

# Minimisation in Logical Form

Nick Bezhanishvili and Marcello Bonsangue and Helle Hvid Hansen and Dexter Kozen and Clemens Kupke and Prakash Panangaden and Alexandra Silva

**Abstract** Recently, two apparently quite different duality-based approaches to automata minimisation have appeared. One is based on ideas that originated from the controllability-observability duality from systems theory, and the other is based on ideas derived from Stone-type dualities specifically linking coalgebras with algebraic structures derived from modal logics. In the present paper, we develop a more abstract view and unify the two approaches. We show that dualities, or more generally dual adjunctions, between categories can be lifted to dual adjunctions between categories of coalgebras and algebras, and from there to automata with initial as well as final states. As in the Stone-duality approach, algebras are essentially logics for reasoning about the automata. By exploiting the ability to pass between these categories, we show that one can minimize the corresponding automata. We give an abstract minimisation algorithm that has several instances, including the celebrated Brzozowski minimisation algorithm. We further develop three examples that have been treated in previous works: deterministic Kripke frames based on a Stone-type duality, weighted automata based on the self-duality of semimodules, and topological automata based on Gelfand duality. As a new example, we develop alternating automata based on the discrete duality between sets and complete atomic Boolean algebras.

---

Nick Bezhanishvili  
University of Amsterdam, The Netherlands, e-mail: [N.Bezhanishvili@uva.nl](mailto:N.Bezhanishvili@uva.nl)

Marcello Bonsangue  
Leiden University, The Netherlands e-mail: [marcello@liacs.nl](mailto:marcello@liacs.nl)

Helle Hvid Hansen  
University of Groningen, The Netherlands e-mail: [h.h.hansen@rug.nl](mailto:h.h.hansen@rug.nl)

Dexter Kozen  
Cornell University, USA e-mail: [kozen@cornell.edu](mailto:kozen@cornell.edu)

Clemens Kupke  
University of Strathclyde, Glasgow, Scotland e-mail: [clemens.kupke@strath.ac.uk](mailto:clemens.kupke@strath.ac.uk)

Prakash Panangaden  
McGill University, Canada e-mail: [prakash@cs.mcgill.ca](mailto:prakash@cs.mcgill.ca)

Alexandra Silva  
University College London, United Kingdom e-mail: [alexandra.silva@gmail.com](mailto:alexandra.silva@gmail.com)

## 1 Introduction

Category theory, algebra, and logic are deepening our understanding of program semantics and algorithms, and Samson has been a pioneer and leader in developing this field. His seminal paper *Domain Theory in Logical Form* [1] studies the connection between program logic and domain theory via Stone duality. This is an example of a fundamental duality in Computer Science between (operational or denotational) semantics and program logic which can be viewed as following the algebraic structure of the syntax.

Building on Stone’s celebrated representation theorems for Boolean algebras [56] and distributive lattices [57], categorical dualities linking algebra and topology [36] have been widely used in logic and theoretical computer science [15, 23, 30]. With algebras corresponding to the syntactic, deductive side of logical systems, and topological spaces to their semantics, Stone-type dualities provide a powerful mathematical framework for studying various properties of logical systems. More recently, it has also been fruitfully explored in more algorithmic applications: notably in understanding minimisation of various types of automata [2, 13, 16, 17, 27, 41, 50]. Among these, [13] and [16] were published around the same time, and they had some similarities, but also some key differences. It was not clear whether the differences could be reconciled in a principled way. The main aim of this paper is to find a unifying perspective on the minimisation constructions in [13] and [16] which we briefly recall here.

In [13], (generalised) Moore automata (without initial state) are modelled as coalgebras on base categories of algebras or topological spaces. The main observation used in [13] is that for many types of such coalgebras, one can define a category of algebras that is dually equivalent to the category of coalgebras. This dual equivalence generalises the Jónsson-Tarski duality known from modal logic, which in turn arises from Stone duality. The algebras in [13] are therefore understood as modal algebras, i.e., they consist of an algebra (that describes a propositional logic, e.g., Boolean logic) expanded with the modal operators. From this coalgebra-algebra duality it follows that maximal quotients of coalgebras correspond to minimal subobjects of algebras. Moreover, it is shown that for a given coalgebra, the minimal subalgebra of its dual modal algebra consists of the algebra of definable subsets. A maximal quotient can therefore be constructed by computing definable subsets and dualising. The minimisation-via-duality approach of [13] was shown to apply to partially observable DFAs (using duality of finite sets and finite Boolean algebras), linear weighted automata (using the self-duality of vector spaces), and belief automata viewed as coalgebras on compact Hausdorff spaces (using Gelfand duality). Moreover, for each of these examples it is shown that the definable subsets are determined by the subsets definable in the trace logic fragment consisting of formulas of the shape  $[a_0] \cdots [a_n]p$ .

In [16], Brzozowski’s double-reversal minimisation algorithm [22] for deterministic finite automata (with both initial and final states) was described categorically and its correctness explained via the duality between reachability and observability known from control theory (cf. [6, 7, 38]). This duality arises from a dual adjunction between algebras and coalgebras, not a full duality, but this is sufficient to formalise Brzozowski’s algorithm in terms of a dual adjunction between categories of automata. This categorical formulation was then used to formulate Brzozowski-style minimisation algorithms for Moore automata (over **Set**) and weighted automata over commutative semirings, which include nondeterministic and linear weighted automata as instances. To be more precise, a weighted automaton is first determined into a generalised Moore automaton with a semimodule statespace to which the double-reversal algorithm is applied, yielding in the end a minimal Moore automaton.

The perspective taken in [16] is language-based; no link is made to modal logic. Conversely, the perspective taken in [13] is logic-based; no link is made to reachability, and language acceptance is only implicitly present via trace logic. Duality will play a central role in our unification of

these approaches. Our work is very much inspired by Samson’s perspective and we hope that he will regard it as being in the spirit of his own approach to formalising theories.

The contributions of the present paper are as follows.

1. A categorical framework within which minimisation algorithms can be understood and different approaches unified (Section 3). We start by illustrating the difference in the approaches from [13] and [16] on classic deterministic automata (Section 3.1), and then proceed to a general setup for different automata types based on algebra and coalgebra (Section 3.2). Section 3.3 includes the categorical picture that unifies the work in [13] and [16]: in a nutshell, it is a stack of three interconnected adjunctions. It starts with a base dual adjunction that is subsequently lifted to a dual adjunction between coalgebras and algebras, and finally to a dual adjunction between automata. Section 3.4 extends this categorical picture to include trace logic. Section 3.5 presents an abstract understanding of reachability and observability, and finally everything is summarised and abstract minimisation algorithms are stated in Section 3.6.
2. A thorough illustration of the general framework instantiated to concrete examples. In Section 4), we revisit a range of examples stemming from previous approaches: deterministic Kripke frames, weighted automata, and topological automata (belief automata). In Section 5, we include an extensive new example on alternating automata, which uses the duality of complete atomic Boolean algebras and sets. For weighted automata, we use our framework to extend a well-known result for weighted automata over a field [54] to weighted automata over a principal ideal domain: the minimal weighted automaton over a principal ideal domain always exists, and, as expected, it has a state space smaller or equal than that of the original automaton.

We conclude the paper with a review of related work (Section 6).

**Acknowledgments** We thank Jan Rutten and Filippo Bonchi for their contributions to the earlier stages this work, and Jurriaan Rot for fruitful discussions.

## 2 Preliminaries

In this section, we fix notation and recall basic definitions of coalgebras and algebras. For a more detailed introduction to coalgebra, we refer to [51]. For general categorical notions, see e.g. [3].

Categories are denoted by  $\mathcal{C}, \mathcal{D}, \dots$ , objects of categories by  $X, Y, Z, \dots$ , and arrows or morphisms of categories by  $f, g, h, \dots$ . We denote by  $\mathbf{Set}$  the category of sets and functions. Let  $X_1, X_2$  be in  $\mathcal{C}$ . The product of  $X_1$  and  $X_2$  (if it exists) is denoted by  $X_1 \times X_2$  with projection maps  $\pi_i: X_1 \times X_2 \rightarrow X_i$ ,  $i = 1, 2$ . Similarly, their coproduct (if it exists) is written  $X_1 + X_2$  with coprojection (inclusion) maps  $\text{in}_i: X_i \rightarrow X_1 + X_2$ . In  $\mathbf{Set}$ ,  $X_1 \times X_2$  and  $X_1 + X_2$  are the usual constructions of cartesian product and disjoint union. Let  $X$  be an object in  $\mathcal{C}$  and  $A$  be a set. Assuming  $\mathcal{C}$  has products, then  $X^A := \prod_A X$  denotes the  $A$ -fold product of  $X$  with itself. Similarly, if  $\mathcal{C}$  has coproducts, then  $A \cdot X := \coprod_A X$  denotes the  $A$ -fold coproduct of  $X$  with itself.

The covariant powerset functor  $\mathcal{P}: \mathbf{Set} \rightarrow \mathbf{Set}$  sends a set  $X$  to its powerset  $\mathcal{P}(X)$  and a function  $f: X \rightarrow Y$  to the direct-image map  $\mathcal{P}(f): \mathcal{P}(X) \rightarrow \mathcal{P}(Y)$ . The contravariant powerset functor  $\mathcal{Q}: \mathbf{Set} \rightarrow \mathbf{Set}^{\text{op}}$  also sends a set  $X$  to its powerset, now denoted  $\mathcal{Q}(X) = 2^X$ , and a function  $f: X \rightarrow Y$  to its inverse-image map  $\mathcal{Q}(f): \mathcal{Q}(Y) \rightarrow \mathcal{Q}(X)$ .

## 2.1 Coalgebras, Algebras and Monads

Given an endofunctor  $F: \mathcal{C} \rightarrow \mathcal{C}$ , an  $F$ -coalgebra is a pair  $(X, \gamma: X \rightarrow FX)$ , where  $X$  is a  $\mathcal{C}$ -object and  $\gamma: X \rightarrow FX$  is a  $\mathcal{C}$ -arrow. The functor  $F$  specifies the type of the coalgebra (which may be thought of as the type of observations and transitions), and the structure map  $\gamma$  specifies the dynamics. An  $F$ -coalgebra morphism from an  $F$ -coalgebra  $(X, \gamma)$  to an  $F$ -coalgebra  $(Y, \delta)$  is a  $\mathcal{C}$ -arrow  $h: X \rightarrow Y$  that preserves the coalgebra structure, i.e.,  $\delta \circ h = Fh \circ \gamma$ .  $F$ -coalgebras and  $F$ -coalgebra morphisms form a category denoted by  $\mathbf{Coalg}_{\mathcal{C}}(F)$ . A final  $F$ -coalgebra is a final object in  $\mathbf{Coalg}_{\mathcal{C}}(F)$ , i.e., an  $F$ -coalgebra  $(Z, \zeta)$  is final if for all  $F$ -coalgebras  $(X, \gamma)$  there is a unique  $F$ -coalgebra morphism  $h: (X, \gamma) \rightarrow (Z, \zeta)$ .

An  $F$ -algebra is a pair  $(X, \alpha)$ , where  $X$  is a  $\mathcal{C}$ -object and  $\alpha: FX \rightarrow X$  is a  $\mathcal{C}$ -arrow. Now, the functor  $F$  can be seen to specify the type of operations of the algebra. An  $F$ -algebra morphism from an  $F$ -algebra  $(X, \alpha)$  to an  $F$ -algebra  $(Y, \beta)$  is a  $\mathcal{C}$ -arrow  $h: X \rightarrow Y$  that preserves the algebra structure, i.e.,  $h \circ \alpha = \beta \circ Fh$ .  $F$ -algebras and  $F$ -algebra morphisms form a category denoted by  $\mathbf{Alg}_{\mathcal{C}}(F)$ . An initial  $F$ -algebra is an initial object  $(A, \alpha)$  in  $\mathbf{Alg}_{\mathcal{C}}(F)$ , i.e., for all  $F$ -algebras  $(X, \beta)$  there is a unique  $F$ -algebra morphism  $h: (A, \alpha) \rightarrow (X, \beta)$ .

A monad (on  $\mathcal{C}$ ) is a triple  $(T, \eta, \mu)$  consisting of a functor  $T: \mathcal{C} \rightarrow \mathcal{C}$  and two natural transformations  $\eta: Id \rightarrow T$  (the unit) and  $\mu: TT \rightarrow T$  (the multiplication) satisfying  $\mu \circ \eta T = id_T = \mu \circ T\eta$  and  $\mu \circ T\mu = \mu \circ \mu T$ . For brevity, we will sometimes refer to a monad simply by its functor part, leaving the unit and multiplication implicit. An Eilenberg-Moore  $T$ -algebra is a  $T$ -algebra  $(A, \alpha)$  such that  $\alpha \circ \eta_A = id_A$  and  $\alpha \circ \mu_A = \alpha \circ T\alpha$ . Eilenberg-Moore  $T$ -algebras and  $T$ -algebra morphisms form a category denoted by  $\mathbf{EM}(T)$ . In particular, for every  $X$  in  $\mathcal{C}$ ,  $(TX, \mu_X)$  is the free Eilenberg-Moore  $T$ -algebra on  $X$ , i.e., for every  $(A, \alpha)$  in  $\mathbf{EM}(T)$  and every  $\mathcal{C}$ -arrow  $f: X \rightarrow A$  there is a unique  $T$ -algebra morphism (called the free extension of  $f$ )  $f^\sharp: (TX, \mu_X) \rightarrow (A, \alpha)$  such that  $f^\sharp \circ \eta_X = f$ . Notice also that we have  $f^\sharp = \alpha \circ Tf$ .

## 2.2 Determinisation

Let  $(T, \eta, \mu)$  be a monad on  $\mathbf{Set}$  and  $F: \mathbf{Set} \rightarrow \mathbf{Set}$  a functor given by  $FX = B \times X^\Sigma$  where  $\Sigma$  is a set and  $B$  is the carrier of an Eilenberg-Moore  $T$ -algebra  $(B, \beta)$ . Then  $FT$ -coalgebras can be seen as automata with input alphabet  $\Sigma$ , output in  $B$  and branching structure given by  $T$ . For example, nondeterministic automata are  $F\mathcal{P}$ -coalgebras where  $FX = 2 \times X^\Sigma$  and  $\beta = \vee: \mathcal{P}2 \rightarrow 2$  is the join (or max). Such  $FT$ -coalgebras can be “determinised” using a generalisation of the classic powerset construction [55], and the result can be seen as an  $F$ -coalgebra in the category  $\mathbf{EM}(T)$ . We follow [12, 34] in explaining this general construction. As shown in [34], there is a so-called distributive law  $\lambda: TF \Rightarrow FT$  of the monad  $(T, \eta, \mu)$  over the functor  $F$  given by

$$\lambda_X: T(B \times X^\Sigma) \xrightarrow{\langle T\pi_1, T\pi_2 \rangle} TB \times T(X^\Sigma) \xrightarrow{\beta \times \mathbf{st}} B \times (TX)^\Sigma \quad (1)$$

where  $\mathbf{st}: T \circ (-)^\Sigma \Rightarrow (-)^\Sigma \circ T$  is the strength natural transformation that exists for all monads on  $\mathbf{Set}$ . Such a distributive law  $\lambda$  corresponds to a lifting of  $F: \mathbf{Set} \rightarrow \mathbf{Set}$  to a functor  $F_\lambda: \mathbf{EM}(T) \rightarrow \mathbf{EM}(T)$  [37], and it induces a functor  $(-)^\sharp: \mathbf{Coalg}_{\mathbf{Set}}(FT) \rightarrow \mathbf{Coalg}_{\mathbf{EM}(T)}(F_\lambda)$  which sends an  $FT$ -coalgebra  $\gamma = \langle o, t \rangle: X \rightarrow B \times (TX)^\Sigma$  to its determinisation  $\gamma^\sharp = F\mu_X \circ \lambda_{TX} \circ T\gamma$ , that is,

$$\gamma^\sharp = TX \xrightarrow{T\gamma} T(B \times (TX)^\Sigma) \xrightarrow{\lambda_{TX}} TB \times (TTX)^\Sigma \xrightarrow{B \times (\mu_X)^\Sigma} B \times (TX)^\Sigma \quad (2)$$

Another perspective is that  $\lambda$  induces an Eilenberg-Moore  $T$ -algebra structure  $\alpha$  on  $FTX$ , and  $\gamma^\sharp: (TX, \mu_S) \rightarrow (FTX, \alpha)$  is the free extension of  $\gamma$  induced by  $\alpha$ . This also justifies our use of the notation  $(-)^\sharp$ . The determinisation  $\gamma^\sharp$  can be seen as a Moore automaton in  $\mathbf{EM}(T)$ . We will use the determinisation construction in order to place alternating automata and weighted automata in our general minimisation framework.

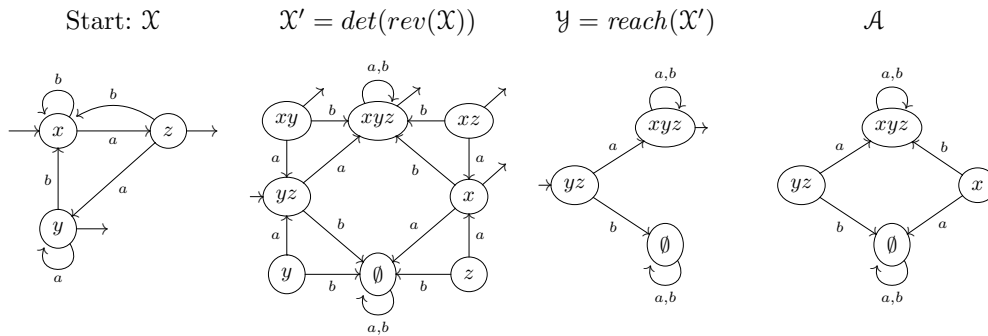
### 3 Minimisation via Dual Adjunctions

In this section, we will present a categorical picture that unifies the approaches of [13] and [16]. In particular, our picture formalises the role of trace logic in the minimisation algorithms. Some of the technical details of this part are known from [16, 32, 39, 50] – precise connections are detailed throughout the subsections below and in Section 6.

#### 3.1 An illustrative example

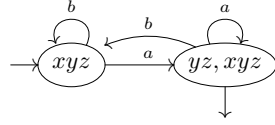
We illustrate the differences between the approaches of [13] and [16] on classic deterministic finite automata (DFAs). A DFA can be minimised via Brzozowski’s algorithm as well as via the approach in [13] using the duality between finite sets and finite Boolean algebras, and observing that a DFA is a PODFA with a single observation (or atomic proposition letter)  $p$  which is true precisely at the accepting states.

We will apply the two minimisation algorithms to the DFA  $\mathcal{X}$  below left which is also found in (11) in [16]. The DFA  $\mathcal{X}$  accepts the language  $(a + b)^*a$ . The result  $\mathcal{X}'$  after the first reverse-determinise step in Brzozowski’s algorithm is shown to the right of  $\mathcal{X}$ . Disregarding final states,  $\mathcal{X}'$  is also the modal algebra obtained from  $\mathcal{X}$ . The reachable part of  $\mathcal{X}'$  is the automaton  $\mathcal{Y}$ , and the algebra  $\mathcal{A}$  is the subalgebra of definable subsets in the modal language with the single proposition letter  $p$ , and a modality for each letter of the alphabet.

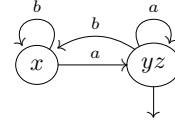


After doing again reverse-determinise-reachability on  $\mathcal{Y}$  to complete the Brzozowski algorithm, we get the automaton below on the left. Taking the dual automaton of atoms of  $\mathcal{A}$ , we get the coalgebra below on the right.

Result of Brzozowski's algorithm:

Accepts  $(a + b)^*a$ 

Result of minimisation-via-duality

State  $x$  accepts  $(a + b)^*a$ 

The two deterministic automata (modulo initial state) are clearly isomorphic, but not identical.

### 3.2 Automata, Algebras and Coalgebras

Throughout this paper, we let  $\Sigma$  be a finite set. We will consider different types of automata, but they will all have input alphabet  $\Sigma$ .

A deterministic finite automaton (DFA), on alphabet  $\Sigma$ , consists of a set  $X$  (the state space), a transition map  $t: X \rightarrow X^\Sigma$  (or equivalently  $t: \Sigma \times X \rightarrow X$ ), an acceptance map  $f: X \rightarrow 2$ , and an initial state  $i: 1 \rightarrow X$ . We generalise this basic definition to arbitrary categories as follows.

**Definition 1** Let  $\mathcal{C}$  be a category, and let  $I$  and  $B$  be objects in  $\mathcal{C}$ . A  $\mathcal{C}$ -automaton (with initialisation in  $I$  and output in  $B$ ) is a quadruple  $\mathcal{X} = (X, t, i, f)$  consisting of a state space object (or carrier)  $X$  in  $\mathcal{C}$ , a  $\Sigma$ -indexed set of transition morphisms  $\{t_a: X \rightarrow X \mid a \in \Sigma\}$ , an initialisation morphism  $i: I \rightarrow X$ , and an output morphism  $f: X \rightarrow B$ . A  $\mathcal{C}$ -automaton morphism from  $\mathcal{X}_1 = (X_1, t_1, i_1, f_1)$  to  $\mathcal{X}_2 = (X_2, t_2, i_2, f_2)$  is a  $\mathcal{C}$ -morphism  $h: X_1 \rightarrow X_2$  such that for all  $a \in \Sigma$ ,  $h \circ t_{1,a} = t_{2,a} \circ h$ ,  $f_1 = f_2 \circ h$ , and  $h \circ i_1 = i_2$ . Together,  $\mathcal{C}$ -automata (with initialisation in  $I$  and output in  $B$ ) and their morphisms form a category which we denote by  $\text{Aut}_{\mathcal{C}}^{I,B}$ .

A deterministic automaton is a  $\text{Set}$ -automaton with output in  $2$  and initialisation in  $1$ .

A central observation in [16] is that automata can be seen as coalgebras with initialisation, or dually, as algebras with output, as we briefly recall now. Assuming that  $\mathcal{C}$  has products and coproducts, the transition morphisms  $\{t_a: X \rightarrow X \mid a \in \Sigma\}$  correspond uniquely to morphisms of the following type:

$$\frac{\langle t_a \rangle_{a \in \Sigma}: X \rightarrow X^\Sigma}{[t_a]_{a \in \Sigma}: \Sigma \cdot X \rightarrow X} \quad (1)$$

Letting  $F$  and  $G$  be endofunctors on  $\mathcal{C}$  given by  $FX = B \times X^\Sigma$  and  $GX = I + \Sigma \cdot X$ , we see that a  $\mathcal{C}$ -automaton is an  $F$ -coalgebra  $\langle f, \langle t_a \rangle_{a \in \Sigma} \rangle: X \rightarrow B \times X^\Sigma$  together with an initialisation morphism  $i: I \rightarrow X$ . Or equivalently, a  $G$ -algebra  $[i, [t_a]_{a \in \Sigma}]: GX \rightarrow X$  together with an output morphism  $f: X \rightarrow B$ .

### 3.3 Dual Adjunctions of Coalgebras, Algebras and Automata

The categorical picture that unifies the work in [13] and [16] is sketched in the diagram (2) below. This picture starts with a base dual adjunction that is lifted to a dual adjunction between coalgebras and algebras. This adjunction captures the construction in [13] for obtaining observable coalgebras via duality. The coalgebra-algebra adjunction is then lifted to a dual adjunction between automata which captures the formalisation of the Brzozowski algorithm from [16], which

uses automata with initial states. In the remainder of the section, we will explain the details of how this picture comes about.

$$\begin{array}{ccc}
 (\text{Aut}_{\mathcal{C}}^{I, S(O)})^{\text{op}} & \xrightarrow{\bar{P}'} & \text{Aut}_{\mathcal{D}}^{O, P(I)} \\
 \downarrow & \begin{array}{c} \xleftarrow{\bar{S}'} \\ \xrightarrow{\bar{P}} \\ \xleftarrow{\bar{S}} \end{array} & \downarrow \\
 \text{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{\text{op}} & \xrightarrow{\bar{P}} & \text{Alg}_{\mathcal{D}}(G_{\mathcal{D}}) \\
 \downarrow & \begin{array}{c} \xleftarrow{\bar{S}} \\ \xrightarrow{\bar{P}} \\ \xleftarrow{\bar{S}} \end{array} & \downarrow \\
 F_{\mathcal{C}}^{\text{op}} \curvearrowright \mathcal{C}^{\text{op}} & \xrightarrow{\bar{P}} & \mathcal{D} \curvearrowright G_{\mathcal{D}} \\
 & \begin{array}{c} \xleftarrow{\bar{S}} \\ \xrightarrow{\bar{P}} \\ \xleftarrow{\bar{S}} \end{array} & 
 \end{array} \tag{2}$$

$$F_{\mathcal{C}} = S(O) \times (-)^{\Sigma}, \quad G_{\mathcal{D}} = O + \Sigma \cdot (-)$$

### 3.3.1 Base dual adjunction

Our starting point is a dual adjunction  $S \dashv P$  between categories  $\mathcal{C}$  and  $\mathcal{D}$  as in the above picture. We will generally try to avoid the use of superscript  $\text{op}$ , and treat  $P$  and  $S$  as contravariant functors. The units of the dual adjunction will be denoted  $\eta: Id \Rightarrow PS$  and  $\varepsilon: Id \Rightarrow SP$ . The natural isomorphism of Hom-sets  $\theta_{X,Y}: \mathcal{C}(X, SY) \rightarrow \mathcal{D}(Y, PX)$ , will sometimes be written in both directions simply as  $f \mapsto f^b$ . For  $f: X \rightarrow SY$ , its adjoint is  $f^b = Pf \circ \eta_Y$ , and for  $g: Y \rightarrow PX$ , its adjoint is  $g^b = Sg \circ \varepsilon_X$ .

In all our examples,  $\mathcal{C}$  and  $\mathcal{D}$  are concrete categories, and the dual adjunction arises from homming into a dualising object  $\Delta$  (cf. [49]), i.e.,  $P = \mathcal{C}(-, \Delta)$  and  $S = \mathcal{D}(-, \Delta)$ , and we will often denote both of them by  $\Delta^{(-)}$ . This means that adjoints are obtained simply by swapping arguments. E.g., for  $f: Y \rightarrow \Delta^X$  we have  $f^b(x)(y) = f(y)(x)$ . Moreover, the units are given by evaluation. E.g.  $\eta_X: X \rightarrow \Delta^{\Delta^X}$  is defined by  $\eta_X(x)(f) = f(x)$ .

*Example 1* Consider the self-dual adjunction of  $\text{Set}$  given by the *contravariant powerset functor*  $\mathcal{Q} = \text{Set}(-, 2)$  which maps a set  $X$  to its powerset  $2^X$  and a function  $f: X \rightarrow Y$  to its inverse image map  $f^{-1}: 2^Y \rightarrow 2^X$ . The functor  $\mathcal{Q}$  is dually self-adjoint with  $\mathcal{Q}^{\text{op}} \dashv \mathcal{Q}$ , and the isomorphism of Hom-sets is given by taking exponential transposes, i.e., for  $f: X \rightarrow 2^Y$  we have  $f^b: Y \rightarrow 2^X$ .

Dual adjunctions are also called *logical connections* as they form the basis of semantics for coalgebraic modal logics [18, 35, 40]. In this logic perspective,  $\mathcal{C}$  is a category of state spaces,  $\mathcal{D}$  is a category of algebras (e.g. Boolean algebras) encoding a propositional logic, and the functor  $G_{\mathcal{D}}$  encodes a modal logic. Intuitively, the adjoint  $P$  maps a state space  $C$  to the predicates over  $C$ , and  $S$  maps an algebra  $A$  to the theories of  $A$ . The logic given by  $G_{\mathcal{D}}$  can be interpreted over  $F_{\mathcal{C}}$ -coalgebras by providing a so-called one-step modal semantics in the form of a natural transformation  $\varrho: G_{\mathcal{D}}P \Rightarrow PF_{\mathcal{C}}$ , or equivalently via its mate  $\xi: F_{\mathcal{C}}S \Rightarrow SG_{\mathcal{D}}$ . The pair  $(G_{\mathcal{D}}, \varrho)$  is referred to as a logic. By assuming that the initial  $G_{\mathcal{D}}$ -algebra  $(A_0, \alpha_0)$  exists, and viewing its elements as formulas, the semantics of formulas in an  $F_{\mathcal{C}}$ -coalgebra  $(C, \gamma)$  is obtained from the initial map  $s^{G_{\mathcal{D}}}: (A_0, \alpha_0) \rightarrow P(\gamma) \circ \varrho_C$ . As an underlying  $\mathcal{D}$ -map, it has type  $s^{G_{\mathcal{D}}}: A_0 \rightarrow P(C)$ , hence it maps formulas to predicates. Alternatively, the semantics can be specified by the theory map  $th^{G_{\mathcal{D}}}: C \rightarrow S(A_0)$  which is defined as the adjoint of  $s^{G_{\mathcal{D}}}$ . We refer to [18, 35, 40] for a more detailed introduction to coalgebraic modal logic via dual adjunctions.

### 3.3.2 Dual adjunction between coalgebras and algebras

The base dual adjunction in (2) lifts to one between coalgebras and algebras due to the shape of the functors  $F_{\mathcal{C}}$  and  $G_{\mathcal{D}}$ . This follows from some basic results in [32, 39] as we explain now.

So assume that  $\mathcal{C}$  has products,  $\mathcal{D}$  has coproducts, and that we have a base dual adjunction  $S \dashv P$  and functors  $F_{\mathcal{C}}$  and  $G_{\mathcal{D}}$  as in (2), in particular,

$$F_{\mathcal{C}}(C) = S(O) \times C^{\Sigma} \quad \text{and} \quad G_{\mathcal{D}}(D) = O + \Sigma \cdot D$$

We know from [32, Cor. 2.15] (see also [39, Thm. 2.5]), that the dual adjunction  $S \dashv P$  lifts to a dual adjunction  $\bar{S} \dashv \bar{P}$  between  $\mathbf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}}) = \mathbf{Alg}_{\mathcal{C}^{\text{op}}}(F_{\mathcal{C}}^{\text{op}})$  and  $\mathbf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$  if there is a natural isomorphism  $\xi: F_{\mathcal{C}}S \xrightarrow{\cong} SG_{\mathcal{D}}$ . For our choice of  $F_{\mathcal{D}}$  and  $G_{\mathcal{C}}$ , we have such a natural isomorphism, since for all  $D \in \mathcal{D}$ ,

$$F_{\mathcal{C}}S(D) = S(O) \times S(D)^{\Sigma} \cong S(O + \Sigma \cdot D) = SG_{\mathcal{D}}(D) \quad (3)$$

since  $S$  (as a dual adjoint functor) turns colimits into limits. Let  $\varrho: G_{\mathcal{D}}P \Rightarrow PF_{\mathcal{C}}$  be the mate of  $\xi$ , i.e., the adjoint of  $\xi_P \circ F_{\mathcal{C}}\varepsilon$ :

$$\varrho = PF_{\mathcal{C}}\varepsilon \circ P\xi_P \circ \eta G_{\mathcal{D}}P \quad (4)$$

The lifted adjoint functors are defined for all  $F_{\mathcal{C}}$ -coalgebras  $\gamma: C \rightarrow F_{\mathcal{C}}(C)$ , all  $F_{\mathcal{C}}$ -coalgebra morphisms  $f$ , all  $G_{\mathcal{D}}$ -algebras  $\alpha: G_{\mathcal{D}}(D) \rightarrow D$ , and all  $G_{\mathcal{D}}$ -algebra morphisms  $g$  by:

$$\begin{aligned} \bar{P}(\gamma) &= P\gamma \circ \varrho_C: G_{\mathcal{D}}PC \rightarrow PC, \quad \bar{P}(f) = P(f) \\ \bar{S}(\alpha) &= \xi_D \circ S\alpha: SD \rightarrow F_{\mathcal{C}}SD, \quad \bar{S}(g) = S(g) \end{aligned} \quad (5)$$

*Remark 1* If  $F'_{\mathcal{C}}: \mathcal{C} \rightarrow \mathcal{C}$  is  $F'_{\mathcal{C}}(C) = B \times C^{\Sigma}$  with  $B \cong S(O)$ , then  $F'_{\mathcal{C}} \xrightarrow{\cong} F_{\mathcal{C}}$ , and hence  $\mathbf{Coalg}_{\mathcal{C}}(F'_{\mathcal{C}}) \cong \mathbf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$ , so we can think of  $F'_{\mathcal{C}}$ -coalgebras as  $F_{\mathcal{C}}$ -coalgebras.

The isomorphism  $\bar{\theta}$  of Hom-sets for  $\bar{S} \dashv \bar{P}$  is simply the restriction of the isomorphism  $\theta$  of Hom-sets for  $S \dashv P$  to the relevant morphisms.

The natural transformation  $\varrho: G_{\mathcal{D}}P \Rightarrow PF_{\mathcal{C}}$  provides the one-step semantics for a modal logic for  $F_{\mathcal{C}}$ -coalgebras as described at the end of Section 3.3.1. This makes most sense when the dual adjunction arises from a dualising object  $\Delta$  in which case  $\Delta$  is a domain of truth-values, i.e., the logic is  $\Delta$ -valued, and when  $\mathcal{D}$  is a category of algebras with operations given by a signature  $Sgn$ . Letting  $\Phi_{\mathcal{D}}(X)$  denote the free algebra in  $\mathcal{D}$  over a set  $X$ , an algebra functor  $G_{\mathcal{D}} = \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot (-)$  then corresponds to a modal language  $\mathbb{L}(G_{\mathcal{D}})$  that has atomic propositions from a finite set  $\Omega$ , labelled modalities  $[a]$ ,  $a \in \Sigma$ , and the propositional connectives are the operations from  $Sgn$ . That is, formulas in  $\mathbb{L}(G_{\mathcal{D}})$  are generated by the following grammar:

$$\varphi ::= q \in \Omega \mid [a]\varphi, a \in \Sigma \mid \sigma(\varphi_1, \dots, \varphi_n), \sigma \in Sgn \quad (6)$$

where  $n$  is the arity of the operation  $\sigma$ .

For our specific choice of functors  $F_{\mathcal{C}}$  and  $G_{\mathcal{D}}$ , and when the adjunction arises from a dualising object  $\Delta$  (i.e.,  $S(\Phi_{\mathcal{D}}(\Omega)) = \Delta^{\Phi_{\mathcal{D}}(\Omega)}$ ), we can compute the concrete definition of  $\varrho$  from (4), and we get the following  $\Delta$ -valued modal semantics of the language  $\mathbb{L}(G_{\mathcal{D}})$ :

$$\begin{aligned} \llbracket q \rrbracket(x) &= j(q), & \text{where } \gamma(x) &= \langle j: \Delta^{\Phi_{\mathcal{D}}(\Omega)}, d: X^{\Sigma} \rangle \\ \llbracket [a]\varphi \rrbracket(x) &= \llbracket \varphi \rrbracket(d(a)), & \text{where } \gamma(x) &= \langle j: \Delta^{\Phi_{\mathcal{D}}(\Omega)}, d: X^{\Sigma} \rangle \\ \llbracket \sigma(\psi_1, \dots, \psi_n) \rrbracket(x) &= \sigma(\llbracket \psi_1 \rrbracket(x), \dots, \llbracket \psi_n \rrbracket(x)) \end{aligned} \quad (7)$$



This shows that  $\varrho$  gives the expected modal semantics for  $F_{\mathcal{C}}$ -coalgebras viewed as deterministic  $\Sigma$ -labelled Kripke frames with observations from  $\Omega$ . In particular, the modalities are “deterministic” Kripke box/diamond-modalities.

*Example 2* We consider the case of DFAs. Here  $\mathcal{C} = \mathcal{D} = \mathbf{Set}$ ,  $F_{\mathbf{Set}} = 2 \times (-)^{\Sigma}$  and  $G_{\mathbf{Set}} = 1 + \Sigma \cdot (-)$ , and the self-dual adjunction of  $\mathbf{Set}$  is given by the contravariant powerset functor  $\mathcal{Q} = \mathbf{Set}(-, 2)$  (Example 1). The formulas of  $\mathbb{L}(G_{\mathbf{Set}})$  are built from a single atomic proposition  $q$ , and a modality  $[a]$  for each  $a \in \Sigma$ , since  $\mathcal{D} = \mathbf{Set}$  means that there are no propositional connectives. The initial  $G_{\mathbf{Set}}$ -algebra is  $\Sigma^*$ , the set of finite words over  $\Sigma$ , which is easily seen to be in bijection with the set of formulas. The logic we obtain is *trace logic* [40], but here interpreted over DFAs rather than labelled transition systems as in [40]. The natural transformation  $\varrho$  has type  $\varrho_X : 1 + \Sigma \cdot 2^X \rightarrow 2^{2 \times X^{\Sigma}}$ , given concretely below together with the induced semantics, where we write  $x \Vdash \varphi$  iff  $\llbracket \varphi \rrbracket(x) = 1$ :

$$\begin{aligned} \varrho_X(q) &= \{(b, d) \in 2 \times X^{\Sigma} \mid b = 1\} & x \Vdash q &\iff x \text{ is accepting} \\ \varrho_X(a, U) &= \{(b, d) \in 2 \times X^{\Sigma} \mid d(a) \in U\} & x \Vdash [a]\varphi &\iff x \xrightarrow{a} y \text{ and } y \Vdash \varphi \end{aligned}$$

### 3.3.3 Dual adjunction between automata

In order to obtain the upper adjunction in (2) (which formalises the Brzozowski algorithm), we will use algebra and coalgebra structure on both sides, hence we assume that  $\mathcal{C}$  and  $\mathcal{D}$  both have products and coproducts. The lifting is a small extension of  $\bar{S} \dashv \bar{P}$  obtained by defining how an initialisation map  $I \rightarrow C$  for an  $F_{\mathcal{C}}$ -coalgebra  $\gamma$  is turned into an observation map  $PC \rightarrow PI$  for the  $G_{\mathcal{D}}$ -algebra  $\bar{P}(\gamma)$ , and vice versa for  $\bar{S}$ .

**Theorem 1** *Assume that  $\mathcal{C}$  and  $\mathcal{D}$  both have products and coproducts, and that we have the dual adjunctions and functors  $F_{\mathcal{C}}$  and  $G_{\mathcal{D}}$  as specified in the two lower parts of (2). The dual adjunction  $\bar{S} \dashv \bar{P}$  between  $\mathbf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$  and  $\mathbf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$  lifts to a dual adjunction  $\bar{S}' \dashv \bar{P}'$  between  $\mathbf{Aut}_{\mathcal{C}}^{I, SO}$  and  $\mathbf{Aut}_{\mathcal{D}}^{O, PI}$  by defining  $\bar{P}'$  and  $\bar{S}'$  as follows for all  $\gamma : C \rightarrow F_{\mathcal{C}}C$  and  $\alpha : G_{\mathcal{D}}D \rightarrow D$ :*

$$\begin{aligned} \bar{P}'(\gamma, i : I \rightarrow C) &= (\bar{P}(\gamma) : G_{\mathcal{D}}PC \rightarrow PC, P(i) : PC \rightarrow PI), \bar{P}'(f) = P(f) \\ \bar{S}'(\alpha, j : D \rightarrow PI) &= (\bar{R}(\alpha) : SD \rightarrow F_{\mathcal{C}}SD, j^{\flat} : I \rightarrow SD), \bar{S}'(g) = S(g) \end{aligned}$$

**Proof** This is a minor generalisation of Prop. 9.1 in [16]. It suffices to show that for all  $\mathcal{C}$ -arrows  $i : I \rightarrow C$ , and all  $\mathcal{D}$ -arrows  $g : D \rightarrow PI$  and  $h : D \rightarrow PX$ :  $g = Pi \circ h$  iff  $g^{\flat} = h^{\flat} \circ i$ . First, if  $g = Pi \circ h$ , then  $g^{\flat} = Sg \circ \varepsilon_I = Sh \circ SPi \circ \varepsilon_I = Sh \circ \varepsilon_X \circ i = h^{\flat} \circ i$ , where the third equality follows from naturality of  $\varepsilon$ . Conversely, if  $g^{\flat} = h^{\flat} \circ i$ , then  $g = Pg^{\flat} \circ \eta_D = Pi \circ Ph^{\flat} \circ \eta_D = Pi \circ h$ .  $\square$

The final  $F_{\mathcal{C}}$ -coalgebra exists and has carrier  $S(O)^{\Sigma^*}$ . The final morphism  $! : C \rightarrow S(O)^{\Sigma^*}$  assigns to each state in  $C$  an  $S(O)$ -weighted language. For  $\mathcal{X} = \langle \gamma, i \rangle \in \mathbf{Aut}_{\mathcal{C}}^{I, S(O)}$ , we define its language semantics as the composition  $I \xrightarrow{i} C \xrightarrow{!_{\gamma}} S(O)^{\Sigma^*}$ . This  $\mathcal{C}$ -morphism can be seen as a  $\Sigma^*$ -indexed family of  $\mathcal{C}$ -morphisms  $\langle \mathcal{X} \rangle_w : I \rightarrow SO$  defined for all  $w = a_1 \cdots a_k \in \Sigma^*$  by

$$\langle \mathcal{X} \rangle_w = I \xrightarrow{i} C \xrightarrow{t_{a_1}} \cdots \xrightarrow{t_{a_k}} C \xrightarrow{f} S(O)$$

Computing the adjoint transpose  $\langle \mathcal{X} \rangle_w^{\flat} = P \langle \mathcal{X} \rangle \circ \eta_O$ , we get the  $\mathcal{D}$ -morphism:

$$\langle \mathcal{X} \rangle_w^{\flat} = P(I) \xleftarrow{P(i)} P(C) \xleftarrow{P(t_{a_1})} \cdots \xleftarrow{P(t_{a_k})} P(C) \xleftarrow{P(f)} P(S(O))$$

Hence  $\langle \mathcal{X} \rangle_w^b = \langle \overline{P'}(\mathcal{X}) \rangle_{w^R}$  where  $w^R = a_k \cdots a_1$  is the reversal of  $w$ . Similarly, we find that for all  $\mathcal{Y} \in \text{Aut}_{\mathcal{D}}^{O, P(I)}$ ,  $\langle \mathcal{Y} \rangle_w^b = \langle \overline{S'}(\mathcal{Y}) \rangle_{w^R}$ . In the case of DFAs from Example 2 where  $I = O = 1$  and  $S(O) = P(I) \cong 2$ , the above says that the adjoint functors reverse the language accepted by the automaton.

### 3.4 Language Semantics and Trace Logic

In this section, we give a general condition on the output sets that ensures that we can link trace logic with the full modal logic via an adjunction. This places trace logic in the general picture. In [13], it was shown in each of the concrete examples that trace logic is equally expressive as the full modal logic. The results of this section give a general explanation of this fact.

Assume that the category  $\mathcal{D}$  is monadic over  $\text{Set}$  with adjunction  $\Phi_{\mathcal{D}} : \mathcal{D} \rightleftarrows \text{Set} : U_{\mathcal{D}}$ . This adjoint situation allows us to relate the  $\text{Set}$ -based language semantics to the final  $F_{\mathcal{C}}$ -coalgebra semantics as we will show now.

Consider the functor  $G : \text{Set} \rightarrow \text{Set}$  defined as  $G(X) = \Omega + \Sigma \cdot X = \Omega + \Sigma \times X$  where  $\Omega$  is a finite set of observations. Then the set  $\Sigma^* \Omega$  is an initial  $G$ -algebra with algebra structure  $\Omega + \Sigma \times (\Sigma^* \Omega) \rightarrow \Sigma^* \Omega$  given by prefixing  $o \in \Omega$  with the empty word  $o \mapsto \varepsilon o$  and concatenation  $(a, w) \mapsto aw$ . Then we can compose the adjunction  $\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}$  with the dual adjunction  $S \dashv P$  to obtain a dual adjunction between  $\mathcal{C}$  and  $\text{Set}$  as follows:

$$\begin{array}{c}
 \begin{array}{ccccc}
 & & P & & U_{\mathcal{D}} \\
 & & \curvearrowright & & \curvearrowright \\
 F_{\mathcal{C}}^{\text{op}} & \xrightarrow{\mathcal{C}^{\text{op}}} & \mathcal{D} & \xrightarrow{\text{Set}} & \text{Set} \\
 & \swarrow S & \downarrow \Phi_{\mathcal{D}} & \searrow G & \\
 & & \mathcal{D} & & 
 \end{array} \\
 \text{(8)}
 \end{array}$$

**Lemma 1** *Assume we have the situation in (8), and that  $F_{\mathcal{C}}$ ,  $G_{\mathcal{D}}$ ,  $G$  are defined by:*

$$F_{\mathcal{C}}(C) = S\Phi_{\mathcal{D}}(\Omega) \times C^{\Sigma}, \quad G_{\mathcal{D}}(D) = \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot D, \quad G(X) = \Omega + \Sigma \cdot X.$$

Then (8) lifts to

$$\begin{array}{c}
 \begin{array}{ccccc}
 & & \overline{P} & & \overline{U}_{\mathcal{D}} \\
 & & \curvearrowright & & \curvearrowright \\
 \text{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{\text{op}} & \xrightarrow{\text{Alg}_{\mathcal{D}}(G_{\mathcal{D}})} & \text{Alg}_{\text{Set}}(G) & \xrightarrow{\text{Alg}_{\text{Set}}(G)} & \text{Alg}_{\text{Set}}(G) \\
 & \swarrow \overline{S} & \downarrow \overline{\Phi}_{\mathcal{D}} & \searrow & \\
 & & \text{Alg}_{\mathcal{D}}(G_{\mathcal{D}}) & & 
 \end{array} \\
 \text{(9)}
 \end{array}$$

**Proof** The dual adjunction on the left lifts because of a special case of (3). For similar reasons, the adjunction on the right lifts, because there is a natural isomorphism  $\kappa : \Phi_{\mathcal{D}}G \xrightarrow{\cong} G_{\mathcal{D}}\Phi_{\mathcal{D}}$  that can be obtained as follows

$$\kappa : \Phi_{\mathcal{D}}GX = \Phi_{\mathcal{D}}(\Omega + \Sigma \cdot X) \cong \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot \Phi_{\mathcal{D}}(X) = G_{\mathcal{D}}\Phi_{\mathcal{D}}(X), \quad (10)$$

since  $\Phi_{\mathcal{D}}$  (being a left adjoint) preserves colimits. By [32, Thm. 2.14],  $\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}$  lifts to an adjunction  $\overline{\Phi}_{\mathcal{D}} \dashv \overline{U}_{\mathcal{D}}$  between  $\text{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$  and  $\text{Alg}_{\text{Set}}(G)$  where the functor  $\overline{\Phi}_{\mathcal{D}}$  maps a  $G$ -algebra  $(X, \alpha)$  to the  $G_{\mathcal{D}}$ -algebra  $(\Phi_{\mathcal{D}}(X), \Phi_{\mathcal{D}}\alpha \circ \kappa^{-1})$ .

By composition of adjunctions, also  $S\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}P$  lifts. This could also be verified by noticing that for all sets  $X$ , there is natural isomorphism

$$\xi^{\text{trc}} := S\kappa \circ \xi\Phi_{\mathcal{D}} : F_{\mathcal{C}}S\Phi_{\mathcal{D}} \xrightarrow{\cong} S\Phi_{\mathcal{D}}G \quad (11)$$

where  $\xi: F_{\mathcal{C}}S \xrightarrow{\cong} SG_{\mathcal{D}}$  from (3) is the mate of the modal logic  $(G_{\mathcal{D}}, \varrho)$ . Hence by [32, Thm. 2.14, Cor. 2.15] (see also [39, Thm. 2.5]), the adjunction  $S\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}P$  lifts to one between  $\text{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{\text{op}}$  and  $\text{Alg}_{\text{Set}}(G)$ .  $\square$

Letting  $\varrho^{\text{trc}}: GU_{\mathcal{D}}P \Rightarrow U_{\mathcal{D}}PF_{\mathcal{C}}$  be the mate of  $\xi^{\text{trc}}$  from (11), then  $(G, \varrho^{\text{trc}})$  is a modal logic for  $F_{\mathcal{C}}$ -coalgebras. Since its formulas are the elements of the initial  $G$ -algebra of traces, we refer to  $(G, \varrho^{\text{trc}})$  as a trace logic.

**Lemma 2** *The theory maps  $th^G$  and  $th^{G^{\mathcal{D}}}$  of the logics  $(G, \varrho^{\text{trc}})$  and  $(G_{\mathcal{D}}, \varrho)$  coincide.*

**Proof** Due to the adjunctions in (8), the initial  $G$ -algebra  $\Sigma^*\Omega$  of traces is mapped by  $\overline{\Phi_{\mathcal{D}}}$  to an initial  $G_{\mathcal{D}}$ -algebra, which in turn is mapped by  $\overline{S}$  to a final  $F_{\mathcal{D}}$ -coalgebra. The coincidence of the theory maps follows from them being adjoints of the initial maps.  $\square$

Since the mates  $\xi$  and  $\xi^{\text{trc}}$  are both natural isomorphisms, it follows from [35,40] (and  $\mathcal{C}$  having a suitable factorisation system, cf. Theorem 2) that the full modal logic  $(G_{\mathcal{D}}, \varrho)$  and trace logic  $(G, \varrho^{\text{trc}})$  are both expressive for  $F_{\mathcal{C}}$ -coalgebras. In other words, the propositional connectives from  $\mathcal{D}$ -structure in the logic language  $\mathbb{L}(G_{\mathcal{D}})$  do not add any expressive power to  $\mathbb{L}(G) = \Sigma^*\Omega$ . In summary, we arrive at the following proposition.

**Proposition 1** *With the above assumptions, the trace logic  $(G, \varrho^{\text{trc}})$  and the full logic  $(G_{\mathcal{D}}, \varrho)$  are equally expressive over  $F_{\mathcal{C}}$ -coalgebras, meaning that for all  $F_{\mathcal{C}}$ -coalgebras  $\gamma: C \rightarrow F_{\mathcal{C}}(C)$ , and all states  $c_1, c_2$  in  $C$  (recall that  $\mathcal{C}$  is a concrete category),  $c_1$  and  $c_2$  are logically equivalent for  $(G, \varrho^{\text{trc}})$  iff they are logically equivalent for  $(G_{\mathcal{D}}, \varrho)$ .*

By the uniqueness of final coalgebras up to isomorphism, it follows that there is an isomorphism  $\sigma: S\Phi_{\mathcal{D}}(\Omega)^{\Sigma^*} \xrightarrow{\cong} S\Phi_{\mathcal{D}}(\Sigma^*\Omega)$  which links the language semantics in the automata/coalgebraic sense with trace logic semantics given by initiality.

**Proposition 2** *For all  $F_{\mathcal{C}}$ -coalgebras  $\gamma$ , its language semantics defined as the unique morphism into the final  $F_{\mathcal{C}}$ -coalgebra  $S\Phi_{\mathcal{D}}(\Omega)^{\Sigma^*}$  corresponds to the trace theory map  $th^G$  into the final  $F_{\mathcal{C}}$ -coalgebra  $\overline{S\Phi_{\mathcal{D}}}(\Sigma^*\Omega)$ , (and with the theory map  $th^{G^{\mathcal{D}}}$ ) via the isomorphism  $\sigma$ .*

We remark that it is straightforward to extend  $\overline{\Phi_{\mathcal{D}}} \dashv \overline{U_{\mathcal{D}}}$  to an adjunction of automata by taking adjoints of additional output maps to the algebras. We omit the details.

Finally, we show that trace logic expressiveness can be extended to coalgebras for what we can think of as subfunctors of  $F_{\mathcal{C}}$ . This will be needed for the topological automata in section 4.3.

*Remark 2* Let  $F'_{\mathcal{C}}$  be a functor on  $\mathcal{C}$  which preserves monos and such that there is a natural transformation  $\tau: F'_{\mathcal{C}} \Rightarrow F_{\mathcal{C}}$  which is abstract mono, i.e., all components are mono. Assume that  $\mathcal{C}$  has factorisation system  $(E, M)$  with  $E \subseteq \text{Epi}$  and  $M \subseteq \text{Mono}$ . Defining  $\xi' = \xi^{\text{trc}} \circ \tau_S$ , then  $\xi': F'_{\mathcal{C}}S\Phi_{\mathcal{D}} \Rightarrow S\Phi_{\mathcal{D}}G$  defines semantics of trace formulas over  $F'_{\mathcal{C}}$ -coalgebras which is essentially the same as the semantics over  $F_{\mathcal{C}}$ -coalgebras. Since  $\tau$  is abstract mono and  $\xi^{\text{trc}}$  is a natural iso, it follows that  $\xi'$  is abstract mono, and hence the associated logic is expressive [35,40].

### 3.5 Reachability and Observability

Recall that a classic DFA is *reachable* if all states are reachable by reading some word from the initial state; it is *observable* if no two states accept the same language; and it is *minimal* if it is both reachable and observable.

A main point emphasised in [16] is that reachability is an algebraic concept, and observability is a coalgebraic concept. We will call an algebra *reachable* if it has no proper subalgebras, and a coalgebra is *observable* if it has no proper quotients. Both concepts apply to  $\mathcal{C}$ -automata as they are both coalgebras and algebras (cf. Section 3.2), and a  $\mathcal{C}$ -automaton is then minimal if its algebraic part is reachable, and its coalgebraic part is observable.

Both [13] and [16] show that a reachable algebra dualises to an observable coalgebra, but the conditions and arguments differ. Note that in [13], observable coalgebras are referred to as minimal automata. In [16], automata were generally considered as automata over  $\mathbf{Set}$ , and the reachable part of an automaton was defined as the image of the initial  $G$ -algebra inside the automaton (using its  $G$ -algebra structure, after possibly forgetting  $\mathcal{D}$ -structure). In [13], an initial  $G_{\mathcal{D}}$ -algebra generally did not exist. Instead, a reachable algebra was obtained by taking a least subalgebra, the existence of which was ensured by assuming that  $\mathcal{D}$  is wellpowered and having an epi-mono factorisation system. It is straightforward to show that when conditions for both are satisfied, the two reachability notions coincide, i.e., if an initial  $G_{\mathcal{D}}$ -algebra exists, and  $\mathcal{D}$  is wellpowered with epi-mono factorisation system, then the least subalgebra is obtained by factorisation of the initial morphism.

The assumptions in Lemma 1 are most closely related to the setup of [16], as we have an initial  $G_{\mathcal{D}}$ -algebra. The connection to the logical perspective of [13] comes from viewing the initial  $G_{\mathcal{D}}$ -algebra as a generalisation of the Lindenbaum algebra. For an  $F_{\mathcal{C}}$ -coalgebra  $(C, \gamma)$ , the factorisation of the initial morphism to  $\overline{P}(\gamma)$  then yields the subalgebra of  $\mathbb{L}(G_{\mathcal{D}})$ -definable subsets of  $C$  (or more abstractly,  $\mathbb{L}(G_{\mathcal{D}})$ -definable  $\Delta$ -valued predicates on  $C$ ). By Lemma 1,  $\Phi_{\mathcal{D}}(\Sigma^* \Omega)$  is also initial, and hence the reachable part of  $\overline{P}(\gamma)$  is equivalently characterised as the factorisation of the unique morphism from  $\Phi_{\mathcal{D}}(\Sigma^* \Omega)$ , and this factorisation is easily seen to be the subalgebra generated by the trace logic definable subsets.

Finally, by Lemma 1, a quotient of an initial  $G_{\mathcal{D}}$ -algebra is mapped by  $\overline{S}$  to a subobject of a final  $F_{\mathcal{C}}$ -coalgebra since the dual adjoint functors turn colimits into limits. A subcoalgebra of a final coalgebra is necessarily observable. The following proposition summarises our discussion.

**Proposition 3** *Under the assumptions of Lemma 1, and assuming further that  $\mathcal{D}$  has a factorisation system  $(E, M)$  such that  $E \subseteq \mathbf{Epi}$  and  $M \subseteq \mathbf{Mono}$ , we then have:*

*For all  $(D, \delta) \in \mathbf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$ , let  $\mathit{reach}(D, \delta)$  be the reachable part of  $(D, \delta)$  obtained by  $(E, M)$ -factorisation of the initial morphism:*

$$\overline{\Phi}_{\mathcal{D}}(\Sigma^* \Omega, \alpha) \xrightarrow{e} \mathit{reach}(D, \delta) \xrightarrow{m} (D, \delta).$$

*The epimorphism  $e: \overline{\Phi}_{\mathcal{D}}(\Sigma^* \Omega, \alpha) \rightarrow \mathit{reach}(D, \delta)$  is mapped by  $\overline{S}$  to a monomorphism*

$$\overline{S}(e): \overline{S}(\mathit{reach}(D, \delta)) \hookrightarrow \overline{S}\overline{\Phi}_{\mathcal{D}}(\Sigma^* \Omega, \alpha).$$

*As a subcoalgebra of a final coalgebra,  $\overline{S}(\mathit{reach}(D, \delta))$  is an observable  $F_{\mathcal{C}}$ -coalgebra.*

Note that  $\overline{S}$  maps epis to monos, but monos are not necessarily mapped to epis, unless we have a full duality. In particular, with only a dual adjunction we cannot argue that a least subalgebra of  $\overline{P}(\gamma)$  is mapped by  $\overline{S}$  to a largest quotient of  $\gamma$ , as in [13].

Extending the notion of reachable part to  $\mathcal{D}$ -automata is done simply by taking the reachable part of their  $G_{\mathcal{D}}$ -algebraic part and restricting the output map. Proposition 3 thus also tells us how to obtain an observable  $\mathcal{C}$ -automaton by taking the reachable part of the dual  $\mathcal{D}$ -automaton.

Brzozowski's algorithm produces a minimal  $\mathcal{C}$ -automaton by also taking the reachable part of the resulting observable  $\mathcal{C}$ -automaton, that is, with respect to the algebraic structure of  $\mathcal{C}$ -automata given by  $G_{\mathcal{C}} = I + \Sigma \cdot (-)$ . In order to do so, we need to assume that also  $\mathcal{C}$  has a suitable factorisation system.

### 3.6 Abstract minimisation algorithms

We now put everything together into one diagram with which we can describe both approaches from [13] and [16] including the role of trace logic.

$$\begin{array}{ccccc}
 (\text{Aut}_{\mathcal{C}}^{I, S\Phi_{\mathcal{D}}(\Omega)})^{\text{op}} & \xrightarrow{\bar{P}'} & \text{Aut}_{\mathcal{D}}^{\Phi_{\mathcal{D}}(\Omega), P(I)} & \xrightarrow{\bar{U}_{\mathcal{D}}'} & \text{Aut}_{\text{Set}}^{\Omega, U_{\mathcal{D}}P(I)} \\
 \downarrow & \begin{array}{c} \xleftarrow{\bar{S}'} \\ \top \\ \xrightarrow{\bar{P}} \end{array} & \downarrow & \begin{array}{c} \xleftarrow{\bar{\Phi}_{\mathcal{D}}'} \\ \top \\ \xrightarrow{\bar{U}_{\mathcal{D}}} \end{array} & \downarrow \\
 \text{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{\text{op}} & \xrightarrow{\bar{P}} & \text{Alg}_{\mathcal{D}}(G_{\mathcal{D}}) & \xrightarrow{\bar{U}_{\mathcal{D}}} & \text{Alg}_{\text{Set}}(G) \\
 \downarrow & \begin{array}{c} \xleftarrow{\bar{S}} \\ \top \\ \xrightarrow{P} \end{array} & \downarrow & \begin{array}{c} \xleftarrow{\bar{\Phi}_{\mathcal{D}}} \\ \top \\ \xrightarrow{U_{\mathcal{D}}} \end{array} & \downarrow \\
 F_{\mathcal{C}}^{\text{op}} \curvearrowright \mathcal{C}^{\text{op}} & \xrightarrow{P} & \mathcal{D} & \xrightarrow{U_{\mathcal{D}}} & \text{Set} \curvearrowright G \\
 & \begin{array}{c} \xleftarrow{S} \\ \top \\ \xrightarrow{G_{\mathcal{D}}} \end{array} & & \begin{array}{c} \xleftarrow{\Phi_{\mathcal{D}}} \\ \top \\ \xrightarrow{G} \end{array} & 
 \end{array} \tag{12}$$

$$F_{\mathcal{C}}(C) = S\Phi_{\mathcal{D}}(\Omega) \times C^{\Sigma}, \quad G_{\mathcal{D}}(D) = \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot D, \quad G(X) = \Omega + \Sigma \cdot X.$$

**Theorem 2** *Let  $\mathcal{C}, \mathcal{D}$  be concrete categories, both having products and coproducts, and both having factorisation systems  $(E, M)$  such that  $E \subseteq \text{Epi}$  and  $M \subseteq \text{Mono}$ . Let  $\Omega$  be a finite set (of observations), and  $I$  an (initialisation) object in  $\mathcal{C}$ , and assume that we have the adjoint situation between  $\mathcal{C}, \mathcal{D}, \text{Set}$  and functors as described at the bottom level of (12). Then the lower adjunctions lift to the upper two levels in (12) as shown in sections 3.3.2, 3.3.3 and 3.4, and we have the following abstract algorithms:*

**Algo1** *Given an  $F_{\mathcal{C}}$ -coalgebra  $\gamma$ , compute  $\bar{S}(\text{reach}(\bar{P}(\gamma)))$  which will be an observable  $F_{\mathcal{C}}$ -coalgebra.*

**Algo2** *Given a  $\mathcal{C}$ -automaton  $(\gamma, i)$ , compute  $\text{reach}(\bar{S}'(\text{reach}(\bar{P}'(\gamma, i))))$ , which will be a reachable and observable (i.e., minimal)  $\mathcal{C}$ -automaton.*

Of course, the abstract algorithms only become actual algorithms, when all structures involved have finite representations.

Concerning **Algo1**, we note that, in general,  $\bar{S}(\text{reach}(\bar{P}(\gamma)))$  can be much larger than  $\gamma$  as the application of both  $\bar{S}$  and  $\bar{P}$  might yield some kind of completion of  $\gamma$ . However, if  $\bar{P} \vdash \bar{S}$  is a dual equivalence, then  $\bar{S}(\text{reach}(\bar{P}(\gamma)))$  is a maximal quotient of  $\gamma$ . All instances of **Algo1** contained in [13] and in this paper are of this form. In the general case, following [50], one can obtain the maximal quotient of  $\gamma$  by factoring a morphism from  $\gamma$  to the result  $\bar{S}(\text{reach}(\bar{P}(\gamma)))$  of **Algo1**.

Also, when  $\bar{P} \vdash \bar{S}$  is a dual equivalence (as in [13]) the initial state is easily found back in the observable coalgebra resulting from **Algo1** as its language equivalence class, so the extension to **Algo2** seems almost trivial. In case  $\bar{P} \vdash \bar{S}$  is not a full duality, the transformation of the initial state goes via the dual adjunction, and factorisation on the dual side. This is formalised in Theorem 1, and illustrated by the example in Section 3.1.

Brzozowski's algorithm and its generalisation to weighted automata in Section 4.2 are instances of **Algo2** as they use initial states. The classic Brzozowski algorithm is the case  $\mathcal{C} = \mathcal{D} = \text{Set}$ ,  $G_{\mathcal{D}} = G$ , and  $\Omega = I = 1$ . The set-based algorithm for weighted automata in [16] is neither

of the above algorithms, but it can be described as constructing  $\text{reach}(\overline{U_{\mathcal{D}}'} \overline{P'}(\gamma, i))$ , that is, reachability is computed over  $\mathbf{Set}$ , and then dualise back (without going through  $\mathcal{D} = \mathbf{SMod}$ ) to get a  $\mathbf{Set}$ -based Moore automaton. As shown in [16], this may result in the reachable part of the reversed automaton being infinite (cf. Example 8.3 of [16]), whereas it might be finitely generated as a coalgebra/automaton over  $\mathcal{D}$ .

*Remark 3* We end this section by observing that the requirements regarding products, coproducts and factorisation systems hold in all our examples, since  $\mathcal{C}$  and  $\mathcal{D}$  are monadic over  $\mathbf{Set}$  meaning that they are equivalent to an Eilenberg-Moore category  $\mathbf{EM}(T)$  for a  $\mathbf{Set}$ -monad  $T$ . For such a category  $\mathbf{EM}(T)$ , we know that it is complete, cocomplete and exact [20, Thm 4.3.5]. W.r.t factorisation systems,  $(\mathit{Epi}, \mathit{Mono})$  is generally not a factorisation system for  $\mathbf{EM}(T)$ , rather  $(\mathit{RegEpi}, \mathit{Mono})$  is. Using the fact that regular epis are the surjective morphisms, and monos are the injective morphisms, one can prove that in  $\mathbf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$  and  $\mathbf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$  the surjective and injective morphisms form a factorisation system.

## 4 Revisiting Examples

### 4.1 Deterministic Kripke Models

A central example from [13] are deterministic Kripke models (in *loc.cit* referred to as PODFAs, i.e., partially observable DFAs). We will first recall the definitions of deterministic Kripke models and their dual Boolean algebras with operators corresponding to a modal logic of tests. After that we will see how this duality can be seen as a special case of our general duality picture, which has as immediate corollary a minimisation algorithm for the case of finite models. In addition, results from Section 3.4 entail that the modal test language without propositional operators is sufficiently expressive to specify deterministic Kripke models up to bisimulation and to compute their observable quotient.

For the remainder of the section we fix an arbitrary finite set  $\Sigma$  of *actions* and an arbitrary finite set  $\Omega$  of *observations*.

**Definition 2** A *deterministic Kripke model* is a quintuple  $\mathcal{S} = (S, \Sigma, \Omega, t : S \rightarrow S^{\Sigma}, f : S \rightarrow 2^{\Omega})$  where  $S$  is a set of *states*,  $t$  is a *transition function* and  $f$  is an *observation function*. A function  $h : S_1 \rightarrow S_2$  is a morphism between Kripke models  $(S_1, \Sigma, \Omega, t_1, f_1)$  and  $(S_2, \Sigma, \Omega, t_2, f_2)$  if for all  $s \in S_1$  and all  $a \in \Sigma$  we have  $h(t_1(s)(a)) = t_2(h(s))(a)$  and  $f_1(s) = f_2(s)$ . We write DKM for the category of deterministic Kripke models.

In other words, deterministic Kripke models are Kripke models where for each action  $a \in \Sigma$  the corresponding relation is the graph of a (total) function. It is well-known that there is a duality between DKM and a suitable category BAO of Boolean algebras. We will now recall the definition of BAO and some known facts concerning this duality.

**Definition 3** The category BAO of (deterministic) *Boolean algebras with operators* (BAOs) has as objects Boolean algebras  $B$  with the usual operations  $\wedge$  and  $\neg$  with a greatest element  $\top$  and least element  $\perp$  together with unary operators  $(a) : B \rightarrow B$ , for each action  $a \in \Sigma$ , such that  $(a)$  is a Boolean homomorphism. For each observation  $o \in \Omega$ , we also have constants  $\underline{o}$ . We denote an object of BAO by

$$\mathcal{B} = (B, \{(a) | a \in \Sigma\}, \{\underline{o} | o \in \Omega\}, \top, \wedge, \neg).$$

The BAO morphisms are the usual Boolean homomorphisms preserving, in addition, the constants and commuting with the unary operators. Finally, we denote by FBAO the category of *finite* Boolean algebras with operators.

The following fact is well-known (cf. e.g. [15, 31]).

**Fact** There is a dual adjunction between  $\mathbf{Set}$  and  $\mathbf{BA}$  as depicted below given by the contravariant functor  $\mathbb{P}$  that maps a set to its Boolean algebra of subsets and the functor  $Uf := \text{Hom}(-, \mathbf{2})$ , i.e., the contravariant functor that maps a Boolean algebra to its collection of ultrafilters. This adjunction restricts to a dual equivalence between the category  $\mathbf{FSet}$  of finite sets and the category  $\mathbf{FBA}$  of finite Boolean algebras.

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \text{Set}^{\text{op}} & \begin{array}{c} \xrightarrow{F^{\text{op}}} \\ \xrightarrow{\mathbb{P}} \\ \xleftarrow{Uf} \end{array} & \text{BA} \\
 & \begin{array}{c} \xleftarrow{\top} \\ \xrightarrow{\text{Uf}} \end{array} & \\
 & & \text{BA}
 \end{array}
 \end{array}
 \quad
 \begin{array}{l}
 F(X) = 2^\Omega \times X^\Sigma \\
 G_{\mathbf{BA}}(X) = \Phi_{\mathbf{BA}}(\Omega) + \Sigma \cdot X \\
 Uf(\Phi_{\mathbf{BA}}(\Omega)) \cong 2^\Omega
 \end{array}$$

We are now going to show how this example fits into our general framework. As a corollary we obtain a minimisation procedure for finite deterministic Kripke models.

**Proposition 4** *We have the following equivalences:*

1.  $\text{DKM} \cong \text{Coalg}_{\mathbf{Set}}(F)$  for  $F = 2^\Omega \times X^\Sigma$
2.  $\text{BAO} \cong \text{Alg}_{\mathbf{BA}}(G_{\mathbf{BA}})$  for  $G_{\mathbf{BA}} = \Phi_{\mathbf{BA}}(\Omega) + \Sigma \cdot X$

Both equivalences are an immediate consequence of the definitions. In the sequel, we will make no distinction between  $F$ -coalgebras and deterministic Kripke models and, likewise, between  $G_{\mathbf{BA}}$ -algebras and BAOs. As a consequence of the proposition we obtain the following duality results by applying our general framework.

**Proposition 5** *The dual adjunction  $Uf \dashv \mathbb{P}$  lifts to a dual adjunction between  $\text{DKM}$  and  $\text{BAO}$  and to an adjunction between  $\text{Aut}_{\mathbf{Set}}^{1,2^\Omega}$  and  $\text{Aut}_{\mathbf{BA}}^{2, \Phi_{\mathbf{BA}}(\Omega)}$ . If we start with the dual equivalence  $\mathbf{FSet} \cong \mathbf{FBA}$ , both liftings are dual equivalences as well.*

**Proof** For the dual adjunction between  $\text{DKM}$  and  $\text{BAO}$  recall from Proposition 4 that both categories are equivalent to categories of  $F$ -coalgebras and  $G_{\mathbf{BA}}$ -algebras for certain functors  $F$  and  $G_{\mathbf{BA}}$ , respectively. Furthermore, we have  $Uf(\Phi_{\mathbf{BA}}(\Omega)) \cong 2^\Omega$ , which follows from the well-known fact that the set of homomorphisms of type  $\Phi_{\mathbf{BA}}(\Omega) \rightarrow \mathbf{2}$  (i.e., ultrafilters) is in one-one correspondence with the set of functions of type  $\Omega \rightarrow 2$ . Therefore the functors  $F$  and  $G_{\mathbf{BA}}$  have the shape required by our general lifting result from Section 3.3.2 and we obtain functors  $\overline{\mathbb{P}} : \text{Coalg}(F)^{\text{op}} \rightarrow \text{Alg}(G_{\mathbf{BA}})$  and  $\overline{Uf} : \text{Alg}(G_{\mathbf{BA}}) \rightarrow \text{Coalg}(F)^{\text{op}}$  with  $\overline{Uf} \dashv \overline{\mathbb{P}}$ .

To extend the adjunction  $\overline{Uf} \dashv \overline{\mathbb{P}}$  between  $\text{Coalg}(F)$  and  $\text{Alg}(G_{\mathbf{BA}})$  further to a dual adjunction  $\overline{Uf}' \dashv \overline{\mathbb{P}}'$  between  $\text{Aut}_{\mathbf{Set}}^{1,2^\Omega}$  and  $\text{Aut}_{\mathbf{BA}}^{\Phi_{\mathbf{BA}}(\Omega), 2}$  — the latter is a slight extension of the former by adding a initial state to deterministic Kripke models and adding an acceptance predicate to BAOs — it suffices to note that  $\mathbb{P}1 \cong 2$  such that the result follows from the general theorem in Section 3.3.3.

The fact that the obtained adjunctions restrict to equivalences when we replace the base categories  $\mathbf{Set}$  and  $\mathbf{BA}$  with  $\mathbf{FSet}$  and  $\mathbf{FBA}$ , respectively, is a matter of routine checking.  $\square$

This shows, in particular, that we get a duality between finite deterministic Kripke models and FBAO's. Proposition 5 is the key for obtaining a minimal realization via logical theories. Towards obtaining observable coalgebras via logical theories, we define a modal logic for DKMs.

**Definition 4** Consider (cf. (6)) the language  $\mathbb{L}(G_{\text{BA}})$ :

$$\varphi ::= \top \mid \hat{o}, o \in \Omega \mid [a]\varphi, a \in \Sigma \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi.$$

with 2-valued semantics defined as in (7) which corresponds to the usual semantics by identifying the predicate  $\llbracket \varphi \rrbracket : S \rightarrow 2$  with the set  $\{s \in S \mid \llbracket \varphi \rrbracket(s) = 1\}$ . For a given deterministic Kripke model  $\mathcal{S} = (S, t, f)$  we say that a subset  $U$  of  $S$  is *definable by*  $\mathbb{L}(G_{\text{BA}})$  if  $U = \llbracket \varphi \rrbracket$  for some  $\varphi \in \mathbb{L}(G_{\text{BA}})$ . We let  $\text{Def}(\mathcal{S}) = (\text{Def}(S), \{(a)_S\}_{a \in \Sigma}, \{\llbracket \hat{o} \rrbracket\}_{o \in \Omega})$  be the BAO-subalgebra of  $\overline{\mathbb{P}}(\mathcal{S})$  based on the definable subsets of  $\mathcal{S}$ , where  $(a)_S(\llbracket \varphi \rrbracket) = \llbracket [a]\varphi \rrbracket$ .

In other words, the modal logic has (deterministic)  $\Sigma$ -indexed modalities and atomic propositions from  $\Omega$ . It might seem strange that we introduced negation and indeed one does not need it, because one can get exactly the same Boolean algebra without having negation explicitly in the logic. This reflects the fact that for deterministic systems, unlike for non-deterministic ones, bisimulation and language equivalence coincide.

For a given DKM  $\mathcal{S}$ , the algebra  $\text{Def}(\mathcal{S})$  of definable subsets is a least BAO-subalgebra (=zero generated subalgebra) of the dual BAO  $\overline{\mathbb{P}}(\mathcal{S})$ . By our definition in Section 3.5,  $\text{Def}(\mathcal{S})$  is clearly reachable, since it has no proper BAO-subalgebras. We instantiate the discussion in Section 3.5 further for DKMs. As is well-known, the Lindenbaum algebra for  $\mathbb{L}(G_{\text{BA}})$  is an initial BAO, and the subalgebra  $\text{Def}(\mathcal{S})$  is also the image of the Lindenbaum algebra in  $\overline{\mathbb{P}}(\mathcal{S})$  under the initial BAO-morphism. This image is obtained from the factorisation of the initial morphism in the factorisation system consisting of surjective and injective BAO-homomorphisms (cf. Remark 3).

Central to [13] was the result that the fragment of trace logic formulas is as expressive as the full modal logic.

**Definition 5** A *trace logic formula* is a formula  $\varphi$  of the form  $[a_1] \dots [a_n]\hat{o}$  for some  $\hat{o} \in \Omega$ ,  $n \in \mathbb{N}$  and  $a_i \in \Sigma$  for  $i \in \{1, \dots, n\}$ . For a DKM  $\mathcal{S}$ , we denote by  $\text{Def}^*(\mathcal{S})$  the Boolean subalgebra of  $\mathbb{P}(\mathcal{S})$  generated by subsets that are definable by a trace formula.

The expressiveness of trace logic was proven in the general setting in Proposition 1, and it is equivalent to the following statement:

$$\text{Def}(\mathcal{S}) = \text{Def}^*(\mathcal{S}) \quad \text{for all DKMs } \mathcal{S}. \quad (1)$$

Note that equation (1) can be seen as a normal form theorem for modal logics over DKMs. This result can also be obtained by observing that  $\text{Def}^*(\mathcal{S})$  is the image of  $\overline{\Phi}_{\text{BA}}(\Sigma^* \Omega)$ , which is an initial BAO by Lemma 1. So both  $\text{Def}(\mathcal{S})$  and  $\text{Def}^*(\mathcal{S})$  are images of an initial object inside  $\overline{\mathbb{P}}(\mathcal{S})$ , hence they must be equal, since initial objects are unique up to isomorphism.

We finish with a key observation from [13] that allows to compute quotients of finite DKMs via duality.

**Corollary 1** *Given a finite DKM  $\mathcal{S}$ , the quotient of  $\mathcal{S}$  modulo bisimulation can be effectively computed as  $\overline{Uf}(\text{Def}^*(\mathcal{S}))$ .*

**Proof** By **Algo1** in Theorem 2 we have that  $\overline{Uf}(\text{reach}(\overline{\mathbb{P}}(\mathcal{S})))$  and thus  $\overline{Uf}(\text{Def}^*(\mathcal{S}))$  are observable. As  $\text{Def}^*(\mathcal{S})$  is a BAO-subalgebra of  $\overline{\mathbb{P}}(\mathcal{S})$  and as **FSet** and **FBA** are dually *equivalent*, we get that  $\overline{Uf}(\text{Def}^*(\mathcal{S}))$  is a quotient of  $\overline{Uf}(\overline{\mathbb{P}}(\mathcal{S})) \cong \mathcal{S}$ . Therefore, as  $\overline{Uf}(\text{Def}^*(\mathcal{S}))$  is an observable quotient of  $\mathcal{S}$ , it is the quotient modulo bisimulation of  $\mathcal{S}$ .  $\square$

**Remark 4** We note that the full duality between finite DKMs and finite BAOs, which was the basis of the minimisation-via-duality in [13], is not an instance of Theorem 2, since the category of finite Boolean algebras is not monadic over **Set**. **Algo1**, of course, applies to finite DKMs as



they are just DKMs, and it will produce the same result as minimisation-via-duality from [13], since the full duality between finite sets and finite Boolean algebras is a restriction of the dual adjunction between **Set** and **BA**.

## 4.2 Weighted Automata

We need some basic definitions on semirings and semimodules to present the example of weighted automata. Recall that a *semiring* is a tuple  $(\mathbb{S}, +, \cdot, 0, 1)$  where  $(\mathbb{S}, +, 0)$  and  $(\mathbb{S}, \cdot, 1)$  are monoids, the former of which is commutative, and multiplication distributes over finite sums:

$$r \cdot 0 = 0 = 0 \cdot r \quad r \cdot (s + t) = r \cdot s + r \cdot t \quad (r + s) \cdot t = r \cdot t + s \cdot t$$

We just write  $\mathbb{S}$  to denote a semiring. Examples of semirings are: every field, the Boolean semiring  $2$ , the semiring  $(\mathbb{N}, +, \cdot, 0, 1)$  of natural numbers, and the tropical semiring  $(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$ . All these semirings are examples of *commutative* semirings, as the  $\cdot$  operation is also commutative.

For a semiring  $\mathbb{S}$ , an  $\mathbb{S}$ -*semimodule* is a commutative monoid  $(M, +, 0)$  with a left-action  $\mathbb{S} \times M \rightarrow M$  denoted by juxtaposition  $rm$  for  $r \in \mathbb{S}$  and  $m \in M$ , such that for every  $r, s \in \mathbb{S}$  and every  $m, n \in M$  the following laws hold:

$$\begin{aligned} (r + s)m &= rm + sm & r(m + n) &= rm + rn \\ 0m &= 0 & r0 &= 0 \\ 1m &= m & r(sm) &= (r \cdot s)m \end{aligned}$$

Every semiring  $\mathbb{S}$  is an  $\mathbb{S}$ -semimodule, where the action is taken to be just the semiring multiplication. Semilattices are another example of semimodules (for the Boolean semiring  $\mathbb{S}$ ).

An  $\mathbb{S}$ -semimodule homomorphism is a monoid homomorphism  $h: M_1 \rightarrow M_2$  such that  $h(rm) = rh(m)$  for each  $r \in \mathbb{S}$  and  $m \in M_1$ .  $\mathbb{S}$ -semimodule homomorphisms are also called  $\mathbb{S}$ -*linear maps* or simply *linear maps*. The set of all linear maps from an  $\mathbb{S}$ -semimodule  $M_1$  to  $M_2$  is denoted by  $\mathbb{S}\text{Mod}(M_1, M_2)$ .

*Free  $\mathbb{S}$ -semimodules* over a set  $X$  exist and can be built using the functor  $\mathcal{V}_{\mathbb{S}}: \mathbf{Set} \rightarrow \mathbf{Set}$  defined on sets  $X$  and maps  $h: X \rightarrow Y$  as follows:

$$\begin{aligned} \mathcal{V}_{\mathbb{S}}(X) &= \{ \varphi: X \rightarrow \mathbb{S} \mid \varphi \text{ has finite support} \}, \\ \mathcal{V}_{\mathbb{S}}(h)(\varphi) &= (y \mapsto \sum_{x \in h^{-1}(y)} \varphi(x)), \end{aligned}$$

where a function  $\varphi: X \rightarrow \mathbb{S}$  is said to have finite support if  $\varphi(x) \neq 0$  holds only for finitely many elements  $x \in X$ .  $\mathcal{V}_{\mathbb{S}}(X)$  is the free  $\mathbb{S}$ -semimodule on  $X$  when equipped with the following pointwise  $\mathbb{S}$ -semimodule structure:

$$(\varphi_1 + \varphi_2)(x) = \varphi_1(x) + \varphi_2(x) \quad (s\varphi_1)(x) = s \cdot \varphi_1(x).$$

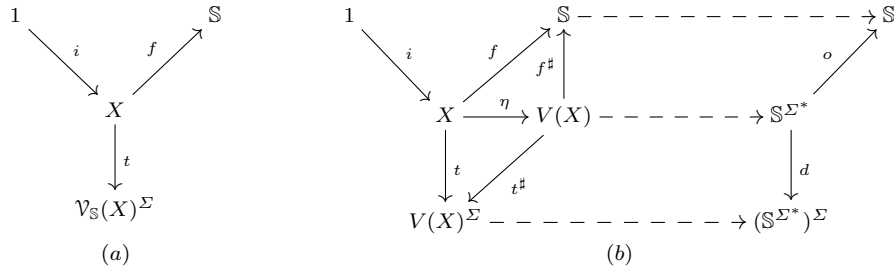
We sometimes write the elements of  $\mathcal{V}_{\mathbb{S}}(X)$  as formal sums  $s_1x_1 + \dots + s_nx_n$  with  $s_i \in \mathbb{S}$  and  $x_i \in X$ .  $\mathcal{V}_{\mathbb{S}}(X)$  is a monad and the category of Eilenberg-Moore algebras is  $\mathbb{S}\text{Mod}$ , the category of  $\mathbb{S}$ -semimodules and  $\mathbb{S}$ -linear maps. As usual, free  $\mathbb{S}$ -semimodules enjoy the following universal property: for every function  $h: X \rightarrow M$  from a set  $X$  to a semimodule  $M$ , there exists a unique linear map  $h^\sharp: \mathcal{V}_{\mathbb{S}}(X) \rightarrow M$  that is called the *linear extension* of  $h$ .

We can define for an  $\mathbb{S}$ -semimodule  $M$  its *dual space*  $M^*$  to be the set  $\mathbb{S}\text{Mod}(M, \mathbb{S})$  of all linear maps between  $M$  and  $\mathbb{S}$ , endowed with the  $\mathbb{S}$ -semimodule structure obtained by taking pointwise

addition and monoidal action:  $(g + h)(m) = g(m) + h(m)$ , and  $(sh)(m) = s \cdot h(m)$ . Note that  $\mathbb{S} \cong \mathcal{V}_{\mathbb{S}}(1)$  and that  $\mathbb{S}^* = \mathbb{S}\text{Mod}(\mathbb{S}, \mathbb{S}) \cong \mathbb{S}$ .

#### 4.2.1 Weighted automata and weighted languages

A *weighted automaton* with finite input alphabet  $\Sigma$  and weights over a semiring  $\mathbb{S}$  is given by a set of states  $X$ , a function  $t: X \rightarrow \mathcal{V}_{\mathbb{S}}(X)^{\Sigma}$  (encoding the transition relation in the following way: the state  $x \in X$  can make a transition to  $y \in X$  with input  $a \in \Sigma$  and weight  $s \in \mathbb{S}$  if and only if  $t(x)(a)(y) = s$ ), a final state function  $f: X \rightarrow \mathbb{S}$  associating an output weight with every state, and an initial state function  $i: 1 \rightarrow \mathcal{V}_{\mathbb{S}}(X)$ . A diagrammatic representation is given in Figure 4.2.1(a).



**Fig. 1** (a) Weighted automata as  $\text{Set}$ -automata, and (b) their determinisation as  $\mathbb{S}\text{Mod}$ -automata.

We see that a weighted automaton is an  $F\mathcal{V}_{\mathbb{S}}$ -coalgebra  $\langle f, t \rangle: X \rightarrow \mathbb{S} \times (\mathcal{V}_{\mathbb{S}}X)^{\Sigma}$ , where  $F: \text{Set} \rightarrow \text{Set}$  is given by  $F(X) = \mathbb{S} \times X^{\Sigma}$ , together with an initialisation map  $i: 1 \rightarrow X$ .

The function  $t: X \rightarrow \mathcal{V}_{\mathbb{S}}(X)^{\Sigma}$  can be inductively extended to words  $w \in \Sigma^*$ :

$$t(x)(\varepsilon) = 1.x$$

$$t(x)(aw) = v_1 t(x_1)(w) + \dots + v_n t(x_n)(w), \text{ where } t(x)(a) = v_1 x_1 + \dots + v_n x_n$$

Weighted automata recognise functions in  $\mathbb{S}^{\Sigma^*}$ , or *formal power series over  $\mathbb{S}$* . More precisely, the formal power series recognised by a weighted automaton  $\mathcal{X} = (X, t, i, f)$  is the function  $\mathcal{L}(\mathcal{X}): \Sigma^* \rightarrow \mathbb{S}$  that maps  $w \in \Sigma^*$  to  $f(t(i)(w)) \in \mathbb{S}$ . More concretely, the value  $\mathcal{L}(\mathcal{X})(w)$ , for  $w = a_1 a_2 \dots a_n$ , is the sum of all  $v_1 \dots v_n \cdot f(x_{n+1})$  over all paths  $p_w = x_1 \xrightarrow{a_1, v_1} \dots \xrightarrow{a_n, v_n} x_{n+1}$  labelled by  $w$ .

Observe that  $\mathbb{S}$  is (isomorphic to) the carrier of the free Eilenberg-Moore  $\mathcal{V}_{\mathbb{S}}$ -algebra on one generator  $\mathcal{V}_{\mathbb{S}}(1)$ . Hence, as described in Section 2.2, we can determinise a weighted automaton  $\mathcal{X}$  into a Moore automaton  $\mathcal{X}^{\#}$  in  $\mathbb{S}\text{Mod}$ . More precisely, letting  $\mathcal{X} = (X, t, i, f)$  be a weighted automaton, we determinise its coalgebraic part  $\langle f, t \rangle: X \rightarrow \mathbb{S} \times \mathcal{V}_{\mathbb{S}}(X)^{\Sigma}$  into  $\langle f^{\#}, t^{\#} \rangle: \mathcal{V}_{\mathbb{S}}(X) \rightarrow \mathbb{S} \times \mathcal{V}_{\mathbb{S}}(X)^{\Sigma}$  and take as initialisation morphism  $\mathcal{V}_{\mathbb{S}}(i): \mathcal{V}_{\mathbb{S}}(1) \rightarrow \mathcal{V}_{\mathbb{S}}(X)$ . The result is an  $\mathbb{S}\text{Mod}$ -automaton  $\mathcal{X}^{\#} = (\mathcal{V}_{\mathbb{S}}(X), t^{\#}, \mathcal{V}_{\mathbb{S}}(i), f^{\#})$  with initialisation in  $\mathbb{S} \cong \mathcal{V}_{\mathbb{S}}(1)$  and output in  $\mathbb{S}$ . We view such automata as Moore automata over  $\mathbb{S}\text{Mod}$ . The construction is illustrated in Figure 4.2.1(b).

The unique map from the determinised Moore automaton into the final Moore automaton of weighted languages gives the language semantics of weighted automata described concretely above. In  $\mathbb{S}\text{Mod}$ , the value of  $\mathcal{L}(\mathcal{X})(w)$  can be computed using the usual matrix representation of linear maps: the initialisation morphism  $\mathcal{V}_{\mathbb{S}}(i)$  corresponds to a column vector  $\eta \circ i: \mathcal{V}_{\mathbb{S}}(1) \rightarrow \mathcal{V}_{\mathbb{S}}(X)$ ,

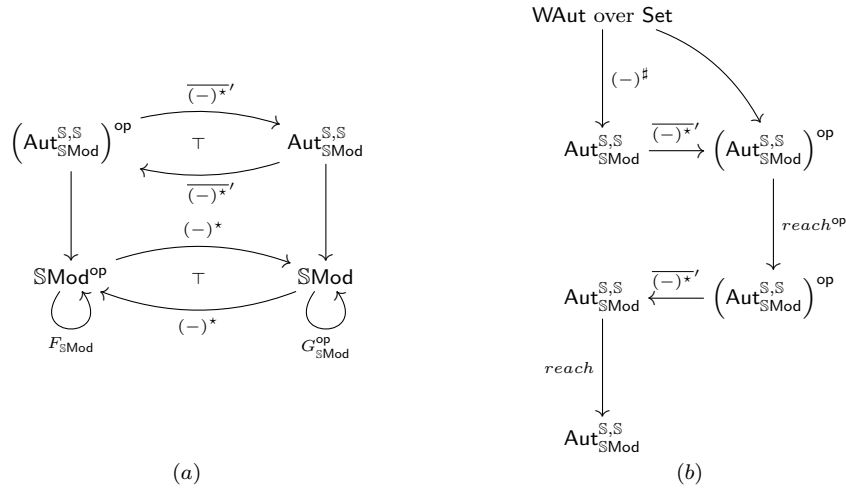
the output morphism  $f^\sharp$  is a row vector, and the transition morphism  $t^\sharp$  can be represented as a  $\Sigma$ -indexed collection of  $X \times X$ -matrices  $t_a$  where  $t_a(y, x) = t(x)(a)(y)$  for all  $x, y \in X$ .  $\mathcal{L}(\mathcal{X})(w)$  is then obtained by the following matrix multiplication  $f \times t_{a_n} \times \dots \times t_{a_0} \times i$ .

#### 4.2.2 Brozowski's Algorithm for Weighted Automata

There is self-dual adjunction of  $\mathbb{S}\text{Mod}$  obtained by taking dual space:  $(-)^* = \mathbb{S}\text{Mod}(-, \mathbb{S})$ . A special case is the self-dual adjunction of vector spaces in case  $\mathbb{S}$  is a field, which restricts to a duality between finite-dimensional vector spaces. This duality was used in [13] to obtain observable Moore automata over vector spaces.

We lift the base adjunction to one between Moore automata in  $\mathbb{S}\text{Mod}$  using Theorem 1. Let  $\mathcal{C} = \mathcal{D} = \mathbb{S}\text{Mod} = \text{EM}(\mathcal{V}_{\mathbb{S}})$  and  $F_{\mathbb{S}\text{Mod}}(X) = \mathbb{S} \times X^\Sigma$  and  $G_{\mathbb{S}\text{Mod}}(X) = \mathbb{S} + \Sigma \cdot X$ . Since  $\mathbb{S}^* \cong \mathbb{S}$ , the conditions for Theorem 1 hold, and the adjunction lifts, as illustrated in Figure 2(a).

We can now give the Brzozowski algorithm for weighted automata by instantiating **Algo2** of Theorem 2 for the determinised automaton. Start with a weighted automaton in  $\text{Set}$ , determinise it into a Moore automaton in  $\text{Aut}_{\mathbb{S}\text{Mod}}^{\mathbb{S}, \mathbb{S}}$  (to have a canonical representative of the accepted language), reverse and determinise, take the reachable part (w.r.t  $G_{\mathbb{S}\text{Mod}}$ -structure over  $\mathbb{S}\text{Mod}$ ), reverse and determinise, take the reachable part again. Diagrammatically, **Algo2** is (putting  $\text{op}$  on the right-hand side to start and end in  $\text{Aut}_{\mathbb{S}\text{Mod}}^{\mathbb{S}, \mathbb{S}}$ ) shown in Figure 2(b).



**Fig. 2** (a) The dual adjunction for weighted automata, and (b) Brzozowski's algorithm for weighted automata.

At this point we have built a minimal Moore automaton  $\min(\mathcal{X}^\sharp)$  over  $\mathbb{S}\text{Mod}$  accepting the same language as the weighted automaton  $\mathcal{X}$  we started with and, moreover, the state space is a subsemimodule of the semimodule generated by the original state space.

The last step missing is to recover a weighted automaton over  $\text{Set}$  with a state space  $Y$  such that  $\mathcal{V}_{\mathbb{S}}(Y)$  is the state space of  $\min(\mathcal{X}^\sharp)$ . Unfortunately, subsemimodules of free, finitely generated semimodules are not necessarily free and finitely generated. Therefore our construction does not guarantee, in general, that the resulting automaton  $\min(\mathcal{X}^\sharp)$  corresponds to a weighted automaton in  $\text{Set}$ . Fortunately, we know from a result of Tan [59] that for a commutative semiring  $\mathbb{S}$ , every nonzero subsemimodule  $N$  of a finitely generated free  $\mathbb{S}$ -semimodule  $M$  is free if and

only if  $\mathbb{S}$  is a principal ideal domain [59, Theorem 4.3]. Furthermore, because  $N$  is free, it follows that it is also finitely generated and of rank smaller than that of  $M$  [59, Theorem 4.3]. In other words, the minimal weighted automaton over a principal ideal domain exists and has a state space smaller or equal than that of the original automaton if the latter is finite.

Recall that a principal ideal domain is an integral domain in which every ideal is principal, i.e., can be generated by a single element. Examples include any Euclidean domain, thus any field, the ring of integers, the ring of polynomials in one variable with coefficients in a field, and the ring of formal power series over a field and one variable. The ring of polynomials in two or more variables and the ring of polynomials with integer coefficients are not principal ideal domains.

### 4.2.3 Logic for weighted automata

The  $\mathbb{S}$ -valued logic (cf. (6)) corresponding to the functor  $G_{\mathbb{S}\text{Mod}}(X) = \mathcal{V}_{\mathbb{S}}(1) + \Sigma \cdot X$  (recall that  $\mathbb{S} \cong \mathcal{V}_{\mathbb{S}}(1)$ , i.e.,  $\Omega = 1$ ), has formulas generated by the following grammar:

$$\varphi ::= \downarrow \mid [a]\varphi, a \in \Sigma \mid 0 \mid s \cdot \varphi, s \in \mathbb{S} \mid \varphi + \varphi$$

where  $\downarrow$  is a single atomic proposition (denoting termination) and the linear propositional connectives are interpreted via semimodule structure. Trace logic is the fragment that is built only from  $\downarrow$  and modalities. The results from Section 3.4 tell us that trace logic is already expressive for  $\mathbb{S}\text{Mod}$ -automata.

## 4.3 Topological Automata via Gelfand Duality

A very popular model heavily used in reinforcement learning is the *partially observable Markov decision process* (POMDP). The idea is that one can only see the observations and not exactly which state the system is in. Many algorithms in machine learning deal with this situation by constructing a new automaton called the *belief automaton*. The state space of this automaton is the set of probability distributions over the original states. When seeking to minimize this using duality [13], the original idea was to exploit the fact that the state space of the belief automaton is a compact Hausdorff space and use Gelfand duality. However, we have since felt that convex duality is a better match for this situation. Nevertheless, the notion of a topological automaton is interesting in its own right and may be the basis for later extensions and examples. This section, therefore develops Gelfand duality and its application to topological automata.

Given a finite set  $X$ , we write  $\mathcal{D}_{\leq 1}(X)$  for the set of discrete *subdistributions* on  $X$  endowed with the relative topology when viewed as a subset of  $[0, 1]^X$ . This is a compact Hausdorff space. As in other sections, we fix a finite set  $\Sigma$  of actions or inputs, and a finite set  $\Omega$  of observations.

**Definition 6** A **compact Hausdorff automaton** is a 5-tuple

$$\mathcal{H} = (S, t : S \rightarrow S^{\Sigma}, f : S \rightarrow \mathcal{D}_{\leq 1}(\Omega), i : S)$$

where  $S$  is a compact Hausdorff space,  $t$  is a *continuous* transition function (to the product space  $S^{\Sigma}$ ),  $f$  is a *continuous* observation function, and  $i : S$  is an initial state.

A compact Hausdorff automaton is easily seen to be a coalgebra for the functor  $F : \text{KHaus} \rightarrow \text{KHaus}$  given by  $F(X) = \mathcal{D}_{\leq 1}(\Omega) \times X^{\Sigma}$  together with an initialisation morphism  $i : 1 \rightarrow S$  where  $1$  is the discrete, one-element space. Since Gelfand duality is a full duality, the initial state plays

only a minor role as mentioned in Section 3.6. In [13], compact Hausdorff automata were defined without initial state.

We recall a few basic facts about  $C^*$ -algebras, and refer to [10, 36] for further information. Usually  $C^*$ -algebras are considered as algebras over the complex field. Here, we are concerned with probabilistic computation, and therefore we consider  $C^*$ -algebras over the field  $\mathbb{R}$  of reals.

A (*real-valued*) *Banach algebra*  $A$  is Banach space (complete normed real vector space) equipped with an associative multiplication such that  $\|xy\| \leq \|x\|\|y\|$  for all  $x, y$ . This requirement makes multiplication continuous in the norm topology. A (*real*)  $C^*$ -algebra is a Banach algebra together with an *involution*  $(-)^*$  which is a linear, norm-preserving map on  $A$  such that  $(xy)^* = y^*x^*$  and  $(x^*)^* = x$ , and which in addition satisfies the  $C^*$ -axiom:  $\|x^*x\| = \|x\|^2$  for all  $x \in A$ . A  $C^*$ -algebra  $A$  is *unital* if it has a multiplicative unit  $1$  whose norm is  $1 \in \mathbb{R}$ , and  $A$  is *commutative* if the multiplication is commutative.

A *homomorphism of  $C^*$ -algebras* is a bounded, linear map that preserves the multiplication and the involution. A homomorphism of unital  $C^*$ -algebras is additionally required to preserve the unit. We denote by  $\text{CUC}^*\text{Alg}$  the category of unital, commutative, real-valued  $C^*$ -algebras and their homomorphisms.

In [48] it was shown<sup>1</sup> that  $U$  has a left adjoint  $M: \text{Set} \rightarrow \text{CUC}^*\text{Alg}$  given by

$$M(X) = C([0, 1]^X) = \{f: [0, 1]^X \rightarrow \mathbb{R} \mid f \text{ continuous}\} \quad (2)$$

$$M(g: X \rightarrow Y) = f(v \circ g) \quad \text{where } f \in C([0, 1]^X), v \in [0, 1]^Y. \quad (3)$$

where  $[0, 1]^X$  is equipped with the product topology.

We denote by  $\text{KHaus}$  the category of compact Hausdorff spaces and continuous maps. Given a compact Hausdorff space  $X$ , the hom-set  $C(X) = \text{Hom}_{\text{KHaus}}(X, \mathbb{R})$  becomes a commutative, unital, real-valued  $C^*$ -algebra by defining operations pointwise. In particular, the unit is the constantly 1 map, and for  $f \in C(X)$ , the norm is  $\|f\| = \sup\{|f(x)| \mid x \in X\}$ ; recall that for a compact space and a continuous function the supremum is attained. For a morphism  $g: X \rightarrow Y$  in  $\text{KHaus}$ , defining  $C(g)(h) = h \circ g$  makes  $C(-)$  a functor from  $\text{KHaus}$  to  $\text{CUC}^*\text{Alg}^{\text{op}}$ .

Conversely, for  $A \in \text{CUC}^*\text{Alg}$ , the set  $\hat{A} = \text{Hom}_{\text{CUC}^*\text{Alg}}(A, \mathbb{R})$  becomes a compact Hausdorff space (called the *spectrum of  $A$* ) by equipping it with the weak  $*$ -topology  $\tau$  which is generated by the sets  $O_x = \{\Phi \in \hat{A} \mid \Phi(x) \neq 0\}$  for all  $x \in A$ . We define  $\text{Spec}(A) = (\hat{A}, \tau)$ . For a morphism  $h: A \rightarrow B$  in  $\text{CUC}^*\text{Alg}$ , defining  $\text{Spec}(h)(\Phi) = \Phi \circ h$  makes  $\text{Spec}$  a functor from  $\text{CUC}^*\text{Alg}^{\text{op}}$  to  $\text{KHaus}$ .

The functors  $C$  and  $\text{Spec}$  establish a dual equivalence between  $\text{KHaus}$  and  $\text{CUC}^*\text{Alg}$  known as Gelfand duality

$$\begin{array}{ccc} & C & \\ & \curvearrowright & \\ \text{KHaus}^{\text{op}} & \cong & \text{CUC}^*\text{Alg} \\ & \curvearrowleft & \\ & \text{Spec} & \end{array} \quad (4)$$

For the purposes of this paper, we only need a dual adjunction. We will take  $C$  to be the right adjoint. As this dual adjunction is in fact a dual equivalence, the unit and the counit of this adjunction are natural isomorphisms. The unit  $\eta_A: A \rightarrow C(\text{Spec}(A))$  is known as the *Gelfand transform*, and is given by  $\eta_A(x)(\Phi) = \Phi(x)$ . For all  $A \in \text{CUC}^*\text{Alg}$ ,  $\eta_A$  is an isometric isomorphism in  $\text{CUC}^*\text{Alg}$ .

We first lift the base dual adjunction between  $\text{KHaus}$  and  $\text{CUC}^*\text{Alg}$  to an a dual adjunction between the category of  $F$ -coalgebras and  $G$ -algebras for the functors

<sup>1</sup> Strictly speaking, she showed it for complex-valued  $C^*$ -algebras, but the result also holds for real-valued ones.

$$\begin{aligned}
F: \mathbf{KHaus} &\rightarrow \mathbf{KHaus}, & F(X) &= \mathcal{D}_{\leq 1}(\Omega) \times X^\Sigma \\
G: \mathbf{CUC}^*\mathbf{Alg} &\rightarrow \mathbf{CUC}^*\mathbf{Alg}, & G(A) &= M(\Omega)/J + \Sigma \cdot A
\end{aligned} \tag{5}$$

Recall from (2) that  $M$  is the left adjoint of the unit interval functor  $U$ . Finally,  $J \subseteq M(\Omega)$  is an ideal of the  $\mathbf{CUC}^*\mathbf{Alg}$ -algebra  $M(\Omega)$  which we describe in a moment. Note that  $\mathbf{CUC}^*\mathbf{Alg}$  has coproducts. This follows from the fact that  $\mathbf{KHaus}$  has products and using Gelfand duality.

In order to lift the base dual adjunction  $\mathit{Spec} \dashv C$  to a dual adjunction between  $\mathit{Coalg}(F)$  and  $\mathit{Alg}(G)$  as in section 3.3.2, we need to show that  $\mathit{Spec}(M(\Omega)/J) \cong \mathcal{D}_{\leq 1}(\Omega)$ . First, we define the ideal  $J$ . Fix a finite set  $Y$  and consider the  $C^*$ -algebra  $M(Y) \in \mathbf{CUC}^*\mathbf{Alg}$  defined by  $[0, 1]^Y$ . For each  $y \in Y$ , we have a projection map  $\pi_y \in M(Y) \rightarrow \mathbb{R}$  given by  $\pi_y(v) = v(y)$ . Let  $\pi = \sum_{y \in Y} \pi_y$ . Then  $\pi: [0, 1]^Y \rightarrow \mathbb{R}$  is linear and  $\pi \in M(Y)$ . We will take  $J$  to be the ideal corresponding to the congruence generated by the equality obtained by rewriting  $\pi \preceq 1$  as an equality as follows:

$$\pi \preceq 1 \iff \pi \vee 1 = 1 \iff \frac{1}{2}(\pi + 1) + \frac{1}{2}|\pi + 1| = 1 \iff |1 - \pi| = 1 - \pi$$

**Definition 7** We define the ideal  $J$  of  $M(Y)$  as the principal ideal generated by the element  $(|\pi^-| - \pi^-)$  where  $\pi^- := 1 - \pi$ . That is,

$$J = \{m \in M(Y) \mid \exists k \in M(Y) : m = k(|\pi^-| - \pi^-)\}.$$

The congruence relation  $\equiv_J$  on  $M(Y)$  arising from the ideal  $J$  is then defined standardly as follows: For  $m, n \in M(Y)$ ,  $m \equiv_J n$  if  $m - n \in J$ . We write  $M(Y)/J$  for the quotient of  $M(Y)$  with respect to  $\equiv_J$ .

Due to space limitations, we omit the rather technical proof.

**Lemma 3** For any set  $Y$ ,  $\mathcal{D}_{\leq 1}(Y) \cong \mathit{Spec}(M(Y)/J)$  in  $\mathbf{KHaus}$ .

From Lemma 3 and section 3.3.2, it follows that the base dual adjunction lifts to one between  $F$ -coalgebras and  $G$ -algebras. This adjunction in turn can be easily lifted to one between automata using Theorem 1 from section 3.3.3. The categories of automata are here the category  $\mathbf{CHA} = \mathbf{Aut}_{\mathbf{KHaus}}^{1, \mathcal{D}_{\leq 1}(\Omega)}$  of compact Hausdorff automata and the category  $\mathbf{CAO} = \mathbf{Aut}_{\mathbf{CUC}^*\mathbf{Alg}}^{M(\text{Obs})/J, \mathbb{R}}$  of  $\mathbf{CUC}^*\mathbf{Alg}$ -automata with initialisation in  $M(\text{Obs})/J$  and output in  $\mathbb{R} \cong C(1)$ .

$$\begin{array}{ccc}
\mathbf{CHA}^{\text{op}} & \xrightleftharpoons[\text{Spec}']{\overline{C}'} & \mathbf{CAO} \\
\downarrow & & \downarrow \\
\mathbf{KHaus}^{\text{op}} & \xrightleftharpoons[\text{Spec}]{C} & \mathbf{CUC}^*\mathbf{Alg}
\end{array}$$

The abstract algorithms **Algo1** and **Algo2** apply since  $\mathbf{KHaus}$  and  $\mathbf{CUC}^*\mathbf{Alg}$  are monadic over  $\mathbf{Set}$  (cf. Section 3.6). In particular,  $\mathbf{KHaus}$  is the Eilenberg-Moore category of the ultrafilter monad [47]. In order to show that the associated trace logic is expressive we need an extra argument, since the functor  $F$  defined in (5) does not have the shape required by Lemma 1 and Theorem 2. However, we can apply Remark 2 after observing the following. Let  $F' := \mathit{Spec}(M(\Omega)) \times (-)^\Sigma$ . Then the associated natural isomorphism  $\xi^{\text{trc}'}: F' \mathit{Spec} M \Rightarrow \mathit{Spec} M G$  specifies semantics of trace logic over  $F'$ -coalgebras. To obtain a suitable  $\tau: F \Rightarrow F'$  note that quotienting with  $J$  in  $\mathbf{CUC}^*\mathbf{Alg}$  yields an epi  $e: M(O) \rightarrow M(O)/J$  from which we get a mono

$Spec(e) : Spec(M(O)/J) \rightarrow Spec(M(O))$  in  $\mathbf{KHaus}$ . Pre-composing  $Spec(e)$  with the isomorphism  $h : \mathcal{D}_{\leq 1}(O) \xrightarrow{\sim} Spec(M(O)/J)$  given by Lemma 3 and defining  $\tau := (Spec(e) \circ h) \times id$ , it follows that  $\tau : F \Rightarrow F'$  has all components mono in  $\mathbf{KHaus}$ . It now follows that trace logic is also expressive for  $F$ -coalgebras, i.e., for compact Hausdorff automata.

*Remark 5* In order to view Gelfand duality (4) as a concrete duality obtained from a dualising object, we need to expand the setting a bit, since  $\mathbb{R}$  is not a compact Hausdorff space. This can be done by considering the dual adjunction between locally compact Hausdorff spaces and not-necessarily unital commutative  $C^*$ -algebras. Gelfand duality is a restriction of this dual adjunction.

## 5 Alternating Automata

*Alternating finite automata* (aka *Boolean automata* or *parallel automata*) were first studied in [21, 24, 25, 42, 46] as a finite-state analog of alternating Turing machines [24]. Let  $\Sigma$  be a fixed finite *input alphabet*. An *alternating finite automaton* (AFA) over  $\Sigma$  is a tuple  $\mathcal{A} = (X, i, t, F)$ , where

- $X$  is a finite set of *states*,
- $F \subseteq X$  are the *final states*,
- $t : \Sigma \rightarrow X \rightarrow 2^X \rightarrow 2$  is the *transition function*, and
- $i : 2^X \rightarrow 2$  is the *acceptance condition*.

Intuitively, the machine  $\mathcal{A}$  operates as follows. Let  $k = |X|$ . Initially  $k$  processes are started, each assigned to a different state, reading the first symbol of the input word  $w \in \Sigma^*$ . In each step, a process at state  $s$  reads the next input symbol  $a$  and spawns  $k$  child processes, each of which moves to a different state and continues in the same fashion, while the parent process at  $s$  waits for the child processes to report back a Boolean value. In this way a  $k$ -branching computation tree is generated. When the end of the input word is reached, a process at state  $s$  reports 1 back to its parent if  $s \in F$ , 0 otherwise. A non-leaf process waiting at state  $s$ , having read input symbol  $a$ , collects the  $k$ -tuple  $b \in 2^X$  of Boolean values reported by its children, computes  $tab$ , and reports that Boolean value back to its parent. When the initial processes have all received values, say  $c \in 2^X$ , the machine accepts if  $ic = 1$ , otherwise it rejects.

Alternating automata accept all and only regular sets. It was shown in [43] by combinatorial means that a language  $L \subseteq \Sigma^*$  is accepted by a  $k$ -state AFA iff its reverse  $\{w^R \mid w \in L\}$  is accepted by a  $2^k$  state deterministic finite automaton (DFA).

Our purpose in this section is to recast this result in the framework of our general duality principle. The duality involves the category  $\mathbf{CABA}$  of *complete atomic Boolean algebras* and the category  $\mathbf{Set}$  of *discrete spaces*, which underlie *powerset Boolean algebras*.

### 5.1 CABA, $\mathbf{EM}(N)$ , and $\mathbf{Set}^{\text{op}}$

#### 5.1.1 CABA

A *complete Boolean algebra* (CBA) is a structure  $(B, \neg, \bigvee, \bigwedge, 0, 1, \leq)$ , where  $B$  is a set,  $\neg$  is a unary operation on  $B$ ,  $\bigvee$  and  $\bigwedge$  are infinitary operations on the powerset of  $B$ , 0 and 1 are constants, and  $\leq$  is a partial order on  $B$ , such that

- $(B, \neg, \vee, \wedge, 0, 1, \leq)$  is an ordinary Boolean algebra (BA), where  $\vee$  and  $\wedge$  are the restrictions of  $\bigvee$  and  $\bigwedge$ , respectively, to two-element sets; and
- $\bigvee A$  and  $\bigwedge A$  give the supremum and infimum of  $A$ , respectively, with respect to  $\leq$ .

The CBA-morphisms are BA-homomorphisms that preserve  $\bigvee$  and  $\bigwedge$ .

An *atom* of a BA is a  $\leq$ -minimal nonzero element. A BA is *atomic* if every nonzero element has an atom  $\leq$ -below it. A *complete atomic Boolean algebra* (CABA) is an atomic CBA. A CABA-morphism is just a CBA-morphism. Together, CABAs and their morphisms form the category CABA.

It is known that every CABA is isomorphic to the powerset Boolean algebra on its atoms, thus every element is the supremum of the atoms below it. CBAs and CABAs satisfy infinitary de Morgan and distributive laws:

$$\begin{aligned} \neg \bigvee a &= \bigwedge \{\neg x \mid x \in a\} & (\bigvee a) \wedge x &= \bigvee \{y \wedge x \mid y \in a\} \\ \neg \bigwedge a &= \bigvee \{\neg x \mid x \in a\} & (\bigwedge a) \vee x &= \bigwedge \{y \vee b \mid y \in a\} \end{aligned}$$

as well as other useful infinitary properties such as commutativity, associativity, and idempotence of  $\bigvee$  and  $\bigwedge$ . The free CABA on generators  $X$  is the powerset CABA  $(2^{2^X}, \bigcup, \bigcap, \sim, \emptyset, 2^X)$ . See, e.g., [31] for further information on the theory of CBAs and CABAs.

### 5.1.2 EM(N)

The self-dual adjunction  $\mathcal{Q}^{\text{op}} \dashv \mathcal{Q}$  of the contravariant powerset functor (Example 1) gives rise to a **Set**-monad  $N = \mathcal{Q} \circ \mathcal{Q}^{\text{op}}$ , where for  $X$  a set and  $f : X \rightarrow Y$  a set function,

$$NX = \mathcal{Q}\mathcal{Q}^{\text{op}}X = 2^{2^X} \qquad Nf = (f^{-1})^{-1} : 2^{2^X} \rightarrow 2^{2^Y}$$

The unit and multiplication are

$$\eta_X(x) = \{a \mid x \in a\}, \quad \mu_X(H) = \{a \mid \eta_{\mathcal{Q}X}(a) \in H\} = \eta_{\mathcal{Q}X}^{-1}(H).$$

This is called the *double powerset* or *neighbourhood monad*. The category of Eilenberg-Moore algebras of  $N$  is denoted **EM(N)**.

### 5.1.3 Equivalence of CABA, Set<sup>op</sup>, and EM(N)

It is known that the Eilenberg-Moore algebras of the double powerset monad  $N$  are exactly the CABAs. These two categories are also dually equivalent to **Set**, that is, equivalent to **Set<sup>op</sup>**, as observed in [60].

The equivalence of the three categories can be shown via the composition of three faithful functors that are injective on objects:

$$\mathbf{Set}^{\text{op}} \xrightarrow{J} \mathbf{EM}(N) \xrightarrow{D} \mathbf{CABA} \xrightarrow{At} \mathbf{Set}^{\text{op}}. \quad (1)$$

Here  $J$  is the Eilenberg-Moore comparison functor [3, 45]. Concretely,  $J$  sends a set  $X$  to the CABA  $2^X$  and a function  $f : X \rightarrow Y$  to its inverse image map. That is,  $J = \mathbf{Set}(-, 2)$  with Boolean structure. The functor  $At$  takes a CABA to its set of atoms and a CABA-morphism



$f : A \rightarrow B$  to  $At f : At B \rightarrow At A$ , where  $At f(b)$  is the unique atom  $a$  of  $A$  such that  $a\uparrow = f^{-1}(b\uparrow)$  and  $a\uparrow$  and  $b\uparrow$  are the principal ultrafilters on atoms  $a$  and  $b$ , respectively. In a CABA, there is a bijection between principal ultrafilters and atoms, and we have that  $At \cong \text{CABA}(-, 2)$ . In other words, the equivalence given by  $J$  and  $At$  is a concrete duality with dualising object  $2$ .

Although the equivalence between  $\text{EM}(N)$  and CABA is fairly well known, the details are rarely provided. We therefore describe the functor  $D$  that produces a CABA from an  $\text{EM}(N)$ -algebra  $(X, \alpha)$ . Let  $TX$  be the term monad for CABA terms over indeterminates  $X$ .<sup>2</sup> Let  $D(X, \alpha) = (X, D\alpha)$ , where

$$D\alpha : TX \rightarrow X \qquad D\alpha = \alpha \circ (\tau N \circ T\eta)_X, \quad (2)$$

where  $T\eta_X : TX \rightarrow TNX$  substitutes  $\eta_X(x)$  for  $x \in X$  in a term and  $\tau_{NX} : TNX \rightarrow NX$  is the evaluation map of the powerset CABA  $(2^{2^X}, \bigcup, \bigcap, \sim, \emptyset, 2^X)$ . In particular (and in more conventional notation), this gives the following definitions of the Boolean operations:

$$\begin{aligned} \bigvee_n x_n &= \alpha(\bigcup_n \eta_X(x_n)) & \bigwedge_n x_n &= \alpha(\bigcap_n \eta_X(x_n)) \\ \neg x &= \alpha(\sim \eta_X(x)) & 0 &= \alpha(\emptyset) & 1 &= \alpha(2^X). \end{aligned} \quad (3)$$

The action of  $D$  on morphisms is the identity.

The natural transformation  $\tau N \circ T\eta : T \rightarrow N$  in (2) relating CABA terms and double powerset is invertible up to CABA equivalence. Consider the natural transformation

$$v : N \rightarrow T \qquad v_X(A) = \bigvee_{a \in A} \left( \bigwedge_{x \in a} x \wedge \bigwedge_{x \notin a} \neg x \right), \quad A \in 2^{2^X}.$$

It can be shown that

$$\tau N \circ T\eta \circ v = id_N \qquad v \circ \tau N \circ T\eta \equiv id_T.$$

By the latter we mean that for any term  $\theta \in TX$ ,  $v_X(\tau_{NX}(T\eta_X(\theta))) \equiv \theta$  modulo the axioms of CABA. This essentially says that there is a disjunctive normal form for CABA terms.

## 5.2 Language acceptance of alternating automata

Let  $\mathcal{A} = (X, t, f, i)$  be an AFA with states  $X$  and components

$$i : 1 \rightarrow 2^{2^X} \qquad t_a : X \rightarrow 2^{2^X}, \quad a \in \Sigma \qquad f : X \rightarrow 2$$

where  $i$  is the (transposed) acceptance condition,  $t_a$  are the transitions, and  $f : X \rightarrow 2$  is the characteristic function for the subset  $F$  of accepting states.

The language accepted by  $\mathcal{A}$  is  $\mathcal{L}(\mathcal{A}) \triangleq \{w \in \Sigma^* \mid i(t'_w(F)) = 1\}$ , where

$$t'_w : 2^X \rightarrow 2^X \qquad t'_\varepsilon(A) = A \qquad t'_{aw}(A)(s) = t_a(s)(t'_w(A)).$$

<sup>2</sup>  $TX$  consists of CBA terms with the arity of the infinitary operations bounded by  $2^{2^{|X|}}$ . There can be no such bound for CBA in general, as there are CBAs of arbitrarily large cardinality generated by  $X$ ; thus there is no term monad for CBA. However, CABAs generated by  $X$  are of cardinality at most double exponential in  $|X|$ , and we can bound arities accordingly.

As constructed in [43], the associated DFA for the reverse language is  $\mathcal{A}'$  with states  $2^X$  and components

$$f^b : 1 \rightarrow 2^X \quad t_a^b : 2^X \rightarrow 2^X, a \in \Sigma \quad i^b : 2^X \rightarrow 2.$$

This is a deterministic automaton, that is, a coalgebra for the functor  $F = 2 \times (-)^\Sigma$  with start state  $f^b$ , transitions  $t_a^b$ , and accept states  $i^b$ . The language accepted by  $\mathcal{A}'$  is  $\mathcal{L}(\mathcal{A}') \triangleq \{w \in \Sigma^* \mid i^b(t_w^b(f^b)) = 1\}$ , where

$$t_\varepsilon^b = id_{2^X} \quad t_{wa}^b = t_a^b \circ t_w^b.$$

The combinatorial construction of [43] amounts to recurring the components of the automata. Denoting the reverse of a string  $w$  by  $w^R$  and using the fact that  $t_a^b = t'_a$ , it can be shown inductively that  $t_w^b = t'_{w^R}$ , therefore the language accepted by  $\mathcal{A}'$  is the reverse of the language accepted by  $\mathcal{A}$ :

$$\mathcal{L}(\mathcal{A}') = \{w \mid i^b(t_w^b(f^b)) = 1\} = \{w \mid i(t'_{w^R}(f)) = 1\} = \{w \mid w^R \in \mathcal{L}(\mathcal{A})\}.$$

### 5.3 Alternating automata as $\mathbf{EM}(N)$ -automata

We now show how the relationship between  $\mathcal{A}$  and  $\mathcal{A}'$  comes about as an instance of a dual adjunction of automata as described in Section 3.3, in particular Section 3.3.3. We use the base equivalence between  $\mathbf{EM}(N)$  and  $\mathbf{Set}^{\text{op}}$  described in Section 5.1. For the sake of uniformity with the general setup in Section 3.3, we take  $R$  as the right adjoint (hence we put the  $\text{op}$  on  $\mathbf{EM}(N)$ ), and consider  $R$  and  $J$  as contravariant functors.

$$\begin{array}{ccc}
 (\text{Aut}_{\mathbf{EM}(N)}^{N(1),2})^{\text{op}} & \xrightarrow{\overline{R}'} & \text{Aut}_{\mathbf{Set}}^{1,2} \\
 \downarrow & \begin{array}{c} \Upsilon \\ \overline{J}' \end{array} & \downarrow \\
 \mathbf{EM}(N)^{\text{op}} & \xrightarrow[\cong]{R} & \mathbf{Set} \\
 & \begin{array}{c} \overleftarrow{J} \end{array} & 
 \end{array} \tag{4}$$

More precisely, we show that  $\mathcal{A}' = \overline{R}'(\mathcal{A}^\sharp)$ , where  $\mathcal{A}^\sharp$  is the deterministic automaton over  $\mathbf{EM}(N)$  obtained by applying the determinisation construction from Section 2.2 for  $N$  to  $\mathcal{A}$ . The functor  $R$  is the composition  $R = At \circ D$  (see (1)).

Recall from Section 2.2 that determinisation for  $N$  takes free extensions of the transition function and output function. That is, given an alternating automaton  $\mathcal{A}$  with states  $X$  and components

$$i : 1 \rightarrow 2^{2^X} \quad t_a : X \rightarrow 2^{2^X}, a \in \Sigma \quad f : X \rightarrow 2$$

over  $\mathbf{Set}$ , we have a deterministic automaton  $\mathcal{A}^\sharp$  with

$$i^\sharp : 2^{2^1} \rightarrow 2^{2^X} \quad t_a^\sharp : 2^{2^X} \rightarrow 2^{2^X}, a \in \Sigma \quad f^\sharp : 2^{2^X} \rightarrow 2$$

over  $\mathbf{EM}(N)$ , using the CABA structure on  $2$ . In  $\mathcal{A}^\sharp$ , we leave algebraic structure on  $2^{2^X}$  and  $2$  implicit. Formally, they are the powerset CABAs on  $2^{2^X}$  and  $2$ , respectively; these are isomorphic to the free  $\mathbf{EM}(N)$ -algebras  $(NX, \mu_X)$  and  $(N\emptyset, \mu_\emptyset)$  on generators  $X$  and  $\emptyset$ , respectively.

We easily see that  $\langle 2^{2^X}, f^\sharp, t^\sharp, i^\sharp \rangle$  instantiates the definition from Section 3.2 of an  $\mathbf{EM}(N)$ -automaton with initialisation in  $2^{2^1}$  and output in  $2$ , i.e.,  $\mathcal{A}^\sharp$  is in  $\mathbf{Aut}_{\mathbf{EM}(N)}^{N(1),2}$ . For ease of notation, we will sometimes write the initialisation morphism  $i^\sharp$  as its corresponding  $\mathbf{Set}$ -function  $i$ .

A dual automaton in  $\mathbf{Aut}_{\mathbf{Set}}^{1,2}$  (with states  $X$ ) is a coalgebra for  $F = 2 \times (-)^\Sigma$  together with an initial state  $j : 1 \rightarrow X$ , or equivalently an algebra for  $G = 1 + \Sigma \times (-)$  with output  $f : X \rightarrow 2$ . It is easy to check that the conditions for Theorem 1 hold. First note that  $I = 2^{2^1}$  and  $O = 1$ . We then easily verify that  $F_{\mathbf{EM}(N)} \cong J(1) \times (-)^\Sigma$  by noting that  $J(1) = 2^1 \cong 2$ . Similarly, to see that  $G \cong R(2^{2^1}) + \Sigma \cdot (-)$ , we note that  $R(2^{2^1}) = At(2^{2^1}) = 2^1 \cong 2$ . Hence the base dual adjunction  $J \dashv R$  lifts to  $\bar{J}' \dashv \bar{R}'$  between automata categories, and the lifted adjoints are given by (5) and Theorem 1. We describe the reversal functor  $\bar{R}'$  a bit more concretely as a contravariant functor from  $\mathbf{Aut}_{\mathbf{EM}(N)}^{N(1),2}$  to  $\mathbf{Aut}_{\mathbf{Set}}^{1,2}$ . The base adjunction of (4) gives us a bijection of homsets:

$$\theta : \mathbf{EM}(N)((A, \alpha), JX) \rightarrow \mathbf{Set}(X, R(A, \alpha))$$

natural in  $(A, \alpha)$  and  $X$ . Given an automaton in  $\mathbf{Aut}_{\mathbf{EM}(N)}^{N(1),2}$

$$i : 1 \rightarrow (A, \alpha) \quad t_a : (A, \alpha) \rightarrow (A, \alpha) \quad f : (A, \alpha) \rightarrow 2$$

(again, we leave the algebraic structure on  $2$  implicit),  $\bar{R}'$  produces the deterministic automaton over  $\mathbf{Set}$

$$\theta f : 1 \rightarrow R(A, \alpha) \quad R(t_a) : R(A, \alpha) \rightarrow R(A, \alpha) \quad Ri^\sharp : R(A, \alpha) \rightarrow 2. \quad (5)$$

Applying  $\bar{R}'$  to  $\mathcal{A}^\sharp$ , which is

$$i : 1 \rightarrow 2^{2^X} \quad t_a^\sharp : 2^{2^X} \rightarrow 2^{2^X} \quad f^\sharp : 2^{2^X} \rightarrow 2$$

we get the reversed, deterministic automaton  $\bar{R}'(\mathcal{A}^\sharp)$  (over  $\mathbf{Set}$ ):

$$\theta f^\sharp : 1 \rightarrow 2^X \quad \bar{R}t_a^\sharp : 2^X \rightarrow 2^X \quad Ri^\sharp : 2^X \rightarrow 2.$$

**Theorem 4** *For any alternating automaton*

$$\mathcal{A} = (X, \{t_a : X \rightarrow 2^{2^X} \mid a \in \Sigma\}, i : 1 \rightarrow 2^{2^X}, f : X \rightarrow 2),$$

*we have that  $\mathcal{A}' \cong \bar{R}'(\mathcal{A}^\sharp)$ .*

**Proof** The state space of  $\mathcal{A}'$  is  $2^X$  and the state space of  $\bar{R}'(\mathcal{A}^\sharp)$  is the set of atoms of the CABA  $D(2^{2^X}, \mu_X)$  which is the set  $\{\{a\} \mid a \subseteq X\}$ . To see that  $\theta(f^\sharp) = f^b$ , observe that  $\theta^{-1}(f^b) = f^{b^{-1}}$ , and expand the definitions to show that  $f^\sharp = f^{b^{-1}}$ . To see that  $R(t_a^\sharp) = t_a^b$ , we show that for all  $d : Y \rightarrow 2^{2^X}$ , and all  $a \subseteq X$ ,  $R(d^\sharp)(\{a\}) = \{d^b(a)\}$ . Thus  $R(d^\sharp) = d^b$  up to bijections relating atoms  $\{a\}$  and their singleton elements  $a$ . The atoms of  $(2^{2^X}, \mu_X)$  are of the form  $\{a\}$  for  $a \subseteq X$ , hence for  $A \in 2^{2^Y}$ ,

$$\begin{aligned}
R(d^\sharp)(\{a\}) &= \bigwedge \{A \in 2^{2^Y} \mid \{a\} \leq d^\sharp(A)\} \\
&= \bigcap \{A \in 2^{2^Y} \mid a \in d^\sharp(A)\} && \text{since } \bigwedge \text{ is } \bigcap \text{ in } (2^{2^X}, \mu_X) \\
&= \bigcap \{A \in 2^{2^Y} \mid d^b(a) \in A\} && \text{since } d^\sharp = d^{b^{-1}} \\
&= \{d^b(a)\} && \text{since } d^b(a) \in 2^Y.
\end{aligned}$$

where  $d^\sharp = d^{b^{-1}}$  can be shown by expanding the definitions. The argument for  $R(i^\sharp) = i^b$  is similar.  $\square$

The relationship between an AFA and its determinised version can be understood as follows. In an AFA, when reading an input word, we generate a computation tree downwards, and once we reach the end of the word, we evaluate the outputs going back up using Boolean functions, and at the top all outputs are aggregated into a single Boolean value with the acceptance condition. In the determinised AFA, we propagate the acceptance condition forwards as a Boolean function (encapsulated in the state) and once we reach the end of the input word, we use the Boolean function to evaluate immediately instead of propagating back up.

The modal logic of alternating automata (cf. (6)) has a single termination predicate, labelled modalities, and no propositional connectives, since  $\mathcal{D} = \mathbf{Set}$ . Hence formulas correspond to words in  $\Sigma^*$ . The dual DFA of an AFA represents its logical semantics, or predicate transformer semantics, where the observations at the end of the word are propagated backwards to the initial state. Since predicate transformers move backwards, the language of an AFA is the reversed language of the dual DFA.

Finally, we note that all conditions for Theorem 2 hold (with  $\mathcal{D} = \mathbf{Set}$  and  $\Phi_{\mathcal{D}} = U_{\mathcal{D}} = Id$ ). Hence we also get a Brzozowski style minimisation algorithm for alternating automata by instantiating **Algo2** of Section 3.6. Reachability in  $\mathbf{Aut}_{\mathbf{Set}}^{1,2}$  is just the standard automata-theoretic notion, whereas now the more abstract algebraic notion from Section 3.5 is relevant “on the left” in the category  $\mathbf{Aut}_{\mathbf{EM}(N)}^{N(1),2}$ . As with weighted automata (cf. Section 4.2), we are not guaranteed that the result of the minimisation algorithm is again an alternating automaton (understood as an  $FN$ -coalgebra over  $\mathbf{Set}$ ), since a subalgebra of a free CABA need not be free.

## 6 Conclusion and Related Work

In this paper, we presented a unifying categorical perspective on the minimisation constructions presented in [13] and [16], revisited some examples from these two papers in light of the general framework, and presented a new example of alternating automata. We also filled in some details regarding topological automata (belief automata) that were missing from [13].

Our starting points are Brzozowski’s algorithm [22] for the minimisation of deterministic automata and the use of Stone-type duality between computational processes and their logical characterisation [1]. The connection between these two seemingly unrelated points is given by the duality principle between reachability and observability originally introduced in systems theory [38] and then extended to automata theory in [7–9].

The duality between reachability and observability has been studied, e.g. in [14], to relate coalgebraic and algebraic specifications in terms of observations and constructors. In this context most notable is the use of Stone-type dualities between automata and varieties of formal languages [28, 29] which recently culminated into a general algebraic and coalgebraic understanding of equations, coequations, Birkhoff’s and Eilenberg-type correspondences [4, 5, 11, 52, 53].

Our unifying categorical perspective is based on a dual adjunction between base categories lifted to a dual adjunction between coalgebras and algebras, as introduced in [19, 39, 40] in the context of coalgebraic modal logic, and in [13, 41] to capture the observable behaviour of a coalgebra. Our novelty is to lift the coalgebra-algebra adjunction to a dual adjunction between automata which generalises the formalisation of Brzozowski’s algorithm from [16], and formalising the relationship of trace logic to the full modal logic and language semantics.

Our paper focuses on comparing and unifying our earlier approaches from [13] and [16] under a common umbrella, but we hasten to remark that the concept of minimisation via logic presented in section 3.3 is already in [50]. At its core, [50] uses a dual adjunction that is lifted to a dual adjunction between coalgebras and algebras. A logic is then used to provide a construction for obtaining observable coalgebras. This is essentially what we call **Algo1**. The setting of [50] is more general as no assumptions are made on the specific shape of the algebra and coalgebra functors involved. Instead the necessary functor requirements are axiomatised. One achievement of [50] is to generalise the setup in [13] from dual equivalences to dual adjunctions. The central contribution in [50] is to combine the duality-based framework with coalgebraic partition-refinement [2] such that a logic-based treatment of Brzozowski and partition refinement is obtained. Compared to [50], our framework is more restricted, as we confine ourselves to functors of certain shapes, but we believe this strikes a good balance between generality and a categorical setting for studying many different types of automata. Furthermore, our categorical framework incorporates a formalisation of the full Brzozowski algorithm via the small extension of the coalgebra-algebra adjunction to the adjunction of automata, i.e., structures that have both initial and final states.

Other categorical approaches to automata minimisation have been proposed in the literature; we mention here just a few. In [26] languages and their acceptors are regarded as functors which provides a different perspective on minimisation in which Brzozowski’s algorithm can also be formulated. In [2] the authors study coalgebras in categories equipped with factorisation structures in order to devise a generic partition refinement algorithm. From the language-theoretic point of view, the relation between the automata constructions resulting from the automata-based congruences, together with the duality between right and left congruences, allows to relate determinisation and minimisation operations [27].

## References

1. Samson Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51(1):1 – 77, 1991.
2. Jirí Adámek, Filippo Bonchi, Mathias Hülsbusch, Barbara König, Stefan Milius, and Alexandra Silva. A coalgebraic perspective on minimization and determinization. In Lars Birkedal, editor, *Foundations of Software Science and Computational Structures - 15th International Conference, (FOSSACS 2012)*, volume 7213 of *Lecture Notes in Computer Science*, pages 58–73. Springer, 2012.
3. Jirí Adámek, Horst Herrlich, and George E. Strecker. *Abstract and Concrete Categories - The Joy of Cats*. Dover Publications, 2009.
4. Jiri Adamek, Robert S. R. Myers, Henning Urbat, and Stefan Milius. Varieties of languages in a category. In *Proceedings of the 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, LICS 15, page 414425. IEEE Computer Society, 2015.
5. Jirí Adámek, Stefan Milius, Robert S.R. Myers, and Henning Urbat. Generalized Eilenberg theorem: Varieties of languages in a category. *ACM Transaction of Computational Logic*, 20(1), 2018.
6. M. A. Arbib and E. G. Manes. Machines in a category: An expository introduction. *SIAM Review*, 16:163–192, 1974.
7. M.A. Arbib and H.P. Zeiger. On the relevance of abstract algebra to control theory. *Automatica*, 5:589–606, 1969.
8. Michael A. Arbib and Ernest G. Manes. Adjoint machines, state-behavior machines, and duality. *J. of Pure and Applied Algebra*, 6(3):313 – 344, 1975.

9. Michael A. Arbib and Ernest G. Manes. Foundations of system theory: The Hankel matrix. *Journal of Computer and System Sciences*, 20:330–378, 1980.
10. William Arveson. *An Invitation to  $C^*$ -Algebras*, volume 39 of *Graduate Texts in Mathematics*. Springer-Verlag, 1976.
11. A. Ballester-Bolinches, E. Cosme-Llópez, and J. Rutten. The dual equivalence of equations and coequations for automata. *Information and Computation*, 244(C):4975, 2015.
12. F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats*. PhD thesis, Vrije Universiteit Amsterdam, 2004.
13. Nick Bezhanishvili, Clemens Kupke, and Prakash Panangaden. Minimization via duality. In L. Ong and R. de Queiroz, editors, *Proceedings of WoLLIC 12*, volume 7456 of *LNCS*, pages 191–205. Springer, 2012.
14. Michel Bidoit, Rolf Hennicker, and Alexander Kurz. On the duality between observability and reachability. In Furio Honsell and Marino Miculan, editors, *FoSSaCS*, volume 2030 of *Lect. Notes in Comp. Sci.*, pages 72–87. Springer, 2001.
15. P. Blackburn, M. de Rijke, and Y. Venema. *Modal logic*. Cambridge University Press, Cambridge, 2001.
16. Filippo Bonchi, Marcello Bonsangue, Helle Hvid Hansen, Prakash Panangaden, Jan Rutten, and Alexandra Silva. Algebra-coalgebra duality in brzozowski’s minimization algorithm. *ACM Transactions on Computational Logic*, 15(1), 2014.
17. Filippo Bonchi, Marcello M. Bonsangue, Michele Boreale, Jan J. M. M. Rutten, and Alexandra Silva. A coalgebraic perspective on linear weighted automata. *Information and Computation*, 211:77–105, 2012.
18. M. M. Bonsangue and A. Kurz. Duality for logics of transition systems. In *FoSSaCS05*, 2005.
19. Marcello M. Bonsangue and Alexander Kurz. Presenting functors by operations and equations. In Luca Aceto and Anna Ingólfssdóttir, editors, *Foundations of Software Science and Computation Structures, 9th International Conference, FOSSACS 2006, Held as Part of the Joint European Conferences on Theory and Practice of Software, (ETAPS 2006)*, volume 3921 of *Lecture Notes in Computer Science*, pages 172–186. Springer, 2006.
20. Francis Borceux. *Handbook of Categorical Algebra 2: Categories and Structure*. Cambridge University Press, 1994.
21. J. A. Brzozowski and E. Leiss. On equations for regular languages, finite automata, and sequential networks. *Theoretical Computer Science*, 10:19–35, 1980.
22. Janusz A. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. In *Mathematical Theory of Automata*, volume 12 of *MRI Symposia Series*, pages 529–561, Polytechnic Institute of Brooklyn, 1962. Polytechnic Press.
23. A. Chagrov and M. Zakharyashev. *Modal logic*, volume 35 of *Oxford Logic Guides*. The Clarendon Press, New York, 1997.
24. Ashok Chandra, Dexter Kozen, and Larry Stockmeyer. Alternation. *J. Assoc. Comput. Mach.*, 28(1):114–133, 1981.
25. Ashok K. Chandra and Larry J. Stockmeyer. Alternation. In *Proc. 17th Symp. Found. Comput. Sci.*, pages 98–108. IEEE, October 1976.
26. Thomas Colcombet and Daniela Petrisan. Automata minimization: a functorial approach. In Filippo Bonchi and Barbara König, editors, *7th Conference on Algebra and Coalgebra in Computer Science, (CALCO 2017)*, volume 72 of *LIPICs*, pages 8:1–8:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.
27. Pierre Ganty, Elena Gutiérrez, and Pedro Valero. A congruence-based perspective on automata minimization algorithms. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, (MFCS 2019)*, volume 138 of *LIPICs*, pages 77:1–77:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
28. Mai Gehrke. Stone duality and the recognisable languages over an algebra. In Alexander Kurz, Marina Lenisa, and Andrzej Tarlecki, editors, *CALCO*, volume 5728 of *Lect. Notes in Comp. Sci.*, pages 236–250. Springer, 2009.
29. Mai Gehrke, Serge Grigorieff, and Jean-Eric Pin. Duality and equational theory of regular languages. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfssdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lect. Notes in Comp. Sci.*, pages 246–257. Springer, 2008.
30. G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *Continuous lattices and domains*, volume 93 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2003.
31. Steven Givant and Paul Halmos. *Introduction to Boolean Algebras*. Springer, 2009.
32. Claudio Hermida and Bart Jacobs. Structural induction and coinduction in a fibrational setting. *Information and Computation*, 145:107–152, 1998.
33. Edward V. Huntington. Sets of independent postulates for the algebra of logic. *Trans. Amer. Math. Soc.*, 5(3):288–309, July 1904.

34. B. Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In K. Futatsugi, J.-P. Jouannaud, and J. Meseguer, editors, *Algebra, Meaning and Computation: Essays dedicated to Joseph A. Goguen on the Occasion of his 65th Birthday*, volume 4060 of *LNCIS*, pages 375–404. Springer, 2006.
35. Bart Jacobs and Ana Sokolova. Exemplaric expressivity of modal logics. *J. Log. Comput.*, 20(5):1041–1068, 2010.
36. P. T. Johnstone. *Stone spaces*. Cambridge University Press, Cambridge, 1982.
37. P.T. Johnstone. Adjoint lifting theorems for categories of algebras. *Bulletin London Mathematical Society*, 7:294–297, 1975.
38. R. Kalman. On the general theory of control systems. *IRE Transactions on Automatic Control*, 4(3):110–110, 1959.
39. Henning Kerstan, Barbara König, and Bram Westerbaan. Lifting adjunctions to coalgebras to (re)discover automata constructions. In Marcello M. Bonsangue, editor, *Coalgebraic Methods in Computer Science*, pages 168–188, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
40. Bartek Klin. Coalgebraic modal logic beyond sets. In Marcelo Fiore, editor, *Proceedings of the 23rd Conference on the Mathematical Foundations of Programming Semantics, (MFPS 2007)*, volume 173 of *Electronic Notes in Theoretical Computer Science*, pages 177–201. Elsevier, 2007.
41. Bartek Klin and Jurriaan Rot. Coalgebraic trace semantics via forgetful logics. *Logical Methods in Computer Science*, 12(4), 2016.
42. Dexter Kozen. On parallelism in Turing machines. In *Proc. 17th Symp. Found. Comput. Sci.*, pages 89–97. IEEE, October 1976.
43. Dexter Kozen. *Theory of Computation*. Springer, New York, 2006.
44. Alexander Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwigs-Maximilians-Universität München, 2000.
45. Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971.
46. Ernst Leiss. Succinct representation of regular languages by Boolean automata. *Theoretical Computer Science*, 13:323–330, 1981.
47. E. Manes. A triple-theoretic construction of compact algebras. In B. Eckman, editor, *Seminar on Triples and Categorical Homology Theory*, number 80 in *Lect. Notes Math.*, pages 91–118. Springer, 1969.
48. Joan W. Negrepointis. Duality in analysis from the point of view of triples. *Journal of Algebra*, 19:228–253, 1971.
49. Hans-E. Porst and Walter Tholen. Concrete dualities. In H. Herrlich and Hans-E. Porst, editors, *Category Theory at Work*. Heldermann Verlag, 1991.
50. Jurriaan Rot. Coalgebraic minimization of automata by initiality and finality. In Lars Birkedal, editor, *The Thirty-second Conference on the Mathematical Foundations of Programming Semantics, (MFPS 2016)*, volume 325 of *Electronic Notes in Theoretical Computer Science*, pages 253–276. Elsevier, 2016.
51. J. J. M. M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.
52. Julian Salamanca, Adolfo Ballester-Bolinches, Marcello M. Bonsangue, Enric Cosme-Llópez, and Jan J. M. M. Rutten. Regular varieties of automata and coequations. In Ralf Hinze and Janis Voigtländer, editors, *Mathematics of Program Construction - 12th International Conference, (MPC 2015), Proceedings*, volume 9129 of *Lecture Notes in Computer Science*, pages 224–237. Springer, 2015.
53. Julian Salamanca, Marcello M. Bonsangue, and Jurriaan Rot. Duality of equations and coequations via contravariant adjunctions. In Ichiro Hasuo, editor, *Coalgebraic Methods in Computer Science - 13th IFIP WG 1.3 International Workshop, (CMCS 2016)*, volume 9608 of *Lecture Notes in Computer Science*, pages 73–93. Springer, 2016.
54. Marcel Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, 1961.
55. A. Silva, F. Bonchi, M.M. Bonsangue, and J.J.M.M. Rutten. Generalizing the powerset construction, coalgebraically. In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, December 15-18, 2010, Chennai, India*, volume 8 of *LIPICs*, pages 272–283. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.
56. M. H. Stone. The theory of representations for Boolean algebras. *Trans. Amer. Math. Soc.*, 40(1):37–111, 1936.
57. M. H. Stone. Topological representation of distributive lattices and Brouwerian logics. *Časopis Pešt. Mat. Fys.*, 67:1–25, 1937.
58. Ross Street. The formal theory of monads. *J. Pure and Applied Algebra*, 2:149–168, 1972.
59. Yi-Jia Tan. Free sets and free subsemimodules in a semimodule. *Linear Algebra and its Applications*, 496:527–548, 2016.
60. Paul Taylor. Subspaces in abstract Stone duality. *Theory and Applications of Categories*, 10(13):300–366, 2002.