow (X \xrightarrow{c} F(X))$ which factors through

$P \hookrightarrow X$ – i.e. *satisfies* $f(y) \in P$ *for all* $y \in Y$ – *also factors through* $\Box P$, *namely as (unique) coalgebra homomorphism* $f' \colon (Y \xrightarrow{d} F(Y)) \to (\Box P \xrightarrow{c_P} F(\Box P))$ *with* $m \circ f' = f$.

Proof The assumption that f factors through $P \subseteq X$ may be rephrased as an inclusion $\mathrm{Im}(f) = \coprod_f(\top) \subseteq P$. But since the image along a homomorphism is an invariant, (see Proposition 6.2.4.2), we get an inclusion $\mathrm{Im}(f) \subseteq \Box P$. This gives the factorisation

$$\big(Y \xrightarrow{\ f\ } X\big) \;=\; \big(Y \xrightarrow{\ f'\ } \Box P \overset{m}{\hookrightarrow} X\big).$$

We have only to show that f' is a homomorphism of coalgebras. But this follows because $F(m)$ is injective; see Lemma 4.2.2. It yields $c_P \circ f' = F(f') \circ d$ since

$$\begin{aligned} F(m) \circ c_P \circ f' &= c \circ m \circ f' \\ &= c \circ f \\ &= F(f) \circ d \\ &= F(m) \circ F(f') \circ d. \end{aligned}$$ ❑

In this section we shall use greatest invariants to prove the existence of limits (equalisers and cartesian products) for coalgebras of Kripke polynomial functors. The constructions can be adapted easily to more general functors, provided the relevant structures, such as $\Box$ and cofree coalgebras, exist.

Recall from Proposition 2.1.5 and Exercise 2.1.14 that colimits (coproducts and coequalisers) of coalgebras are easy: they are constructed just like for sets. The product structure of coalgebras, however, is less trivial. First results appeared in [468], for 'bounded' endofunctors on **Sets**; see Definition 4.6.5 later on. This was generalised in [186, 283, 231] and [242] (which is followed below). We begin with equalisers, which are easy using henceforth $\Box$.

Theorem 6.3.4 (Equalisers of coalgebras) *The category* **CoAlg**(F) *of coalgebras of a Kripke polynomial functor* $F \colon$ **Sets** $\to$ **Sets** *has equalisers: for two coalgebras* $X \xrightarrow{c} F(X)$ *and* $Y \xrightarrow{d} F(Y)$ *with two homomorphisms* $f, g \colon X \to Y$ *between them, there is an equaliser diagram in* **CoAlg**(F),

$$\begin{pmatrix} F(\Box E(f,g)) \\ \uparrow \\ \Box E(f,g) \end{pmatrix} \overset{m}{\rightarrowtail} \begin{pmatrix} F(X) \\ \uparrow c \\ X \end{pmatrix} \underset{g}{\overset{f}{\rightrightarrows}} \begin{pmatrix} F(Y) \\ \uparrow d \\ Y \end{pmatrix}$$

where $E(f,g) \hookrightarrow X$ *is the equaliser* $\{x \in X \mid f(x) = g(x)\}$ *as in* **Sets**. *The greatest invariant invariant* $\Box E(f,g) \hookrightarrow E(f,g)$ *carries a subcoalgebra structure by the previous proposition.*

Proof We show that the diagram above is universal in **CoAlg**(F): for each coalgebra $e\colon Z \to F(Z)$ with homomorphism $h\colon Z \to X$ satisfying $f \circ h = g \circ h$, the map h factors through $Z \to \mathrm{E}(f,g)$ via a unique function. By Proposition 6.3.3 this h restricts to a (unique) coalgebra homomorphism $Z \to \Box\, \mathrm{E}(f,g)$. ❑

The next result requires a restriction to *finite* polynomial functors because the proof uses cofree coalgebras; see Proposition 2.5.3.

Theorem 6.3.5 (Products of coalgebras) *For a finite Kripke polynomial functor* $F\colon$ **Sets** $\to$ **Sets**, *the category* **CoAlg**(F) *of coalgebras has arbitrary products* $\prod$.

Proof We shall construct the product of two coalgebras $c_i\colon X_i \to F(X_i)$, for $i = 1,2$, and leave the general case to the reader. We first form the product $X_1 \times X_2$ of the underlying sets and consider the cofree coalgebra on it; see Proposition 2.5.3. It will be written as $e\colon UG(X_1 \times X_2) \to F(UG(X_1 \times X_2))$, where $U\colon$ **CoAlg**$(F) \to$ **Sets** is the forgetful functor, and G its right adjoint. This coalgebra e comes with a universal map $\varepsilon\colon UG(X_1 \times X_2) \to X_1 \times X_2$. We write $\varepsilon_i = \pi_i \circ \varepsilon\colon UG(X_1 \times X_2) \to X_i$.

Next we form the following equaliser (in **Sets**):

$$E \;=\; \{u \in UG(X_1 \times X_2) \mid \forall i \in \{1,2\}.\, (c_i \circ \varepsilon_i)(u) = (F(\varepsilon_i) \circ e)(u)\}.$$

Then we take its greatest invariant $\Box E \subseteq E$, as in the diagram below, describing E explicitly as equaliser:

$$\begin{tikzcd}[column sep=small]
 & & & F(UG(X_1 \times X_2)) \arrow[dr, "\langle F(\varepsilon_1), F(\varepsilon_2)\rangle"] & \\
\Box E \arrow[r, hook, "n"] & E \arrow[r, hook, "m"] & UG(X_1 \times X_2) \arrow[ur, "e"] \arrow[dr, "\varepsilon"'] & & F(X_1) \times F(X_2) \\
 & & & X_1 \times X_2 \arrow[ur, "c_1 \times c_2"'] &
\end{tikzcd} \tag{6.5}$$

By Proposition 6.3.3, the subset $\Box E \hookrightarrow UG(X_1 \times X_2)$ carries an F-subcoalgebra structure, for which we write $c_1 \dot{\times} c_2$ in

$$\begin{tikzcd}
F(\Box E) \arrow[r, tail, "F(m \circ n)"] & F(UG(X_1 \times X_2)) \\
\Box E \arrow[u, "c_1 \dot{\times} c_2"] \arrow[r, tail, "m \circ n"'] & UG(X_1 \times X_2) \arrow[u, "e"']
\end{tikzcd} \tag{6.6}$$

The dot in $\dot{\times}$ is ad hoc notation used to distinguish this product of objects (coalgebras) from the product $c_1 \times c_2$ of functions, as used in the equaliser diagram above.

We claim this coalgebra $c_1 \dot{\times} c_2 \colon \Box E \to F(\Box E)$ is the product of the two coalgebras c_1 and c_2, in the category **CoAlg**(F). We thus follow the categorical description of product, from Definition 2.1.1. The two projection maps are

$$p_i \stackrel{\text{def}}{=} \big(\Box E \xrightarrow{\;n\;} E \xrightarrow{\;m\;} UG(X_1 \times X_2) \xrightarrow{\;\varepsilon_i\;} X_i\big).$$

We have to show that they are homomorphisms of coalgebras $c_1 \dot{\times} c_2 \to c_i$. This follows from easy calculations:

$$\begin{aligned} F(p_i) \circ (c_1 \dot{\times} c_2) &= F(\varepsilon_i) \circ F(m \circ n) \circ (c_1 \dot{\times} c_2) \\ &= F(\varepsilon_i) \circ e \circ m \circ n && \text{see Diagram (6.6) above} \\ &= \pi_i \circ (c_1 \times c_2) \circ \varepsilon \circ m \circ n && m \text{ is an equaliser in (6.5)} \\ &= c_i \circ \pi_i \circ \varepsilon \circ m \circ n \\ &= c_i \circ p_i. \end{aligned}$$

Next we have to construct pairs, for coalgebra homomorphisms $f_i \colon (Y \stackrel{d}{\to} F(Y)) \longrightarrow (X_i \stackrel{c_i}{\to} F(X_i))$. To start, we can form the ordinary pair $\langle f_1, f_2\rangle \colon Y \to X_1 \times X_2$ in **Sets**. By cofreeness it gives rise to unique function $g \colon Y \to UG(X_1 \times X_2)$ forming a coalgebra homomorphism $d \to e$, with $\varepsilon \circ g = \langle f_1, f_2\rangle$. This g has the following equalising property in (6.5):

$$\begin{aligned} \langle F(\varepsilon_1), F(\varepsilon_2)\rangle \circ e \circ g &= \langle F(\pi_1 \circ \varepsilon), F(\pi_2 \circ \varepsilon)\rangle \circ F(g) \circ d \\ &\qquad \text{since } g \text{ is a coalgebra homomorphism } d \to e \\ &= \langle F(\pi_1 \circ \varepsilon \circ g) \circ d, F(\pi_2 \circ \varepsilon \circ g) \circ d\rangle \\ &= \langle F(f_1) \circ d, F(f_2) \circ d\rangle \\ &= \langle c_1 \circ f_1, c_2 \circ f_2\rangle \\ &\qquad \text{because } f_i \text{ is a coalgebra map } d \to c_i \\ &= \langle c_1 \circ \pi_1 \circ \varepsilon \circ g, c_2 \circ \pi_2 \circ \varepsilon \circ g\rangle \\ &= (c_1 \times c_2) \circ \varepsilon \circ g. \end{aligned}$$

As a result, g factors through $m \colon E \hookrightarrow UG(X_1 \times X_2)$, say as $g = m \circ g'$. But then, by Proposition 6.3.3, g' also factors through $\Box E$. This yields the pair we seek: we write $\langle\langle f_1, f_2\rangle\rangle$ for the unique map $Y \to \Box E$ with $n \circ \langle\langle f_1, f_2\rangle\rangle = g'$.

We still have to show that this pair $\langle\langle f_1, f_2\rangle\rangle$ satisfies the required properties.

- The equations $p_i \circ \langle\langle f_1, f_2\rangle\rangle = f_i$ hold, since

$$\begin{aligned} p_i \circ \langle\langle f_1, f_2\rangle\rangle &= \pi_i \circ \varepsilon \circ m \circ n \circ \langle\langle f_1, f_2\rangle\rangle \\ &= \pi_i \circ \varepsilon \circ m \circ g' \\ &= \pi_i \circ \varepsilon \circ g \\ &= \pi_i \circ \langle f_1, f_2\rangle \\ &= f_i. \end{aligned}$$

- The pair $\langle\langle f_1, f_2\rangle\rangle$ is the unique homomorphism with $p_i \circ \langle\langle f_1, f_2\rangle\rangle = f_i$. Indeed, if $h\colon Y \to \square E$ is also a coalgebra map $d \to (c_1 \dot{\times} c_2)$ with $p_i \circ h = f_i$, then $m \circ n \circ h$ is a coalgebra map $d \to e$ which satisfies

$$\begin{aligned} \varepsilon \circ m \circ n \circ h &= \langle \pi_1 \circ \varepsilon \circ m \circ n \circ h, \pi_2 \circ \varepsilon \circ m \circ n \circ h\rangle \\ &= \langle p_1 \circ h, p_2 \circ h\rangle \\ &= \langle f_1, f_2\rangle. \end{aligned}$$

  Hence by definition of g:

$$m \circ n \circ h = g = m \circ g' = m \circ n \circ \langle\langle f_1, f_2\rangle\rangle,$$

  Because both m and n are injections we get the required uniqueness: $h = \langle\langle f_1, f_2\rangle\rangle$. ❑

Since we have already seen that equalisers exist for coalgebras, we now know that all limits exist (see for instance [344, V, 2]). Proposition 2.1.5 and Exercise 2.1.14 showed that colimits also exist. Hence we can summarise the situation as follows.

Corollary 6.3.6 *The category* **CoAlg**(F) *of coalgebras of a finite Kripke polynomial functor is both complete and cocomplete.* ❑

Structure in categories of coalgebras is investigated further in [283], for endofunctors on more general categories than **Sets**. For instance, a construction of a 'subobject classifier' is given. It captures the correspondence between predicates $P \subseteq X$ and classifying maps $X \to 2$ in general categorical terms. Such subobject classifiers are an essential ingredient of a 'topos'. However, not all topos structure is present in categories of coalgebras (of functors preserving weak pullbacks): effectivity of equivalence relations may fail.

6.3.1 Greatest Invariants and Subcoalgebras, Categorically

The goal of the remainder of this section is to define in the abstract categorical setting of factorisation systems what it means to have greatest invariants $\square$. Since in this setting invariants and subcoalgebras need not be the same – see Lemma 6.2.8 – we shall also describe greatest subcoalgebras (via comprehension). In principle, an easy direct characterisation of $\square P$ is possible, namely as the greatest invariant $Q \leq P$. Below we shall give a more fancy description via an adjunction. This subsection is mostly an exercise in categorical formulations and is not of direct relevance in the sequel. It starts by describing the set-theoretic situation that we have dealt with so far a bit more systematically.

For an endofunctor $F\colon$ **Sets** $\to$ **Sets** we introduce a category **PredCoAlg**(F) of 'predicates on coalgebras'. Its objects are coalgebra-predicate pairs

$\langle X \to F(X), P \subseteq X\rangle$. And its morphisms $\langle X \to F(X), P \subseteq X\rangle \longrightarrow \langle Y \to F(Y), Q \subseteq Y\rangle$ are coalgebra homomorphisms $f\colon (X \to F(X)) \longrightarrow (Y \to F(Y))$ which are at the same time morphisms of predicates: $P \subseteq f^{-1}(Q)$, or equivalently, $\coprod_f(P) \subseteq Q$.

From this new category $\mathbf{PredCoAlg}(F)$ there are obvious forgetful functors to the categories of coalgebras and of predicates. Moreover, one can show that they form a pullback of categories:

$$\begin{tikzcd}
\mathbf{PredCoAlg}(F) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{Pred} \arrow[d] \\
\mathbf{CoAlg}(F) \arrow[r] & \mathbf{Sets}
\end{tikzcd} \tag{6.7}$$

Lemma 6.3.7 *For a Kripke polynomial functor $F\colon$ **Sets** $\to$ **Sets** in the context described above, the greatest invariant operation $\square$ yields a right adjoint in a commuting triangle:*

$$\begin{tikzcd}
\mathbf{PredCoAlg}(F) \arrow[rr, bend left, "\square"] \arrow[dr] & \top & \mathbf{CoAlg}(\mathrm{Pred}(F)) \arrow[ll] \arrow[dl] \\
& \mathbf{Sets} &
\end{tikzcd}$$

Proof Let $X \xrightarrow{c} F(X)$ and $Y \xrightarrow{d} F(Y)$ be two coalgebras with a map of coalgebras $f\colon X \to Y$ between them. Let $P \subseteq X$ be an invariant, and $Q \subseteq Y$ an ordinary predicate. The above adjunction then involves a bijective correspondence:

$$\begin{array}{cl}
(c, P) \xrightarrow{\;f\;} (d, Q) & \text{in } \mathbf{PredCoAlg}(F) \\
\hline\hline
(c, P) \xrightarrow[\;f\;]{} (d, \square\, Q) & \text{in } \mathbf{CoAlg}(\mathrm{Pred}(F)).
\end{array}$$

Above the double lines we have $\coprod_f(P) \subseteq Q$. But since P is an invariant and $\coprod_f$ preserves invariants, this is equivalent to having $\coprod_f(P) \subseteq \square\, Q$, like below the lines. ❑

This leads to the following obvious generalisation.

Definition 6.3.8 Let $F\colon \mathbb{C} \to \mathbb{C}$ be a functor on a category $\mathbb{C}$ with a factorisation system $(\mathfrak{M}, \mathfrak{E})$, inducing a lifting $\mathrm{Pred}(F)\colon \mathrm{Pred}(\mathbb{C}) \to \mathrm{Pred}(\mathbb{C})$ as in Definition 6.1.11. Form the category $\mathbf{PredCoAlg}(F)$ as on the left below:

$$
\begin{tikzcd}
\mathbf{PredCoAlg}(F) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{Pred}(\mathbb{C}) \arrow[d] \\
\mathbf{CoAlg}(F) \arrow[r] & \mathbb{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{PredCoAlg}(F) \arrow[rr, bend left, "\Box"] \arrow[dr] & \top & \mathbf{CoAlg}(\mathrm{Pred}(F)) \arrow[ll] \arrow[dl] \\
& \mathbb{C} &
\end{tikzcd}
$$

We say that the functor F admits **greatest invariants** if there is a right adjoint $\Box$ making the triangle on the right commute.

We turn to greatest subcoalgebras. Recall from Theorem 6.2.5 that they coincide with invariants in the set-theoretic case. But more generally, they require a different description, which we provide in terms of a comprehension functor $\{-\}$. As before, we first recall the set-theoretic situation. A systematic account of comprehension can be found in [239].

Consider the forgetful functor $\mathbf{Pred} \to \mathbf{Sets}$ that sends a predicate $(P \subseteq X)$ to its underlying set X. There is an obvious 'truth' predicate functor $\top\colon \mathbf{Sets} \to \mathbf{Pred}$ sending a set X to the truth predicate $\top(X) = (X \subseteq X)$. It is not hard to see that $\top$ is right adjoint to the forgetful functor $\mathbf{Pred} \to \mathbf{Sets}$.

In this situation there is another adjoint, namely the 'comprehension' or 'subset type' functor $\{-\}\colon \mathbf{Pred} \to \mathbf{Sets}$, given by $(P \subseteq X) \mapsto P$. One can prove that $\{-\}$ is right adjoint to truth $\top$, so that there is a situation:

$$
\begin{tikzcd}
\mathbf{Pred} \arrow[d, shift right=2] \arrow[d, phantom, "\dashv"] \arrow[d, bend left=80, "\{-\}"] \arrow[d, phantom, shift left=6, "\top\dashv"] \\
\mathbf{Sets} \arrow[u, bend left]
\end{tikzcd}
\tag{6.8}
$$

Lemma 6.3.9 *For a Kripke polynomial functor F:* **Sets** $\to$ **Sets**, *consider the category* **PredCoAlg**(F) *described in (6.7).*

1. *There is a truth predicate functor $\top$:* **CoAlg**$(F) \to$ **PredCoAlg**(F) *which is right adjoint to the forgetful functor* **PredCoAlg**$(F) \to$ **CoAlg**(F).
2. *This functor $\top$ has a right adjoint $\{-\}$:* **PredCoAlg**$(F) \to$ **CoAlg**(F) *given by*

$$\langle X \xrightarrow{c} F(X), P \subseteq X\rangle \longmapsto (\Box P \xrightarrow{c_P} F(\Box P))$$

using the induced coalgebra c_P on the greatest invariant $\Box P$ from Proposition 6.3.3.

Proof The truth functor $\top\colon \mathbf{CoAlg}(F) \to \mathbf{PredCoAlg}(F)$ is given by

$$(X \xrightarrow{c} F(X)) \longmapsto (c, \top) \qquad \text{and} \qquad f \longmapsto f.$$

Next let $c\colon X \to F(X)$ and $d\colon Y \to F(Y)$ be two coalgebras, with a predicate $Q \subseteq Y$. We write $d_Q\colon \Box Q \to F(\Box Q)$ for the induced coalgebra on the greatest invariant $\pi_Q\colon \Box Q \rightarrowtail Y$. We prove the comprehension adjunction:

$$\frac{\left\langle \begin{pmatrix} F(X) \\ \uparrow c \\ X \end{pmatrix}, \top \right\rangle \xrightarrow{\quad f \quad} \left\langle \begin{pmatrix} F(Y) \\ \uparrow d \\ Y \end{pmatrix}, Q \right\rangle}{\begin{pmatrix} F(X) \\ \uparrow c \\ X \end{pmatrix} \xrightarrow[\quad g \quad]{} \begin{pmatrix} F(\Box\, Q) \\ \uparrow d' \\ \Box\, Q \end{pmatrix}} \; .$$

This correspondence works as follows.

- Given a map f in the category **PredCoAlg**(F), we have $\top \subseteq f^{-1}(Q)$, so that $f(x) \in Q$, for all $x \in X$. By Proposition 6.3.3 f then restricts to a unique coalgebra homomorphism $\overline{f}\colon X \to \Box\, Q$ with $\pi_Q \circ \overline{f} = f$.
- Conversely, given a coalgebra homomorphism $g\colon X \to \Box\, Q$, we get a homomorphism $\overline{g} = \pi_Q \circ g\colon X \to Y$. By construction its image is contained in Q. ❑

It is easy to generalise this situation.

Definition 6.3.10 For a functor $F\colon \mathbb{C} \to \mathbb{C}$ on a category $\mathbb{C}$ with a factorisation system $(\mathfrak{M}, \mathfrak{E})$, consider the category **PredCoAlg**(F) described in Definition 6.3.8. There is an obvious truth functor $\top\colon$ **CoAlg**$(F) \to$ **PredCoAlg**(F). We say that the functor F admits **greatest subcoalgebras** if this truth functor $\top$ has a right adjoint $\{-\}$.

Exercises

6.3.1 Fix two sets A, B and consider the associated functor $F(X) = X^A \times B$ for deterministic automata.

1. Check that the cofree coalgebra functor $G\colon$ **Sets** $\to$ **CoAlg**(F) is given by $Y \mapsto (B \times Y)^{A^\star}$.
2. Assume two automata $\langle \delta_i, \epsilon_i \rangle\colon X_i \to X_i^A \times B$. Show that the product coalgebra, as constructed in the proof of Theorem 6.3.5, has carrier W given by the pullback of the maps to the final coalgebra:

$$\begin{tikzcd} W \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X_2 \arrow[d, "\mathrm{beh}_{\langle \delta_2, \epsilon_2 \rangle}"] \\ X_1 \arrow[r, "\mathrm{beh}_{\langle \delta_1, \epsilon_1 \rangle}"'] & B^{A^\star} \end{tikzcd}$$

   (Proposition 2.3.5 describes $B^{A^\star}$ as the final F-coalgebra.)
3. Describe the product coalgebra structure on W explicitly.
4. Explain this pullback outcome using Propositions 4.2.5 and 4.2.6.1 and the construction of products from pullbacks in Diagram (4.6).

6.3.2 Let $c\colon X \to F(X)$ be a coalgebra of a Kripke polynomial functor F. For two predicates $P, Q \subseteq X$ define a new predicate P andthen $Q = P \wedge \Box Q$. Prove that andthen forms a monoid on the poset $\mathcal{P}(X)$ of predicates on X, with truth as neutral element.

6.3.3 The next categorical result is a mild generalisation of [411, theorem 17.1]. It involves an arbitrary functor K between categories of coalgebras, instead of a special functor induced by a natural transformation, as in Proposition 2.5.5. Also the proof hint that we give leads to a slightly more elementary proof than in [411] because it avoids bisimilarity and uses an equaliser (in **Sets**) instead, much as in the proof of Theorem 6.3.5.

Consider two finite Kripke polynomial functors $F, H\colon \mathbf{Sets} \to \mathbf{Sets}$. Assume that there is a functor K between categories of coalgebras, commuting with the corresponding forgetful functors U_F and U_H, as in

$$\begin{tikzcd}
\mathbf{CoAlg}(F) \arrow[rr, "K"] \arrow[dr, bend right, "U_F"'] & & \mathbf{CoAlg}(H) \arrow[dl, bend left, "U_H"] \\
& \mathbf{Sets} \arrow[ul, bend right, "G"'] \arrow[ul, phantom, "\dashv"] &
\end{tikzcd}$$

Prove that if F has cofree coalgebras, given by a right adjoint G to the forgetful functor U_F as in the diagram (and as in Proposition 2.5.3), then K has a right adjoint. *Hint*: For an arbitrary H-coalgebra $d\colon Y \to H(Y)$, first consider the cofree F-coalgebra on Y, say $e\colon U_F G(Y) \to F(U_F G(Y))$, and then form the equaliser

$$E = \{u \in U_F G(Y) \mid (K(e) \circ H(\varepsilon_Y))(u) = (d \circ \varepsilon_Y)(u)\}.$$

The greatest invariant $\Box E$ is then the carrier of the required F-coalgebra.

6.4 Temporal Logic for Coalgebras

Modal logic is a branch of logic in which the notions of necessity and possibility are investigated, via special modal operators. It has developed into a field in which other notions such as time, knowledge, program execution and provability are analysed in comparable manners; see for instance [128, 167, 228, 313, 198, 437, 73]. The use of temporal logic for reasoning about (reactive) state-based systems is advocated especially in [389, 390, 343], concentrating on temporal operators for transition systems – which may be seen as special instances of coalgebras (see Section 2.2.4). The coalgebraic approach to temporal logic extends these operators from transition systems to other coalgebras, in a uniform manner, following ideas first put forward by Moss [359], Kurz [322], Pattinson [373] and many others; see the overview papers [318, 323, 99]. This section will consider what we call *temporal* logic of coalgebras, involving logical modalities that cover all possible transitions by a particular coalgebra. Section 6.5 deals with a more refined *modal* logic, with modalities capturing specific moves to successor states. In this section we focus on (Kripke) polynomial functors on **Sets**.

We have already seen a few constructions with predicate lifting and invariants. Here we will elaborate the logical aspects and will in particular illustrate how a tailor-made temporal logic can be associated with a coalgebra, via a generic definition. This follows [243]. The exposition starts with 'forwards' temporal operators, talking about future states, and will continue with 'backwards' operators in Section 6.4.1.

The logic in this section will deal with predicates on the state spaces of coalgebras. We extend the usual boolean connectives to predicates, via pointwise definitions: for $P, Q \subseteq X$,

$$\begin{aligned} \neg P &= \{x \in X \mid \neg P(x)\} \\ P \wedge Q &= \{x \in X \mid P(x) \wedge Q(x)\} \\ P \Rightarrow Q &= \{x \in X \mid P(x) \Rightarrow Q(x)\}, \quad \text{etc.} \end{aligned}$$

In Section 1.3 we described a nexttime operator $\bigcirc$ for sequences. We start by generalising it to other coalgebras. This $\bigcirc$ will be used to construct more temporal operators.

Definition 6.4.1 Let $c\colon X \to F(X)$ be a coalgebra of a Kripke polynomial functor F. We define the **nexttime** operator $\bigcirc\colon \mathcal{P}(X) \to \mathcal{P}(X)$ as

$$\bigcirc P = c^{-1}(\mathrm{Pred}(F)(P)) = \{x \in X \mid c(x) \in \mathrm{Pred}(F)(P)\}.$$

That is, the operation $\bigcirc\colon \mathcal{P}(X) \to \mathcal{P}(X)$ is defined as the composite

$$\bigcirc = \Big(\mathcal{P}(X) \xrightarrow{\mathrm{Pred}(F)} \mathcal{P}(FX) \xrightarrow{c^{-1}} \mathcal{P}(X)\Big).$$

We understand the predicate $\bigcirc P$ as true for those states x, all of whose immediate successor states, if any, satisfy P. This will be made precise in Proposition 6.4.7 below. Notice that we leave the dependence of the operator $\bigcirc$ on the coalgebra c (and the functor) implicit. Usually, this does not lead to confusion.

Here are some obvious results.

Lemma 6.4.2 *The above nexttime operator $\bigcirc$ satisfies the following properties.*

1. *It is monotone: $P \subseteq Q \Rightarrow \bigcirc P \subseteq \bigcirc Q$. Hence $\bigcirc$ is an endofunctor $\mathcal{P}(X) \to \mathcal{P}(X)$ on the poset category of predicates ordered by inclusion.*
2. *It commutes with inverse images: $\bigcirc(f^{-1}Q) = f^{-1}(\bigcirc Q)$.*
3. *It has invariants as its coalgebras: $P \subseteq X$ is an invariant if and only if $P \subseteq \bigcirc P$.*
4. *It preserves meets (intersections) of predicates.*
5. *The greatest invariant $\Box P$ from Definition 6.3.1 is the cofree $\bigcirc$-coalgebra on P: it is the final coalgebra – or greatest fixed point – of the operator $S \mapsto P \wedge \bigcirc S$ on $\mathcal{P}(X)$.*

Proof We illustrate only the second and the last point. For a homomorphism of coalgebras $(X \xrightarrow{c} FX) \xrightarrow{f} (Y \xrightarrow{d} FY)$ and a predicate $Q \subseteq Y$ we have:

$$\begin{array}{rcll} \bigcirc(f^{-1}Q) & = & c^{-1}\mathrm{Pred}(F)(f^{-1}Q) & \\ & = & c^{-1}F(f)^{-1}\mathrm{Pred}(F)(Q) & \text{by Lemma 6.1.3.2} \\ & = & f^{-1}d^{-1}\mathrm{Pred}(F)(Q) & \text{since } f \text{ is a homomorphism} \\ & = & f^{-1}(\bigcirc Q). & \end{array}$$

For the last point of the lemma, first note that the predicate $\Box P$ is a coalgebra of the functor $P \wedge \bigcirc(-)$ on $\mathcal{P}(X)$. Indeed, $\Box P \subseteq P \wedge \bigcirc(\Box P)$, because $\Box P$ is contained in P and is an invariant. Next, $\Box P$ is the greatest such coalgebra, and hence the final one: if $Q \subseteq P \wedge \bigcirc Q$, then Q is an invariant contained in P, so that $Q \subseteq \Box P$. We conclude that $\Box P$ is the cofree $\bigcirc(-)$-coalgebra. ❑

The nexttime operator $\bigcirc$ is fundamental in temporal logic. By combining it with negations, least fixed points μ and greatest fixed points ν one can define other temporal operators. For instance $\neg \bigcirc \neg$ is the so-called **strong nexttime** operator. It holds for those states for which there actually is a successor state satisfying P. Figure 6.1 shows a few standard operators.

We shall next illustrate the temporal logic of coalgebras in two examples.

Example 6.4.3 Douglas Hofstadter explains in his famous book *Gödel, Escher, Bach* [224] the object- and meta-level perspective on formal systems

Notation	Meaning	Definition
$\bigcirc P$	Nexttime P	$c^{-1}\mathrm{Pred}(F)(P)$
$\Box P$	Henceforth P	$\nu S.(P \wedge \bigcirc S)$
$\Diamond P$	Eventually P	$\neg \Box \neg P$
$P\, \mathcal{U}\, Q$	P until Q	$\mu S.(Q \vee (P \wedge \neg \bigcirc \neg S))$

Figure 6.1 Standard (forwards) temporal operators, where μ and ν are the least and greatest fixed point operators, respectively.

using a simple 'MU-puzzle'. It consists of a simple 'Post' production system (see e.g. [114, section 5.1]) or rewriting system for generating certain strings containing the symbols M, I, U. The meta-question that is considered is whether the string MU can be produced. Both this production system and this question (and also its answer) can be (re)formulated in coalgebraic terminology.

Let therefore our alphabet A be the set $\{\mathsf{M}, \mathsf{I}, \mathsf{U}\}$ of relevant symbols. We will describe the production system as an unlabelled transition system (UTS) $A^\star \to \mathcal{P}_{\mathrm{fin}}(A^\star)$ on the set $A^\star$ of strings over this alphabet. This is given by the following transitions (from [224]), which are parametrised by strings $x, y \in A^\star$:

$$x\mathsf{I} \to x\mathsf{IU} \qquad \mathsf{M}x \to \mathsf{M}xx \qquad x\mathsf{III}y \to x\mathsf{U}y \qquad x\mathsf{UU}y \to xy.$$

Thus, the associated transition system $A^\star \to \mathcal{P}_{\mathrm{fin}}(A^\star)$ is given by

$$\begin{aligned} w \longmapsto \{z \in A^\star \mid \exists x \in A^\star.\ & (w = x\mathsf{I} \wedge z = x\mathsf{IU}) \\ & \vee (w = \mathsf{M}x \wedge z = \mathsf{M}xx) \\ \vee\ \exists x, y \in A^\star.\ & (w = x\mathsf{III}y \wedge z = x\mathsf{U}y) \\ & \vee (w = x\mathsf{UU}y \wedge z = xy)\}. \end{aligned}$$

It is not hard to see that for each word $w \in A^\star$ this set on the right-hand side is finite.

The question considered in [224] is whether the string MU can be obtained from MI; that is, whether $\mathsf{MI} \longrightarrow^* \mathsf{MU}$. Or to put it in temporal terminology, whether the predicate 'equal to MU' eventually holds, starting from MI:

$$\begin{aligned} & \Diamond(\{x \in A^\star \mid x = \mathsf{MU}\})(\mathsf{MI}) \\ \iff & \neg\,\Box(\neg\{x \in A^\star \mid x = \mathsf{MU}\})(\mathsf{MI}) \\ \iff & \neg\exists P \text{ invariant}.\ P \subseteq \neg\{x \in A^\star \mid x = \mathsf{MU}\} \wedge P(\mathsf{MI}) \\ \iff & \forall P \text{ invariant}.\ \neg(\forall x \in P.\ x \neq \mathsf{MU}) \wedge P(\mathsf{MI})) \\ \iff & \forall P \text{ invariant}.\ P(\mathsf{MI}) \implies \exists x \in P.\ x = \mathsf{MU} \\ \iff & \forall P \text{ invariant}.\ P(\mathsf{MI}) \implies P(\mathsf{MU}). \end{aligned}$$

Hofstadter [224] provides a counter-example, by producing an invariant $P \subseteq A^\star$ for which $P(\mathsf{MI})$, but not $P(\mathsf{MU})$, namely:

$$P(x) \overset{\text{def}}{\iff} \text{the number of Is in } x \text{ is not a multiple of 3.}$$

This P is clearly an invariant: of the above four parametrised transitions, the first and last one do not change the number of Is; in the second transition $\mathsf{M}x \to \mathsf{M}xx$, if the number of Is in the right-hand side, i.e. in xx, is $3n$, then n must be even, so that the number of Is in x (and hence in $\mathsf{M}x$) must already be a multiple of 3; a similar argument applies to the third transition. Thus, property P is an invariant. Once we have reached this stage we have P as counter example: clearly $P(\mathsf{MI})$, but not $P(\mathsf{MU})$. Thus MU cannot be obtained from MI.

This proof is essentially the same proof that Hofstadter provides, but of course he does not use the same coalgebraic formulation and terminology. However, he does call the property P 'hereditary'.

This concludes the example. The relation we have used between $\longrightarrow^*$ and $\Diamond$ will be investigated more systematically below; see especially in Proposition 6.4.7.

Here is another, more technical, illustration.

Example 6.4.4 This example assumes some familiarity with the untyped lambda-calculus, and especially with its theory of Böhm trees; see [54, chapter 10]. It involves an operational model for head normal form reduction, consisting of a final coalgebra of certain trees. Temporal logic will be used to define an appropriate notion of 'free variable' on these trees.

We fix a set V and think of its elements as variables. We consider the polynomial functor $F\colon \mathbf{Sets} \to \mathbf{Sets}$ given by

$$F(X) = 1 + (V^\star \times V \times X^\star). \tag{6.9}$$

In this example we shall often omit the coprojections κ_i and simply write $*$ for $\kappa_1(*) \in 1 + (V^\star \times V \times X^\star)$ and $(\vec{v}, w, \vec{x})$ for $\kappa_2(\vec{v}, w, \vec{x}) \in 1 + (V^\star \times V \times X^\star)$. Also, we shall write $\zeta\colon \mathcal{B} \xrightarrow{\cong} F(\mathcal{B})$ for the final F-coalgebra – which exists by Theorem 2.3.9.

Lambda terms are obtained from variables $x \in V$, application MN of two λ-terms M, N, and abstraction $\lambda x.\, M$. The main reduction rule is $(\lambda x.\, M)N \to M[N/x]$. By an easy induction on the structure of λ-terms one then sees that an arbitrary term can be written of the form $\lambda x_1 \ldots x_n.\, yM_1 \ldots M_m$. The set Λ of λ-terms thus carries an F-coalgebra structure, given by the head-normal-form function $\mathsf{hnf}\colon \Lambda \to F(\Lambda)$ (see [54, section 8.3]): for $M \in \Lambda$,

$$\mathsf{hnf}(M) = \begin{cases} * & \text{if } M \text{ has no head normal form} \\ (\langle x_1, \ldots, x_n\rangle, y, \langle M_1, \ldots, M_m\rangle) & \text{if } M \text{ has head normal form} \\ & \lambda x_1 \ldots x_1.\, yM_1 \ldots M_m. \end{cases}$$

We can now define the Böhm tree $\mathsf{BT}(M)$ of a λ-term M via finality:

$$\begin{tikzcd} 1 + (V^\star \times V \times \Lambda^\star) \arrow[r, dashed, "\mathrm{id} + (\mathrm{id} \times \mathrm{id} \times \mathsf{BT}^\star)"] & 1 + (V^\star \times V \times \mathcal{B}^\star) \\ \Lambda \arrow[u, "\mathsf{hnf}"] \arrow[r, dashed, "\mathsf{BT}"'] & \mathcal{B} \arrow[u, "\cong"', "\zeta"] \end{tikzcd}$$

We call the elements of $\mathcal{B}$ (abstract[1]) Böhm trees. We do not really need to know what these elements look like, because we can work with the universal property of $\mathcal{B}$, namely finality. A picture may be useful. For $A \in \mathcal{B}$ we can write

$$\zeta(A) = \bot \qquad \text{or} \qquad \zeta(A) = \left(\begin{array}{ccc} & \lambda x_1 \ldots x_n.\, y & \\ / & & \backslash \\ \zeta(A_1) & \cdots & \zeta(A_m) \end{array}\right)$$

where the second picture applies when $\zeta(A) = (\langle x_1, \ldots, x_n \rangle, y, \langle A_1, \ldots, A_m \rangle)$. The '$\lambda$' is just syntactic sugar, used to suggest the analogy with the standard notation for Böhm trees [54]. The elements of $\mathcal{B}$ are thus finitely branching, possibly infinite, rooted trees, with labels of the form $\lambda x_1 \ldots x_n.\, y$, for variables $x_i, y \in V$.

Using the inverse $\zeta^{-1} \colon 1 + (\mathcal{B}^\star \times V \times \mathcal{B}^\star) \to \mathcal{B}$ of the final coalgebra we can explicitly construct Böhm trees. We give a few examples.

- Let us write $\bot_{\mathcal{B}} \in \mathcal{B}$ for $\zeta^{-1}(*)$. This the 'empty' Böhm tree.
- The Böhm tree $\lambda x.\, x$ is obtained as $\zeta^{-1}(\langle x \rangle, x, \langle\rangle)$. In a similar way one can construct various kinds of finite Böhm trees. For instance, the **S** combinator $\lambda xyz.\, xz(yz)$ is obtained as

  $$\zeta^{-1}(\langle x, y, z \rangle, x, \langle \zeta^{-1}(\langle\rangle, z, \langle\rangle), \zeta^{-1}(\langle\rangle, y, \langle \zeta^{-1}(\langle\rangle, z, \langle\rangle)\rangle)\rangle).$$

  Its picture is:

  $$\begin{array}{ccc} & \lambda xyz.\, x & \\ / & & \backslash \\ z & & y \\ & & | \\ & & z \end{array}$$

- Given an arbitrary Böhm tree $A \in \mathcal{B}$, we can define a new tree $\lambda x.\, A \in \mathcal{B}$ via λ-abstraction:

  $$\lambda x.\, A \;=\; \begin{cases} \bot_{\mathcal{B}} & \text{if } \zeta(A) = * \\ \zeta^{-1}(x \cdot \vec{y}, z, \vec{B}) & \text{if } \zeta(A) = (\vec{y}, z, \vec{B}). \end{cases}$$

We proceed by using temporal logic to define free variables for Böhm trees. Such a definition is non-trivial since Böhm trees may be infinite objects. Some preliminary definitions are required. Let $x \in V$ be an arbitrary variable. It will

[1] One may have a more restricted view and call 'Böhm tree' only those elements in $\mathcal{B}$ which actually come from λ-terms, i.e. which are in the image of the function $\mathsf{BT} \colon \Lambda \to \mathcal{B}$. Then one may wish to call the elements of the whole set $\mathcal{B}$ 'abstract' Böhm trees. We shall not do so. But it is good to keep in mind that the function BT is not surjective. For example, Böhm trees coming from λ-terms can have only a finite number of free variables (as defined below), whereas elements of $\mathcal{B}$ can have arbitrarily many.

be used in the auxiliary predicates Abs_x and Hv_x on Böhm trees, which are defined as follows: for $B \in \mathcal{B}$,

$$\begin{aligned} \mathsf{Abs}_x(B) &\iff \exists x_1, \ldots, x_n.\, \exists B_1, \ldots, B_m. \\ &\qquad\qquad B = \lambda x_1 \ldots x_n.\, yB_1 \ldots B_m \text{ and } x = x_i \text{ for some } i \\ \mathsf{Hv}_x(B) &\iff \exists x_1, \ldots, x_n.\, \exists B_1, \ldots, B_m. \\ &\qquad\qquad B = \lambda x_1 \ldots x_n.\, yB_1 \ldots B_m \text{ and } x = y. \end{aligned}$$

Thus the predicate Abs_x describes the occurrence of x in the abstracted variables, and Hv_x captures that x is the head variable.

For a Böhm tree $A \in \mathcal{B}$ we can now define the set $\mathsf{FV}(A) \subseteq V$ of *free variables* in A via the until operator $\mathcal{U}$ from Figure 6.1:

$$x \in \mathsf{FV}(A) \iff (\neg\mathsf{Abs}_x \;\mathcal{U}\; (\mathsf{Hv}_x \wedge \neg\mathsf{Abs}_x))(A).$$

In words: a variable x is free in a Böhm tree A if there is a successor tree B of A in which x occurs as 'head variable', and in all successor trees of A until that tree B is reached, including B itself, x is not used in a lambda abstraction. This until formula then defines a predicate on $\mathcal{B}$, namely '$x \in \mathsf{FV}(-)$'.

There are then two crucial properties that we would like to hold for a Böhm tree A.

1. If $A = \perp_{\mathcal{B}}$, then
$$\mathsf{FV}(A) = \emptyset.$$
This holds because if $A = \perp_{\mathcal{B}}$, then both $\mathsf{Abs}_x(A)$ and $\mathsf{Hv}_x(A)$ are false, so that the least fixed point in Figure 6.1 defining $\mathcal{U}$ at A in $x \in \mathsf{FV}(A)$ is $\mu S.\, \neg \bigcirc \neg S$. This yields the empty set.
2. If $A = \lambda x_1 \ldots x_n.\, yA_1 \ldots A_m$, then
$$\mathsf{FV}(A) = (\{y\} \cup \mathsf{FV}(A_1) \cup \cdots \cup \mathsf{FV}(A_m)) - \{x_1, \ldots, x_n\}.$$
This result follows from the fixed point property (indicated as 'f.p.' below) defining the until operator $\mathcal{U}$ in Figure 6.1:
$$\begin{aligned} x \in \mathsf{FV}(A) &\overset{\text{def}}{\iff} [\neg\mathsf{Abs}_x \;\mathcal{U}\; (\mathsf{Hv}_x \wedge \neg\mathsf{Abs}_x)](A) \\ &\overset{\text{f.p.}}{\iff} [(\mathsf{Hv}_x \wedge \neg\mathsf{Abs}_x) \vee (\neg\mathsf{Abs}_x \wedge \neg \bigcirc \neg(x \in \mathsf{FV}(-)))](A) \\ &\iff \neg\mathsf{Abs}_x(A) \wedge (\mathsf{Hv}_x(A) \vee \neg \bigcirc \neg(x \in \mathsf{FV}(-))(A)) \\ &\iff x \notin \{x_1, \ldots, x_n\} \wedge \big(x = y \vee \exists j \leq m.\, x \in \mathsf{FV}(A_j)\big) \\ &\iff x \in (\{y\} \cup \mathsf{FV}(A_1) \cup \cdots \cup \mathsf{FV}(A_m)) - \{x_1, \ldots, x_n\}. \end{aligned}$$

This shows how temporal operators can be used to define sensible predicates on infinite structures. The generic definitions provide adequate expressive power in concrete situations. We should emphasise however that the final coalgebra $\mathcal{B}$ of Böhm trees is only an operational model of the lambda calculus and not a denotational one: for instance, it is not clear how to define an application operation $\mathcal{B} \times \mathcal{B} \to \mathcal{B}$ on our abstract Böhm trees via coinduction. Such application is defined on the Böhm model used in [54, section 18.3] via finite approximations. For more information on models of the (untyped) λ-calculus, see e.g. [54, part V], [239, section 2.5], or [136]. See also [324] for a coalgebraic approach to the infinitary λ-calculus.

Our next application of temporal logic does not involve a specific functor, as for Böhm trees above, but is generic. It involves an (unlabelled) transition relation for an arbitrary coalgebra. Before we give the definition it is useful to introduce some special notation and some associated results.

Lemma 6.4.5 *For an arbitrary set X and an element $x \in X$ we define a 'singleton' and 'non-singleton' predicate on X as*

$$\begin{aligned}(\cdot = x) &= \{y \in X \mid y = x\} = \{x\}.\\ (\cdot \neq x) &= \{y \in X \mid y \neq x\} = \neg(\cdot = x).\end{aligned}$$

Then

1. *For a predicate $P \subseteq X$,*

$$P \subseteq (\cdot \neq x) \iff \neg P(x).$$

2. *For a function $f \colon Y \to X$,*

$$f^{-1}(\cdot \neq x) = \bigcap_{y \in f^{-1}(x)} (\cdot \neq y).$$

3. *And for a Kripke polynomial functor F and a predicate $Q \subseteq F(X)$,*

$$\underleftarrow{\mathrm{Pred}}(F)(Q) = \{x \in X \mid Q \not\subseteq \mathrm{Pred}(F)(\cdot \neq x)\}$$

where $\underleftarrow{\mathrm{Pred}}(F)$ is the 'predicate lowering' left adjoint to predicate lifting $\mathrm{Pred}(F)$ from Section 6.1.1.

These 'non-singletons' $(\cdot \neq x)$ are also called coequations, for instance in [177, 17, 422], and pronounced as 'avoid x'. They are used for a sound and complete logical deduction calculus for coalgebras via a 'child' rule (capturing henceforth $\Box$) and a 'recolouring' rule (capturing $\boxtimes$; see Exercise 6.8.8). Here these coequations $(\cdot \neq x)$ arise naturally in a characterisation of predicate lowering in point (3).

Proof Points (1) and (2) follow immediately from the definition. For (3) we use (1) and the adjoint/Galois correspondence in

$$\begin{aligned} x \in \underleftarrow{\mathrm{Pred}}(F)(Q) &\iff \underleftarrow{\mathrm{Pred}}(F)(Q) \not\subseteq (\cdot \neq x) \\ &\iff Q \not\subseteq \mathrm{Pred}(F)(\cdot \neq x). \end{aligned}$$

❑

In Proposition 6.1.9 we have seen an abstract way to turn an arbitrary coalgebra into an unlabelled transition system. Here, and later on in Theorem 6.4.9, we shall reconsider this topic from a temporal perspective.

Definition 6.4.6 Let $c\colon X \to F(X)$ be a coalgebra of a polynomial functor F. On states $x, x' \in X$ we define a transition relation via the strong nexttime operator, as

$$\begin{aligned} x \to x' &\iff x \in (\neg \bigcirc \neg)(\cdot = x') \\ &\iff x \notin \bigcirc(\cdot \neq x') \\ &\iff c(x) \notin \mathrm{Pred}(F)((\cdot \neq x')). \end{aligned}$$

This says that there is a transition $x \to x'$ if and only if there is successor state of x which is equal to x'. In this way we turn an arbitrary coalgebra into an unlabelled transition system.

We shall first investigate the properties of this new transition system $\to$, and only later in Theorem 6.4.9 show that it is actually the same as the earlier translation from coalgebras to transition systems from Section 6.1.1.

So let us first consider what we get for a coalgebra $c\colon X \to \mathcal{P}(X)$ of the powerset functor. Then the notation $x \to x'$ is standardly used for $x' \in c(x)$. This coincides with Definition 6.4.6 since

$$\begin{aligned} x \notin \bigcirc(\cdot \neq x') &\iff x \notin c^{-1}(\mathrm{Pred}(\mathcal{P})(\cdot \neq x')) \\ &\iff c(x) \notin \{a \mid a \subseteq (\cdot \neq x')\} \\ &\iff c(x) \notin \{a \mid x' \notin a\}, \qquad \text{by Lemma 6.4.5.1} \\ &\iff x' \in c(x). \end{aligned}$$

Now that we have gained some confidence in this temporal transition definition, we consider further properties. It turns out that the temporal operators can be expressed in terms of the new transition relation.

Proposition 6.4.7 *The transition relation $\to$ from Definition 6.4.6, induced by a coalgebra $X \to F(X)$, and its reflexive transitive closure $\longrightarrow^*$, satisfy the following properties.*

1. *For a predicate $P \subseteq X$,*

   a. $\bigcirc P = \{x \in X \mid \forall x'.\, x \to x' \implies P(x')\}$.
   b. $\Box P = \{x \in X \mid \forall x'.\, x \longrightarrow^* x' \implies P(x')\}$.
   c. $\Diamond P = \{x \in X \mid \exists x'.\, x \longrightarrow^* x' \wedge P(x')\}$.

This says that the temporal operators on the original coalgebra are the same as the ones on the induced unlabelled transition system.

2. *For a predicate $P \subseteq X$, the following three statements are equivalent.*

   a. *P is an invariant.*
   b. $\forall x, x' \in X.\, P(x) \wedge x \to x' \implies P(x')$.
   c. $\forall x, x' \in X.\, P(x) \wedge x \longrightarrow^* x' \implies P(x')$.

3. *For arbitrary states $x, x' \in X$, the following are equivalent.*

   a. $x \longrightarrow^* x'$.
   b. $P(x) \Rightarrow P(x')$, *for all invariants $P \subseteq X$.*
   c. $x \in \Diamond(\cdot = x')$, i.e. *eventually there is a successor state of x that is equal to x'.*

Proof 1. We reason as follows.

$$\begin{aligned} x \in \bigcirc P &\iff c(x) \in \mathrm{Pred}(F)(P) \\ &\iff \{c(x)\} \subseteq \mathrm{Pred}(F)(P) \\ &\iff \overleftarrow{\mathrm{Pred}}(F)(\{c(x)\}) \subseteq P \\ &\iff \forall x'.\, x' \in \overleftarrow{\mathrm{Pred}}(F)(\{c(x)\}) \Rightarrow P(x') \\ &\iff \forall x'.\, \{c(x)\} \not\subseteq \mathrm{Pred}(F)(\cdot \neq x') \Rightarrow P(x'), \text{ by Lemma 6.4.5.3} \\ &\iff \forall x'.\, c(x) \notin \mathrm{Pred}(F)(\cdot \neq x') \Rightarrow P(x') \\ &\iff \forall x'.\, x \to x' \Rightarrow P(x'). \end{aligned}$$

For the inclusion ($\subseteq$) of (b) we can use (a) an appropriate number of times since $\Box P \subseteq \bigcirc \Box P$ and $\Box P \subseteq P$. For ($\supseteq$) we use that the predicate $\{x \mid \forall x'.\, x \longrightarrow^* x' \implies P(x')\}$ contains P and is an invariant, via (a); hence it is contained in $\Box P$.

The third point (c) follows directly from (b) since $\Diamond = \neg \Box \neg$.

2. This is Immediate from (1) since P is an invariant if and only if $P \subseteq \bigcirc P$, if and only if $P \subseteq \Box P$.
3. The equivalence (b) $\Leftrightarrow$ (c) follows by unfolding the definitions. The implication (a) $\Rightarrow$ (b) follows directly from (2), but for the reverse we have to do a bit of work. Assume $P(x) \Rightarrow P(x')$ for all invariants P. In order to prove $x \longrightarrow^* x'$, consider the predicate Q defined by $Q(y) \Leftrightarrow x \longrightarrow^* y$. Clearly $Q(x)$, so $Q(x')$ follows once we have established that Q is an invariant. But this is an easy consequence using (2): if $Q(y)$, i.e. $x \longrightarrow^* y$, and $y \to y'$, then clearly $x \longrightarrow^* y'$, which is $Q(y')$. ❑

6.4.1 Backwards Reasoning

So far in this section we have concentrated on ‘forwards’ reasoning, by considering only operators that talk about future states. However, within the setting of coalgebras there is also a natural way to reason about previous states. This happens via predicate lowering instead of via predicate lifting, i.e. via the left adjoint $\underleftarrow{\mathrm{Pred}}(F)$ to $\mathrm{Pred}(F)$, introduced in Section 6.1.1.

It turns out that the forwards temporal operators have backwards counterparts. We shall use notation with backwards underarrows for these analogues: $\underleftarrow{\bigcirc}$, $\underleftarrow{\Box}$ and $\underleftarrow{\Diamond}$ are backwards versions of $\bigcirc$, $\Box$ and $\Diamond$.

Definition 6.4.8 For a coalgebra $c\colon X \to F(X)$ of a polynomial functor F, and a predicate $P \subseteq X$ on its carrier X, we define a new predicate **lasttime** P on X by

$$\underleftarrow{\bigcirc}\, P \;=\; \underleftarrow{\mathrm{Pred}}(F)(\textstyle\coprod_c P) \;=\; \underleftarrow{\mathrm{Pred}}(F)(\{c(x) \mid x \in P\}).$$

Thus

$$\underleftarrow{\bigcirc} = \Big(\mathcal{P}(X) \xrightarrow{\;\coprod_c\;} \mathcal{P}(FX) \xrightarrow{\;\underleftarrow{\mathrm{Pred}}(F)\;} \mathcal{P}(X)\Big).$$

This is the so-called **strong lasttime** operator, which holds of a state x if there is an (immediate) predecessor state of x which satisfies P. The corresponding **weak lasttime** is $\neg\underleftarrow{\bigcirc}\neg$.

One can easily define an infinite extension of $\underleftarrow{\bigcirc}$, called **earlier**:

$$\begin{aligned} \underleftarrow{\Diamond}\, P &= \text{'the least invariant containing } P\text{'} \\ &= \{x \in X \mid \forall Q, \text{invariant}.\, P \subseteq Q \implies Q(x)\} \\ &= \textstyle\bigcap\{Q \supseteq P \mid Q \text{ is an invariant}\}. \end{aligned}$$

This predicate $\underleftarrow{\Diamond}\, P$ holds of a state x if there is some (non-immediate) predecessor state of x for which P holds.

Figure 6.2 gives a brief overview of the main backwards temporal operators. In the remainder of this section we shall concentrate on the relation between the backwards temporal operators and transitions.

But first we give a result that was already announced. It states an equivalence between various (unlabelled) transition systems induced by coalgebras.

Notation	Meaning	Definition	Galois Connection
$\underleftarrow{\bigcirc}\, P$	lasttime P	$\underleftarrow{\mathrm{Pred}}(F)(\coprod_c P)$	$\underleftarrow{\bigcirc} \dashv \bigcirc$
$\underleftarrow{\Diamond}\, P$	(sometime) earlier P	$\mu S.\,(P \vee \underleftarrow{\bigcirc}\, S)$	$\underleftarrow{\Diamond} \dashv \Box$
$\underleftarrow{\Box}\, P$	(always) before P	$\neg\underleftarrow{\Diamond}\,\neg P$	$\Diamond \dashv \underleftarrow{\Box}$
$P\,\mathcal{S}\,Q$	P since Q	$\mu S.\,(Q \vee (P \wedge \underleftarrow{\bigcirc}\, S))$	

Figure 6.2 Standard (backwards) temporal operators.

Theorem 6.4.9 *Consider a coalgebra* $c: X \to F(X)$ *of a Kripke polynomial functor* F. *Using the lasttime operator* $\underleftarrow{\bigcirc}$ *one can also define an unlabelled transition system by*

$$\begin{aligned} x \to x' &\iff \text{‘there is an immediate predecessor state of } x'\text{, equal to } x\text{’} \\ &\iff x' \in \underleftarrow{\bigcirc}(\cdot = x). \end{aligned}$$

This transition relation is then the same as

1. $x \notin \bigcirc(\cdot \neq x')$ *from Definition 6.4.6*
2. $x' \in \mathrm{sts}(c(x)) = \underleftarrow{\mathrm{Pred}}(F)(\{c(x)\})$, *used in the translation in (6.4).*

Proof All these forwards and backwards transition definitions are equivalent because

$$\begin{aligned} x' \in \underleftarrow{\bigcirc}(\cdot = x) &\iff x' \in \underleftarrow{\mathrm{Pred}}(F)(\textstyle\coprod_c(\cdot = x)) && \text{by Definition 6.4.8} \\ &\iff x' \in \underleftarrow{\mathrm{Pred}}(F)(\{c(x)\}) && \text{as used in (6.4)} \\ &\iff \{c(x)\} \not\subseteq \mathrm{Pred}(F)(\cdot \neq x') && \text{by Lemma 6.4.5.3} \\ &\iff x \notin \bigcirc(\cdot \neq x') && \text{as used in Definition 6.4.6.} \quad \square \end{aligned}$$

Finally we mention the descriptions of the backwards temporal operators in terms of transitions, as in Proposition 6.4.7.1.

Proposition 6.4.10 *For a predicate* $P \subseteq X$ *on the state space of a coalgebra,*

1. $\underleftarrow{\bigcirc} P = \{x \in X \mid \exists y.\, y \to x \wedge P(y)\}$.
2. $\underleftarrow{\Diamond} P = \{x \in X \mid \exists y.\, y \longrightarrow^* x \wedge P(y)\}$.
3. $\underleftarrow{\Box} P = \{x \in X \mid \forall y.\, y \longrightarrow^* x \implies P(y)\}$.

Proof Let $c: X \to F(X)$ be the underlying coalgebra.

1. $$\begin{aligned} \underleftarrow{\bigcirc} P &= \underleftarrow{\mathrm{Pred}}(F)(\{c(y) \mid y \in P\}) \\ &= \underleftarrow{\mathrm{Pred}}(F)(\textstyle\bigcup_{y \in P}\{c(y)\}) \\ &= \textstyle\bigcup_{y \in P} \underleftarrow{\mathrm{Pred}}(F)(\{c(y)\}) && \text{since } \underleftarrow{\mathrm{Pred}}(F) \text{ is a left adjoint} \\ &= \{x \in X \mid \exists y \in P.\, x \in \underleftarrow{\bigcirc}(\cdot = y)\} \\ &= \{x \in X \mid \exists y.\, y \to x \wedge P(y)\}. \end{aligned}$$
2. Let us write $P' = \{x \in X \mid \exists y.\, y \longrightarrow^* x \wedge P(y)\}$ for the right-hand side. We have to prove that P' is the least invariant containing P.
   - Clearly $P \subseteq P'$, by taking no transition.
   - Also P' is an invariant, by Proposition 6.4.7.2: if $P'(x)$, say with $y \longrightarrow^* x$ where $P(y)$, and $x \to x'$, then also $y \longrightarrow^* x'$ and thus $P'(x')$.
   - If $Q \subseteq X$ is an invariant containing P, then $P' \subseteq Q$: if $P'(x)$, say with $y \longrightarrow^* x$ where $P(y)$, then $Q(y)$, and thus $Q(x)$ by Proposition 6.4.7.2.
3. Immediately from the definition $\underleftarrow{\Box} = \neg \underleftarrow{\Diamond} \neg$. $\square$

Exercises

6.4.1 Consider the transition system $A^\star \to \mathcal{P}_{\text{fin}}(A^\star)$ from Example 6.4.3 and prove

$$\square(\{x \in A^\star \mid \exists y \in A^\star.\, x = \mathsf{M}y\})(\mathsf{MI}).$$

This property is also mentioned in [224]. In words: each successor of MI starts with M.

6.4.2 Prove the following ‘induction rule of temporal logic’:

$$P \wedge \square(P \Rightarrow \bigcirc P) \subseteq \square P.$$

(Aside: using the term ‘induction’ for a rule that follows from a greatest fixed point property is maybe not very fortunate.)

6.4.3 Prove that for a predicate P on the state space of coalgebra of a Kripke polynomial functor,

$$\square P = \bigcap_{n\in\mathbb{N}} \bigcirc^n P \qquad \text{and} \qquad \underleftarrow{\Diamond} P = \bigcup_{n\in\mathbb{N}} \underleftarrow{\bigcirc}^n P$$

(where $\bigcirc^0 P = P$, and $\bigcirc^{n+1} P = \bigcirc \bigcirc^n P$, and similarly for $\underleftarrow{\bigcirc}$).

6.4.4 Prove that

$$\begin{aligned} \square(f^{-1}Q) &= f^{-1}(\square Q) \\ \underleftarrow{\bigcirc}(\textstyle\coprod_f P) &= \textstyle\coprod_f(\underleftarrow{\bigcirc} P) \\ \underleftarrow{\Diamond}(\textstyle\coprod_f P) &= \textstyle\coprod_f(\underleftarrow{\Diamond} P) \end{aligned}$$

when f is a homomorphism of coalgebras.

6.4.5 Consider coalgebras $c\colon X \to \mathcal{M}_M(X)$ and $d\colon Y \to \mathcal{D}(Y)$ of the multiset and distribution functors. Use Exercise 6.1.4 to prove

$$\bigcirc(P \subseteq X) = \{x \mid \forall x'.\, c(x)(x') \neq 0 \Rightarrow P(x')\}.$$

What is $\square(Q \subseteq Y)$?

6.4.6 Demonstrate that the least fixed point $\mu\bigcirc$ of the nexttime operator $\bigcirc\colon \mathcal{P}(X) \to \mathcal{P}(X)$ can be characterised as

$$\mu\bigcirc = \{x \in X \mid \text{there are no infinite paths } x \to x_1 \to x_2 \to \cdots\}.$$

6.4.7 Consider the transition relation $\longrightarrow$ from Definition 6.4.6 and use Lemma 6.4.5.2 to prove that for a homomorphism $f\colon X \to Y$ of coalgebras,

$$f(x) \to y \iff \exists x'.\, x \to x' \wedge f(x') = y.$$

Note that this captures the functoriality of the translation from coalgebras to transition systems as in (6.4).

6.4.8 Show that each subset can be written as intersection of non-singletons (coequations): for $U \subseteq X$,

$$U = \bigcap_{x \in \neg U} (\cdot \neq x).$$

This result forms the basis for formulating a (predicate) logic for coalgebras in terms of these ‘coequations’ $(\cdot \neq x)$; see [177].

6.4.9 Prove – and explain in words – that

$$x \longrightarrow^{*} x' \iff x' \in \underleftarrow{\Diamond}(\cdot = x).$$

(The notation $\langle x \rangle = \{x' \mid x \longrightarrow^{*} x'\}$ and $\langle P \rangle = \{x' \mid \exists x \in P.\, x \longrightarrow^{*} x'\}$ is used in [411] for the least invariants $\underleftarrow{\Diamond}\{x\} = \underleftarrow{\Diamond}(\cdot = x)$ and $\underleftarrow{\Diamond}\, P$ containing an element x or a predicate P.)

6.4.10 Verify the Galois connections in Figure 6.2. (Such Galois connections for temporal logic are studied systematically in [288, 243].)

6.4.11 Check that $P \,\mathcal{U}\, P = P$.

6.4.12 The following is taken from [120, section 5], where it is referred to as the Whisky Problem. It is used there as a challenge in proof automation in linear temporal logic. Here it will be formulated in the temporal logic of an arbitrary coalgebra (of a Kripke polynomial functor). Consider an arbitrary set A with an endofunction $h\colon A \to A$. Let $P\colon A \to \mathcal{P}(X)$ be a parametrised predicate on the state space of a coalgebra X, satisfying for a specific $a \in A$ and $y \in X$:

1. $P(a)(y)$
2. $\forall b \in A.\, P(b)(y) \Rightarrow P(h(b))(y)$
3. $\Box\,(\{x \in X \mid \forall b \in A.\, P(h(b))(x) \Rightarrow \bigcirc(P(b))(x)\})(y)$.

Prove then that $\Box(P(a))(y)$ *Hint*: Use Exercise 6.4.3.

6.4.13 Describe the nexttime operator $\bigcirc$ as a natural transformation $\mathcal{P}U \Rightarrow \mathcal{P}U$, as in Corollary 6.1.4, where $U\colon \mathbf{CoAlg}(F) \to \mathbf{Sets}$ is the forgetful functor and $\mathcal{P}$ is the contravariant powerset functor. Show that it can be described as a composition of natural transformations:

$$\begin{tikzcd} \mathcal{P}U \arrow[rr, Rightarrow, "\bigcirc"] \arrow[dr, Rightarrow, "\mathrm{Pred}(F)U"'] & & \mathcal{P}U \\ & \mathcal{P}FU \arrow[ur, Rightarrow, "c^{-1}U"'] & \end{tikzcd}$$

6.4.14 Prove that the ‘until’ and ‘since’ operators $\mathcal{U}, \mathcal{S}\colon \mathcal{P}(X) \times \mathcal{P}(X) \to \mathcal{P}(X)$ on the state space X of a coalgebra (see Figures 6.1 and 6.2) can be described in the following way in terms of the transition relation $\to \subseteq X \times X$ from Definition 6.4.6:

$$\begin{aligned} P \,\mathcal{U}\, Q &= \{x \mid \exists n.\, \exists x_0, \ldots, x_n.\, x_0 = x \wedge (\forall i < n.\, x_i \to x_{i+1}) \wedge Q(x_n) \\ &\qquad \wedge \forall i < n.\, P(x_i)\} \\ P \,\mathcal{S}\, Q &= \{x \mid \exists n.\, \exists x_0, \ldots, x_n.\, x_n = x \wedge (\forall i < n.\, x_i \to x_{i+1}) \wedge Q(x_0) \\ &\qquad \wedge \forall i > 0.\, P(x_i)\}. \end{aligned}$$

6.4.15 We consider the strong nexttime operator $\neg\bigcirc\neg$ associated with a coalgebra and call a predicate P **maintainable** if $P \subseteq \neg\bigcirc\neg P$. Notice that such a predicate is a $\neg\bigcirc\neg$-coalgebra.

1. Investigate what this requirement means, for instance for a few concrete coalgebras.
2. Let us use the notation $\mathsf{EA}\,P$ for the greatest maintainable predicate contained in P. Describe $\mathsf{EA}\,P$ in terms of the transition relation $\to$ from Definition 6.4.6.
3. Similarly for $\mathsf{AE}\,P \stackrel{\text{def}}{=} \neg\mathsf{EA}\,\neg P$

(Operators like such as EA and AE are used in computation tree logic (CTL); see e.g. [128] to reason about paths in trees of computations. The interpretations we use here involve infinite paths.)

6.5 Modal Logic for Coalgebras

In the previous section we have seen a temporal logic for coalgebras based on the nexttime operator; see Definition 6.4.1. The meaning of $\bigcirc P$ is that the predicate P holds in *all* direct successor states. This operator is very useful for expressing safety and liveness properties, via the derived henceforth and eventually operators $\Box$ and $\Diamond$, expressing ‘for all/some future states . . . ’. But this temporal logic is not very useful for expressing more refined properties dealing for instance with one particular branch. Modal logics (including dynamic logics [198]) are widely used in computer science to reason about various kinds of dynamical systems. Often they are tailored to a specific domain of reasoning. As usual in coalgebra, the goal is to capture many of these variations in a common abstract framework.

Consider a simple coalgebra $c = \langle c_1, c_2\rangle\colon X \to X \times X$ involving two transition maps $c_1, c_2\colon X \to X$. The meaning of $\bigcirc(P)$ from temporal logic for this coalgebra is

$$\bigcirc(P) = \{x \mid c_1(x) \in P \wedge c_2(x) \in P\}.$$

Thus it contains those states x *all* of whose successors $c_1(x)$ and $c_2(x)$ satisfy P. It would be useful to have two separate logical operators, say $\bigcirc_1$ and $\bigcirc_2$, talking specifically about transition maps c_1 and c_2, respectively, as in

$$\bigcirc_1(P) = \{x \mid c_1(x) \in P\} \quad \text{and} \quad \bigcirc_2(P) = \{x \mid c_2(x) \in P\}. \tag{6.10}$$

Coalgebraic modal logic allows us to describe such operators.

To see another example/motivation, consider a coalgebra $c: X \to \mathcal{M}_{\mathbb{N}}(X)$ of the multiset (or bag) functor $\mathcal{M}_{\mathbb{N}}$, counting in the natural numbers $\mathbb{N}$. Thus we may write $x \xrightarrow{n} x'$ if $c(x)(x') = n \in \mathbb{N}$, expressing a transition which costs for instance n resources. For each $N \in \mathbb{N}$, a so-called graded modality $\bigcirc_N$ (see [131]) is defined as

$$\bigcirc_N(P) = \{x \mid \forall x'.\, c(x)(x') \geq N \Rightarrow P(x')\}.$$

This section describes how to obtain such operators in the research field known as 'coalgebraic modal logic'. The approach that we follow is rather concrete and 'hands-on'. The literature on the topic is extensive (see e.g. [359, 322, 404, 373, 317, 296, 315, 273] or the overview papers [318, 323, 99]), but there is a tendency to wander off into meta-theory and to omit specific examples (and what the modal logic might be useful for).

We begin with an abstract definition describing the common way that coalgebraic logics are now understood. It involves a generalised form of predicate lifting, as will be explained and illustrated subsequently. An even more abstract approach will be described later on, in Section 6.5.1.

Definition 6.5.1 For a functor $F: \mathbf{Sets} \to \mathbf{Sets}$, a **coalgebraic modal logic** is given by a 'modal signature functor' $L: \mathbf{Sets} \to \mathbf{Sets}$ and a natural transformation

$$L\mathcal{P} \xRightarrow{\;\delta\;} \mathcal{P}F,$$

where $\mathcal{P} = 2^{(-)}: \mathbf{Sets}^{\mathrm{op}} \to \mathbf{Sets}$ is the contravariant powerset functor.

Given such a $\delta: L\mathcal{P} \Rightarrow \mathcal{P}F$, each F-coalgebra $c: X \to F(X)$ yields an L-algebra on the set of predicates $\mathcal{P}(X)$, namely:

$$L(\mathcal{P}(X)) \xrightarrow{\;\delta_X\;} \mathcal{P}(F(X)) \xrightarrow{\;c^{-1} = \mathcal{P}(c)\;} \mathcal{P}(X).$$

This yields a functor $\mathbf{CoAlg}(F)^{\mathrm{op}} \to \mathbf{Alg}(L)$ in a commuting diagram:

$$\begin{tikzcd}
\mathbf{CoAlg}(F)^{\mathrm{op}} \arrow[r] \arrow[d] & \mathbf{Alg}(L) \arrow[d] \\
\mathbf{Sets}^{\mathrm{op}} \arrow[r, "\mathcal{P}"] & \mathbf{Sets}
\end{tikzcd} \tag{6.11}$$

Recall from Corollary 6.1.4 that predicate lifting is described as a functor $\mathrm{Pred}(F): \mathcal{P} \Rightarrow \mathcal{P}F$. In the above definition this is generalised by adding a functor L in front, yielding $L\mathcal{P} \Rightarrow \mathcal{P}F$. This L makes more flexible liftings – and thus more flexible modal operators – possible, as will be illustrated next.

A less general and more clumsy approach using 'ingredients' of functors is used in [241]. One can think of the functor L as describing the type of these ingredient operators, which becomes explicit if one considers the induced L-algebra $L(\mathcal{P}(X)) \to \mathcal{P}(X)$. For instance, for the two operators from (6.10) we use $L(Y) = Y + Y$, so that we can describe the pair of nexttime operators in (6.10) as L-algebra:

$$L(\mathcal{P}(X)) = \mathcal{P}(X) + \mathcal{P}(X) \xrightarrow{[\bigcirc_1, \bigcirc_2]} \mathcal{P}(X),$$

arising according to the pattern in the definition via a map δ in

$$L(\mathcal{P}(X)) = \mathcal{P}(X) + \mathcal{P}(X) \xrightarrow{\delta} \mathcal{P}(X \times X) \xrightarrow{\langle c_1, c_2\rangle^{-1}} \mathcal{P}(X),$$

where $\delta = [\delta_1, \delta_2] \colon L(\mathcal{P}(X)) \to \mathcal{P}(X \times X)$ is given by

$$\begin{aligned} \delta_1(P) &= \pi_1^{-1} = \mathcal{P}(\pi_1) \\ &= \{(x_1, x_2) \in X \times X \mid P(x_1)\} \end{aligned} \qquad \begin{aligned} \delta_2(P) &= \pi_2^{-1} = \mathcal{P}(\pi_2) \\ &= \{(x_1, x_2) \in X \times X \mid P(x_2)\}. \end{aligned}$$

Then indeed, $\bigcirc_i(P) = \langle c_1, c_2\rangle^{-1} \circ \delta \circ \kappa_i = c_i^{-1}$.

We turn to a more elaborate illustration.

Example 6.5.2 Suppose we wish to describe a simple bank account coalgebraically. It involves a state space X, considered as black box, accessible only via the next three balance, deposit and withdraw operations.

$\mathsf{bal} \colon X \longrightarrow \mathbb{N}$	for learning the balance of the account, which, for simplicity, is represented as a natural number
$\mathsf{dep} \colon X \times \mathbb{N} \longrightarrow X$	for depositing a certain amount of money, given as parameter, into the account
$\mathsf{wdw} \colon X \times \mathbb{N} \longrightarrow X + X$	for withdrawing a certain amount of money, given as parameter, from the account; the first/left output option of + is used for a successful withdrawal, when the balance before the withdrawal exceeds the retrieved amount. The second/right +-option is used for unsuccessful withdrawals; in that case the balance remains unchanged.

Together these maps form a coalgebra of the form

$$X \xrightarrow{\langle \mathsf{bal}, \mathsf{dep}, \mathsf{wdw}\rangle} F(X) \quad \text{for} \quad F(X) = \mathbb{N} \times X^{\mathbb{N}} \times (X + X)^{\mathbb{N}}.$$

We would like to express the above informal descriptions of the behaviour of this coalgebra in precise logical terms, using modal operators. Therefore we define predicates:

$$\begin{aligned} \mathsf{bal}\downarrow n &= \{x \in X \mid \mathsf{bal}(x) = n\} \\ [\mathsf{dep}(n)](P) &= \{x \in X \mid \mathsf{dep}(x, n) \in P\}. \end{aligned}$$

Now we can require

$$\mathsf{bal}\downarrow m \vdash [\mathsf{dep}(n)](\mathsf{bal}\downarrow(m+n)),$$

where $\vdash$ should be understood as subset inclusion $\subseteq$. Obviously this captures the intended behaviour of 'deposit'.

The modal operator for withdrawal is more subtle because of the two output options in $X+X$. Therefore we define two modal operators, one for each option:

$$\begin{aligned} [\mathsf{wdw}(n)]_1(P) &= \{x \in X \mid \forall x'.\, \mathsf{wdw}(x, n) = \kappa_1 x' \Rightarrow P(x')\} \\ [\mathsf{wdw}(n)]_2(P) &= \{x \in X \mid \forall x'.\, \mathsf{wdw}(x, n) = \kappa_2 x' \Rightarrow P(x')\}. \end{aligned}$$

One may now expect a requirement:

$$\mathsf{bal}\downarrow(m+n) \vdash [\mathsf{wdw}(n)]_1(\mathsf{bal}\downarrow m).$$

But a little thought reveals that this is too weak, since it does not enforce that a successful withdrawal yields an output in the first/left +-option. We need to use the derived operator $\langle f\rangle(P) = \neg[f](\neg P)$ so that

$$\langle\mathsf{wdw}(n)\rangle_i(P) = \neg[\mathsf{wdw}(n)]_i(\neg P) = \{x \in X \mid \exists x'.\, \mathsf{wdw}(x, n) = \kappa_i x' \wedge P(x')\}.$$

Now we can express the remaining requirements as

$$\begin{aligned} \mathsf{bal}\downarrow(m+n) &\vdash \langle\mathsf{wdw}(n)\rangle_1(\mathsf{bal}\downarrow m) \\ \mathsf{bal}\downarrow(m) &\vdash \langle\mathsf{wdw}(m+n+1)\rangle_2(\mathsf{bal}\downarrow m). \end{aligned}$$

Thus we have used four logical operators to lay down the behaviour of a bank account, via restriction of the underlying coalgebra $X \to F(X)$. Such restrictions will be studied more systematically in Section 6.8. Here we concentrate on the modal/dynamic operations. The four of them can be described jointly in the format of Definition 6.5.1, namely as 4-cotuple:

$$[\,\mathsf{bal}\downarrow(-),\ [\mathsf{dep}(-)],\ [\mathsf{wdw}_1(-)],\ [\mathsf{wdw}_1(-)]\,],$$

forming an algebra:

$$\mathbb{N} + (\mathbb{N} \times \mathcal{P}(X)) + (\mathbb{N} \times \mathcal{P}(X)) + (\mathbb{N} \times \mathcal{P}(X)) \longrightarrow \mathcal{P}(X). \tag{6.12}$$

It is an algebra $L(\mathcal{P}(X)) \to \mathcal{P}(X)$ for the modal signature functor $L\colon \mathbf{Sets} \to \mathbf{Sets}$ given by

$$L(Y) \;=\; \mathbb{N} + (\mathbb{N} \times Y) + (\mathbb{N} \times Y) + (\mathbb{N} \times Y).$$

The algebra of operators (6.12) is the composition of two maps:

$$L(\mathcal{P}(X)) \xrightarrow{\quad\delta\quad} \mathcal{P}(F(X)) \xrightarrow{\quad c^{-1}\quad} \mathcal{P}(X),$$

where $c = \langle \mathsf{bal}, \mathsf{dep}, \mathsf{wdw}\rangle\colon X \to F(X) = \mathbb{N} \times X^{\mathbb{N}} \times (X + X)^{\mathbb{N}}$ is the coalgebra involved. The map $\delta\colon L(\mathcal{P}(X)) \to \mathcal{P}(F(X))$ is a 4-cotuple of the form $\delta = [\delta_{\mathsf{bal}}, \delta_{\mathsf{dep}}, \delta_{\mathsf{wdw},1}, \delta_{\mathsf{wdw},2}]$, where

$$\begin{aligned}
\delta_{\mathsf{bal}}(n) &= \{(m, f, g) \in F(X) \mid m = n\} \\
\delta_{\mathsf{dep}}(n, P) &= \{(m, f, g) \in F(X) \mid f(n) \in P\} \\
\delta_{\mathsf{wdw},1}(n, P) &= \{(m, f, g) \in F(X) \mid \forall x.\, g(n) = \kappa_1 x \Rightarrow P(x)\} \\
\delta_{\mathsf{wdw},2}(n, P) &= \{(m, f, g) \in F(X) \mid \forall x.\, g(n) = \kappa_2 x \Rightarrow P(x)\}.
\end{aligned}$$

It is not hard to see that this δ is a natural transformation $L\mathcal{P} \Rightarrow \mathcal{P}F$.

A next step is to see how the modal signature functor L and the natural transformation $\delta\colon L\mathcal{P} \Rightarrow \mathcal{P}F$ in Definition 6.5.1 arise. In general, this is a matter of choice. But in many cases, for instance when the functor is a polynomial, there are some canonical choices. We shall illustrate this in two steps, namely by first defining coalgebraic logics for several basic functors, and subsequently showing that coalgebraic logics can be combined via several constructions, such as composition and (co)product. A similar, but more language-oriented, modular approach is described in [100].

Definition 6.5.3 Coalgebraic logics $(L, L\mathcal{P} \overset{\delta}{\Rightarrow} \mathcal{P}F)$ can be defined when F is the identity/constant/powerset/multiset/distribution functor in the following manner.

1. For the identity functor $\mathrm{id}\colon \mathbf{Sets} \to \mathbf{Sets}$ we take $L = \mathrm{id}$, with identity natural transformation:

$$L(\mathcal{P}(X)) = \mathcal{P}(X) \xrightarrow{\quad\delta\, =\, \mathrm{id}\quad} \mathcal{P}(X) = \mathcal{P}(\mathrm{id}(X)).$$

2. For a constant functor $A\colon \mathbf{Sets} \to \mathbf{Sets}$, given by $X \mapsto A$, we also take $L = A$, with singleton (unit) map:

$$L(\mathcal{P}(X)) = A \xrightarrow{\quad\delta\, =\, \{-\}\quad} \mathcal{P}(A) = \mathcal{P}(A(X)).$$

3. For the (covariant) powerset functor $\mathcal{P}$ we take $L = \mathrm{id}$, with

$$L(\mathcal{P}(X)) = \mathcal{P}(X) \xrightarrow{\quad\delta\quad} \mathcal{P}(\mathcal{P}(X))$$

given by $\delta(P) = \{U \in \mathcal{P}(X) \mid U \subseteq P\}$. For the finite powerset functor $\mathcal{P}_{\mathrm{fin}}$ an analogous map $\mathcal{P}(X) \to \mathcal{P}(\mathcal{P}_{\mathrm{fin}}(X))$ is used.
4. For the distribution functor $\mathcal{D}\colon \mathbf{Sets} \to \mathbf{Sets}$ take $L(Y) = [0,1]_{\mathbb{Q}} \times Y$, where $[0,1]_{\mathbb{Q}} = [0,1] \cap \mathbb{Q}$ is the unit interval of rational numbers. The associated natural transformation follows the ideas of [210]:

$$L(\mathcal{P}(X)) = [0,1]_{\mathbb{Q}} \times \mathcal{P}(X) \xrightarrow{\quad\delta\quad} \mathcal{P}(\mathcal{D}X),$$

where $\delta(r, P) = \{\varphi \in \mathcal{D}(X) \mid \forall x'.\, \varphi(x)(x') \geq r \Rightarrow P(x')\}$.
5. The same coalgebraic modal logic can be used for the multiset functor $\mathcal{M}_M$, assuming the monoid carries an order. This yields the graded operators, as sketched in the beginning of this section for the multiset/bag functor $\mathcal{M}_{\mathbb{N}}$ over the natural numbers.
6. For the neighbourhood functor $\mathcal{N}(X) = 2^{(2^X)}$ from Exercise 2.2.7 one takes $L = \mathrm{id}$ with natural transformation

$$L(\mathcal{P}(X)) = \mathcal{P}(X) \xrightarrow{\quad\delta\quad} \mathcal{P}(\mathcal{N}X) = \mathcal{P}\mathcal{P}\mathcal{P}(X),$$

given by $\delta(P) = \{V \in \mathcal{P}\mathcal{P}(X) \mid P \in V\}$.

Lemma 6.5.4 *Coalgebraic modal logics can be combined in the following ways.*

1. *For two functors $F_1, F_2\colon$ **Sets** $\to$ **Sets**, each with a coalgebraic modal logic (L_i, δ_i), there is also a coalgebraic modal logic for the composite functor $F_1 \circ F_2$, namely $L_1 \circ L_2$ with natural transformation given by*

$$L_1 L_2 \mathcal{P}(X) \xrightarrow{\;L_1(\delta_{2,X})\;} L_1 \mathcal{P} F_2(X) \xrightarrow{\;\delta_{1,F_2(X)}\;} \mathcal{P} F_1 F_2(X).$$

2. *Given an I-indexed collection of functors $(F_i)_{i\in I}$ with logics (L_i, δ_i) we define a modal signature functor $L(Y) = \coprod_{i\in I} L_i(Y)$ for the coproduct functor $\coprod_{i\in I} F_i$, with natural transformation*

$$L(\mathcal{P}(X)) = \coprod_{i\in I} L_i(\mathcal{P}(X)) \xrightarrow{\;[\coprod_{\kappa_i} \circ\, \delta_i]_{i\in I}\;} \mathcal{P}(\coprod_{i\in I} F_i(X)).$$

We write $\coprod_{\kappa_i}\colon \mathcal{P}(F_i(X)) \to \mathcal{P}(\coprod_i F_i(X))$ for the direct *image, using $\mathcal{P}$ as a covariant functor.*

3. *Similarly, for a product functor* $\prod_{i\in I} F_i$ *we use the coproduct* $\coprod_{i\in I} L_i$ *of associated modal signature functors, with*

$$L(\mathcal{P}(X)) = \coprod_{i\in I} L_i(\mathcal{P}(X)) \xrightarrow{\;[\pi_i^{-1}\circ\delta_i]_{i\in I}\;} \mathcal{P}(\prod_{i\in I} F_i(X)).$$

4. *As a special case of the previous point we make the exponent functor* F^A *explicit. It has a modal signature functor* $A \times L(-)$*, assuming a coalgebraic modal logic* (L,δ) *for* F*, with natural transformation*

$$A \times L(\mathcal{P}(X)) \longrightarrow \mathcal{P}(F(X)^A),$$

given by $(a,u) \mapsto \{f \in F(X)^A \mid f(a) \in \delta(u)\}$.

Proof One has only to check that the new δs are natural transformations; this is easy. ❑

As a result, each Kripke polynomial functor has a (canonical) coalgebraic modal logic.

6.5.1 Coalgebraic Modal logic, More Abstractly

In Definition 6.5.1 we have introduced a coalgebraic modal logic for a functor F as a pair (L,δ), where $\delta\colon L\mathcal{P} \Rightarrow \mathcal{P}F$. As we have seen, this approach works well for many endofunctors on **Sets**. Still, the picture is a bit too simple, for two reasons.

- Modal operators usually satisfy certain preservation properties. In particular, almost all of them preserve finite conjunctions $(\top,\wedge)$. This can be captured by restricting the modal signature functor L, from an endofunctor on **Sets** to an endofunctor on the category **MSL** of meet semilattices.
- The approach is defined for endofunctors on **Sets** and not for endofunctors on an arbitrary category. In order to take care of this additional generality we will use the more general logic described in terms of factorisation systems in Section 4.3.

We will tackle both these issues at the same time. We proceed in a somewhat informal manner; refer to the literature [323] for further details and ramifications.

Let $F\colon \mathbb{C} \to \mathbb{C}$ be an endofunctor on a category $\mathbb{C}$ with a logical factorisation system $(\mathfrak{M},\mathfrak{E})$. Recall from Lemma 4.3.4 that it gives rise to an indexed category

$$\mathbb{C}^{\mathrm{op}} \xrightarrow{\;\mathrm{Pred}(-)\;} \mathbf{MSL}.$$

For each object $X \in \mathbb{C}$ the poset $\mathrm{Pred}(X)$ of predicates $(U \rightarrowtail X) \in \mathfrak{M}$ on X has finite meets $\top, \wedge$; additionally, a map $f\colon X \to Y$ in $\mathbb{C}$ yields a finite meet-preserving substitution functor $\mathrm{Pred}(f) = f^{-1}\colon \mathrm{Pred}(Y) \to \mathrm{Pred}(X)$ by pullback.

In this general situation, a **coalgebraic modal logic** for $F\colon \mathbb{C} \to \mathbb{C}$ consists of a functor $L\colon \mathbf{MSL} \to \mathbf{MSL}$ together with a natural transformation $\delta\colon L\mathrm{Pred} \Rightarrow \mathrm{Pred}F$. Thus, δ is a natural transformation between the following two parallel functors:

$$\begin{tikzcd} & \mathbf{MSL} \arrow[dr, "L"] & \\ \mathbb{C}^{\mathrm{op}} \arrow[ur, "\mathrm{Pred}"] \arrow[dr, "F"'] & & \mathbf{MSL} \\ & \mathbb{C}^{\mathrm{op}} \arrow[ur, "\mathrm{Pred}"'] & \end{tikzcd}$$

In Exercise 6.5.9 it is shown that all coalgebraic models described in Definition 6.5.3 live in this way in the category **MSL**, except for the neighbourhood functor $\mathcal{N}$. Also, by Exercise 6.5.8 all the constructions on coalgebraic modal logics in Lemma 6.5.4 can be performed in the category **MSL**.

A coalgebra $c\colon X \to F(X)$ in the base category $\mathbb{C}$ gives rise to an L-algebra, as before in Definition 6.5.1, but this time in the category **MSL**:

$$L(\mathrm{Pred}(X)) \xrightarrow{\ \delta\ } \mathrm{Pred}(F(X)) \xrightarrow{\ c^{-1} = \mathrm{Pred}(c)\ } \mathrm{Pred}(X).$$

This yields a lifted functor $\mathrm{Pred}\colon \mathbf{CoAlg}(F)^{\mathrm{op}} \to \mathbf{Alg}(L)$, in a slightly different diagram from (6.11):

$$\begin{tikzcd} \mathbf{CoAlg}(F)^{\mathrm{op}} \arrow[r, "\mathrm{Pred}"] \arrow[d] & \mathbf{Alg}(L) \arrow[d] \\ \mathbf{Sets}^{\mathrm{op}} \arrow[r, "\mathrm{Pred}"] & \mathbf{MSL} \end{tikzcd}$$

Next assume that the functor $L\colon \mathbf{MSL} \to \mathbf{MSL}$ has an initial algebra. We shall write it as

$$L(\mathit{Form}) \xrightarrow{\ \cong\ } \mathit{Form} \tag{6.13}$$

where '*Form*' stands for 'formulas'. This set $\mathit{Form} \in \mathbf{MSL}$ is by construction closed under finite conjunctions $(\top, \wedge)$ and comes equipped with modal operators via the above algebra $L(\mathit{Form}) \to \mathit{Form}$. By initiality we get a unique homomorphism in **MSL**:

$$\begin{tikzcd} L(\mathit{Form}) \arrow[r, dashed, "L(\llbracket - \rrbracket_c)"] \arrow[d, "\cong"'] & L(\mathrm{Pred}(X)) \arrow[d, "\mathrm{Pred}(c) \circ \delta"] \\ \mathit{Form} \arrow[r, dashed, "\llbracket - \rrbracket_c"'] & \mathrm{Pred}(X) \end{tikzcd} \tag{6.14}$$

It maps a formula $\varphi \in \mathit{Form}$ to its interpretation $\llbracket \varphi \rrbracket_c \in \mathrm{Pred}(X)$, as an $\mathfrak{M}$-subobject of the state space X. This map $\llbracket - \rrbracket$ preserves finite meets and preserves the modal operators.

Remark 6.5.5 The collection $\mathit{Form} \in \mathbf{MSL}$ of logical formulas defined in (6.13) has finite meets by construction. What if we would like to have all Boolean operations on formulas? The obvious way would be to construct *Form* as an initial algebra in the category **BA** of Boolean algebras. But this approach does not work, because in general the modal operations $L(\mathit{Form}) \to \mathit{Form}$ preserve only finite meets and for instance not $\neg$ or $\vee$.

There is a neat trick around this; see e.g. [273, 323]. We use that the forgetful functor $U\colon \mathbf{BA} \to \mathbf{MSL}$ from Boolean algebras to meet semilattices has a left adjoint $F\colon \mathbf{MSL} \to \mathbf{BA}$ – which follows from Exercise 5.4.15.3. Now we consider the functor

$$L' = \Big(\mathbf{BA} \xrightarrow{\;U\;} \mathbf{MSL} \xrightarrow{\;L\;} \mathbf{MSL} \xrightarrow{\;F\;} \mathbf{BA}\Big).$$

There is a direct (adjoint) correspondence between L'- and L-algebras: for a Boolean algebra B,

$$\begin{array}{lc} FLU(B) = \dfrac{L'(B) \longrightarrow B}{\overline{\overline{\;L(UB) \longrightarrow UB\;}}} & \begin{array}{l}\text{in } \mathbf{BA} \\[1ex] \text{in } \mathbf{MSL}.\end{array} \end{array}$$

Thus if we now define the collection Form' as initial algebra of the functor L' in **BA**, then Form' carries all Boolean structure and has modal operators $L(\mathit{Form}') \to \mathit{Form}'$ that preserve only finite meets.

In order to proceed further we need another assumption, namely that the predicate functor $\mathrm{Pred}\colon \mathbb{C}^{\mathrm{op}} \to \mathbf{MSL}$ has a left adjoint $\mathcal{S}$. Thus we have an adjoint situation:

$$\begin{tikzcd} \mathbb{C}^{\mathrm{op}} \arrow[r, bend left, "\mathrm{Pred}"] \arrow[r, phantom, "\top"] & \mathbf{MSL}. \arrow[l, bend left, "\mathcal{S}"] \end{tikzcd} \tag{6.15}$$

Exercise 6.5.7 deals with some situations where this is the case. Such 'dual' adjunctions form the basis for many dualities; see [280], relating predicates and states, for instance in domain theory [6], probabilistic computing [308], or in quantum computing [121, 257] and see also Exercise 5.4.11.

In presence of this adjunction (6.15), two things are relevant.

- The natural transformation $\delta\colon L\mathrm{Pred} \Rightarrow \mathrm{Pred}F$, forming the modal coalgebraic logic for the functor F, bijectively corresponds to another natural transformation $\overline{\delta}$, as in
$$\frac{L\mathrm{Pred} \overset{\delta}{\Longrightarrow} \mathrm{Pred}F}{F\mathcal{S} \underset{\overline{\delta}}{\Longrightarrow} \mathcal{S}L} \;. \tag{6.16}$$
Working out this correspondence is left to the interested reader, in Exercise 6.5.10.
- We can take the transpose of the interpretation map $[\![-]\!]_c\colon \mathit{Form} \to \mathrm{Pred}(X)$ from (6.14). It yields a 'theory' map $th_c : X \to \mathcal{S}(\mathit{Form})$ that intuitively sends a state to the formulas that hold for this state. The relation containing the states for which the same formulas hold is given as kernel/equaliser $\equiv_c \rightarrowtail X \times X$ in $\mathbb{C}$:
$$\equiv_c = \mathrm{Ker}(th_c) \rightarrowtail X \times X \underset{th_c \circ \pi_2}{\overset{th_c \circ \pi_1}{\rightrightarrows}} \mathcal{S}(\mathit{Form}). \tag{6.17}$$

An important question is how this relation $\equiv_c$ relates to the notions of indistinguishability that we have seen for coalgebras. It turns out that the behavioural equivalence (cospan) definition works best in this situation. The next result is based on [419] and also, in more categorical form, on [377, 297, 273].

Theorem 6.5.6 *Consider the situation described above, where the functor* $F\colon \mathbb{C} \to \mathbb{C}$ *has a modal coalgebraic logic* $\delta\colon L\mathrm{Pred} \Rightarrow \mathrm{Pred}F$ *with initial algebra* $L(\mathit{Form}) \xrightarrow{\cong} \mathit{Form}$, *and where there is a left adjoint* $\mathcal{S}$ *to the indexed category* $\mathrm{Pred}\colon \mathbb{C}^{\mathrm{op}} \to \mathbf{MSL}$ *associated with the logical factorisation system* $(\mathfrak{M}, \mathfrak{E})$ *on* $\mathbb{C}$, *as in (6.15).*

1. *Observationally equivalent states satisfy the same logical formulas: each kernel of a coalgebra map factors through the equaliser* $\equiv_c \rightarrowtail X \times X$ *in (6.17).*
2. *If the functor* F *preserves abstract monos (in* $\mathfrak{M}$*) and the transpose* $\overline{\delta}\colon F\mathcal{S} \Rightarrow \mathcal{S}L$ *in (6.16) consists of abstract monos, then the converse is also true: states that make the same formulas true are observationally equivalent.*

The latter property is usually called **expressivity** of the logic, or the **Hennessy–Milner** property. Originally, this property was proven in [212] but only for finitely branching transition systems. The most significant assumption in this much more general theorem for coalgebras is the $\overline{\delta}$-injectivity property of the coalgebra modal logic.

Proof 1. Let $f\colon X \to Y$ be a map of coalgebras, from $c\colon X \to F(X)$ to $d\colon Y \to F(Y)$, with kernel

$$\begin{tikzcd} \mathrm{Ker}(f) \arrow[r, tail, "\langle k_1, k_2\rangle"] & X \times X \arrow[r, shift left, "f \circ \pi_1"] \arrow[r, shift right, "f \circ \pi_2"'] & Y. \end{tikzcd}$$

By initiality we get $\mathrm{Pred}(f) \circ [\![-]\!]_d = [\![-]\!]_c$ in

$$\begin{tikzcd} L(\mathit{Form}) \arrow[r] \arrow[d, "\cong"'] & L(\mathrm{Pred}(Y)) \arrow[r] \arrow[d, "\mathrm{Pred}(d) \circ \delta_Y"] & L(\mathrm{Pred}(X)) \arrow[d, "\mathrm{Pred}(c) \circ \delta_X"] \\ \mathit{Form} \arrow[r, "{[\![-]\!]_d}"] \arrow[rr, bend right, "{[\![-]\!]_c}"'] & \mathrm{Pred}(Y) \arrow[r, "\mathrm{Pred}(f)"] & \mathrm{Pred}(X) \end{tikzcd}$$

Now we can see that $\mathrm{Ker}(f)$ factors through the equaliser $\equiv_c \rightarrowtail X \times X$ in (6.17):

$$\begin{aligned} th_c \circ k_1 &= \mathcal{S}([\![-]\!]_c) \circ \eta \circ k_1 \\ &= \mathcal{S}(\mathrm{Pred}(f) \circ [\![-]\!]_d) \circ \mathcal{S}(\mathrm{Pred}(k_1)) \circ \eta \\ &= \mathcal{S}(\mathrm{Pred}(k_1) \circ \mathrm{Pred}(f) \circ [\![-]\!]_d) \circ \eta \\ &= \mathcal{S}(\mathrm{Pred}(f \circ k_1) \circ [\![-]\!]_d) \circ \eta \\ &= \mathcal{S}(\mathrm{Pred}(f \circ k_2) \circ [\![-]\!]_d) \circ \eta \\ &= \cdots \\ &= th_c \circ k_2. \end{aligned}$$

2. We first observe that the transpose $\overline{\delta}$ makes the following diagram commute:

$$\begin{tikzcd} F(X) \arrow[r, "F(th_c)"] & F\mathcal{S}(\mathit{Form}) \arrow[r, tail, "\overline{\delta}"] & \mathcal{S}L(\mathit{Form}) \\ X \arrow[u, "c"] \arrow[rr, "th_c"'] & & \mathcal{S}(\mathit{Form}) \arrow[u, "\cong"', "\mathcal{S}(\alpha)"] \end{tikzcd}$$

where we write the initial algebra map in (6.13) as $\alpha\colon L(\mathit{Form}) \xrightarrow{\cong} \mathit{Form}$. Commutation of this rectangle follows from the explicit description $\overline{\delta} = \mathcal{S}L(\eta) \circ \mathcal{S}(\delta) \circ \varepsilon$ in Exercise 6.5.10:

$$\begin{aligned}
\overline{\delta} \circ F(th_c) \circ c &= \mathcal{S}L(\eta) \circ \mathcal{S}(\delta) \circ \varepsilon \circ F(th_c) \circ c \\
&= \mathcal{S}L(\eta) \circ \mathcal{S}(\delta) \circ \mathcal{S}\mathrm{Pred}(F(th_c) \circ c) \circ \varepsilon \\
&= \mathcal{S}(\mathrm{Pred}(c) \circ \mathrm{Pred}F(th_c) \circ \delta \circ L(\eta)) \circ \varepsilon \\
&= \mathcal{S}(\mathrm{Pred}(c) \circ \delta \circ L\mathrm{Pred}(th_c) \circ L(\eta)) \circ \varepsilon \\
&= \mathcal{S}(\mathrm{Pred}(c) \circ \delta \circ L(\llbracket - \rrbracket_c)) \circ \varepsilon \\
&= \mathcal{S}(\llbracket - \rrbracket_c \circ \alpha) \circ \varepsilon \qquad\qquad \text{by (6.14)} \\
&= \mathcal{S}(\alpha) \circ \mathcal{S}(\mathrm{Pred}(th_c) \circ \eta) \circ \varepsilon \\
&= \mathcal{S}(\alpha) \circ \mathcal{S}(\eta) \circ \mathcal{S}\mathrm{Pred}(th_c) \circ \varepsilon \\
&= \mathcal{S}(\alpha) \circ \mathcal{S}(\eta) \circ \varepsilon \circ th_c \\
&= \mathcal{S}(\alpha) \circ th_c.
\end{aligned}$$

Next we take the factorisation of the theory map th_c in

$$th_c = \Big(X \xrightarrow{\;e\;} Y \rightarrowtail^{\;m\;} \mathcal{S}(Form)\Big).$$

Since the functor F preserves maps in $\mathfrak{M}$ a coalgebra d can be defined on the image Y via diagonal-fill-in:

$$\begin{tikzcd}
X \arrow[rr, two heads, "e"] \arrow[d, "c"'] & & Y \arrow[d, tail, "m"] \arrow[ddll, dashed, "d"'] \\
F(X) \arrow[d, "F(e)"'] & & \mathcal{S}(Form) \arrow[d, "\mathcal{S}(\alpha)", "\cong"'] \\
F(Y) \arrow[r, tail, "F(m)"'] & F\mathcal{S}(Form) \arrow[r, tail, "\overline{\delta}"'] & \mathcal{S}L(Form)
\end{tikzcd}$$

In particular, the abstract epi e is a map of coalgebras $c \to d$.

If we write $\langle r_1, r_2\rangle \colon \equiv_c \rightarrowtail X \times X$ for the equaliser map in (6.17), then, by construction:

$$m \circ e \circ r_1 = th_c \circ r_1 = th_c \circ r_2 = m \circ e \circ r_2.$$

Since m is monic, this yields $e \circ r_1 = e \circ r_2$. Thus $\equiv_c$ is contained in the kernel $\mathrm{Ker}(e)$ of a map of coalgebras. Hence, states related by $\equiv_c$ are behaviourally equivalent. ❑

6.5.2 Modal Logic Based on Relation Lifting

What we have described above is coalgebraic modal logic based on (an extension of) *predicate* lifting. The first form of coalgebraic modal logic, introduced by Moss [359], was based however on *relation* lifting. This lifting is applied to the set membership relation $\in$, as in Lemma 5.2.7. In fact, the distributive law $\nabla \colon F\mathcal{P} \Rightarrow \mathcal{P}F$ described there captures the essence of the logic. This ∇ is understood as a so-called logical 'cover' operator. It leads to a non-standard syntax, which we briefly illustrate.

- Consider the functor $F(X) = X \times X$, as used in the beginning of this section. It leads to an operator $\nabla\colon \mathcal{P}(X) \times \mathcal{P}(X) \to \mathcal{P}(X \times X)$, given by

$$\nabla(P, Q) \;=\; \{(x, x') \mid P(x) \wedge Q(x')\}.$$

  We see that ∇ works on multiple predicates at the same time and returns a predicate that combines the application of these predicates. (In fact, in this case we get the 'double strength' operator dst for the powerset from Exercise 5.2.13.)
- Next consider the 'bank account' functor $F(X) = \mathbb{N} \times X^{\mathbb{N}} \times (X + X)^{\mathbb{N}}$ from Example 6.5.2. The associated $\nabla\colon F(\mathcal{P}(X)) \to \mathcal{P}(F(X))$ takes the form

$$\begin{aligned}\nabla(n, P, Q) \;=\; \{(m, f, g) \in F(X) \mid\, & m = n \wedge \forall k \in \mathbb{N}.\, f(k) \in P(k) \,\wedge \\ & \forall x, U.\, g(k) = \kappa_i x \Rightarrow Q(k) = \kappa_i U \wedge x \in U\}.\end{aligned}$$

  With some effort one can recognise within this formulation the four logical operators $\mathsf{bal}\downarrow(-)$, $[\mathsf{dep}(-)]$, $[\mathsf{wdw}_1(-)]$, $[\mathsf{wdw}_1(-)]$ that we used in Example 6.5.2. But certainly, this ∇ is not very convenient or illuminating: just try to formulate the bank requirements using ∇.

Hence this ∇-based modal logic is mostly of theoretical interest. There are ways of translating between modal logic based on predicate lifting and this ∇-logic based on relation lifting; see [323]. In that case it is convenient to use also n-ary coalgebraic modal logics, involving maps $\delta\colon \mathcal{P}(X^n) \to \mathcal{P}(FX)$.

One advantage that is sometimes claimed for this ∇-logic is that it is generic, in the sense that the logical syntax is obtained directly from the functor and does not require a choice of (L, δ) as in Definition 6.5.1. However, this argument is hardly convincing if it leads to such a non-standard syntax. Moreover, having more choice and flexibility can be both useful and convenient – in the presence of good default choices, as offered by Definition 6.5.3 and Lemma 6.5.4. For instance, consider the multiset/bag functor $\mathcal{M}_{\mathbb{N}}$. If we wish for $n, m \in \mathbb{N}$ a modality $\bigcirc_{n,m}$ that selects the outcomes in the interval $[n, m]$, we can do so easily via the predicate-lifting-based coalgebraic modal logic, via the functor $L(Y) = \mathbb{N} \times \mathbb{N} \times Y$, with $\delta\colon L\mathcal{P} \Rightarrow \mathcal{P}\mathcal{M}_{\mathbb{N}}$ given by

$$\delta(n, m, P) \;=\; \{\varphi \in \mathcal{M}_{\mathbb{N}}(X) \mid \forall x.\, n \leq \varphi(x) \leq m \Rightarrow P(x)\}.$$

Here we conclude our brief introduction to coalgebraic modal logic. It is one of the more active subfields in coalgebra, involving much more than was covered here, such as proof theory, decidability and extensions like fixed point logic. Refer to [318] for more information and references.

Exercises

6.5.1 Check that coalgebraic temporal logic (see Section 6.4) is a special case of coalgebraic modal logic.

6.5.2 Show that a coalgebraic modal logic $\delta\colon L\mathcal{P} \Rightarrow \mathcal{P}F$ induces a functor $\mathbf{CoAlg}(F)^{\mathrm{op}} \to \mathbf{Alg}(L)$, as claimed in Definition 6.5.1, and that it makes the diagram (6.11) commute.

6.5.3 Consider in Example 6.5.2 the (senseless) action $\mathsf{wdw}(0)$ of withdrawing nothing. According to the requirements given there, does $\mathsf{wdw}(0)$ end up in the left or in the right option in $X + X$? Reformulate the requirements in such a way that $\mathsf{wdw}(0)$ is handled differently, via the other $+$-option.

6.5.4 Give two different implementations of the bank account coalgebra in Example 6.5.2:

1. One with the natural numbers $\mathbb{N}$ as state space, and $\mathsf{bal} = \mathrm{id}\colon \mathbb{N} \to \mathbb{N}$;
2. A 'history' model with non-empty lists $\mathbb{N}^+$ of natural numbers as states, where $\mathsf{bal} = \mathsf{last}\colon \mathbb{N}^+ \to \mathbb{N}$.

Of course, the requirements in Example 6.5.2 must hold for these implementations.

6.5.5 Recall from Proposition 2.2.3 that each simple polynomial functor F can be written as an arity functor of the form $F_\#(X) = \coprod_{i\in I} X^{\#i}$, for an arity $\#\colon I \to \mathbb{N}$. Show that the modal signature functor for this arity is $L_\#(Y) = \coprod_{i\in I}(\#i) \times X$ and describe the natural transformation $\delta_\#\colon L_\#(\mathcal{P}(X)) \to \mathcal{P}(F_\#(X))$ according to Definition 6.5.3 and Lemma 6.5.4.

6.5.6 Apply the previous exercise to the functor $F(X) = 1 + X + (E \times X)$ used in Section 1.1 for a simplified representation of statements of the Java programming language. What are the associated modal operations, and how are they interpreted in a Java context?

6.5.7
1. Show that the powerset functor $\mathcal{P}\colon \mathbf{Sets}^{\mathrm{op}} \to \mathbf{MSL}$ has a left adjoint $\mathcal{S}$ that sends a meet semilattice D to the set of filters in D: upwards closed subsets $U \subseteq D$ with $\top \in U$ and $x, y \in U \Rightarrow x \wedge y \in U$.
2. Sending a topological space X to its opens $\mathcal{O}(X)$ yields a functor $\mathbf{Sp}^{\mathrm{op}} \to \mathbf{MSL}$ from the opposite of the category $\mathbf{Sp}$ of topological spaces and continuous maps to the category $\mathbf{MSL}$. Show that it also has the filter functor $\mathcal{S}$ as left adjoint, where $\mathcal{S}(D)$ carries the smallest topology that makes the subsets of filters $\eta(a) = \{U \in \mathcal{S}(L) \mid a \in U\}$ open, for $a \in D$.

3. (See also [273]) Similarly show that sending a measurable space to its measurable subsets yields a functor **Meas** → **MSL**, with the filter functor $\mathcal{S}$ as left adjoint.

6.5.8 This exercise looks at products and coproducts in the category **MSL** of meet semilattices.

1. Check that **MSL** has products $\prod_{i \in I} D_i$ as in **Sets**, with componentwise order.
2. Show that there is an inclusion functor **MSL** → **CMon** of meet semilattices in commutative monoids and that the category **MSL** has finite biproducts ⊕ – just like **CMon** has; see Exercise 2.1.6.
3. For arbitrary, set-indexed products, show that the following construction works:

$$\coprod_{i \in I} D_i \;=\; \{\varphi \colon I \to \bigcup_i D_i \mid \forall i.\, \varphi(i) \in D_i \text{ and } \mathrm{supp}(\varphi) \text{ is finite}\},$$

where $\mathrm{supp}(\varphi) = \{i \mid \varphi(i) \neq \top\}$. Top and meet are defined pointwise.

4. For a set A, describe the copower $A \cdot D = \coprod_{a \in A} D$ and power $D^A = \prod_{a \in A} D$ for $D \in$ **MSL** explicitly.

6.5.9 Prove that the various functors $L \colon$ **Sets** → **Sets** introduced in Definition 6.5.3 can in fact be understood as functors $L \colon$ **MSL** → **MSL** and that the associated maps $\delta_X \colon L\mathcal{P}(X) \to \mathcal{P}(F(X))$ are maps in **MSL**, i.e. preserve finite conjunctions – except for the neighbourhood functor $\mathcal{N}$.

6.5.10 In the bijective correspondence (6.16), one defines $\overline{\delta}$ as

$$\overline{\delta} \stackrel{\text{def}}{=} \Big(F\mathcal{S} \xRightarrow{\varepsilon F\mathcal{S}} \mathcal{S}\mathrm{Pred}F\mathcal{S} \xRightarrow{\mathcal{S}\delta\mathcal{S}} \mathcal{S}L\mathrm{Pred}\mathcal{S} \xRightarrow{\mathcal{S}L\eta} \mathcal{S}L\Big).$$

Define the correspondence also in the other direction, and prove that these constructions are each other's, inverses.

6.5.11 Check that the map $\overline{\delta} \colon F\mathcal{S} \Rightarrow L\mathcal{S}$ is injective for $F_\#, L_\#, \delta_\#$ from Exercise 6.5.5, and $\mathcal{S} \colon$ **MSL** → **Sets**$^{\mathrm{op}}$ as in Exercise 6.5.7.1.

6.6 Algebras and Terms

At this stage we take a step back and look at the traditional way to handle logical assertions. Such assertions are predicates (or, more generally, relations) on carriers describing restrictions for algebraic (or coalgebraic) operations. This section starts with algebras and first reviews some basic constructions and definitions involving terms. The next section will look at assertions in an algebraic context. Subsequent sections will deal with the coalgebraic situation.

Traditionally in universal algebra, the material at hand is presented using terms and equations between them. Here we quickly move to a more abstract level and use (free) monads, culminating in Theorem 6.6.3, as the main result of this section. But we make a gentle start by first describing free algebras for arity functors $F_\#$. The elements of such a free algebra can be described as terms, built up inductively from variables and operations. This yields an explicit description of the free monad $F_\#^*$ on the functor $F_\#$; see Proposition 5.1.8. The term 'description' will be used to introduce equational logic for algebras.

Let $\#\colon I \to \mathbb{N}$ be an arity, as introduced in Definition 2.2.2, with associated endofunctor $F_\#(X) = \coprod_{i\in I} X^{\#i}$ on **Sets**. For each $i \in I$ we choose a function symbol, say f_i, and consider it with arity $\#i$. This yields a collection $(\mathsf{f}_i)_{i\in I}$. If V is a set 'of variables', we can form terms in the familiar way: we define the set $\mathcal{T}_\#(V)$ to be the least set satisfying the following two requirements:

- $V \subseteq \mathcal{T}_\#(V)$
- For each $i \in I$, if $t_1, \ldots t_{\#i} \in \mathcal{T}_\#(V)$, then $\mathsf{f}_i(t_1, \ldots, t_{\#i}) \in \mathcal{T}_\#(V)$.

The first requirement yields a map $V \to \mathcal{T}_\#(V)$. The second requirement provides the set of terms with an algebra structure $F_\#(\mathcal{T}_\#(V)) \to \mathcal{T}_\#(V)$ of the functor $F_\#$ associated with the arity, as in (2.18). Together these two maps yield a (cotuple) algebra structure

$$\begin{array}{rcl} V + F_\#(\mathcal{T}_\#(V)) & \longrightarrow & \mathcal{T}_\#(V) \\ v & \longmapsto & v \\ \langle i, (t_1, \ldots, t_{\#i})\rangle & \longmapsto & \mathsf{f}_i(t_1, \ldots, t_{\#i}). \end{array}$$

Thus we have an algebra of the functor $V + F_\#(-)$. It turns out that terms $\mathcal{T}_\#(V)$ form the initial algebra, and thus yield the free monad on the functor $F_\#$, following the characterisation of such monads in Proposition 5.1.8.

Proposition 6.6.1 *The set of terms $\mathcal{T}_\#(V)$ built from an arity $\#\colon I \to \mathbb{N}$ and a set of variables V is the free $F_\#$-algebra on the set V. The induced monad is the free monad $F_\#^*$ on $F_\#$ with* $\mathbf{Alg}(F_\#) \cong \mathcal{EM}(F_\#^*)$ *by Proposition 5.4.7, as summarised in*

$$\begin{tikzcd} \mathbf{Alg}(F_\#) \cong \mathcal{EM}(F_\#^*) \arrow[d, shift left=2, "U"] \arrow[d, phantom, "\dashv"] \\ \mathbf{Sets} \arrow[u, bend left=50, "\mathcal{T}_\#"] \arrow[loop right, "F_\#^* = U\mathcal{T}_\#"] \end{tikzcd}$$

In essence this adjunction captures inductively defined compositional semantics, written as interpretation function $[\![-]\!]$: given an arbitrary algebra

$F_\#(X) \to X$, for each 'valuation' function $\rho\colon V \to X$ that sends variables to elements of the algebra's carrier X, there is a unique homomorphism of algebras $[\![-]\!]_\rho\colon \mathcal{T}_\#(V) \to X$ with $[\![-]\!]_\rho \circ \eta_V = \rho$, where the unit $\eta_V\colon V \to \mathcal{T}_\#(V)$ is obtained from the first bullet above.

Proof For an algebra $F_\#(X) \to X$, the interpretation map $[\![-]\!]_\rho\colon \mathcal{T}_\#(V) \to X$ extends the valuation function $\rho\colon V \to X$ from variables to terms, via the (inductive) definition

$$\begin{aligned} [\![v]\!]_\rho &= \rho(v), \qquad \text{for } v \in V \\ [\![\mathsf{f}_i(t_1, \ldots, t_{\#i})]\!]_\rho &= f_i([\![t_1]\!]_\rho, \ldots, [\![t_{\#i}]\!]_\rho), \end{aligned}$$

where the function $f_i\colon X^{\#i} \to X$ is the ith component of the algebra structure $F_\#(X) = \coprod_{j\in I} X^{\#j} \to X$. By construction $[\![-]\!]_\rho$ is a homomorphism of algebras $\mathcal{T}_\#(V) \to X$ such that $V \hookrightarrow \mathcal{T}_\#(V) \to X$ is ρ. Uniqueness is trivial. We have an adjunction because we have established a bijective correspondence:

$$\frac{V \xrightarrow{\quad\rho\quad} X}{\left(\begin{array}{c} F_\#(\mathcal{T}_\#(V)) \\ \downarrow \\ \mathcal{T}_\#(V) \end{array}\right) \xrightarrow[{[\![-]\!]_\rho}]{\qquad\qquad} \left(\begin{array}{c} F_\#(X) \\ \downarrow \\ X \end{array}\right)}\,.$$

❑

From the uniqueness of the interpretation maps $[\![-]\!]$ we can easily derive the following properties:

$$\begin{cases} [\![t]\!]_\eta = t \\ h([\![t]\!]_\rho) = [\![t]\!]_{h\circ\rho}, \end{cases} \quad \text{i.e.} \quad \begin{cases} [\![-]\!]_\eta = \mathrm{id} \\ h \circ [\![-]\!]_\rho = [\![-]\!]_{h\circ\rho}, \end{cases} \tag{6.18}$$

where η is the inclusion $V \hookrightarrow \mathcal{T}_\#(V)$ and h is a homomorphism of algebras. (Implicitly, these properties already played a role in Exercise 2.5.17.)

From now on we shall use the free monad notation $F_\#^*$ instead of the terms notation $\mathcal{T}_\#$. More generally, for an arbitrary functor $F\colon \mathbf{Sets} \to \mathbf{Sets}$, we understand $F^*(V)$ as the algebra of terms with operations given by the functor F. These terms can be organised in a category, via a basic categorical construction described in the previous chapter, namely the Kleisli category $\mathcal{K}\ell(-)$. Since terms contain only finitely many variables we restrict the objects to finite sets $n = \{0, 1, \ldots, n-1\}$. We shall write $\mathcal{K}\ell_{\mathbb{N}}(-)$ when such restrictions are applied. One can understand the number/set n as a context with n variables $v_0, \ldots, v_{n-1}$.

Definition 6.6.2 For a monad T on **Sets**, we write $\mathcal{K}\ell_{\mathbb{N}}(T) \hookrightarrow \mathcal{K}\ell(T)$ for the full subcategory with $n \in \mathbb{N}$ as objects. It will be called the **finitary** Kleisli category of T. We write $\mathbf{Law}(T) = \mathcal{K}\ell_{\mathbb{N}}(T)^{\mathrm{op}}$ for the **Lawvere theory** associated with the monad T.

A **model** of this monad is a finite product preserving functor $\mathbf{Law}(T) \to \mathbf{Sets}$. We write $\mathbf{Model}(T) = [\mathbf{Law}(T), \mathbf{Sets}]_{\mathrm{fp}}$ for the category of finite product preserving functors $\mathbf{Law}(T) \to \mathbf{Sets}$, and natural transformations between them.

Lawvere theories were been introduced in [329] (see also [304, 234]) as categories with natural numbers $n \in \mathbb{N}$ as objects and finite products given by $(0, +)$. The main examples are opposites of finitary Kleisli categories, as above. Since these theories involve both the opposite $(-)^{\mathrm{op}}$ and the Kleisli construction $\mathcal{K}\ell_{\mathbb{N}}(-)$ it requires some unravelling to get a handle on their morphisms. Below we argue step by step that, nevertheless, these categories $\mathbf{Law}(T)$ form a natural way of representing terms.

- The category $\mathcal{K}\ell_{\mathbb{N}}(T)$ has coproducts, inherited from the underlying category **Sets**; see Proposition 5.2.2.3. They can simply be described as sums of natural numbers: $n + m \in \mathcal{K}\ell_{\mathbb{N}}(T)$ is the coproduct of objects $n, m \in \mathcal{K}\ell_{\mathbb{N}}(T)$, and $0 \in \mathcal{K}\ell_{\mathbb{N}}(T)$ is the initial object. As a result $(0, +)$ yield products in the opposite category $\mathcal{K}\ell_{\mathbb{N}}(T)^{\mathrm{op}} = \mathbf{Law}(T)$.
- Using that $n = 1 + \cdots + 1$ there are bijective correspondences:

$$\dfrac{\begin{array}{cl} \dfrac{\dfrac{m \longrightarrow n}{n \longrightarrow m}}{n \longrightarrow T(m)} & \begin{array}{l} \text{in } \mathbf{Law}(T) \\ \text{in } \mathcal{K}\ell_{\mathbb{N}}(T) \\ \text{in } \mathbf{Sets} \end{array} \end{array}}{n \text{ `terms' } t_1, \ldots, t_n \in T(m) \text{ with } m \text{ `free variables'}}$$

  Since free variables form inputs and are usually positioned on the left (of a turnstile $\vdash$ or of an arrow), the direction of arrows in the category $\mathbf{Law}(T)$ is the most natural one for organising terms.

  Speaking of 'terms' and 'free variables' is justified for the free monad $F_{\#}^{*}$ on an arity functor $F_{\#}$; see Proposition 6.6.1. Here we stretch the terminology and use it for an arbitrary monad T on **Sets**.
- The coprojections $\kappa_i \colon 1 \to m$ in **Sets** yield coprojections $\eta \circ \kappa_i \colon 1 \to m$ in $\mathcal{K}\ell_{\mathbb{N}}(T)$; see point (3) in the proof of Proposition 5.2.2. In the category $\mathbf{Law}(T)$ this map $m \to 1$ is a projection, which may be understood as the ith variable v_i which is projected out of a context of m variables $v_1, \ldots, v_m$.
- Kleisli composition corresponds to substitution in terms. This can best be illustrated for a free monad $F_{\#}^{*}$ on an arity functor $F_{\#}$. Assume we have composable maps in the Lawvere theory $\mathbf{Law}(F_{\#}^{*})$:

$$k \xrightarrow{\;s = \langle s_1, \ldots, s_m \rangle\;} m \qquad \text{and} \qquad m \xrightarrow{\;t = \langle t_1, \ldots, t_n \rangle\;} n.$$

That is, we have maps of cotuples:

$$n \xrightarrow{t = [t_1, \ldots, t_n]} F_\#^*(m) \qquad \text{and} \qquad m \xrightarrow{s = [s_1, \ldots, s_m]} F_\#^*(k).$$

The Kleisli composition $s \circledcirc t \colon n \to F_\#^*(k)$ is given by the n-cotuple of maps

$$t_i[s_1/v_1, \ldots, s_m/v_m] \in F_\#^*(k),$$

where we write $v_1, \ldots, v_m$ for the m variables in the terms $t_i \in F_\#^*(m)$. Thus in the Lawvere theory $\mathbf{Law}(F_\#^*)$ we have as composite

$$k \xrightarrow{t \circ s = \langle t_1[\vec{s}/\vec{v}], \ldots, t_n[\vec{s}/\vec{v}]\rangle} n.$$

These terms $t_i[\vec{s}/\vec{v}]$ are the result of substituting s_j for all occurrences of v_j in t_i (if any). They are defined by induction on the structure

$$\begin{aligned} w[\vec{s}/\vec{v}] &= \begin{cases} s_i & \text{if } w = v_i \\ w & \text{otherwise} \end{cases} \\ \mathsf{f}(r_1, \ldots, r_m)[\vec{s}/\vec{v}] &= \mathsf{f}(r_1[\vec{s}/\vec{v}], \ldots, r_m[\vec{s}/\vec{v}]), \end{aligned} \tag{6.19}$$

where f is a function symbol with arity $m = \#\mathsf{f}$.

- Weakening involves moving a term to a bigger context with additional variables (which don't occur in the term). In Kleisli categories this happens via post-composition with a coprojection $\kappa_1 \colon m \to m + k$, as in

$$n \xrightarrow{t} T(m) \xrightarrow{T(\kappa_1)} T(m + k).$$

That is, weakening in $\mathbf{Law}(T) = \mathcal{K}\ell_{\mathbb{N}}(T)^{\mathrm{op}}$ is described as

$$m + k \xrightarrow{\pi_1} m \xrightarrow{t} n.$$

Similarly, contraction involves replacing multiple occurrences v, v' of variables by a single variable via substitution $[w/v, w/v']$. In Kleisli categories this is done via post composition with a codiagonal $\nabla = [\mathrm{id}, \mathrm{id}]$, and in the associated Lawvere theory via a diagonal $\Delta = \langle \mathrm{id}, \mathrm{id}\rangle$.

- As mentioned, categories with natural numbers as objects and finite products given by sums $(0, +)$ are called **Lawvere theories**; see [329, 304, 386, 233, 234]. A (set-theoretic) model of such a theory is a finite product preserving functor to **Sets**. Here we consider only such models in **Sets**, but the definition of model of a Lawvere theory easily generalises to arbitrary categories with finite products. Understanding theories as categories and models as structure-preserving functors is the essence of Lawvere's functorial semantics.

(Sometimes people use an ‘opposite’ description of Lawvere theories, as categories with natural numbers as objects and finite coproducts (see [111]); in that case the finitary Kleisli categories $\mathcal{K}\ell_{\mathbb{N}}(T)$ are prime examples.)

We have described algebras of a functor or monad as models of certain operations. This model-theoretic aspect is made explicit in the next two (standard) results, connecting algebras and functorial semantics. For more general, enriched, versions, see [393]. The proof of the theorem below is quite long, even if we leave many details to the reader. A non-finitary analogue of this result is described in Exercise 6.6.5.

Theorem 6.6.3 *For a monad T :* **Sets** $\to$ **Sets** *there is a faithful functor from Eilenberg–Moore algebras to models:*

$$\begin{array}{rcl} \mathcal{EM}(T) & \xrightarrow{\quad\mathcal{L}\quad} & \mathbf{Model}(T) = \ [\mathbf{Law}(T), \mathbf{Sets}]_{\mathrm{fp}} \\ (T(X) \stackrel{\alpha}{\to} X) & \longmapsto & (n \longmapsto X^n). \end{array}$$

This $\mathcal{L}$ is an equivalence if T is finitary functor.

Each category of Eilenberg–Moore algebras (over **Sets**) can thus be embedded in a category of presheaves.

Proof On objects, the functor $\mathcal{L}\colon \mathcal{EM}(T) \to [\mathbf{Law}(T), \mathbf{Sets}]_{\mathrm{fp}}$ is described as follows. Given an Eilenberg–Moore algebra $\alpha\colon T(X) \to X$, we obtain a functor

$$\begin{array}{rcl} \mathbf{Law}(T) & \xrightarrow{\qquad\mathcal{L}(X,\alpha)\qquad} & \mathbf{Sets} \\ n & \longmapsto & X^n \\ (n \xrightarrow{\langle t_1,\ldots,t_m\rangle} m) & \longmapsto & (X^n \xrightarrow{\langle [\![t_1]\!],\ldots,[\![t_m]\!]\rangle} X^m). \end{array}$$

The interpretation $[\![t]\!]\colon X^n \to X$ of a term $t \in T(n)$ is obtained on $h \in X^n$ as

$$[\![t]\!](h) = \alpha(T(h)(t)) = \alpha \circ T(h) \circ t\colon\ 1 \longrightarrow T(n) \longrightarrow T(X) \longrightarrow X.$$

The identity $n \to n$ in $\mathbf{Law}(T)$ is given by the unit $\eta\colon n \to T(n)$ in $\mathcal{K}\ell_{\mathbb{N}}(T)$, consisting of n terms $\eta(i) \in T(n)$, for $i \in n$. They are interpreted as projections π_i since

$$[\![\eta(i)]\!](h) = \alpha \circ T(h) \circ \eta \circ \kappa_i = \alpha \circ \eta \circ h \circ \kappa_i = h(i) = \pi_i(h).$$

Hence $\mathcal{L}(X, \alpha)$ preserves identities. Similarly, this functor preserves composition in $\mathbf{Law}(T)$. Clearly, the functor $\mathcal{L}(X, \alpha)$ preserves products; i.e. it sends products $(0, +)$ in $\mathbf{Law}(T)$ to products in **Sets**: $\mathcal{L}(X, a)(0) = X^0 \cong 1$ and

$$\mathcal{L}(X,a)(n+m) \;=\; X^{n+m} \;\cong\; X^n \times X^m \;=\; \mathcal{L}(X,a)(n) \times \mathcal{L}(X,a)(m).$$

For a map of algebras $f\colon (T(X) \xrightarrow{\alpha} X) \to (T(Y) \xrightarrow{\beta} Y)$ one obtains a natural transformation $\mathcal{L}(f)\colon \mathcal{L}(X,\alpha) \Rightarrow \mathcal{L}(Y,\beta)$ with components

$$\mathcal{L}(X,a)(n) = X^n \xrightarrow{\quad \mathcal{L}(f)_n \,=\, f^n \quad} Y^n = \mathcal{L}(Y,b).$$

This is natural in n: for a map $t = \langle t_1, \ldots, t_m\rangle\colon n \to m$ in $\mathbf{Law}(T)$ one easily checks that there is a commuting diagram:

$$\begin{tikzcd}
X^n \arrow[r, "f^n"] \arrow[d, "{\langle [\![t_1]\!]^{\alpha}, \ldots, [\![t_m]\!]^{\alpha}\rangle}"'] & Y^m \arrow[d, "{\langle [\![t_1]\!]^{\beta}, \ldots, [\![t_m]\!]^{\beta}\rangle}"] \\
X^m \arrow[r, "f^m"'] & Y^m
\end{tikzcd}.$$

Obviously, the mapping $f \mapsto \mathcal{L}(f) = (f^n)_n$ is injective, making $\mathcal{L}$ a faithful functor.

We now assume that T is finitary. Our first goal is to show that the functor $\mathcal{L}\colon \mathcal{EM}(T) \to [\mathbf{Law}(T), \mathbf{Sets}]_{\mathrm{fp}}$ is full: if we have a natural transformation $\sigma\colon \mathcal{L}(X,\alpha) \to \mathcal{L}(Y,\beta)$, then the component at the object 1 yields a map between the two carriers:

$$X = X^1 = \mathcal{L}(X,\alpha)(1) \xrightarrow{\quad \sigma_1 \quad} \mathcal{L}(Y,\beta)(1) = Y^1 = Y.$$

We show in a moment that σ_1 is an algebra map. But first we check that $\mathcal{L}(\sigma_1) = \sigma$. For each $n \in \mathbb{N}$ and $i \in n$ we have a term $\eta(i) \in T(n)$, forming a map $n \to 1$ in $\mathbf{Law}(T)$, with interpretation $[\![\eta(i)]\!] = \pi_i$. Therefore, the following naturality square commutes:

$$\begin{tikzcd}
X^n = \mathcal{L}(X,\alpha)(n) \arrow[r, "\sigma_n"] \arrow[d, "{\pi_i = [\![\eta(i)]\!]^{\alpha}}"'] & \mathcal{L}(Y,\beta)(n) = Y^n \arrow[d, "{[\![\eta(i)]\!]^{\beta} = \pi_i}"] \\
X = \mathcal{L}(X,\alpha)(1) \arrow[r, "\sigma_1"'] & \mathcal{L}(Y,\beta)(1) = Y
\end{tikzcd}$$

Hence $\sigma_n = \sigma_1^n = \mathcal{L}(\sigma_1)_n$.

We can now check that σ_1 is an algebra map. For an element $u \in T(X)$ we need to prove $\beta(T(\sigma_1)(u)) = \sigma_1(\alpha(u))$. Since T is finitary, we may assume $u = T(h)(t)$, for some $n \in \mathbb{N}$, $h\colon n \hookrightarrow X$ and $t \in T(n)$. Hence

$$\begin{aligned}
\beta(T(\sigma_1)(u)) &= \beta(T(\sigma_1)(T(h)(t))) \\
&= \beta(T(\sigma_1 \circ h)(t))) \\
&= [\![t]\!]^{\beta}(\sigma_1 \circ h) \\
&= [\![t]\!]^{\beta}((\sigma_1)^n(h)) \\
&= [\![t]\!]^{\beta}(\sigma_n(h)) && \text{as just shown} \\
&= \sigma_1([\![t]\!]^{\alpha}(h)) && \text{by naturality of } \sigma \\
&= \sigma_1(\alpha(T(h)(t))) \\
&= \sigma_1(\alpha(u)).
\end{aligned}$$

In order to show that $\mathcal{L}\colon \mathcal{EM}(T) \to [\mathbf{Law}(T), \mathbf{Sets}]_{\mathrm{fp}}$ is an equivalence, it suffices to show that it is 'essentially surjective' (see e.g. [46, prop. 7.25]): for each $M \in [\mathbf{Law}(T), \mathbf{Sets}]_{\mathrm{fp}}$ we need to find an Eilenberg–Moore algebra $\alpha\colon F(X) \to X$ such that $\mathcal{L}(X, \alpha) \cong M$.

Given a finite product preserving $M\colon \mathbf{Law}(T) \to \mathbf{Sets}$, we take $X = M(1) \in \mathbf{Sets}$ as carrier. On this carrier X an algebra structure $\alpha_M\colon T(X) \to X$ can be defined by using (again) that T is finitary. For $u \in T(X)$ there is an $n \in \mathbb{N}$ and $h\colon n \hookrightarrow X$ with $t \in T(n)$ such that $T(h)(t) = u$. This t forms a map $t\colon n \to 1$ in $\mathbf{Law}(T)$. By applying the functor M we get in $\mathbf{Sets}$

$$X^n = M(1)^n \xrightarrow[\cong]{\varphi_n^{-1}} M(n) \xrightarrow{\quad M(t) \quad} M(1) = X,$$

where the product-preservation isomorphism $\varphi_n\colon M(n) \to M(1)^n$ can be described explicitly as $\varphi_n(y)(i) = M(n \xrightarrow{\pi_i} 1)(y)$.

We define an algebra structure $\alpha_M\colon T(X) \to X$ as $\alpha_M(u) = M(t)(\varphi_n^{-1}(h))$. This outcome does not depend on the choice of n, h, t, as long as $T(h)(t) = u$.

We check that $\alpha_M \circ \eta = \mathrm{id}$. Fix $x \in X$ and consider $\eta(x) \in T(X)$. We can take $t = \eta(*) \in T(1)$ and $h = x\colon 1 \to X$ satisfying

$$T(h)(t) = T(h)(\eta(*)) = \eta(h(*)) = \eta(x).$$

Hence we get our required equality:

$$\alpha_M(\eta(x)) = \alpha_M(T(h)(t)) = M(t)(\varphi_1^{-1}(h)) = M(\eta(*))(h) = M(\mathrm{id})(x) = x.$$

Similarly one proves $\alpha_M \circ \mu = \alpha_M \circ T(\alpha_M)$.

Finally, in order to show that $\mathcal{L}(X, \alpha_M) \cong M$, we have on objects $n \in \mathbb{N}$:

$$\mathcal{L}(X, \alpha_M)(n) = X^n = M(1)^n \cong M(n).$$

On morphisms one has $[\![t]\!]^{\alpha_M} = M(t) \circ \varphi_n^{-1}\colon X^n \to X$, for $t \in F^*(n)$. ❑

The previous result is stated for monads but can be adapted to endofunctors F, via the associated free monad F^*. In order to do so we use the following result.

Lemma 6.6.4 *Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor for which the free monad F^* on F exists. If F is finitary, then so is F^*.*

Proof We use initiality of the algebra $\alpha_X\colon X + F(F^*(X)) \xrightarrow{\cong} F^*(X)$, defining the free monad F^*; see Proposition 5.4.7. Fix a set X and define the subset/predicate $i\colon P \hookrightarrow F^*(X)$ as

$$P = \{u \in F^*(X) \mid \exists(n,h,t).\, n \in \mathbb{N}, h \in X^n, t \in F^*(n) \text{ with } F^*(h)(t) = u\}.$$

Our aim is to define an algebra structure $b\colon X+F(P) \to P$ making the inclusion $i\colon P \hookrightarrow F^*(X)$ a map of algebras $b \to \alpha_X$. Then by initiality we also get a map of algebras $\mathrm{int}_b\colon F^*(X) \to W$ with $i \circ \mathrm{int}_b = \mathrm{id}$. This yields $P = F^*(X)$ and makes F^* finitary.

We define the required algebra $b = [b_1, b_2]\colon X + F(P) \to P$ in two steps.

- For $x \in X$ we define $b_1(x) = \eta_X(x) \in F^*(X)$, where $\eta_X = \alpha_X \circ \kappa_1\colon X \to F^*(X)$ is the unit of the monad F^*. This $b_1(x)$ is in the subset P via the triple $(1, x, \eta_1(*))$, where $x\colon 1 \to X$, since by naturality of η:

$$F^*(x)(\eta(*)) = (\eta \circ x)(*) = \eta(x).$$

  Moreover, by construction, $i(b_1(x)) = \eta(x) = \alpha(\kappa_1 x)$.
- For an element $v \in F(P)$ we use that the functor F is finitary to get $m \in \mathbb{N}$ with $g\colon m \to P$ and $s \in F(m)$ such that $F(g)(s) = v$. For each $i \in m$ we pick a triple (n_i, h_i, t_i) with $F^*(h_i)(t_i) = g(i) \in F^*(X)$. Next we take $n = n_1 + \cdots + n_m$ and $h = [h_1, \ldots, h_m]\colon n \to X$ and $t = [F^*(\kappa_1) \circ t_1, \ldots F^*(\kappa_m) \circ t_m]\colon m \to F^*(n)$, where $\kappa_i\colon n_i \to n$ is the appropriate insertion/coprojection map. We use the universal map $\theta\colon F \Rightarrow F^*$ from the proof of Proposition 5.1.8 to get $\theta(s) \in F^*(m)$ and then $r = \alpha \circ \kappa_2 \circ F(t) \circ s\colon 1 \to F^*(n)$. This yields a new element $b_2(v) = \alpha(\kappa_2 v) \in F^*(X)$, which is in P via the triple (n, h, r), since

$$\begin{aligned} F^*(h) \circ r &= F^*(h) \circ \alpha \circ \kappa_2 \circ F(t) \circ s \\ &= \alpha \circ \kappa_2 \circ F(F^*(h)) \circ F(t) \circ s \\ &= \alpha \circ \kappa_2 \circ F([F^*(h_1) \circ t_1, \ldots, F^*(h_n) \circ t_n]) \circ s \\ &= \alpha \circ \kappa_2 \circ F(g) \circ s \\ &= b_2(v). \end{aligned}$$

❑

The following is now an easy consequence of Theorem 6.6.3.

Corollary 6.6.5 *Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor with free monad F^*. Then there is a faithful functor from functor algebras to models:*

$$\mathbf{Alg}(F) \xrightarrow{\quad \mathcal{L} \quad} \mathbf{Model}(F^*) = [\mathbf{Law}(F^*), \mathbf{Sets}]_{\mathrm{fp}}.$$

This $\mathcal{L}$ is an equivalence if the functor F is finitary.

Proof Proposition 5.4.7 describes an isomorphism of categories $\mathbf{Alg}(F) \cong \mathcal{EM}(F^*)$, which immediately gives the first part of the result. For the second part we use the previous lemma. ❑

With these results in place we are ready to consider assertions relating terms in the next section.

Exercises

6.6.1 Show that the action of the functor $F^*_\# = \mathcal{T}_\#$ from Proposition 6.6.1 on functions f can be described by simultaneous substitution:

$$F_\#(f)(t) = t[f(x_1)/x_1, \ldots, f(x_n)/x_n],$$

if $x_1, \ldots, x_n$ are the free variables in the term t.

6.6.2 Conclude from the fact that each term $t \in F^*_\#(V)$ contains only finitely many (free) variables that the functor/monad $F^*_\#$ is finitary.

6.6.3 Consider the functor $\mathcal{L}\colon \mathbf{Alg}(F) \to [\mathbf{Law}(F^*), \mathbf{Sets}]_{\mathrm{fp}}$ from Corollary 6.6.5. Let $a\colon F(X) \to X$ be an algebra, with associated functor $\mathcal{L}(X, a)\colon \mathbf{Law}(F^*) \to \mathbf{Sets}$. Show that the interpretation $[\![t]\!]\colon X^m \to X$ of a term $t \in F^*(m)$, considered as a map $m \to 1$ in the Lawvere theory $\mathbf{Law}(F^*)$, is obtained by initiality in

$$\begin{tikzcd}
m + F(F^*(m)) \arrow[r, dashed] \arrow[d, "\alpha_m"', "\cong"] & m + F(X^{X^m}) \arrow[d, "a_m"] \\
F^*(m) \arrow[r, dashed, "{[\![-]\!]}"'] & X^{X^m}
\end{tikzcd}$$

where the algebra $a_m = [a_{m,1}, a_{m,2}]\colon m + F(X^{X^m}) \to X^{X^m}$ on the right-hand side is given by

$$\begin{cases} a_{m,1} = \lambda i \in m.\, \lambda h \in X^m.\, h(i) : m \longrightarrow X^{X^m} \\ a_{m,2} = \Big(F(X^{X^m}) \xrightarrow{r} F(X)^{X^m} \xrightarrow{a^{X^m}} X^{X^m}\Big). \end{cases}$$

The r map is the 'exponent version' of strength from Exercise 5.2.16.

6.6.4 The aim of this exercise is to elaborate a concrete instance of Corollary 6.6.5, stating the correspondence between algebras and models. We chose a simple (arity) functor $F(X) = A + X$, for a fixed set A.

1. Check that the free monad F^* on F is of the form $F^*(V) = \mathbb{N} \times (V + A)$. Describe the unit $\eta\colon V \to F^*(V)$, multiplication $\mu\colon F^*F^*(V) \to F^*(V)$ and universal map $\theta\colon F(V) \Rightarrow F^*(V)$ explicitly.

2. Describe morphisms $t : n \to m$ in the categories $\mathcal{K}\ell_{\mathbb{N}}(F^*)$ and $\mathbf{Law}(F^*)$ explicitly. Give concrete descriptions of identity maps and of composition.
3. Describe the two functors $\mathbf{Alg}(F) \leftrightarrows [\mathbf{Law}(F^*), \mathbf{Sets}]_{\mathrm{fp}}$ of the equivalence of Corollary 6.6.5 concretely.

6.6.5 Let $\mathbb{C}$ a category with arbitrary products.

1. Fix an object $A \in \mathbb{C}$. Write $C_A : \mathbf{Sets} \to \mathbf{Sets}$ for the functor given by

$$C_A(X) = \mathbb{C}(A^X, A).$$

Show that C_A is a monad, generalising the continuation monad from Example 5.1.3.7.

2. Let T be a monad on **Sets**. Prove that there is a bijective correspondence (up-to-isomorphism) between

$$\frac{\mathcal{K}\ell(T)^{\mathrm{op}} \longrightarrow \mathbb{C} \;\; \text{preserving arbitrary products}}{\overline{A \in \mathbb{C} \;\; \text{with a map of monads} \;\; T \Longrightarrow C_A}}.$$

This gives an infinitary version of Theorem 6.6.3, after [221].

6.6.6 Show that each term $t \in F^*(V)$ gives rise to a functor in a commuting triangle:

$$\begin{tikzcd}
\mathbf{Alg}(F) \arrow[rr, "\llbracket t \rrbracket"] \arrow[dr] & & \mathbf{Alg}((-)^V) \arrow[dl] \\
& \mathbf{Sets} &
\end{tikzcd}$$

This view on terms is elaborated in [135].

6.7 Algebras and Assertions

This section covers logical assertions in an algebraic context, at first in the form of equations between terms. This material forms the basic theory of (untyped, single-sorted) algebraic specifications and may be found in many places in the literature, such as [463, 126, 465, 398, 340]. Our presentation follows the monad-based approach from the previous section, so that the similarity/duality with the coalgebraic situation in subsequent sections becomes clear. The main result of this section, Theorem 6.7.11, shows how logical assertions give rise to a quotient monad, whose Eilenberg–Moore algebras are models of the assertions. These are standard results in the theory of monads. What is new here is the systematic presentation in terms of relation lifting and quotients (of equivalence relations and of congruence equivalences).

In the end, the most important point is that operations are captured by algebras of *functors* and that operations with assertions require algebras of *monads*. This same distinction applies in the coalgebraic case.

We start with an illustration, giving a formal description of groups. Their arity function # can be seen as a map $\#\colon \{\mathsf{e}, \mathsf{m}, \mathsf{i}\} \to \mathbb{N}$, where

- e is the symbol of the unit element, with arity $\#\mathsf{e} = 0$
- m is used for multiplication, with $\#\mathsf{m} = 2$
- i is the symbol for the inverse operation, whose arity is one: $\#\mathsf{i} = 1$.

An algebra for the arity functor $F_\#(X) = 1 + (X + X) + X$ associated with # consists of a set A with a map $1 + (A \times A) + A \to A$, i.e. with interpretations $1 \to A$, $A \times A \to A$ and $A \to A$ of the three function symbols $\mathsf{e}, \mathsf{m}, \mathsf{i}$.

Until now we have talked only about interpretation of the function symbols and not about validity of the familiar group axioms:

$$\begin{array}{rclcrcl} \mathsf{m}(\mathsf{e}, v) & = & v & \qquad & \mathsf{m}(\mathsf{i}(v), v) & = & \mathsf{e} \\ \mathsf{m}(v, \mathsf{e}) & = & v & & \mathsf{m}(v, \mathsf{i}(v)) & = & \mathsf{e} \\ \multicolumn{7}{c}{\mathsf{m}(v_1, \mathsf{m}(v_2, v_3)) \; = \; \mathsf{m}(\mathsf{m}(v_1, v_2), v_3).} \end{array} \tag{6.20}$$

These equations consist of pairs of terms (t_1, t_2) in the free algebra $F_\#^*(V)$, for a set of variables V.

Such axioms form a relation $Ax_V \subseteq F_\#^*(V) \times F_\#^*(V)$ on the carrier of the free algebra on V, given explicitly as

$$\begin{array}{rcl} Ax_V & = & \{\langle \mathsf{m}(\mathsf{e}, v), v\rangle \mid v \in V\} \cup \{\langle \mathsf{m}(v, \mathsf{e}), v\rangle \mid v \in V\} \\ & & \cup\, \{\langle \mathsf{m}(\mathsf{i}(v), v), \mathsf{e}\rangle \mid v \in V\} \cup \{\langle \mathsf{m}(v, \mathsf{i}(v)), \mathsf{e}\rangle \mid v \in V\} \\ & & \cup\, \{\langle \mathsf{m}(v_1, \mathsf{m}(v_2, v_3)), \mathsf{m}(\mathsf{m}(v_1, v_2), v_3)\rangle \mid v_1, v_2, v_3 \in V\}. \end{array} \tag{6.21}$$

In the computer science literature such a pair $(\#, Ax)$ is usually called an algebraic specification.

Our main focus will be on models of such specifications. But we also wish to use axioms for reasoning and proving results such as uniqueness of inverses:

$$\mathsf{m}(v, w) = \mathsf{e} \implies v = \mathsf{i}(w). \tag{6.22}$$

In order to do so we need derivation rules for equations $t_1 = t_2$. In general, assuming an arity # and a set of axioms $Ax_V \subseteq F_\#^*(V) \times F_\#^*(V)$, as in (6.20), one standardly uses the following logical rules:

$$
\begin{array}{ll}
(a) & \dfrac{Ax}{t_1 = t_2} \quad (\text{ if } (t_1, t_2) \in Ax_V; \text{ but see (6.24) below }) \\[2ex]
(b) & \dfrac{}{t = t} \qquad \dfrac{t_1 = t_2}{t_2 = t_1} \qquad \dfrac{t_1 = t_2 \quad t_2 = t_3}{t_2 = t_3} \\[2ex]
(c) & \dfrac{t_1 = t'_1 \quad \cdots \quad t_m = t'_m}{\mathsf{f}(t_1, \ldots t_m) = \mathsf{f}(t'_1, \ldots t'_m)} \quad (\text{ for a function symbol } \mathsf{f} \text{ of arity } m).
\end{array}
\tag{6.23}
$$

The rules in (b) turn the equality relation = into an equivalence relation. And the rules in (c) make it a

congruence. This can be expressed as: the equality relation = is an algebra of the relation lifting functor $\mathrm{EqRel}(F_\#)\colon \mathrm{EqRel}(\mathbf{Sets}) \to \mathrm{EqRel}(\mathbf{Sets})$, restricted to equivalence relations as in Corollary 4.4.4.

In general an equation $t_1 = t_2$ is said to be derivable from a collection $Ax = (Ax_V)_V$ of relations on terms if there is a derivation tree structured by these rules with $t_1 = t_2$ as conclusion. One then often writes $Ax \vdash t_1 = t_2$. Derivable equations are also called theorems. We write $\mathrm{Th}(Ax) \subseteq F^*_\#(V) \times F^*_\#(V)$ for the relation containing precisely the equations that are derivable from Ax. It is the free congruence equivalence on Ax.

Example 6.7.1 The implication $\mathsf{m}(v, w) = \mathsf{e} \Rightarrow v = \mathsf{i}(w)$ from (6.22) has a formal derivation in the theory of groups: Figure 6.3 shows a derivation tree with the equation $v = \mathsf{i}(w)$ as conclusion, and with $\mathsf{m}(v, w) = \mathsf{e}$ as only assumption. In this tree the term $\mathsf{i}(v)$ is written as $\mathsf{i}v$ in order to save parentheses. The relation *GrAx* refers to the group axioms from (6.20).

Remark 6.7.2 The meticulous reader may have noticed that we have cheated a bit, namely in the two rightmost occurrences of the 'axiom' rule in Figure 6.3. They use instantiations of axioms. For instance the rightmost rule involves an equation $\mathsf{m}(\mathsf{e}, \mathsf{i}w) = \mathsf{i}w$. Strictly speaking this is not an axiom but a *substitution instance* $\mathsf{m}(\mathsf{e}, v)[\mathsf{i}w/v] = v[\mathsf{i}w/v]$ of the axiom $\mathsf{m}(\mathsf{e}, v) = v$, namely with $\mathsf{i}w$ in place of v. This can be formalised as follows.

The improved 'axiom rule' (1) in (6.23) now reads

$$
(a') \quad \frac{Ax}{t_1[\vec{s}/\vec{v}] = t_2[\vec{s}/\vec{v}]} \quad (\text{for } (t_1, t_2) \in Ax) \tag{6.24}
$$

For convenience we will assume from now on that our sets of axioms are closed under substitutions, so that there is no difference between the rules (a) in (6.23) and (a') in (6.24). Basically this means that the axioms are formulated in a slightly different manner. For groups this would involve replacing the formulation used in (6.21) by

$$
\begin{prooftree}
\AxiomC{$GrAx$}
\UnaryInfC{$\mathsf{m}(v,\mathsf{e}) = v$}
\UnaryInfC{$v = \mathsf{m}(v,\mathsf{e})$}
\AxiomC{}
\UnaryInfC{$v = v$}
\AxiomC{$GrAx$}
\UnaryInfC{$\mathsf{m}(w,\mathsf{i}w) = \mathsf{e}$}
\UnaryInfC{$\mathsf{e} = \mathsf{m}(w,\mathsf{i}w)$}
\BinaryInfC{$\mathsf{m}(v,\mathsf{e}) = \mathsf{m}(v,\mathsf{m}(w,\mathsf{i}w))$}
\BinaryInfC{$v = \mathsf{m}(v,\mathsf{m}(w,\mathsf{i}w))$}
\AxiomC{$GrAx$}
\UnaryInfC{$\mathsf{m}(v,\mathsf{m}(w,\mathsf{i}w)) = \mathsf{m}(\mathsf{m}(v,w),\mathsf{i}w)$}
\AxiomC{$\mathsf{m}(v,w) = \mathsf{e}$}
\AxiomC{}
\UnaryInfC{$\mathsf{i}w = \mathsf{i}w$}
\BinaryInfC{$\mathsf{m}(\mathsf{m}(v,w),\mathsf{i}w) = \mathsf{m}(\mathsf{e},\mathsf{i}w)$}
\AxiomC{$GrAx$}
\UnaryInfC{$\mathsf{m}(\mathsf{e},\mathsf{i}w) = \mathsf{i}w$}
\BinaryInfC{$\mathsf{m}(\mathsf{m}(v,w),\mathsf{i}w) = \mathsf{i}w$}
\BinaryInfC{$\mathsf{m}(v,\mathsf{m}(w,\mathsf{i}w)) = \mathsf{i}w$}
\BinaryInfC{$v = \mathsf{i}w$}
\end{prooftree}
$$

Figure 6.3 Derivation of the implication (6.22) from the group axioms *GrAx*.

$$\begin{aligned} Ax_V &= \{\langle \mathsf{m}(\mathsf{e}, t), t\rangle \mid t \in F^*_\#(V)\} \cup \{\langle \mathsf{m}(t, \mathsf{e}), t\rangle \mid t \in F^*_\#(V)\} \\ &\quad \cup \{\langle \mathsf{m}(\mathsf{i}(t), t), \mathsf{e}\rangle \mid t \in F^*_\#(V)\} \cup \{\langle \mathsf{m}(t, \mathsf{i}(t)), \mathsf{e}\rangle \mid t \in F^*_\#(V)\} \\ &\quad \cup \{\langle \mathsf{m}(t_1, \mathsf{m}(t_2, t_3)), \mathsf{m}(\mathsf{m}(t_1, t_2), t_3)\rangle \mid t_1, t_2, t_3 \in F^*_\#(V)\}. \end{aligned}$$

If we assume that axioms Ax are closed under substitution, then also derivable equations are closed under substitution, in the sense that

$$Ax \vdash t_1 = t_2 \implies Ax \vdash t_1[\vec{s}/\vec{v}] = t_2[\vec{s}/\vec{v}]. \tag{6.25}$$

This is obtained by induction on the length of the derivation.

In more categorical fashion one can formulate axioms as a functor. We shall do so for the finitary Kleisli categories $\mathcal{K}\ell_{\mathbb{N}}(T)$ from Definition 6.6.2, because (terms in) axioms involve only finitely many variables. Closure of axioms under substitution is guaranteed, by construction.

Definition 6.7.3 An **axiom system** for a monad T on **Sets** is a functor $\mathcal{A}$ in a commuting triangle

$$\begin{tikzcd} & \mathbf{EnRel} \arrow[d] \\ \mathcal{K}\ell_{\mathbb{N}}(T) \arrow[ur, "\mathcal{A}"] \arrow[r, "U"'] & \mathbf{Sets} \end{tikzcd} \tag{6.26}$$

where **EnRel** is the category with endorelations $R \rightarrowtail X \times X$ on a single carrier as objects, and where $U : \mathcal{K}\ell_{\mathbb{N}}(T) \to \mathbf{Sets}$ is the standard (right adjoint) functor from Proposition 5.2.2.

An axiom system for an *endofunctor* F is an axiom system for the associated free monad F^* and is thus given by a functor $\mathcal{A} : \mathcal{K}\ell_{\mathbb{N}}(F^*) \to \mathbf{EnRel}$ as above.

In a (functorial) model of the monad T, in the form of a finite product preserving functor

$$\mathbf{Law}(T) = \mathcal{K}\ell_{\mathbb{N}}(T)^{\mathrm{op}} \xrightarrow{\quad M \quad} \mathbf{Sets},$$

the **axioms hold** – or, equivalently, M **satisfies the axioms** – if for each parallel pair of maps

$$\begin{tikzcd} n \arrow[r, shift left, "s = \langle s_1, \ldots, s_m\rangle"] \arrow[r, shift right, "t = \langle t_1, \ldots, t_m\rangle"'] & m \end{tikzcd}$$

in $\mathbf{Law}(T)$, one has

$$[\forall i \in m.\, (s_i, t_i) \in \mathcal{A}(n)] \implies M(s) = M(t).$$

In that case we write $M \models \mathcal{A}$. This determines a full subcategory

$$\mathbf{Model}(T, \mathcal{A}) \hookrightarrow \mathbf{Model}(T)$$

of models in which the axioms $\mathcal{A}$ hold.

Similarly, for a monad T and functor F, we have full subcategories

$$\mathcal{EM}(T, \mathcal{A}) \hookrightarrow \mathcal{EM}(T) \qquad \text{and} \qquad \mathbf{Alg}(F, \mathcal{A}) \hookrightarrow \mathbf{Alg}(F)$$

of monad and functor algebras satisfying $\mathcal{A}$. This means that the axioms $\mathcal{A}$ hold in the corresponding functorial models, obtained via Theorem 6.6.3 and Corollary 6.6.5.

The functorial description of axioms (6.26) can be unravelled as follows. For each $n \in \mathbb{N}$, considered as object $n \in \mathcal{K}\ell_{\mathbb{N}}(T)$ in the finitary Kleisli category, there is a relation $\mathcal{A}(n) \rightarrowtail T(n) \times T(n)$ on $U(n) = T(n) \in \mathbf{Sets}$, containing the pairs of terms with n variables that are equated by the axioms in $\mathcal{A}$. And for each Kleisli map $f : n \to T(m)$ there is a commuting diagram

$$\begin{tikzcd}
\mathcal{A}(n) \arrow[r, dashed] \arrow[d, tail] & \mathcal{A}(m) \arrow[d, tail] \\
T(n) \times T(n) \arrow[r, "f^{\$} \times f^{\$}"'] & T(m) \times T(m)
\end{tikzcd}$$

where $U(f) = f^{\$} = \mu \circ T(f)$ is the Kleisli extension of f; see Proposition 5.2.3. This guarantees that the relations $\mathcal{A}(n)$ are closed under substitution. If this is too abstract, it may be helpful to elaborate on the details of this closure condition for the special case when T is a free monad $F_{\#}^{*}$ on an arity functor.

A slightly different formulation of axiom systems is given in Exercise 6.7.3.

We briefly reformulate validity of axioms in more traditional terms, along the lines of the free construction in Proposition 6.6.1 using terms.

Lemma 6.7.4 *Let # be an arity and $\mathcal{A} : \mathcal{K}\ell_{\mathbb{N}}(F_{\#}^{*}) \to$ **EnRel** an axiom system for the associated arity functor $F_{\#}$. For an algebra $a : F_{\#}(X) \to X$ the following two statements are equivalent.*

- *The axiom system $\mathcal{A}$ holds in the algebra $a : F_{\#}(X) \to X$.*
- *For each $n \in \mathbb{N}$ and for each pair of terms $t, s \in F_{\#}^{*}(n)$ containing at most n variables, if $(t, s) \in \mathcal{A}(n)$, then $[\![t]\!]_{\rho} = [\![s]\!]_{\rho}$ for each valuation function $\rho : n \to X$, with interpretations $[\![-]\!]_{\rho}$ as defined in the proof of Proposition 6.6.1.*

Diagrammatically this means that each valuation yields a map of relations, from axioms to equality:

$$\begin{tikzcd}
\mathcal{A}(n) \arrow[rr, dashed] \arrow[d, tail] & & Eq(X) = X \arrow[d, tail, "\Delta = \langle \mathrm{id}, \mathrm{id} \rangle"] \\
F_{\#}^{*}(n) \times F_{\#}^{*}(n) \arrow[rr, "{[\![-]\!]_{\rho} \times [\![-]\!]_{\rho}}"'] & & X \times X
\end{tikzcd}$$

Proof The functor $\mathcal{L}(X, a) \colon \mathbf{Law}(F_{\#}^{*}) \to \mathbf{Sets}$ associated in Corollary 6.6.5 with the algebra sends the terms $t, s \in F_{\#}(n)$ to functions $[\![t]\!], [\![s]\!] \colon X^n \to X$, as described explicitly in Exercise 6.6.4. Validity of $\mathcal{A}$ in the model $\mathcal{L}(X, a)$ means that $[\![t]\!] = [\![s]\!]$. The resulting mapping $\rho \mapsto [\![t]\!](\rho)$ yields the adjoint transpose in the setting of Proposition 6.6.1, since $[\![\eta(i)]\!](\rho) = \pi_i(\rho) = \rho(i)$. Thus, the (validity) equation $[\![t]\!] = [\![s]\!]$ is equivalent to $[\![t]\!]_{\rho} = [\![t]\!]_{\rho}$ for any valuation $\rho \colon n \to X$. ❑

Example 6.7.5 For the functor $F(X) = 1 + (X \times X) + X$ capturing the group operations and the group axioms *GrAx* described in (6.20) we obtain that the category **Grp** of groups and group homomorphisms can be described as **Grp** = **Alg**$(F, GrAx)$.

We now generalise from free monads $F_{\#}^{*}$ on arity functors to arbitrary monads T (on **Sets**) and consider congruences for such monads. Recall that such congruences correspond to algebras of a relation-lifting functor – defined here with respect to the standard set-theoretic logical factorisation system of injections and surjections. For a monad T we use Eilenberg–Moore algebras of the associated lifting $\mathrm{Rel}(T)$ as T-congruences. Recall that $\mathrm{Rel}(T)$ is a monad by Exercise 4.4.6. Using Eilenberg–Moore algebras properly generalises the situation for endofunctors F because $\mathbf{Alg}(\mathrm{Rel}(F)) \cong \mathcal{EM}(\mathrm{Rel}(F^{*}))$, by Exercise 5.4.18. In the remainder we restrict to liftings to endorelations; more specifically, we consider liftings $\mathrm{EqRel}(T)$ to equivalence relations, as in Corollary 4.4.4.

Lemma 6.7.6 *Let* $T \colon \mathbf{Sets} \to \mathbf{Sets}$ *be a monad, with associated monad*

$$\mathbf{EqRel} \xrightarrow{\mathrm{EqRel}(T)} \mathbf{EqRel}$$

obtained by lifting to equivalence relations. For an axiom system $\mathcal{A} \colon \mathcal{K}\ell_{\mathbb{N}}(T) \to \mathbf{EnRel}$ *we define for each* $n \in \mathbb{N}$

$$\mathrm{Th}(\mathcal{A})(n) \;=\; [\, \textit{the free } \mathrm{EqRel}(T)\textit{-algebra on } \mathcal{A}(n) \,].$$

This yields a new axiom system $\mathrm{Th}(\mathcal{A})$ *of congruence equivalences in*

$$\begin{tikzcd} & & \mathcal{EM}(\mathrm{EqRel}(T)) \arrow[d] \\ & & \mathbf{EqRel} \arrow[d] \arrow[loop right, "\mathrm{EqRel}(T)"] \\ \mathcal{K}\ell_{\mathbb{N}}(T) \arrow[uurr, "\mathrm{Th}(\mathcal{A})"] \arrow[rr, "U"'] & & \mathbf{Sets} \arrow[loop right, "T"] \end{tikzcd}$$

If T is finitary, then for each model $M \in \mathbf{Model}(T) = [\mathbf{Law}(T), \mathbf{Sets}]_{\mathrm{fp}}$ *we have*

$$M \models \mathcal{A} \iff M \models \mathrm{Th}(\mathcal{A}).$$

Proof For each $n \in \mathbb{N}$ there is a monotone function between posets of relations:

$$\mathcal{P}(T(n) \times T(n)) \xrightarrow{\ \overline{\mathcal{A}(n)} \vee \coprod_{\mu\times\mu} \mathrm{EqRel}(T)(-)\ } \mathcal{P}(T(n) \times T(n))$$

where $\overline{\mathcal{A}(n)}$ is the least equivalence relation containing the axioms $\mathcal{A}(n) \subseteq T(n)\times T(n)$. This function has a least fixed point, by the Knaster–Tarski fixpoint theorem (see e.g. [119, chapter 4]). By Proposition 5.1.8 this is the free algebra $\coprod_{\mu\times\mu} \mathrm{EqRel}(T)(\mathrm{Th}(\mathcal{A}(n))) \subseteq \mathrm{Th}(\mathcal{A})(n)$ on $\overline{\mathcal{A}(n)}$. The inclusion expresses that $\mathrm{Th}(\mathcal{A})(n)$ is a $\mathrm{Rel}(T)$-functor algebra in

$$\begin{tikzcd} \mathrm{EqRel}(T)(\mathrm{Th}(\mathcal{A})(n)) \arrow[r, dashed] \arrow[d, tail] & \mathrm{Th}(\mathcal{A})(n) \arrow[d, tail] \\ T^2(n) \times T^2(n) \arrow[r, "\mu\times\mu"'] & T(n) \times T(n) \end{tikzcd} \tag{6.27}$$

By Exercise 6.2.6 this relation $\mathrm{Th}(\mathcal{A})(n)$ is automatically an Eilenberg–Moore algebra and thus a T-congruence.

Suppose $M \models \mathcal{A}$. We wish to prove $M \models \mathrm{Th}(\mathcal{A})$. The other direction is trivial since $\mathcal{A}(n) \subseteq \overline{\mathcal{A}(n)} \subseteq \mathrm{Th}(\mathcal{A})(n)$. For each pair of terms $t, s \in \mathcal{A}(n)$, considered as parallel maps $n \rightrightarrows 1$ in $\mathbf{Law}(T)$, we have $M(s) = M(t)\colon M(n) \to M(1)$. The relation

$$R = \{(s,t) \mid s,t \in T(n) \text{ with } M(s) = M(t)\}$$

thus contains $\mathcal{A}(n)$. This R is clearly an equivalence relation, so also $\overline{\mathcal{A}(n)} \subseteq R$. In order to show that R is also a T-congruence, we need to define an Eilenberg–Moore algebra structure $\beta\colon T(R) \to R$. First we name the inclusion explicitly as $\langle r_1, r_2\rangle\colon R \hookrightarrow T(n) \times T(n)$. Since T is finitary, an element $u \in T(R)$ can be written as $u = T(h)(v)$, for some $m \in \mathbb{N}$, $h\colon m \to R$ and $v \in T(m)$. Write

$h(i) = (s_i, t_i) \in R$. Then $M(s_i) = M(t_i)$. Moreover, these terms yield cotuples $[t_1, \ldots, t_m] = r_1 \circ h$ and $[s_1, \ldots, s_m] = r_2 \circ h$, forming functions $m \to T(n)$ and thus tuples $n \to m$ in $\mathbf{Law}(T)$. Since the model M preserves products, we get $M(r_1 \circ h) = M(r_2 \circ h)$, as functions $M(n) \to M(m) \cong M(1)^m$.

Now we can prove that the pair $\langle \mu(T(r_1)(u)), \mu(T(r_2)(u))\rangle \in T(n) \times T(n)$ is in the relation R:

$$\begin{aligned} M(\mu(T(r_1)(u))) \;=\; M(\mu \circ T(r_1) \circ T(h) \circ v) &= M((r_1 \circ h) \circledcirc v) \\ &= M(v) \circ M(r_1 \circ h) \\ &= M(v) \circ M(r_2 \circ h) \\ &= M((r_2 \circ h) \circledcirc v) \\ &= M(\mu(T(r_2)(u))). \end{aligned}$$

We thus get a unique $\beta(u) \in R$ with $\langle r_1, r_2\rangle(\beta(u)) = \langle \mu(T(r_1)(u)), \mu(T(r_2)(u))\rangle$. It is easy to check that the resulting function $\beta \colon T(R) \to R$ is an Eilenberg–Moore algebra.

Thus we have that R is a congruence containing $\overline{\mathcal{A}(n)}$. Hence $\mathrm{Th}(\mathcal{A})(n) \subseteq R$. This means $M \models \mathrm{Th}(\mathcal{A})$. ❑

The next lemma is a categorical analogue of Exercise 6.7.2, saying that simultaneous substitution of equal terms yields equal results.

Lemma 6.7.7 *Let T be a monad with axiom system $\mathcal{A}$. Then*

$$\begin{tikzcd} & \mathrm{Th}(\mathcal{A})(m) \arrow[d, tail] \\ n \arrow[ur, dashed] \arrow[r, "\langle f, g\rangle"'] & T(m) \times T(m) \end{tikzcd} \qquad \textit{implies} \qquad \begin{tikzcd} & \mathrm{Th}(\mathcal{A})(m) \arrow[d, tail] \\ T(n) \arrow[ur, dashed] \arrow[r, "\langle f^{\$}, g^{\$}\rangle"'] & T(m) \times T(m) \end{tikzcd}$$

Proof We use that $f^{\$} = \mu \circ T(f)$ and that the theory is a congruence. Let $h \colon n \to \mathrm{Th}(\mathcal{A})(m)$ be the dashed map in the above triangle on the left. Then

$$\begin{tikzcd} & T(\mathrm{Th}(\mathcal{A})(m)) \arrow[d, two heads] & \\ & \mathrm{EqRel}(T)(\mathrm{Th}(\mathcal{A})(m)) \arrow[r, dashed] \arrow[d, tail] & \mathrm{Th}(\mathcal{A})(m) \arrow[d, tail] \\ T(n) \arrow[uur, bend left, "T(h)"] \arrow[r, "\langle T(f), T(g)\rangle"'] & T^2(m) \times T^2(m) \arrow[r, "\mu \times \mu"'] & T(m) \times T(m) \end{tikzcd}$$

The dashed arrow in this diagram comes from (6.27). ❑

We continue with quotients in **Sets** and first collect some basic results. We use the description of quotients as left adjoint to equality from Definition 4.5.7. Such quotients exist for **Sets** by Exercise 4.5.5.

Lemma 6.7.8 *Consider the quotient functor Q, as left adjoint to equality in*

$$\begin{tikzcd} \mathbf{EqRel} \arrow[d, bend right=60, "Q"'] \arrow[d, phantom, "\dashv"] \\ \mathbf{Sets} \arrow[u, "Eq"'] \end{tikzcd} \qquad \textit{so that} \qquad \frac{R \longrightarrow Eq(Y)}{\overline{\overline{Q(R) \longrightarrow Y}}} \; .$$

This functor Q sends an equivalence relation $R \subseteq X \times X$ to the quotient $Q(R) = X/R$, with canonical (unit) map $[-]_R \colon X \to Q(R)$. This set-theoretic quotient Q satisfies the following three properties.

1. *Each equivalence relation $R \subseteq X \times X$ is the kernel of its quotient map $[-]_R$:*

$$R \;=\; \mathrm{Ker}([-]_R) \;=\; ([-]_R \times [-]_R)^{-1}(Eq(Q(R))).$$

More concretely, this means

$$R(x, x') \iff [x]_R = [x']_R.$$

2. *Each surjection $f \colon X \twoheadrightarrow Y$ is (isomorphic to) the quotient map of its kernel* $\mathrm{Ker}(f) \subseteq X \times X$*, as in*

$$\begin{tikzcd} \mathrm{Ker}(f) \arrow[r, shift left] \arrow[r, shift right] & X \arrow[r, two heads, "{[-]}"] \arrow[dr, two heads, "f"'] & Q(\mathrm{Ker}(f)) \arrow[d, dashed, "\cong"] \\ & & Y \end{tikzcd}$$

3. *For each weak pullback preserving functor $F \colon$ **Sets** $\to$ **Sets**, with associated lifting* $\mathrm{EqRel}(F) \colon$ **EqRel** $\to$ **EqRel**, *one has*

$$Q(\mathrm{EqRel}(F)(R)) \;\cong\; F(Q(R)).$$

Proof The first two points are completely standard, so we concentrate on the third one, using Proposition 4.4.3: it says that equality relations are preserved by set-theoretic relation lifting and that inverse images $(-)^{-1}$ are preserved because the functor preserves weak pullbacks. Thus

$$\begin{aligned} Q\big(\mathrm{EqRel}(F)(R)\big) &\cong Q\big(\mathrm{EqRel}(F)(([-]_R \times [-]_R)^{-1}(\mathrm{Eq}(Q(R))))\big) && \text{by (1)} \\ &\cong Q\Big((F([-]_R) \times F([-]_R))^{-1}(\mathrm{EqRel}(F)(\mathrm{Eq}(Q(R))))\Big) \\ &= Q\Big((F([-]_R) \times F([-]_R))^{-1}(\mathrm{Eq}(F(Q(R))))\Big) \\ &= Q(\mathrm{Ker}(F([-]_R))) \\ &\cong F(Q(R)) && \text{by (2).} \end{aligned}$$

The final step is justified because by the axiom of choice ('each surjection is split') the functor F preserves surjections; see Lemma 2.1.7. ❑

These observations are used to show that quotients lift to algebras.

Proposition 6.7.9 *For a weak pullback preserving functor $F\colon \mathbf{Sets} \to \mathbf{Sets}$ the quotient functor Q lifts to (functor) algebras as on the left below:*

$$
\begin{tikzcd}
\mathbf{Alg}(F) \arrow[d] \arrow[r, "\perp", "Eq"'] & \mathbf{Alg}(\mathrm{EqRel}(F)) \arrow[l, bend right, "Q"'] \arrow[d] \\
\mathbf{Sets} \arrow[loop below, "F"] \arrow[r, "\perp", "Eq"'] & \mathbf{EqRel} \arrow[l, bend right, "Q"'] \arrow[loop below, "\mathrm{EqRel}(F)"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{EM}(T) \arrow[d] \arrow[r, "\perp", "Eq"'] & \mathcal{EM}(\mathrm{EqRel}(T)) \arrow[l, bend right, "Q"'] \arrow[d] \\
\mathbf{Sets} \arrow[loop below, "T"] \arrow[r, "\perp", "Eq"'] & \mathbf{EqRel} \arrow[l, bend right, "Q"'] \arrow[loop below, "\mathrm{EqRel}(T)"]
\end{tikzcd}
$$

For a weak pullback preserving monad $T\colon \mathbf{Sets} \to \mathbf{Sets}$ this lifting specialises to (monad) algebras as on the right above.

Proof The lifted adjunction is a consequence of Theorem 2.5.9, using the isomorphism $FQ \cong Q\mathrm{EqRel}(F)$ from the previous lemma. For the restriction to monad algebras we need to show that the algebra equations remain valid. But this is obvious because the unit and multiplication of T and $\mathrm{EqRel}(T)$ are essentially the same; see Exercise 4.4.6. ❑

It is useful to extend axioms from relations on carriers $T(n)$ for $n \in \mathbb{N}$ to carriers $T(X)$ with arbitrary sets X.

Lemma 6.7.10 *Let T be a monad on* **Sets** *that is finitary and preserves weak pullbacks, and let $\mathcal{A}$ be an axiom system for T. For an arbitrary set X define a relation $\mathcal{A}_X \subseteq T(X) \times T(X)$ as*

$$\mathcal{A}_X = \bigcup_{n\in\mathbb{N}} \bigcup_{h\in X^n} \Big\{ \langle T(h)(s), T(h)(t)\rangle \;\Big|\; \langle s, t\rangle \in \mathrm{Th}(\mathcal{A})(n) \Big\}.$$

1. *These relations $\mathcal{A}_X$ are congruence equivalences.*
2. *For $m \in \mathbb{N}$ one has $\mathcal{A}_m = \mathrm{Th}(\mathcal{A})(m)$.*
3. *The axioms $\mathcal{A}$ hold in an arbitrary Eilenberg–Moore algebra $\alpha\colon T(X) \to X$ iff there is a map of relations*

$$
\begin{tikzcd}
\mathcal{A}_X \arrow[r, dashed] \arrow[d, tail] & Eq(X) = X \arrow[d, tail, "\Delta = \langle \mathrm{id}, \mathrm{id}\rangle"] \\
T(X) \times T(X) \arrow[r, "\alpha \times \alpha"'] & X \times X
\end{tikzcd}
$$

4. *For each Kleisli map* $f\colon X \to T(Y)$ *there is a map of relations*

$$\begin{tikzcd}
\mathcal{A}_X \arrow[rr, dashed] \arrow[d, tail] & & \mathcal{A}_Y \arrow[d, tail] \\
T(X) \times T(X) \arrow[rr, "f^\$ \times f^\$"'] & & T(Y) \times T(Y)
\end{tikzcd}$$

where $f^\$ = \mu \circ T(f)$ *is the Kleisli extension of* f. *The mapping* $X \mapsto \mathcal{A}_X$ *thus yields a functor* $\mathcal{K}\ell(T) \to \mathcal{EM}(\mathrm{EqRel}(T))$.

Proof 1. Showing that $\mathcal{A}_X$ is a congruence equivalence involves some low-level reasoning, where weak pullback preservation is used for transitivity. Details are left to the interested reader.

2. The inclusion $\mathrm{Th}(\mathcal{A})(m) \subseteq \mathcal{A}_m$ is obvious. For the other direction, assume $\langle u, v\rangle \in \mathcal{A}_m$, say with $u = T(h)(s), v = T(h)(t)$ for $h\colon n \to m$ and $\langle s, t\rangle \in \mathrm{Th}(\mathcal{A})(n)$. Since $T(h) = (\eta \circ h)^\$$, functoriality of $\mathrm{Th}(\mathcal{A})$ in Lemma 6.7.6 yields that $T(h)$ is a map of relations $\mathrm{Th}(\mathcal{A})(n) \to \mathrm{Th}(\mathcal{A})(m)$. Then $\langle u, v\rangle = \langle T(h)(s), T(h)(t)\rangle \in \mathrm{Th}(\mathcal{A})(m)$.
3. Validity of $\mathcal{A}$ in an algebra $\alpha\colon T(X) \to X$ means that for each pair $(s, t) \in \mathcal{A}(n)$ one has $[\![s]\!] = [\![t]\!]\colon X^n \to X$, where $[\![s]\!](h) = \alpha(T(h)(s))$. This precisely means for each pair $(u, v) \in \mathcal{A}_X$ one has $\alpha(u) = \alpha(v)$.
4. Assume $f\colon X \to T(Y)$ and $(u_1, u_2) \in \mathcal{A}_X$, say with $u_i = T(h)(s_i)$ for $h \in X^n$ and $(s_1, s_2) \in \mathrm{Th}(\mathcal{A})(n)$. We get $f \circ h \in T(Y)^n$. Since T is finitary we can choose for each $j \in n$ a $g_j \in Y^{m_j}$ and $r_j \in T(m_j)$ with $T(g_j)(r_j) = f(h(j))$. Put $m = m_1 + \cdots + m_n$ and $g = [g_1, \ldots, g_n]\colon m \to Y$, and $r = [T(\kappa_1) \circ r_1, \ldots, T(\kappa_n) \circ r_n]\colon n \to T(m)$, where $\kappa_i\colon m_i \rightarrowtail m$ is the appropriate insertion map. Since axioms are closed under substitution we get $(r^\$(s_1), r^\$(s_2)) \in \mathrm{Th}(\mathcal{A})(m)$. These elements prove that the pair $(f^\$(u_1), f^\$(u_2))$ is in $\mathcal{A}_Y$, since

$$\begin{aligned}
f^\$(u_i) &= \mu \circ T(f) \circ T(h) \circ s_i \\
&= \mu \circ T([T(g_1) \circ r_1, \ldots, T(g_n) \circ r_n]) \circ s_i \\
&= \mu \circ T^2(g) \circ T([T(\kappa_1) \circ r_1, \ldots, T(\kappa_n) \circ r_n]) \circ s_i \\
&= T(g) \circ \mu \circ T(r) \circ s_i \\
&= T(g)(r^\$(s_i)).
\end{aligned}$$

❑

We now come to the main technical result of this section, showing that axioms can be captured by quotient monads.

Theorem 6.7.11 *Let* T : **Sets** $\to$ **Sets** *be a monad that is finitary and weak pullback preserving, with an axiom system* $\mathcal{A}\colon \mathcal{K}\ell_{\mathbb{N}}(T) \to$ **EnRel**. *Then there is a 'quotient' monad* $T/\mathcal{A}$: **Sets** $\to$ **Sets** *with a monad map* $[-]\colon T \Rightarrow T/\mathcal{A}$, *giving rise to an isomorphism* $\mathcal{EM}(T/\mathcal{A}) \xrightarrow{\cong} \mathcal{EM}(T, \mathcal{A})$ *in*

$$\begin{tikzcd}
\mathcal{EM}(T) \arrow[dr] & \mathcal{EM}(T,\mathcal{A}) \arrow[l, hook'] \arrow[r, "\cong"] \arrow[d] & \mathcal{EM}(T/\mathcal{A}) \arrow[dl] \\
& \mathbf{Sets} \arrow[ul, bend left, "\dashv"] \arrow[ur, bend right, "\dashv"'] \arrow[loop left, "T"] \arrow[loop right, "T/\mathcal{A}"] &
\end{tikzcd}$$

Proof For a set X consider the relation $\mathcal{A}_X \hookrightarrow T(X) \times T(X)$ from the previous lemma, as algebra of the lifted monad $\mathrm{EqRel}(T)$. The lifted quotient functor $Q\colon \mathcal{EM}(\mathrm{EqRel}(F)) \to \mathcal{EM}(T)$ from Proposition 6.7.9 yields a map of Eilenberg–Moore algebras:

$$\begin{tikzcd}
T^2(X) \arrow[r, two heads, "T({[-]_{\mathcal{A}_X}})"] \arrow[d, "\mu"'] & T(T/\mathcal{A}(X)) \arrow[d, "\xi_X"] \\
T(X) \arrow[r, two heads, "{[-]_{\mathcal{A}_X}}"'] & T/\mathcal{A}(X) \stackrel{\text{def}}{=} Q(\mathcal{A}_X) = T(X)/\mathcal{A}_X
\end{tikzcd}$$

The mapping $X \mapsto T/\mathcal{A}(X)$ is functorial, since $X \mapsto \mathcal{A}_X$ is functorial, so that $[-]\colon T \Rightarrow T/\mathcal{A}$ becomes a natural transformation. Moreover, by construction, the axioms $\mathcal{A}$ hold in the algebra $\xi_X\colon T(T/\mathcal{A}(X)) \to T/\mathcal{A}(X)$, via Lemma 6.7.10: consider $(u, v) \in \mathcal{A}_{T/\mathcal{A}(X)}$, say $u = T(h)(s), v = T(h)(t)$ for $h \in (T/\mathcal{A}(X))^n$ and $(s, t) \in \mathrm{Th}(\mathcal{A})(n)$. We need to show $\xi_X(u) = \xi_X(v)$. We can choose a $g \in T(X)^n$ with $h = [-] \circ g$. Define

$$s' = \mu(T(g)(s)) = g^{\$}(s) \qquad t' = \mu(T(g)(t)) = g^{\$}(t).$$

By Lemma 6.7.10.4 we get $(s', t') \in \mathcal{A}_X$. Hence $[s'] = [t'] \in T/\mathcal{A}(X)$. But now we can finish the validity argument:

$$\begin{aligned}
\xi(u) = \xi \circ T(h) \circ s = \xi \circ T([-]) \circ T(g) \circ s &= [-] \circ \mu \circ T(g) \circ s \\
&= [-] \circ g^{\$} \circ s \\
&= [-] \circ g^{\$} \circ t = \cdots = \xi(v).
\end{aligned}$$

Next we show that the mapping $X \mapsto \xi_X$ is left adjoint to the forgetful functor $\mathcal{EM}(T, \mathcal{A}) \to \mathbf{Sets}$. For an Eilenberg–Moore algebra $\beta\colon T(Y) \to Y$ satisfying $\mathcal{A}$ we have a bijective correspondence

$$\frac{X \xrightarrow{\;\rho\;} Y}{\left(\begin{array}{c} T(T/\mathcal{A}(X)) \\ \downarrow \xi_X \\ T/\mathcal{A}(X) \end{array}\right) \xrightarrow{\;[\![-]\!]_\rho\;} \left(\begin{array}{c} T(Y) \\ \downarrow \beta \\ Y \end{array}\right)} \;. \tag{6.28}$$

Given a 'valuation' $\rho\colon X \to Y$, we obtain $\beta \circ T(\rho)\colon T(X) \to Y$, forming a map of algebras $\mu_X \to \beta$. By Lemma 6.7.10.4 the function $T(\rho) = (\eta \circ$

$\rho)^{\$}: T(X) \to T(Y)$ is a map of relations on the left below. The map on the right exists because $\beta \models \mathcal{A}$:

$$\begin{tikzcd}
\mathcal{A}_X \arrow[r, dashed] \arrow[d, tail] & \mathcal{A}_Y \arrow[r, dashed] \arrow[d, tail] & \mathrm{Eq}(Y) = Y \arrow[d, tail, "\langle \mathrm{id}, \mathrm{id} \rangle"] \\
T(X) \times T(X) \arrow[r, "T(\rho) \times T(\rho)"'] & T(Y) \times T(Y) \arrow[r, "\beta \times \beta"'] & Y \times Y
\end{tikzcd}$$

Hence there is a unique map $[\![-]\!]_\rho : T/\mathcal{A}(X) \to Y$ with $[\![-]\!]_\rho \circ [-] = \beta \circ T(\rho)$.

The monad arising from this adjunction $\mathcal{EM}(T, \mathcal{A}) \rightleftarrows \mathbf{Sets}$ is what we call the quotient monad $T/\mathcal{A} : \mathbf{Sets} \to \mathbf{Sets}$. We have as unit $\eta^{T/\mathcal{A}} = [-] \circ \eta^T$ and as multiplication $\mu^{T/\mathcal{A}} = [\![-]\!]_{\mathrm{id}} : (T/\mathcal{A})^2(X) \to T/\mathcal{A}(X)$; this is the map associated via the correspondence (6.28) with the identity map on $T/\mathcal{A}(X)$ as valuation. Thus, $\mu^{T/\mathcal{A}}$ is the unique map with $\mu^{T/\mathcal{A}} \circ [-] = \xi_X$. This allows us to show that $[-]: T \Rightarrow T/\mathcal{A}$ is a map of monads (see Definition 5.1.7):

$$\mu^{T/\mathcal{A}} \circ [-] \circ T([-]) = \xi \circ T([-]) = [-] \circ \mu^T.$$

Our next step is to prove that the comparison functor $K: \mathcal{EM}(T, \mathcal{A}) \to \mathcal{EM}(T/\mathcal{A})$ – see Exercise 5.4.19 – is an isomorphism. This functor K sends an algebra $\beta: T(Y) \to Y$ satisfying $\mathcal{A}$ to the interpretation $[\![-]\!]_{\mathrm{id}} : T/\mathcal{A}(Y) \to Y$ arising from the identity $Y \to Y$ as valuation. This $K(\beta)$ is thus the unique map with $K(\beta) \circ [-] = \beta$.

In the other direction, given an Eilenberg–Moore algebra $\alpha: T/\mathcal{A}(X) \to X$ there is a map $K^{-1}(\alpha) = \alpha \circ [-]: T(X) \to X$, which is an algebra because $[-]: T \Rightarrow T/\mathcal{A}$ is a map of monads. We have $K^{-1}(\alpha) \models \mathcal{A}$ since

$$\begin{tikzcd}
\mathcal{A}_X \arrow[r, dashed] \arrow[d, tail] & T/\mathcal{A}(X) \arrow[r, dashed] \arrow[d, tail, "\langle \mathrm{id}, \mathrm{id} \rangle"] & X \arrow[d, tail, "\langle \mathrm{id}, \mathrm{id} \rangle"] \\
T(X) \times T(X) \arrow[r, "{[-] \times [-]}"'] & T/\mathcal{A}(X) \times T/\mathcal{A}(X) \arrow[r, "\alpha \times \alpha"'] & X \times X
\end{tikzcd}$$

This functor K is an isomorphism, since

$$K^{-1}(K(\beta)) = [\![-]\!]_{\mathrm{id}} \circ [-] = \beta \circ T(\mathrm{id}) = \beta.$$

The equation $K(K^{-1}(\alpha)) = \alpha$ follows directly from the definition of K. ❑

Before describing things more abstractly we consider terms from arities again, together with axioms. Quotienting terms by axioms gives another free construction, analogously to Proposition 6.6.1.

Corollary 6.7.12 *Let # be an arity with axioms $\mathcal{A}$. Then there is a finitary monad T:* **Sets** $\to$ **Sets** *with a natural transformation $F_\# \Rightarrow T$, such that $F_\#$-algebras satisfying $\mathcal{A}$ correspond to Eilenberg–Moore T-algebras, as in*

$$\begin{tikzcd} \mathbf{Alg}(F_\#, \mathcal{A}) \cong \mathcal{EM}(T) \arrow[d] \\ \mathbf{Sets} \arrow[u, bend left] \arrow[u, phantom, "\dashv"] \arrow[loop right, "T"] \end{tikzcd}$$

Proof The free monad $F_\#^*$ is finitary by Exercise 6.6.2, and it preserves weak pullbacks by Exercise 6.7.4. Hence we can take T to be the quotient monad $F_\#^*/\mathcal{A}$ from Theorem 6.7.11, giving us a natural transformation $F_\# \Rightarrow F_\#^* \Rightarrow F_\#^*/\mathcal{A} = T$ and an isomorphism $\mathbf{Alg}(F_\#, \mathcal{A}) \cong \mathcal{EM}(F_\#^*, \mathcal{A}) \cong \mathcal{EM}(T)$. The monad T is finitary because $F_\#^*$ is finitary and the map $F_\#^* \Rightarrow T$ consists of surjective quotient maps $[-]$. ❑

The monad T in this corollary can be defined explicitly on a finite set $n \in \mathbb{N}$ as the quotient

$$T(n) = F_\#^*(n)/\mathrm{Th}(\mathcal{A})(n).$$

This holds since $\mathcal{A}_n = \mathrm{Th}(\mathcal{A})(n)$ by Lemma 6.7.10.2.

For a term $[t] \in T(n)$ and a valuation $\rho\colon n \to T(m)$ with $\rho(i) = [s_i]$ we get

$$[\![\, t \,]\!]_\rho = \big[\, t[s_1/v_1, \ldots, s_n/v_n] \,\big]_{\mathrm{Th}(\mathcal{A})}, \tag{6.29}$$

where $v_1, \ldots, v_n$ are used as variables in t. It is used in the following basic result.

Theorem 6.7.13 (Soundness and completeness) *Consider an arity # with axioms $\mathcal{A}$. For two terms $t_1, t_2 \in F_\#^*(n)$ the following are equivalent.*

1. $\mathcal{A} \vdash t_1 = t_2$
2. *For each model $M \in$ **Model**$(F_\#, \mathcal{A})$, one has $M(t_1) = M(t_2)$*
3. *For each algebra $F_\#(X) \xrightarrow{a} X$ in* **Alg**$(F_\#, \mathcal{A})$*, one has $[\![\, t_1 \,]\!]_\rho = [\![\, t_2 \,]\!]_\rho$, for every $\rho\colon n \to X$.*

Proof The implication (1) $\Rightarrow$ (2) is soundness and follows from the implication $M \models \mathcal{A} \Rightarrow M \models \mathrm{Th}(\mathcal{A})$ from Lemma 6.7.6. The implication (2) $\Rightarrow$ (3) is based on Corollary 6.6.5 (and Exercise 6.6.4), relating algebras and models. For the implication (3) $\Rightarrow$ (1) we choose the quotient $T(n) = F_\#^*(n)/\mathrm{Th}(\mathcal{A})(n)$ from Corollary 6.7.12, with algebra structure

$$F_\#(T(n)) \xrightarrow{\;\theta\;} F_\#^*(T(n)) \xrightarrow{\;\xi_n\;} T(n),$$

as in the proof of Theorem 6.7.11. We now take the monad's unit as valuation $\eta\colon n \to T(n)$. It yields $[\![t_1]\!]_\eta = [\![t_2]\!]_\eta$, by assumption. This means $[t_1] = [t_2]$, by (6.29), and thus $(t_1, t_2) \in \mathrm{Th}(\mathcal{A})(n)$. The latter says $\mathcal{A} \vdash t_1 = t_2$, as required. ❑

In Corollary 6.7.12 we have seen that an arity with equations can be described by a finitary monad. We conclude this section by showing that each finitary monad is essentially given by an arity with equations.

Recall from Lemma 4.7.3 that for a functor $F\colon \mathbf{Sets} \to \mathbf{Sets}$ there is an arity $\#_F = \pi_1\colon (\coprod_{i\in\mathbb{N}} F(i)) \to \mathbb{N}$, together with a natural transformation $\mathrm{ap}\colon F_{\#_F} \Rightarrow F$. This natural transformation has components $\mathrm{ap}_n\colon F(n) \times X^n \to F(X)$ given by $\mathrm{ap}_n(u, h) = F(h)(u)$. Lemma 4.7.3 states that these components are surjective if and only if F is finitary.

If we apply these constructions to a monad $T\colon \mathbf{Sets} \to \mathbf{Sets}$ we get an arity $\#_T$ and a natural transformation $\mathrm{ap}\colon F_{\#_T} \Rightarrow T$. The free monad $F^*_{\#_T}$ on $F_{\#_T}$ thus yields a map of monads $\overline{\mathrm{ap}}\colon F^*_{\#_T} \Rightarrow T$ with $\overline{\mathrm{ap}} \circ \theta = \mathrm{ap}$; see Proposition 5.1.8. It can be described also via initiality, as in

$$\begin{tikzcd}
X + F_{\#_T}(F^*_{\#_T}(X)) \arrow[r, dashed] \arrow[d, "\alpha_X"', "\cong"] & X + F_{\#_T}(T(X)) \arrow[d, "{[\eta, \mu \circ \mathrm{ap}]}"] \\
F^*_{\#_T}(X) \arrow[r, dashed, "\overline{\mathrm{ap}}_X"'] & T(X)
\end{tikzcd} \tag{6.30}$$

Proposition 6.7.14 *For a monad T on* **Sets** *with monad map $\overline{\mathrm{ap}}\colon F^*_{\#_T} \Rightarrow T$ from* (6.30)*, we define the* ***theory of*** *T as the axiom system $\mathrm{Th}(T)\colon \mathcal{K}\ell_{\mathbb{N}}(F^*_{\#_T}) \to$* **EqRel** *defined as follows. For $n \in \mathbb{N}$,*

$$\mathrm{Th}(T)(n) = \mathrm{Ker}(\overline{\mathrm{ap}}_n) = \{(t_1, t_2) \in F^*_{\#_T}(n) \times F^*_{\#_T}(n) \mid \overline{\mathrm{ap}}_n(t_1) = \overline{\mathrm{ap}}_n(t_2)\}.$$

Then

1. *The relations $\mathrm{Th}(T)(n)$ are congruence equivalences (and hence theories).*
2. *Assuming T is finitary, the maps $\overline{\mathrm{ap}}_X\colon F^*_{\#_T}(X) \Rightarrow T(X)$ are surjective.*

Proof 1. For convenience we shall write F for $F_{\#_T}$. We first need to check that the mapping $n \mapsto \mathrm{Th}(T)(n)$ yields a functor $\mathcal{K}\ell_{\mathbb{N}}(F^*) \to$ **EqRel**. The relations $\mathrm{Th}(T)(n)$ are obviously equivalence relations. We check functoriality: for a map $f\colon n \to F^*(m)$ we need to show $\langle t_1, t_2\rangle \in \mathrm{Th}(T)(n) \Rightarrow \langle f^\$(t_1), f^\$(t_2)\rangle \in \mathrm{Th}(T)(m)$, where $f^\$ = \mu^{F^*} \circ F^*(f)$ is the Kleisli extension of f. We have

$$\begin{aligned}
\overline{\mathrm{ap}}_m(f^{\$}(t_1)) &= (\overline{\mathrm{ap}}_m \circ \mu^{F^*} \circ F^*(f))(t_1) \\
&= (\mu^T \circ T(\overline{\mathrm{ap}}_m) \circ \overline{\mathrm{ap}}_m \circ F^*(f))(t_1) && \overline{\mathrm{ap}} \text{ is a map of monads} \\
&= (\mu^T \circ T(\overline{\mathrm{ap}}_m) \circ T(f) \circ \overline{\mathrm{ap}}_n)(t_1) \\
&= (\mu^T \circ T(\overline{\mathrm{ap}}_m) \circ T(f) \circ \overline{\mathrm{ap}}_n)(t_2) && \text{since } \langle t_1, t_2\rangle \in \mathrm{Th}(T)(n) \\
&= \cdots \\
&= \overline{\mathrm{ap}}_m(f^{\$}(t_2)).
\end{aligned}$$

Next we need to check that $\mathrm{Th}(T)(n)$ is a congruence. We do so by defining an algebra $b\colon F(\mathrm{Th}(T)(n)) \to \mathrm{Th}(T)(n)$. So assume we have a triple $\langle m \in \mathbb{N}, t \in T(m), h \in \mathrm{Th}(T)(n)^m\rangle \in F(\mathrm{Th}(T)(n))$, using that F is the arity functor $F_{\#_T}$. We can consider h as a tuple $h = \langle h_1, h_2\rangle$ of maps $h_i\colon m \to F^*(n)$. Hence we get two elements:

$$\langle m \in \mathbb{N}, t \in T(m), h_i \in F^*(n)^m\rangle \in F(F^*(n)).$$

By applying the second component of the initial algebra $\alpha_n\colon n + F(F^*(n)) \xrightarrow{\cong} F^*(n)$ we can define

$$b(m, t, h) = \langle \alpha_n(\kappa_2(m, t, h_1)),\ \alpha_n(\kappa_2(m, t, h_2))\rangle.$$

Using Diagram (6.30) one can show $b(m, t, h) \in \mathrm{Th}(T)(n)$.

2. Assuming the monad T is finitary, the maps $\overline{\mathrm{ap}}\colon F^*(X) \Rightarrow T(X)$ are surjective: for $u \in T(X)$ we can find $n \in \mathbb{N}, h \in X^n$ and $t \in T(n)$ with $T(h)(t) = u$. The triple $\langle n, t, h\rangle$ is then an element of $F_{\#_T}(X)$, which we simply write as $F(X)$. Using the free extension map $\theta\colon F \Rightarrow F^*$ from Proposition 5.1.8 we get $\theta(n, t, h) \in F^*(X)$. It proves surjectivity of $\overline{\mathrm{ap}}$:

$$\begin{aligned}
\overline{\mathrm{ap}}(\theta(n, t, h)) &= \mathrm{ap}(n, t, h) && \text{by definition of } \overline{\mathrm{ap}} \\
&= T(h)(t) && \text{by definition of ap; see Lemma 4.7.3} \\
&= u.
\end{aligned}$$

❑

We now obtain a standard result in the theory of monads see e.g. [344, 340]. The restriction to finitary monads can be avoided if one allows operations with arbitrary arities – and not just finite arities as we use here.

Corollary 6.7.15 *Each finitary monad T can be described via operations and equations, namely: via the arity $\#_T$ and theory* $\mathrm{Th}(T)$ *from the previous result one has*

$$T \cong F^*_{\#_T}/\mathrm{Th}(T),$$

using the quotient monad of Theorem 6.7.11.

Proof Since both $F^*_{\#_T}$ and T are finitary, it suffices to prove the isomorphism for finite sets $n \in \mathbb{N}$. We know the maps $\overline{\mathrm{ap}}\colon F^*_{\#_T}(n) \to T(n)$ are surjective. Hence, by Proposition 6.7.8.2, $T(n)$ is isomorphic to the quotient $F^*_{\#_T}(n)/\mathrm{Th}(T)(n)$, via the kernel

$$\mathrm{Ker}(\overline{\mathrm{ap}}) = \{(s,t) \in F^*_{\#_T}(n) \times F^*_{\#_T}(n) \mid \overline{\mathrm{ap}}(s) = \overline{\mathrm{ap}}(t)\} = \mathrm{Th}(T)(n). \qquad \square$$

Remark 6.7.16 A more abstract approach may be found in [334]. There, equations for a monad T are given by an endofunctor E together with two natural transformations $\tau_1, \tau_2 \colon E \Rightarrow T$. Terms $t, s \in T(X)$ are then related if they are in the image of the tuple $\langle \tau_1, \tau_2 \rangle \colon E(X) \to T(X) \times T(X)$. The free monad E^* yields two maps of monads $\overline{\tau_1}, \overline{\tau_2} \colon E^* \Rightarrow T$, as in Proposition 5.1.8. Next, a quotient monad T/E is obtained by taking the coequaliser of $\overline{\tau_1}, \overline{\tau_2}$ *in the category of monads*.

The approach followed here, especially in Theorem 6.7.11, is a bit more concrete. Moreover, the notion of equation that we use, in Definition 6.7.3, has closure under substitution built in.

Exercises

6.7.1 Let $a \colon F(X) \to X$ and $b \colon F(Y) \to Y$ be two algebras with a surjective homomorphism $X \twoheadrightarrow Y$ between them. Use (6.18) to prove

$$a \models \mathcal{A} \implies b \models \mathcal{A}.$$

6.7.2 Let $\mathcal{A}$ be an axiom system for an arity functor $F_\#$. Prove

$$\mathcal{A} \vdash s_i = r_i \implies \mathcal{A} \vdash t[\vec{s}/\vec{v}] = t[\vec{r}/\vec{v}].$$

6.7.3 For a monad T on **Sets** consider the category $\mathbf{EnRel}_{\mathbb{N}}(T)$ obtained by pullback in

$$\begin{tikzcd}
\mathbf{EnRel}_{\mathbb{N}}(T) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{EnRel} \arrow[d] \\
\mathcal{K}\ell_{\mathbb{N}}(T) \arrow[r] & \mathbf{Sets}
\end{tikzcd}$$

1. Give an explicit description of this category $\mathbf{EnRel}_{\mathbb{N}}(T)$.
2. Check that an axiom system as introduced in Definition 6.7.3 corresponds to a section of the forgetful functor in

$$\begin{tikzcd}
\mathbf{EnRel}_{\mathbb{N}}(T) \arrow[d] \\
\mathcal{K}\ell_{\mathbb{N}}(T) \arrow[u, bend left, "\mathcal{A}"]
\end{tikzcd}$$

6.7.4 Consider an arity $\# \colon I \to \mathbb{N}$ with associated arity functor $F_\#$ and free monad $F^*_\#$, sending a set V to the set of terms $F^*_\#(V) = \mathcal{T}_\#(V)$, as in Proposition 6.6.1. Prove that $F_\#^*\colon \mathbf{Sets} \to \mathbf{Sets}$ preserves weak pullbacks. *Hint*: Recall Exercise 6.6.1.

6.8 Coalgebras and Assertions

This section returns to the study of coalgebras, in particular, coalgebras with assertions. It follows the main lines of the previous two sections on algebras and assertions, using predicates instead of relations as axioms.

First we look into cofree coalgebras for arity functors. They always exist, by Proposition 2.5.3, and are built from suitable observations. The explicit construction is a bit more complicated than for free algebras.

For an arity $\#\colon I \to \mathbb{N}$ and a set C we shall consider infinite, finitely branching trees of the form

$$\begin{tikzcd}[column sep=small]
 & & & (c,i)\in C\times I \arrow[dll] \arrow[drr] & & & \\
 & (c_1,i_1)\in C\times I \arrow[dl] \arrow[dr] & & \overset{\begin{array}{c} m=\#i \\ \text{subtrees}\end{array}}{\cdots} & & (c_m,i_m)\in C\times I \arrow[dl] \arrow[dr] & \\
 {} & \cdots & {} & & {} & \cdots & {} \\
 & \#i_1 \text{ subtrees} & & & & \#i_m \text{ subtrees} &
\end{tikzcd} \tag{6.31}$$

More formally, we describe the set of these trees as

$$\mathcal{O}_\#(C) \;=\; \Big\{\varphi\colon \mathbb{N} \to (C\times I)^\star \;\Big|\; |\varphi(0)| = 1 \wedge \forall n.\, |\varphi(n+1)| = \#\varphi(n)\Big\}, \tag{6.32}$$

where $|\sigma| \in \mathbb{N}$ denotes the length of a list σ, and

$$\#\langle (c_1,i_1),\ldots,(c_n,i_n)\rangle \;=\; \#i_1 + \cdots + \#i_n.$$

The elements of the set C are sometimes called colours, because they occur as labels throughout the trees in $\mathcal{O}_\#(C)$. They give rise to the following dual version of Proposition 6.6.1; see also [29].

Proposition 6.8.1 *For an arity* $\#\colon I \to \mathbb{N}$ *and a set* C*, the above set of trees* $\mathcal{O}_\#(C)$ *describes the cofree coalgebra on* C *for the arity functor* $F_\#$*. This gives rise to a right adjoint, as below, where the induced comonad is the cofree comonad* $F_\#^\infty$ *on the arity functor* $F_\#$*, with isomorphism* $\mathbf{CoAlg}(F_\#) \cong \mathcal{EM}(F_\#^\infty)$ *from Proposition 5.4.7:*

$$\begin{tikzcd} \mathcal{EM}(F_\#^\infty) \cong \mathbf{CoAlg}(F_\#) \arrow[d, shift right, "U"'] \arrow[d, phantom, "\dashv"] \\ F_\#^\infty = U\mathcal{O}_\# \circlearrowright \mathbf{Sets} \arrow[u, bend right, "\mathcal{O}_\#"'] \end{tikzcd}$$

Proof The map $\varepsilon_C : \mathcal{O}_\#(C) \to C$ takes the C-element at the root of the tree, as in

$$\varepsilon_C(\varphi) = \pi_1\varphi(0) \in C.$$

The coalgebra $\zeta_C : \mathcal{O}_\#(C) \to \coprod_{i\in I} \mathcal{O}_\#(C)^{\#i}$ sends a tree φ to $i = \pi_2\varphi(0) \in I$ with its $m = \#i$ subtrees $\varphi_1, \ldots, \varphi_m$, as described in (6.31).

If we have an arbitrary coalgebra $d : X \to F_\#(X)$ with a function $f : X \to C$, there is a map of coalgebras $\mathrm{beh}_d : X \to \mathcal{O}_\#(C)$ given as

$$\mathrm{beh}_d(x) = \left(\begin{tikzcd}[column sep=small] & (f(x), i) \arrow[dl] \arrow[dr] & \\ \mathrm{beh}_d(x_1) & \cdots & \mathrm{beh}_d(x_m) \end{tikzcd} \right)$$

where $d(x) = (i, \langle x_1, \ldots, x_m\rangle)$ with $m = \#i \in \mathbb{N}$. Clearly, $\varepsilon_C \circ \mathrm{beh}_d = f$. It is not hard to see that beh_d is the unique coalgebra homomorphism $X \to \mathcal{O}_\#(C)$. ❑

The next result is the analogue of Theorem 6.6.3 for comonads, in the style of Exercise 6.6.5. It shows that coalgebras can also be understood as functorial models. In the algebraic case we restricted ourselves to *finitary* Kleisli categories $\mathcal{K}\ell_{\mathbb{N}}(T)$, with natural numbers as objects. This is natural because terms in an algebraic context typically involve only finitely many variables. In a coalgebraic context the corresponding restriction to only finitely many observable outcomes is less natural. Therefore we use ordinary Kleisli categories in the coalgebraic setting. A consequence is that we do not get exactly the same result as Theorem 6.6.3.

Theorem 6.8.2 *For a comonad S : **Sets** → **Sets** there is a full and faithful functor from Eilenberg–Moore coalgebras to models:*

$$\begin{array}{ccc} \mathcal{EM}(S)^{\mathrm{op}} & \xrightarrow{\quad\mathcal{L}\quad} & [\mathcal{K}\ell(S), \mathbf{Sets}]_{\mathrm{fp}} \\ (X \xrightarrow{\gamma} S(X)) & \longmapsto & (U \longmapsto U^X). \end{array}$$

Proof The action $\mathcal{L}(X,\gamma)(U) = U^X$ is functorial in U, since each Kleisli map $f : S(U) \to V$ yields a map $\mathcal{L}(X,\gamma)(f) : U^X \to V^X$ given by abstraction and strength as

$$\Lambda\Big(U^X \times X \xrightarrow{\ \mathrm{id}\times\gamma\ } U^X \times S(X) \xrightarrow{\mathrm{st}'} S(U^X \times X) \xrightarrow{S(\mathrm{ev})} S(U) \xrightarrow{\ f\ } V\Big),$$

where the swapped strength map st′ is as in (5.12). Explicitly,

$$\mathcal{L}(X,\gamma)(f) = \lambda g \in U^X.\, \lambda x \in X.\, f(S(g)(\gamma(x))). \tag{6.33}$$

We show that (Kleisli) identities are preserved, using that comonads on **Sets** are strong, (see Exercise 5.2.14); preservation of composition is proven similarly and is left to the reader.

$$\begin{aligned}
\mathcal{L}(X,\gamma)(\mathrm{id}) &= \Lambda(\varepsilon \circ S(\mathrm{ev}) \circ \mathrm{st}' \circ (\mathrm{id} \times \gamma)) \\
&= \Lambda(\mathrm{ev} \circ \varepsilon \circ \mathrm{st}' \circ (\mathrm{id} \times \gamma)) \\
&= \Lambda(\mathrm{ev} \circ (\mathrm{id} \times \varepsilon) \circ (\mathrm{id} \times \gamma)) \\
&= \Lambda(\mathrm{ev}) \\
&= \mathrm{id}.
\end{aligned}$$

This $\mathcal{L}$ is also functorial: for a map of coalgebras $(X \xrightarrow{\gamma} S(X)) \xrightarrow{h} (Y \xrightarrow{\beta} S(Y))$ we get a natural transformation $\mathcal{L}(h)\colon \mathcal{L}(Y,\beta) \Rightarrow \mathcal{L}(X,\gamma)$ with components $\mathcal{L}(h)_U = U^h = (-) \circ h\colon U^Y \to U^X$.

Clearly, this yields a faithful functor: if $U^h = U^k\colon U^Y \to U^X$ for each U, then by taking $U = Y$ and precomposing with the identity on Y we get $h = Y^h(\mathrm{id}_Y) = Y^k(\mathrm{id}_Y) = k$. For fullness we have to do more work. Let $\sigma\colon \mathcal{L}(Y,\beta) \Rightarrow \mathcal{L}(X,\gamma)$ be a natural transformation. Applying the component $\sigma_Y\colon Y^Y \to Y^X$ at Y to the identity function yields a map $h = \sigma_Y(\mathrm{id}_Y)\colon X \to Y$. We must show two things: h is a map of coalgebras, and $\sigma_U = U^h\colon U^Y \to U^X$. We start with the latter.

- First we notice that for a function $f\colon U \to V$ we have

  $$\mathcal{L}(X,\gamma)(f \circ \varepsilon) = f^X : U^X \longrightarrow V^X.$$

  We need to show $\sigma_U(g) = U^h(g) = g \circ h$ for each $g\colon Y \to U$. But such a function g yields a naturality square:

  $$\begin{tikzcd}
  \mathcal{L}(Y,\beta)(Y) = Y^Y \arrow[r, "\sigma_Y"] \arrow[d, "\mathcal{L}(Y,\beta)(g \circ \varepsilon) = g^Y"'] & Y^X = \mathcal{L}(X,\gamma)(Y) \arrow[d, "g^X = \mathcal{L}(X,\gamma)(g \circ \varepsilon)"] \\
  \mathcal{L}(Y,\beta)(V) = U^Y \arrow[r, "\sigma_U"'] & U^X = \mathcal{L}(X,\gamma)(U)
  \end{tikzcd}$$

  Now we are done:

  $$U^h(g) = g \circ h = g^X(h) = (g^X \circ \sigma_Y)(\mathrm{id}_Y) = (\sigma_U \circ g^Y)(\mathrm{id}_Y) = \sigma_U(g).$$

- In order to show that $h = \sigma_Y(\mathrm{id}_Y)\colon X \to Y$ is a map of coalgebras we use the naturality diagram below; it involves the identity function $S(Y) \to S(Y)$, considered as map $Y \to S(Y)$ in the Kleisli category $\mathcal{K}\ell(S)$:

$$\begin{tikzcd}
\mathcal{L}(Y,\beta)(Y) = Y^Y \arrow[r, "\sigma_Y"] \arrow[d, "\mathcal{L}(Y,\beta)(\mathrm{id}_{S(Y)})"'] & Y^X = \mathcal{L}(X,\gamma)(Y) \arrow[d, "\mathcal{L}(X,\gamma)(\mathrm{id}_{S(Y)})"] \\
\mathcal{L}(Y,\beta)(S(Y)) = S(Y)^Y \arrow[r, "\sigma_{S(Y)}"'] & U^X = \mathcal{L}(X,\gamma)(S(Y))
\end{tikzcd}$$

The description (6.33) now yields the required equation:

$$\begin{array}{rcll}
S(h) \circ \gamma & = & \mathcal{L}(X,\gamma)(\mathrm{id}_{S(Y)})(h) & \text{by (6.33)} \\
& = & (\mathcal{L}(X,\gamma)(\mathrm{id}_{S(Y)}) \circ \sigma_Y)(\mathrm{id}_Y) & \\
& = & (\sigma_{S(Y)} \circ \mathcal{L}(Y,\beta)(\mathrm{id}_{S(Y)}))(\mathrm{id}_Y) & \text{by naturality} \\
& = & \sigma_{S(Y)}(\beta) & \text{by (6.33)} \\
& = & S(Y)^h(\beta) & \text{by the previous point} \\
& = & \beta \circ h. &
\end{array}$$

□

We turn to assertions for coalgebras. Here we are interested not in the way that assertions are formed syntactically, e.g. via modal operators as in Section 6.5, but in their meaning. There are several significant differences with the approach for algebras.

- In the coalgebraic case we will be using predicates instead of relations (as for algebras).
- We distinguish *axioms* and *axiom systems*. Axioms are simply given by a subset of a final coalgebra. Axiom systems on the other hand involve collections of subsets, given in a functorial manner, as in Definition 6.7.3.

  Axioms are easier to work with in the kind of coalgebraic 'class-style' specifications that we are interested in here – see also the next section – and will thus receive most attention. Following [183] they may be called *behavioural* axioms (or *sinks*, as in [405]).

  In particular, this emphasis on subsets of a final coalgebra as axioms means that arbitrary cofree coalgebras on colours (and 'recolouring'; see Exercise 6.8.8) do not play a big role here – unlike in other work [183, 47, 17].
- We do not describe a deductive calculus for predicates on coalgebras. Such a calculus is described in [17, 422] via a 'child' and a 'recolouring' rule, corresponding to the temporal operators □ and ⊠ (see Exercise 6.8.8). These rules are defined via closure under *all* operations of some sort (successor and recolouring) and thus involve greatest fixed points. Hence they are not deductive logics in a traditional sense, with finite proof trees.
- As already explained above, before Theorem 6.8.2, we use ordinary (non–finitary) Kleisli categories in the coalgebraic case, since it is natural to have

finitely many variables in (algebraic) terms but not to restrict observations to finite sets.

Definition 6.8.3 1. Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor with final coalgebra $Z \xrightarrow{\cong} F(Z)$. **Axioms** for F are given by a (single) subset $\mathcal{A} \subseteq Z$.

A coalgebra $c\colon X \to F(X)$ satisfies $\mathcal{A}$ if the image of the unique coalgebra map $\mathrm{beh}_c\colon X \to Z$ is contained in $\mathcal{A}$; that if there is a map of predicates $\top(X) \to \mathcal{A}$ in

$$\begin{tikzcd} X \arrow[r, dashed] \arrow[d, equal, "\top(X)"'] & \mathcal{A} \arrow[d, tail] \\ X \arrow[r, "\mathrm{beh}_c"] & Z \end{tikzcd} \qquad \text{or} \qquad \textstyle\coprod_{\mathrm{beh}_c}(\top) \leq \mathcal{A} \qquad \text{or} \qquad \mathrm{beh}_c^{-1}(\mathcal{A}) = \top,$$

where $\top(X) = (X \subseteq X)$ for the truth predicate on X. In that case we write $c \models \mathcal{A}$.

2. Similarly, axioms for a comonad $S\colon \mathbf{Sets} \to \mathbf{Sets}$ are given by a subset $\mathcal{A} \subseteq S(1)$ of the carrier of the final coalgebra $\delta\colon S(1) \to S^2(1)$, that is, of the cofree coalgebra on a final/singleton set 1.

An Eilenberg–Moore coalgebra $\gamma\colon X \to S(X)$ satisfies these axioms, written as $\gamma \models \mathcal{A}$, if the image of the unique coalgebra map $S(!_X) \circ \gamma\colon X \to S(1)$ is contained in $\mathcal{A}$.

3. In this way we get full subcategories:

$$\mathbf{CoAlg}(F, \mathcal{A}) \hookrightarrow \mathbf{CoAlg}(F) \qquad \text{and} \qquad \mathcal{EM}(S, \mathcal{A}) \hookrightarrow \mathcal{EM}(S)$$

of functor and comonad coalgebras satisfying $\mathcal{A}$.

In case the comonad S is a cofree comonad F^∞ on a functor F, then the two ways of describing axioms coincide, because the final F-coalgebra Z is the cofree comonad $F^\infty(1)$ at 1. Moreover, validity for a functor coalgebra $X \to F(X)$ is the same as validity for the associated Eilenberg–Moore coalgebra $X \to F^\infty(X)$; see Exercise 6.8.2.

Axioms as defined above can also be described in a functorial manner. We shall do so for comonads and leave the analogue for functors as an exercise below.

Lemma 6.8.4 *Let $S\colon \mathbf{Sets} \to \mathbf{Sets}$ be a comonad with axioms $\mathcal{A} \subseteq S(1)$. Then one can define two functors $\mathcal{A}^{\mathcal{K}\ell}$ and $\mathcal{A}^{\mathcal{EM}}$ as below:*

$$\begin{tikzcd} & \mathbf{Pred} \arrow[d] \\ \mathcal{K}\ell(S) \arrow[ur, "\mathcal{A}^{\mathcal{K}\ell}"] \arrow[r] & \mathbf{Sets} \end{tikzcd} \qquad\qquad \begin{tikzcd} & \mathbf{Pred} \arrow[d] \\ \mathcal{EM}(S) \arrow[ur, "\mathcal{A}^{\mathcal{EM}}"] \arrow[r] & \mathbf{Sets} \end{tikzcd}$$

Applying these functors to an arbitrary morphism yields a pullback diagram (between predicates).

For an Eilenberg–Moore coalgebra $\gamma\colon X \to S(X)$ the following statements are then equivalent.

a. $\gamma \models \mathcal{A}$.
b. $\mathcal{A}^{\mathcal{EM}}(X,\gamma) = \top$.
c. *γ is a map of predicates* $\top(X) \to \mathcal{A}^{\mathcal{K}\ell}(X)$, i.e. $\mathrm{Im}(\gamma) = \coprod_\gamma(\top) \leq \mathcal{A}^{\mathcal{K}\ell}(X)$.
d. For each $f\colon X \to U$, the Kleisli extension $S(f) \circ \gamma\colon X \to S(U)$ is a map of predicates $\top(X) \to \mathcal{A}^{\mathcal{K}\ell}(U)$.

This last formulation is the coalgebraic analogue of validity of axiom systems in algebras from Definition 6.7.3. Hence in the present context we shall call a collection of predicates $\mathcal{B}(U) \subseteq S(U)$ forming a functor $\mathcal{B}\colon \mathcal{K}\ell(S) \to$ **Pred** as in the above triangle on the left an **axiom system**. Such a system is automatically closed under substitution – and in particular under recolourings; see Exercise 6.8.8. The system holds in a coalgebra $X \to S(X)$ if the image of the coalgebra is contained in $\mathcal{B}(X) \subseteq S(X)$, as in (c) above. Equivalently, condition (d) may be used.

Proof For an object (set) $U \in \mathcal{K}\ell(S)$ use the unique map $!_U\colon U \to 1$ to define a predicate by pullback:

$$\mathcal{A}^{\mathcal{K}\ell}(U) = S(!_U)^{-1}(\mathcal{A}) \subseteq S(U).$$

We have to show that for each map $g\colon U \to V$ in the Kleisli category $\mathcal{K}\ell(S)$, the resulting Kleisli extension $g^\$ = S(g) \circ \delta\colon S(U) \to S(V)$ is a map of predicates $\mathcal{A}^{\mathcal{K}\ell}(U) \to \mathcal{A}^{\mathcal{K}\ell}(V)$. First we have

$$S(!_V) \circ g^\$ = S(!_V \circ g) \circ \delta = S(!_{S(U)}) \circ \delta = S(!_U \circ \varepsilon) \circ \delta = S(!_U).$$

Next we get the required dashed map in

$$\begin{tikzcd}
\mathcal{A}^{\mathcal{K}\ell}(U) \arrow[r, dashed] \arrow[d, tail] \arrow[rr, bend left] & \mathcal{A}^{\mathcal{K}\ell}(V) \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{A} \arrow[d, tail] \\
S(U) \arrow[r, "g^\$"] \arrow[rr, bend right, "S(!_U)"'] & S(V) \arrow[r, "S(!_V)"] & S(1)
\end{tikzcd}$$

By the pullback lemma (Exercise 4.2.6) the rectangle on the left is a pullback.

Similarly, for an Eilenberg–Moore algebra $\gamma\colon X \to S(X)$ one defines a predicate

$$\mathcal{A}^{\mathcal{EM}}(X,\gamma) = (S(!_X) \circ \gamma)^{-1}(\mathcal{A}) \subseteq X.$$

For a map of coalgebras $(X \xrightarrow{\gamma} S(X)) \xrightarrow{h} (Y \xrightarrow{\beta} S(Y))$ one obtains a pullback on the left below:

$$\begin{tikzcd}
\mathcal{A}^{\mathcal{EM}}(X,\gamma) \arrow[r, dashed] \arrow[d, tail] \arrow[rr, bend left] & \mathcal{A}^{\mathcal{EM}}(Y,\beta) \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{A} \arrow[d, tail] \\
X \arrow[r, "h"] \arrow[rr, bend right, "S(!_X) \circ \gamma"'] & Y \arrow[r, "S(!_Y) \circ \beta"] & S(1)
\end{tikzcd}$$

This works since

$$S(!_Y) \circ \beta \circ h = S(!_Y) \circ S(h) \circ \gamma = S(!_X) \circ \gamma.$$

Finally, we prove the equivalence of the statements (a)–(d). Validity $\gamma \models \mathcal{A}$ in (a) means $\coprod_{S(!_X)\circ\gamma}(\top) \leq \mathcal{A}$. Equivalently, $\top = (S(!_X) \circ \gamma)^{-1}(\mathcal{A}) = \mathcal{A}^{\mathcal{EM}}(X,\gamma)$ as in point (b). This is equivalent to the statement in (c), namely $\coprod_\gamma(\top) \subseteq \mathcal{A}^{\mathcal{K}\ell}(X)$, since

$$(S(!_X) \circ \gamma)^{-1}(\mathcal{A}) = \gamma^{-1} S(!_X)^{-1}(\mathcal{A}) = \gamma^{-1}(\mathcal{A}^{\mathcal{K}\ell}(X)).$$

For each function $f\colon X \to U$ the map $S(f) = (f \circ \varepsilon)^{\$}\colon S(X) \to S(Y)$ is a map of predicates $\mathcal{A}^{\mathcal{K}\ell}(X) \to \mathcal{A}^{\mathcal{K}\ell}(U)$ by functoriality of $\mathcal{A}^{\mathcal{K}\ell}$. Hence by precomposing this map with the map in (c) we get (d). In order to get from (d) to (a) one instantiates with $f = \mathrm{id}_X$. ❑

Example 6.8.5 We consider a simplified version of the bank account specification from Example 6.5.2, with only a balance $\mathsf{bal}\colon X \to \mathbb{N}$ and deposit operation $\mathsf{dep}\colon X \times \mathbb{N} \to X$. The relevant assertion

$$\mathsf{bal}(\mathsf{dep}(x,n)) = \mathsf{bal}(x) + n \tag{6.34}$$

was written in Example 6.5.2 in modal logic style as

$$\mathsf{bal} \downarrow m \vdash [\mathsf{dep}(n)](\mathsf{bal} \downarrow (m+n)).$$

The assertion (6.34) can be interpreted directly in any coalgebra of the form $\langle \mathsf{dep}, \mathsf{bal}\rangle\colon X \to F(X)$, for $F = (-)^{\mathbb{N}} \times \mathbb{N}$, namely as subset of the state space:

$$\{x \in X \mid \forall n \in \mathbb{N}.\, \mathsf{bal}(\mathsf{dep}(x,n)) = \mathsf{bal}(x) + n\}. \tag{6.35}$$

One expects that assertion (6.34) holds in the coalgebra if this set is the whole of X. We investigate this situation in some detail.

Definition 6.8.3 involves assertions as subsets of final coalgebras. By Proposition 2.3.5 the final F-coalgebra is $\mathbb{N}^{\mathbb{N}^\star}$ with operations

$$\mathsf{bal}(\varphi) = \varphi(\langle\rangle) \qquad \text{and} \qquad \mathsf{dep}(\varphi, n) = \lambda\sigma \in \mathbb{N}^\star.\, \varphi(n \cdot \sigma).$$

The assertion (6.34) can be described as a subset of this final coalgebra:

$$\mathcal{A} = \{\varphi \in \mathbb{N}^{\mathbb{N}^\star} \mid \forall n \in \mathbb{N}.\, \varphi(\langle n\rangle) = \varphi(\langle\rangle) + n\}.$$

For an arbitrary coalgebra $\langle \mathsf{dep}, \mathsf{bal}\rangle \colon X \to X^{\mathbb{N}} \times \mathbb{N}$ the unique coalgebra map $\mathrm{beh}\colon X \to \mathbb{N}^{\mathbb{N}^\star}$ is, following the proof of Proposition 2.3.5, given by

$$\begin{aligned}\mathrm{beh}(x) &= \lambda\sigma \in \mathbb{N}^\star.\, \mathsf{bal}(\mathsf{dep}^*(x,\sigma)) \\ &= \lambda\langle n_1, \ldots, n_k\rangle \in \mathbb{N}^\star.\, \mathsf{bal}(\mathsf{dep}(\cdots \mathsf{dep}(x, n_1)\cdots, n_k)).\end{aligned}$$

We can now see that the predicate (6.35) is simply $\mathrm{beh}^{-1}(\mathcal{A})$. Hence our earlier statement that validity of the assertion means that the predicate (6.35) is all of X is precisely what is formulated in Definition 6.8.3.

We add two more observations.

1. A simple example of a model of this specification takes the natural numbers $\mathbb{N}$ as state space. The balance operation $\mathsf{bal}\colon \mathbb{N} \to \mathbb{N}$ is the identity function, and the deposit operation $\mathsf{dep}\colon \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ is addition. Clearly, assertion (6.34) holds. It is not hard to see that this model is the final one satisfying this assertion: for an arbitrary coalgebra $c\colon X \to X^{\mathbb{N}} \times \mathbb{N}$ satisfying (6.34), its 'balance' map $\pi_2 \circ c\colon X \to \mathbb{N}$ is a map of coalgebras. Exercise 6.8.4 deals with an alternative 'history' model that keeps track of past transactions.
2. We briefly describe the functorial description $\mathcal{A}^{\mathcal{K}\ell}\colon \mathcal{K}\ell(F^\infty) \to \mathbf{Pred}$ of assertions, from Lemma 6.8.4, for this bank example. An explicit description of the cofree comonad $F^\infty(U) = (U \times \mathbb{N})^{\mathbb{N}^\star}$ is obtained via Propositions 2.5.3 and 2.3.5. Its operations are

   $$\mathsf{bal}(\varphi) = \pi_2\varphi(\langle\rangle) \quad\text{and}\quad \mathsf{dep}(\varphi, n) = \lambda\sigma \in \mathbb{N}^\star.\, \varphi(n\cdot\sigma).$$

   The axiom system $\mathcal{A}^{\mathcal{K}\ell}\colon \mathcal{K}\ell(F^\infty) \to \mathbf{Pred}$ from Lemma 6.8.4 determined by (6.34) is given by

   $$\begin{aligned}\mathcal{A}^{\mathcal{K}\ell}(U) &= F^\infty(!_U)^{-1}(\mathcal{A}) \\ &= \{\varphi \in F^\infty(U) \mid \forall n \in \mathbb{N}.\, \pi_2\varphi(\langle n\rangle) = \pi_2\varphi(\langle\rangle) + n\}.\end{aligned}$$

This bank account will be investigated further in Example 6.8.10 below.

Recall from Exercise 6.2.5 that for a comonad S on **Sets** the associated predicate lifting $\mathrm{Pred}(S)\colon \mathbf{Pred} \to \mathbf{Pred}$ is also a comonad. Hence we can consider its category of coalgebras. These Eilenberg–Moore coalgebras capture S-invariants.

Lemma 6.8.6 *Let $S\colon$ **Sets** $\to$ **Sets** be a comonad with axiom system $\mathcal{A} \subseteq S(1)$. Recall the induced functor $\mathcal{A}^{\mathcal{K}\ell}\colon \mathcal{K}\ell(S) \to$ **Pred** from Lemma 6.8.4. For* *each set U we define a new predicate $\Box\mathcal{A}^{\mathcal{K}\ell}(U) \subseteq S(U)$ via the Knaster–Tarski fixpoint theorem (see* e.g. *[119, chapter 4]), namely as the greatest fixed point of the following monotone operator:*

$$\mathcal{P}(S(U)) \xrightarrow{\mathcal{A}^{\mathcal{K}\ell}(U) \wedge \delta^{-1}(\mathrm{Pred}(S)(-))} \mathcal{P}(S(U)).$$

Hence, $\Box\mathcal{A}^{\mathcal{K}\ell}(U)$ is the greatest invariant for the coalgebra $\delta\colon S(U) \to S^2(U)$, contained in $\mathcal{A}^{\mathcal{K}\ell}(U)$; see Figure 6.4.

This yields a functor (or axiom system) $\Box\mathcal{A}^{\mathcal{K}\ell}$ in a situation

$$\begin{tikzcd}
 & & \mathcal{EM}(\mathrm{Pred}(S)) \arrow[d] \\
 & & \mathbf{Pred} \arrow[d] \arrow[loop right, "\mathrm{Pred}(S)"] \\
\mathcal{K}\ell(S) \arrow[uurr, "\Box\mathcal{A}^{\mathcal{K}\ell}"] \arrow[urr, "\mathcal{A}^{\mathcal{K}\ell}"'] \arrow[rr] & & \mathbf{Sets} \arrow[loop right, "S"]
\end{tikzcd}$$

For a coalgebra $\gamma\colon X \to S(X)$ one has

$$\gamma \models \mathcal{A} \iff \gamma \models \Box\mathcal{A}^{\mathcal{K}\ell}.$$

The role of invariants (or subcoalgebras) for validity, as expressed by the last result, is emphasised in [47].

Proof The (greatest) fixed point

$$\Box\mathcal{A}^{\mathcal{K}\ell}(U) = \mathcal{A}^{\mathcal{K}\ell}(U) \wedge \delta^{-1}(\mathrm{Pred}(S)(\Box\mathcal{A}^{\mathcal{K}\ell}(U)))$$

is by construction contained in the subset $\mathcal{A}^{\mathcal{K}\ell}(U) \subseteq S(U)$. The inclusion $\Box\mathcal{A}^{\mathcal{K}\ell}(U) \subseteq \delta^{-1}(\mathrm{Pred}(S)(\Box\mathcal{A}^{\mathcal{K}\ell}(U)))$ gives a factorisation

$$\begin{tikzcd}
\Box\mathcal{A}^{\mathcal{K}\ell}(U) \arrow[r, dashed] \arrow[d, tail] & \mathrm{Pred}(S)(\Box\mathcal{A}^{\mathcal{K}\ell}(U)) \arrow[d, tail] \\
S(U) \arrow[r, "\delta"'] & S^2(U)
\end{tikzcd}$$

By Exercise 6.2.5 this is actually an Eilenberg–Moore coalgebra of the comonad $\mathrm{Pred}(S)$ and thus an S-invariant.

In order to show that $\Box\mathcal{A}^{\mathcal{K}\ell}$ is a functor, assume a map $f\colon S(U) \to V$. We have to show $f^{\$} = S(f) \circ \delta$ is a map of predicates $\Box\mathcal{A}^{\mathcal{K}\ell}(U) \to \Box\mathcal{A}^{\mathcal{K}\ell}(V)$. We have

$$\textstyle\coprod_{f^{\$}}(\Box\mathcal{A}^{\mathcal{K}\ell}(U)) \subseteq \coprod_{f^{\$}}(\mathcal{A}^{\mathcal{K}\ell}(U)) \subseteq \mathcal{A}^{\mathcal{K}\ell}(V).$$

The second inclusion follows from the fact that $\mathcal{A}^{\mathcal{K}\ell}$ is a functor. Since $\coprod_{f^{\$}}$ preserves invariants, by Exercise 6.2.5, we get $\coprod_{f^{\$}}(\Box\mathcal{A}^{\mathcal{K}\ell}(U)) \subseteq \Box\mathcal{A}^{\mathcal{K}\ell}(V))$, because the latter is the greatest invariant contained in $\mathcal{A}^{\mathcal{K}\ell}(V)$.

Let $\gamma\colon X \to S(X)$ be a coalgebra. If $\gamma \models \Box\mathcal{A}^{\mathcal{K}}$, then $\gamma \models \mathcal{A}$ because of the inclusions $\Box\mathcal{A}^{\mathcal{K}}(1) \subseteq \mathcal{A}^{\mathcal{K}}(1) = \mathcal{A}$, as subsets of $S(1)$. For the other direction, from $\gamma \models \mathcal{A}$ we obtain $\coprod_\gamma(\top) \subseteq \mathcal{A}^{\mathcal{K}}(X)$ as in Lemma 6.8.4(c). But since the image $\coprod_\gamma(\top)$ is an invariant, again by Exercise 6.2.5, we get $\coprod_\gamma(\top) \subseteq \Box\mathcal{A}^{\mathcal{K}}(X)$. ❑

We recall from (6.8) that comprehension $\{-\}$ can be described categorically as a functor **Pred** $\to$ **Sets**, sending a predicate $P \rightarrowtail X$ to its domain P, considered as a set itself. This functor $\{-\}$ is right adjoint to the truth functor $\top\colon$ **Sets** $\to$ **Sets**. Also in the present context comprehension is a highly relevant operation that takes invariants to (sub)coalgebras.

Proposition 6.8.7 *Let S be a comonad on* **Sets**.

1. *As a functor, S preserves injections.*
2. *Thus* $\{\mathrm{Pred}(S)(P)\} \cong S(\{P\})$, *for each predicate P.*
3. *Comprehension $\{-\}$ lifts to Eilenberg–Moore coalgebras in*

$$\begin{tikzcd}
\mathcal{EM}(S) \arrow[r, "\top"] \arrow[d] & \mathcal{EM}(\mathrm{Pred}(S)) \arrow[l, bend left, "\{-\}"] \arrow[d] \\
\mathbf{Sets} \arrow[r, "\top"] \arrow[loop left, "S"] & \mathbf{Pred} \arrow[l, bend left, "\{-\}"] \arrow[loop right, "\mathrm{Pred}(S)"]
\end{tikzcd}$$

Proof 1. By Lemma 2.1.7 the functor S preserves injections $f\colon X \to Y$ with $X \neq \emptyset$, so we need to consider only the case where $X = \emptyset$. There is a counit map $S(\emptyset) \to \emptyset$, which gives $S(\emptyset) = \emptyset$ since $\emptyset$ is a strict initial object in **Sets**. Hence S preserves all injections.

2. Recall that predicate lifting $\mathrm{Pred}(S)$ applied to a predicate $m\colon P \rightarrowtail X$ is obtained by epi-mono factorisation of $S(m)\colon S(P) \to S(X)$. But since $S(m)$ is injective by (1), we get $\mathrm{Pred}(S)(P) = S(P)$. Or, a bit more formally, $\{\mathrm{Pred}(S)(m)\} = S(\{m\})$.
3. This is proved by Exercise 2.5.13, using the isomorphism from the previous point. ❑

Theorem 6.8.8 *Let* $S\colon$ **Sets** $\to$ **Sets** *be a comonad with an axiom system* $\mathcal{A} \subseteq S(1)$. *Then there is a 'subset' comonad* $\{S \mid \mathcal{A}\}\colon$ **Sets** $\to$ **Sets** *with a comonad map* $\pi\colon \{S \mid \mathcal{A}\} \Rightarrow S$ *and with an isomorphism of categories* $\mathcal{EM}(\{S \mid \mathcal{A}\}) \xrightarrow{\cong} \mathcal{EM}(S, \mathcal{A})$ *in*

$$\begin{tikzcd}
\mathcal{EM}(S) \arrow[dr, bend right=10] \arrow[dr, phantom, "\dashv"] & \mathcal{EM}(S,\mathcal{A}) \arrow[l, hook'] \arrow[r, "\cong"] \arrow[d] & \mathcal{EM}(\{S \mid \mathcal{A}\}) \arrow[dl, bend left=10] \arrow[dl, phantom, "\dashv"] \\
& \mathbf{Sets} \arrow[ul, bend left] \arrow[ur, bend right] \arrow[loop left, "S"] \arrow[loop right, "\{S \mid \mathcal{A}\}"] &
\end{tikzcd}$$

A similar result involving an induced subcomonad occurs in [169], starting from a covariety (a suitably closed class of coalgebras), instead from axioms.

Proof On a set X, define $\{S \mid \mathcal{A}\}(X)$ to be the carrier of the Eilenberg–Moore subcoalgebra obtained by comprehension from the invariant $\Box\,\mathcal{A}^{\mathcal{K}}(X)$ from Lemma 6.8.6, in

$$\begin{tikzcd}
& S(\{S \mid \mathcal{A}\}(X)) \arrow[r, tail, "S(\pi_X)"] & S^2(X) \\
\{\Box\,\mathcal{A}^{\mathcal{K}}(X)\} = \{S \mid \mathcal{A}\}(X) \arrow[ur, phantom] \arrow[r, tail, "\pi_X"'] & S(X) \arrow[u, "\delta_X"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
S(\{S \mid \mathcal{A}\}(X)) \arrow[r, tail, "S(\pi_X)"] & S^2(X) \\
\{\Box\,\mathcal{A}^{\mathcal{K}}(X)\} = \{S \mid \mathcal{A}\}(X) \arrow[u, "\vartheta_X"] \arrow[r, tail, "\pi_X"'] & S(X) \arrow[u, "\delta_X"']
\end{tikzcd}$$

The axioms $\mathcal{A}$ hold by construction in the coalgebra ϑ_X: by functoriality of $\mathcal{A}^{\mathcal{K}}$ we get a map $\{S \mid \mathcal{A}\}(X) \to \mathcal{A}(S(X))$ in

$$\begin{tikzcd}
\{S \mid \mathcal{A}\}(X) = \Box\,\mathcal{A}^{\mathcal{K}}(X) \arrow[r, hook] \arrow[dr, tail, "\pi_X"'] & \mathcal{A}^{\mathcal{K}}(X) \arrow[r] \arrow[d, tail] & \mathcal{A}^{\mathcal{K}}(S(X)) \arrow[d, tail] \\
& S(X) \arrow[r, "\delta = \mathrm{id}^{\$}"'] & S^2(X)
\end{tikzcd}$$

It gives us the required factorisation below, using the pullback property from Lemma 6.8.4:

$$\begin{tikzcd}
\{S \mid \mathcal{A}\}(X) \arrow[dr, dashed] \arrow[drr, bend left] \arrow[ddr, bend right, "\vartheta_X"'] & & \\
& \mathcal{A}^{\mathcal{K}}(\{S \mid \mathcal{A}\}) \arrow[r] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & \mathcal{A}^{\mathcal{K}}(S(X)) \arrow[d, tail] \\
& S(\{S \mid \mathcal{A}\}(X)) \arrow[r, "S(\pi_X)"'] & S^2(X)
\end{tikzcd}$$

Next we show that the mapping $X \mapsto \vartheta_X$ forms a right adjoint to the forgetful functor $\mathcal{EM}(S,\mathcal{A}) \to \mathbf{Sets}$. For a coalgebra $\beta\colon Y \to S(Y)$ satisfying $\mathcal{A}$ there is a bijective correspondence:

$$\frac{Y \xrightarrow{\quad g \quad} X}{\begin{pmatrix} S(Y) \\ \beta\uparrow \\ Y \end{pmatrix} \xrightarrow[\quad h \quad]{} \begin{pmatrix} S(\{S \mid \mathcal{A}\}(X)) \\ \uparrow\vartheta_X \\ \{S \mid \mathcal{A}\}(X) \end{pmatrix}} \;.$$

This works as follows.

- Given a function $g\colon Y \to X$ we obtain $S(g) \circ \beta\colon Y \to S(X)$, which is a map of coalgebras $\beta \to \delta$. We get $\coprod_{S(g)\circ\beta}(\top) \subseteq \mathcal{A}^{\mathcal{K}}(X)$ in

$$\begin{tikzcd}
& \mathcal{A}^{\mathcal{K}}(Y) \arrow[r, dashed] \arrow[d, tail] & \mathcal{A}^{\mathcal{K}}(X) \arrow[d, tail] \\
Y \arrow[ur, dashed] \arrow[r, "\beta"'] & S(Y) \arrow[r, "S(g) = (g \circ \varepsilon)^{\$}"'] & S(X)
\end{tikzcd}$$

  Since $\coprod_{S(g)\circ\beta}(\top)$ is an invariant by Exercise 6.2.5, it is included in the greatest invariant: $\coprod_{S(g)\circ\beta}(\top) \subseteq \Box\mathcal{A}^{\mathcal{K}}(X) = \{S \mid \mathcal{A}\}(X)$. Hence there is a unique map of coalgebras $\overline{g}\colon Y \to \{S \mid \mathcal{A}\}(X)$ with $\pi_X \circ \overline{g} = S(g) \circ \beta$.
- Given a map of coalgebras $h\colon Y \to \{S \mid \mathcal{A}\}(X)$, take $\overline{h} = \varepsilon \circ \pi_X \circ h\colon Y \to X$.

We leave it to the reader to check the remaining details of this correspondence, and proceed to show that $\{S \mid \mathcal{A}\}$ is a comonad. We have a counit and comultiplication:

$$\begin{aligned}
\varepsilon_X^{\{S \mid \mathcal{A}\}} &= \Big(\{S \mid \mathcal{A}\}(X) \xrightarrow{\pi_X} S(X) \xrightarrow{\varepsilon_X^S} X\Big) \\
\delta_X^{\{S \mid \mathcal{A}\}} &= \Big(\{S \mid \mathcal{A}\}(X) \xrightarrow{\overline{\mathrm{id}_{\{S \mid \mathcal{A}\}}}} \{S \mid \mathcal{A}\}^2(X)\Big).
\end{aligned}$$

The comultiplication is obtained as map of coalgebras via the above adjoint correspondence. Then, by construction it satisfies: $\pi \circ \delta_X^{\{S \mid \mathcal{A}\}} = \vartheta_X$. This makes $\pi\colon \{S \mid \mathcal{A}\} \Rightarrow S$ a map of comonads, since

$$S(\pi) \circ \pi \circ \delta^{\{S \mid \mathcal{A}\}} = S(\pi) \circ \vartheta = \delta^S \circ \pi.$$

What remains is to show that the comparison functor $K\colon \mathcal{EM}(S, \mathcal{A}) \to \mathcal{EM}(\{S \mid \mathcal{A}\})$ is an isomorphism. This functor K sends a coalgebra $\beta\colon Y \to S(Y)$ satisfying $\mathcal{A}$ to the coalgebra $K(\beta) = \overline{\mathrm{id}_Y}\colon Y \to \{S \mid \mathcal{A}\}(Y)$ obtained via the above adjunction. Thus, by construction, $K(\beta)$ is unique with $\pi \circ K(\beta) = \beta$.

In the other direction we have a functor K^{-1} that sends a coalgebra $\gamma\colon X \to \{S \mid \mathcal{A}\}(X)$ to $K^{-1}(\gamma) = \pi \circ \gamma\colon X \to S(X)$. This coalgebra satisfies $\mathcal{A}$ since its image is contained in $\mathcal{A}^{\mathcal{K}}(X)$; see

$$\begin{tikzcd}
 & & \mathcal{A}^{\mathcal{K}}(X) \arrow[d, tail] \\
X \arrow[r, "\gamma"'] & \{S \mid \mathcal{A}\}(X) \arrow[ur, hook] \arrow[r, "\pi"'] & S(X)
\end{tikzcd}$$

Obviously, $K^{-1}(K(\beta)) = \beta$. And the equation $K(K^{-1}(\beta)) = \beta$ holds since $K(K^{-1}(\beta))$ is the unique map f with $\pi \circ f = K^{-1}(\beta)$. Since $K^{-1}(\beta) = \pi \circ \beta$ this unique map must be β itself. ❑

The next result comes from [236], where it occurs in more limited form, with a direct proof (not via the previous theorem).

Corollary 6.8.9 *Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor with cofree comonad F^{∞} and axioms $\mathcal{A} \subseteq F^{\infty}(1)$. Then there is a comonad S with natural transformation $S \Rightarrow F$ and an isomorphism of categories*

$$\begin{tikzcd}
\mathbf{CoAlg}(F, \mathcal{A}) \cong \mathcal{EM}(S) \arrow[d, shift right=2] \\
\mathbf{Sets} \arrow[u, bend right, shift right=2] \arrow[u, phantom, "\dashv"] \arrow[loop right, "S"]
\end{tikzcd}$$

Proof Of course one defines S as subset comonad $S = \{F^{\infty} \mid \mathcal{A}\}$ from Theorem 6.8.8, with isomorphism $\mathbf{CoAlg}(F^{\infty}, \mathcal{A}) \cong \mathcal{EM}(S)$ and (composite) natural transformations $S \Rightarrow F^{\infty} \Rightarrow F$. Further, by Exercise 6.8.2 and Proposition 5.4.7 there is an isomorphism $\mathbf{CoAlg}(F, \mathcal{A}) \cong \mathcal{EM}(F^{\infty}, \mathcal{A})$. Hence we are done. ❑

We return to our earlier bank account example, in order to determine the associated comonad that captures the assertion.

Example 6.8.10 In the context of Example 6.8.5 we will explicitly calculate what the comonad is whose Eilenberg–Moore coalgebras are precisely the models of the bank assertion $\mathsf{bal}(\mathsf{dep}(x, n)) = \mathsf{bal}(x) + n$ from (6.34), for coalgebras $\langle \mathsf{dep}, \mathsf{bal}\rangle\colon X \to F(X) = X^{\mathbb{N}} \times \mathbb{N}$. We already saw the cofree comonad comonad F^{∞} is given by $F^{\infty}(X) = (X \times \mathbb{N})^{\mathbb{N}^{\star}}$, with predicates:

$$\mathcal{A}^{\mathcal{K}}(X) = \{\varphi \in F^{\infty}(X) \mid \forall n \in \mathbb{N}.\, \pi_2\varphi(\langle n\rangle) = \pi_2\varphi(\langle\rangle) + n\}.$$

It is not hard to see that the greatest invariant contained in it is

$$\Box\,\mathcal{A}^{\mathcal{K}}(X) = \{\varphi \in F^{\infty}(X) \mid \forall \sigma \in \mathbb{N}^{\star}.\, \forall n \in \mathbb{N}.\, \pi_2\varphi(n \cdot \sigma) = \pi_2\varphi(\sigma) + n\}.$$

For elements φ in this subset the function $\pi_2 \circ \varphi\colon \mathbb{N}^{\star} \to \mathbb{N}$ is thus completely determined by the value $\pi_2\varphi(\langle\rangle) \in \mathbb{N}$ on the empty sequence. Hence this invariant can be identified with

$$S(X) = X^{\mathbb{N}^\star} \times \mathbb{N}.$$

This is the comonad we seek. Its counit and comultiplication are

$$\varepsilon(\varphi, m) = \varphi(\langle\rangle) \qquad \delta(\varphi, m) = \langle\, \lambda\sigma \in \mathbb{N}^\star.\, \langle\, \lambda\tau \in \mathbb{N}^\star.\, \varphi(\sigma \cdot \tau),\ m + \Sigma\sigma \rangle,\ m\,\rangle,$$

where $\Sigma\sigma \in \mathbb{N}$ is the sum of all the numbers in the sequence $\sigma \in \mathbb{N}^\star$.

Now assume we have an Eilenberg–Moore coalgebra $\gamma = \langle\gamma_1, \gamma_2\rangle \colon X \to S(X) = X^{\mathbb{N}^\star} \times \mathbb{N}$. The coalgebra equations $\varepsilon \circ \gamma = \mathrm{id}$ and $S(\gamma) \circ \gamma = \delta \circ \gamma$ amount to the following two equations:

$$\begin{aligned} \gamma_1(x)(\langle\rangle) &= x \\ \langle\, \lambda\sigma.\, \gamma(\gamma_1(x)(\sigma)),\ \gamma_2(x)\,\rangle &= \langle\, \lambda\sigma.\, \langle\, \lambda\tau.\, \gamma_1(x)(\sigma \cdot \tau),\ \gamma_2(x) + \Sigma\sigma\,\rangle,\ \gamma_2(x)\,\rangle. \end{aligned}$$

The latter equation can be split into two equations: for all $\sigma, \tau \in \mathbb{N}^\star$,

$$\gamma_2(\gamma_1(x)(\sigma)) = \gamma_2(x) + \Sigma\sigma \qquad \gamma_1(\gamma_1(x)(\sigma))(\tau) = \gamma_1(x)(\sigma \cdot \tau).$$

These equations show that γ_2 is the balance operation bal and that γ_1 is the iterated deposit operation dep^*, from (2.22), described as monoid action.

We conclude with some additional observations.

Remark 6.8.11 1. In Corollary 6.7.15 we have seen that each finitary monad can be described via operations and assertions. There is no similar result for comonads. The construction for a monad T relies on natural maps of the form $\coprod_{n\in\mathbb{N}} T(n) \times X^n \to T(X)$, from an arity functor to T; they are surjective if and only if T is finitary. Dually, for a comonad S, one may consider maps of the form

$$\begin{array}{ccc} S(X) & \longrightarrow & \prod_{n\in\mathbb{N}} S(n)^{(n^X)} \\ u & \longmapsto & \lambda n.\, \lambda h \in n^X.\, S(h)(u). \end{array}$$

One can require that these maps are injective, and then proceed with the functor $F(X) = \prod_{n\in\mathbb{N}} S(n)^{(n^X)}$ on the right-hand side. However, this functor F does not seem to have cofree coalgebras, so that the subset comonad construction from Theorem 6.8.8 does not work here. Additionally, the functor F is not a standard arity functor of the form that we have worked with in this book. The associated (non-standard) operations have been investigated to some extent in [325], to which we refer for further information. See also [62] for the notion of comodel.

2. Much work has been done towards a ‘coalgebraic Birkhoff theorem’, characterising a class of coalgebras that is defined by assertions via suitable closure properties; see for instance [183, 411, 168, 405, 321, 47, 29, 17,

169, 422]. Colours and cofree coalgebras play an important role in this work – unlike here.

3. In the end, the categorically inclined reader may wish to try to describe the quotient monad $T/\mathcal{A}$ from Theorem 6.7.11 and the subset comonad $\{S \mid \mathcal{A}\}$ from Theorem 6.8.8 via quotient and comprehension adjunctions between suitable categories of (co)monads.
4. In this book we use quotients of *relations* and comprehension of *predicates*. In a linear (or quantum) setting it is more natural to consider quotients also of predicates. For instance, for a linear subspace $P \subseteq V$ of a vector space V there is the quotient vector space V/P determined by the induced equivalence relation $\sim_P$ on V given by $x \sim_P y \Longleftrightarrow y - x \in P$. This setting allows a more uniform logical approach, with predicates only; see [96, 95].

Exercises

6.8.1 Describe the action on morphisms of the functor $\mathcal{O}_\#$ from Proposition 6.8.1.

6.8.2 Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor with cofree comonad F^∞ and axioms $\mathcal{A} \subseteq F^\infty(1)$, where $F^\infty(1)$ is the final F-coalgebra. Recall from the proof of Proposition 5.4.7 how the Eilenberg–Moore coalgebra $K(c)\colon X \to F^\infty(X)$ associated with the functor coalgebra $c\colon X \to F(X)$ is obtained. Prove $c \models \mathcal{A} \Longleftrightarrow K(c) \models \mathcal{A}$.

6.8.3 Let $F\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor with axioms $\mathcal{A} \subseteq Z$, where $Z \xrightarrow{\cong} F(Z)$ is a final coalgebra. Define, as in Lemma 6.8.4, a functor:

$$\begin{tikzcd} & \mathbf{Pred} \arrow[d] \\ \mathbf{CoAlg}(F) \arrow[ur, "\mathcal{A}"] \arrow[r] & \mathbf{Sets} \end{tikzcd}$$

6.8.4 Consider the bank account specification from Example 6.8.5. Use the set $\mathbb{N}^+$ of non-empty sequences of natural numbers as state space for a 'history' model, with balance operation $\mathsf{bal} = \mathsf{last}\colon \mathbb{N}^+ \to \mathbb{N}$; define a deposit operation $\mathsf{dep}\colon \mathbb{N}^+ \times \mathbb{N} \to \mathbb{N}^+$ such that assertion (6.34) holds.

6.8.5 Prove explicitly that the functor S with ε, δ, as described in Example 6.8.10, is a comonad on $\mathbf{Sets}$.

6.8.6 Let $S\colon \mathbf{Sets} \to \mathbf{Sets}$ be a comonad with axiom system $\mathcal{A}$. Define validity of $\mathcal{A}$ in a functorial model $\mathcal{K}\ell(S) \to \mathbf{Sets}$, in such a way that for a coalgebra $\gamma\colon X \to S(X)$,

$$\gamma \models \mathcal{A} \Longleftrightarrow \mathcal{L}(X, \gamma) \models \mathcal{A},$$

where $\mathcal{L}$ is the functor from Theorem 6.8.2.

6.8.7 Consider the more elaborate bank account from Example 6.5.2, with balance, deposit and withdraw operations combined as coalgebra of the functor

$$F(X) = \mathbb{N} \times X^{\mathbb{N}} \times (X + X)^{\mathbb{N}}.$$

1. Prove that this functor can be massaged into the isomorphic form

$$X \longmapsto \mathbb{N} \times \mathcal{P}(\mathbb{N}) \times X^{\mathbb{N}+\mathbb{N}}.$$

2. Use Propositions 2.5.3 and 2.3.5 to determine the cofree coalgebra F^∞ on F as

$$F^\infty(X) = (X \times \mathbb{N} \times \mathcal{P}(\mathbb{N}))^{(\mathbb{N}+\mathbb{N})^\star}.$$

   Describe the balance, deposit and withdraw operations from Example 6.5.2 explicitly on $F^\infty(X)$.
3. Interpret the assertions from Example 6.5.2 as subset $\mathcal{A} \subseteq F^\infty(1)$ and as invariant $\Box\mathcal{A}^{\mathcal{K}}(X) \subseteq F^\infty(X)$.
4. Prove that the resulting comonad, as in Corollary 6.8.9, is $S(X) = X^{(\mathbb{N}+\mathbb{N})^\star} \times \mathbb{N}$. (Thus, the final coalgebra $S(1)$ is $\mathbb{N}$, as for the bank account specification in Example 6.8.5.)

6.8.8 Let S be a comonad on **Sets**. Following [47], we write, for a subset $P \subseteq S(X)$,

$$\boxtimes P = \bigcap\{h^{-1}(P) \mid h\colon S(X) \to S(X) \text{ is a coalgebra map}\}.$$

1. Check that $\boxtimes P \subseteq P$. This $\boxtimes P$ is the greatest subset of P closed under all recolourings h of P.
2. Axiom systems are automatically closed under $\boxtimes$: prove that for an axiom system $\mathcal{A}\colon \mathcal{K}\ell(S) \to$ **Pred**, where $\mathcal{A}(X) \subseteq S(X)$, one has $\boxtimes\mathcal{A}(X) = \mathcal{A}(X)$.

6.9 Coalgebraic Class Specifications

This final section illustrates the use of assertions for coalgebras in the description of state-based systems in computer science, in particular in object–oriented programming. We start with the well-known mathematical structure of Fibonacci numbers, formulated coalgebraically. Figure 6.4 presents a simple illustration of a ‘coalgebraic specification’. It will be explained step by step. A coalgebraic specification is a structured text with a name (here: ‘Fibonacci’) that describes coalgebras with an initial state satisfying assertions.

More formally, a coalgebraic specification consists of three parts or sections, labelled 'operations', 'assertions', 'creation'. Here we give only an informal description; refer to [406, 445] for more details.

The operations section consists of a finite list of coalgebras $c_i \colon X \to F_i(X)$ of simple polynomial functors F_i. Of course, they can be described jointly as a single coalgebra of the product functor $F_1 \times \cdots \times F_n$, but in these specifications it is clearer to describe these operations separately, with their own names. Among the operations one sometimes distinguishes between 'fields' (or 'observers') and 'methods'. Fields are coalgebras of the form $X \longrightarrow A$ whose result type A is a constant that does not contain the state space X. Hence these fields do not change the state, but only give some information about it. In contrast, methods have the state X in their result type and can change the state, i.e. have a side effect. Hence, in Figure 6.4, the operation val is a field and next is a method.

In object-oriented programming languages a class is a basic notion that combines data with associated operations on such data. A coalgebraic specification can be seen as specification of such a class, where the fields capture the data and the methods their operations.

The assertions section contains assertions about the coalgebras in the operations section. They involve a distinguished variable $x \colon X$, so that they can be interpreted as predicates on the state space X, much as in the banking illustration in Example 6.8.5. The assertions are meant to constrain the behaviour of the coalgebras in a suitable manner.

Finally, the creation section of a coalgebraic specification contains assertions about the assumed initial state new. These assertions may involve the coalgebras from the operations section.

A **model** of a coalgebraic specification consists of (1) a coalgebra c of the (combined, product) type described in the operations section of the specification that satisfies the assertions and (2) an initial state that satisfies the creation conditions.

Coalgebraic Specification *Fibonacci*

Operations

$\mathsf{val} \colon X \longrightarrow \mathbb{N}$

$\mathsf{next} \colon X \longrightarrow X$

Assertions

$\mathsf{val}(\mathsf{next}(\mathsf{next}(x))) = \mathsf{val}(\mathsf{next}(x)) + \mathsf{val}(x)$

Creation

$\mathsf{val}(\mathsf{new}) = 1$

$\mathsf{val}(\mathsf{next}(\mathsf{new})) = 1$

Figure 6.4 A coalgebraic specification of a Fibonacci system.

Here is a possible model of the *Fibonacci* specification from Figure 6.4. As state space we take

$$X = \{(f, n) \in \mathbb{N}^{\mathbb{N}} \times \mathbb{N} \mid \forall m \geq n.\, f(m+2) = f(m+1) + f(m)\} \qquad (6.36)$$

with operations

$$\mathsf{val}(f, n) = f(n) \quad \text{and} \quad \mathsf{next}(f, n) = (f, n+1).$$

It is clear that the resulting coalgebra $\langle \mathsf{val}, \mathsf{next} \rangle \colon X \to \mathbb{N} \times X$ satisfies the assertion from Figure 6.4. As initial state we can take $\mathsf{new} = (\mathit{fib}, 0) \in X$, where $\mathit{fib} \colon \mathbb{N} \to \mathbb{N}$ is the well-known recursively defined Fibonacci function satisfying $\mathit{fib}(0) = 1$, $\mathit{fib}(1) = 1$, and $\mathit{fib}(m+2) = \mathit{fib}(m+1) + \mathit{fib}(m)$, for all $m \in \mathbb{N}$. Notice that our states $(f, n) \in X$ implicitly keep track of the stage n in the infinite sequence of Fibonacci numbers $\langle \mathit{fib}(0), \mathit{fib}(1), \mathit{fib}(2), \ldots \rangle$. But this stage is not directly visible from the outside. The specification requires only that the current value is available and that a next state can be computed.

Earlier we mentioned that coalgebraic specifications can be understood as specifications of classes in object-oriented programming languages. We shall sketch how this works by describing a class in the object-oriented programming language Java [43] that can be understood as 'implementation' of the *Fibonacci* specification from Figure 6.4. It is presented in Figure 6.5. First we note that this Java implementation uses bounded integers `int`

```
class Fibonacci {

   private int current_value;
   private int previous_value;

   public int val() {
      return current_value;
   }

   public void next() {
      int next_value = current_value + previous_value;
      previous_value = current_value;
      current_value = next_value;
   }

   public Fibonacci() {
      current_value = 1;
      previous_value = 0;
   }
}
```

Figure 6.5 A Java implementation for the Fibonacci specification from Figure 6.4.

where the specification uses (unbounded) natural numbers $\mathbb{N}$, since $\mathbb{N}$ is not available in Java.[2] This already leads to a mismatch. Further, the Java implementation uses an auxiliary field `previous_value` which is not present in the specification. However, since it is `private` and since it has no 'get' method, this `previous_value` is not visible from the outside. Apart from overflow (caused by the bounded nature of `int`), the assertion from Figure 6.4 seems to hold for the implementation. Also, the creation conditions seem to hold for the initial state resulting from the constructor `Fibonacci()` in Figure 6.5.

Continuing the discussion of this Java implementation a bit further, one can ask whether there is a way to make it mathematically precise that the Java implementation from Figure 6.5 yields a model (as defined above) for the coalgebraic specification in Figure 6.4. One way is to give a 'coalgebraic semantics' to Java by interpreting Java programs as suitable coalgebras. This happened for instance in [66, 265, 256]. However, from a programming perspective it makes more sense to incorporate assertions as used in coalgebraic specification into the programming language. This can be done for instance via the specification language JML [91]. It involves assertions, such as class invariants and pre- and post-conditions for methods, that can be checked and verified with the aid of various tools. Figure 6.6 contains an annotated version of the Java Fibonacci class, where logical assertions are preceded by special comment signs `//@` making them recognisable for special JML compilers and tools. The assertions themselves are mostly self-explanatory, except possibly for two keywords: `\old(-)` refers to the value of a field before a method call, and `\result` refers to the outcome of a (non-void) method.

We return to our more mathematically oriented approach to coalgebraic specifications and ask ourselves what the final coalgebra is satisfying the assertion from Figure 6.4 – ignoring the initial state for a moment. The general approach of Theorem 6.8.8, concretely described in Example 6.8.10, tells that we should first look at the final coalgebra of the functor $X \mapsto \mathbb{N} \times X$ – which is $\mathbb{N}^{\mathbb{N}}$ by Proposition 2.3.5 – and consider the greatest invariant $P = \Box(\mathit{assertion}) \subseteq \mathbb{N}^{\mathbb{N}}$ as subcoalgebra. It is not hard to see that $P = \{f \in \mathbb{N}^{\mathbb{N}} \mid \forall m.\, f(m+2) = f(m+1) + f(m)\}$. This means that any $f \in P$ is completely determined by its first two values $f(0)$ and $f(1)$. Hence the final coalgebra satisfying the assertions can be identified with $\mathbb{N}^2 = \mathbb{N} \times \mathbb{N}$, with operations $\mathsf{val}\colon \mathbb{N}^2 \to \mathbb{N}$ and $\mathsf{next}\colon \mathbb{N}^2 \to \mathbb{N}^2$ given by

[2] The integral type `int` in Java uses 32-bit 'signed' numbers, which are in the interval $[-2^{31}, 2^{31} - 1] = [-2147483648, 2147483647]$.

```
class Fibonacci {

    //@ invariant previous_value >= 0 &&
    //@           current_value >= previous_value;

    private int current_value;
    private int previous_value;

    //@ ensures \result == current_value;
    public int val() {
       return current_value;
    }

    //@ assignable previous_value, next_value;
    //@ ensures previous_value == \old(current_value) &&
    //@         current_value == \old(current_value) +
    //@                          \old(previous_value);
    public void next() {
       int next_value = current_value + previous_value;
       previous_value = current_value;
       current_value = next_value;
    }

    //@ assignable previous_value, next_value;
    //@ ensures previous_value == 0 && current_value == 1;
    public Fibonacci() {
       current_value = 1;
       previous_value = 0;
    }
}
```

Figure 6.6 The Java *Fibonacci* class from Figure 6.5 with JML annotations.

$$\mathsf{val}(n_1, n_2) = n_2 \quad \text{and} \quad \mathsf{next}(n_1, n_2) = (n_2, n_2 + n_1).$$

It satisfies the assertion from Figure 6.4:

$$\begin{aligned} \mathsf{val}(\mathsf{next}(\mathsf{next}(n_1, n_2))) &= \mathsf{val}(\mathsf{next}(n_2, n_2 + n_1)) \\ &= \mathsf{val}(n_1 + n_2, (n_2 + n_1) + n_2) \\ &= (n_2 + n_1) + n_2 \\ &= \mathsf{val}(\mathsf{next}(n_1, n_2)) + \mathsf{val}(n_1, n_2). \end{aligned}$$

Within this final coalgebra we also find the initial state $\mathsf{new} = (0, 1) \in \mathbb{N}^2$ satisfying the creation condition from Figure 6.4.

Interestingly, this final coalgebra with initial state corresponds closely to the Java implementation from Figure 6.5. It forms the 'minimal realisation' of the required behaviour, which needs to involve only two natural numbers.

Exercises

6.9.1 Describe the unique coalgebra map from the state space (6.36) to the final coalgebra $\mathbb{N}^2$ of the Fibonacci specification in Figure 6.4. Check that it preserves the initial state.

6.9.2 Derive from the Fibonacci specification in Figure 6.4:

$$\Box\,(\{x \in X \mid \mathsf{val}(\mathsf{next}(x)) \geq \mathsf{val}(x)\})(\mathsf{new}).$$

6.9.3 Consider the functor $F(X) = \mathbb{N} \times X$ for the Fibonacci specification in Figure 6.4.

1. Show that the cofree comonad F^∞ on F is given by $F^\infty(X) = (X \times \mathbb{N})^{\mathbb{N}}$; describe the coalgebra structure $F^\infty(X) \to F(F^\infty(X))$.
2. Interpret the assertion from Figure 6.4 as a subset $\mathcal{A}(X) \subseteq F^\infty(X)$ and determine the greatest invariant $\Box\,\mathcal{A}(X) \subseteq F^\infty(X)$.
3. Prove that the comonad induced by Corollary 6.8.9 is

$$S\,(X) \;=\; X^{\mathbb{N}} \times \mathbb{N} \times \mathbb{N},$$

with counit and comultiplication

$$\begin{aligned} \varepsilon(\varphi, m, n) \;&=\; \varphi(0) \\ \delta(\varphi, m, n) \;&=\; \langle\, \lambda k \in \mathbb{N}.\, \langle\, \varphi(k + (-)),\ \mathit{Fib}(k, m, n)\,\rangle,\ m,\ n\,\rangle, \end{aligned}$$

where $\mathit{Fib}(k, -)\colon \mathbb{N}^2 \to \mathbb{N}^2$ is the outcome of the monoid action obtained by doing k Fibonacci steps starting from the input:

$$\mathit{Fib}(0, m, n) \;=\; (m, n) \qquad\qquad \mathit{Fib}(k + 1, m, n) \;=\; \mathit{Fib}(k, n, m + n).$$

Check that the Eilenberg–Moore coalgebras of S correspond to Fibonacci coalgebras (without initial state).

6.9.4 Consider a coalgebra $\langle \mathsf{val}, \mathsf{next}\rangle\colon X \to \mathbb{N} \times X$ satisfying the assertion from the Fibonacci specification in Figure 6.4. Prove that for each $x \in X$ and $\epsilon > 0$,

$$\Diamond\left(\Box\left(\{y \in X \mid \left|\frac{\mathsf{val}(\mathsf{next}(y))}{\mathsf{val}(y)} - \frac{1 + \sqrt{5}}{2}\right| < \epsilon\}\right)\right)(x).$$

(This is coalgebraic/temporal way of saying that the limit of the quotient $\mathsf{val}(\mathsf{next}^{(n+1)}(x))/\mathsf{val}(\mathsf{next}^{(n)}(x))$ is the golden ratio $1 + \sqrt{5}/2$, as n goes to infinity.)