Minimisation in Logical Form

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

Published in:
Samson Abramsky on Logic and Structure in Computer Science and Beyond

Minimisation in Logical Form

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

May 26, 2020

1 Introduction

The role of category theory, algebra, and logic in deepening our understanding of semantics and algorithms in Computer Science has long been one of Samson's flagships. 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 semantics, operational or denotational, and syntax, provided as a logic or specification language.

Building on Stone's celebrated representation theorems for Boolean algebras [68] and distributive lattices [69], categorical dualities linking algebra and topology [43] have been widely used in logic and theoretical computer science [19, 27, 35]. 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, 16, 20, 21, 31, 48, 58]. Among these, [16] and [20] had striking similarities in the approach yet it was not clear whether the differences could be harmonised in a uniform way. The main aim of this paper is to find a unifying perspective on the minimisation constructions in [16] and [20]. Duality will play a central role in achieving our aim of unification of approaches, which puts us on the path forged by Samson.

In [16], the authors adopt the coalgebraic perspective on automata and use a dual equivalence between the category of coalgebras and a category of algebras to explain minimisation. The key observation is that the algebras considered provide semantics of a modal logic. This algebraic semantics is dual to coalgebraic semantics in which logical equivalence coincides with trace equivalence. From this coalgebra-algebra duality it follows that maximal quotients of coalgebras correspond to minimal subobjects of algebras. In order to explain the minimisation algorithm, the authors exploit duality to prove that the maximal quotient of a coalgebra can be constructed by computing the subalgebra of definable predicates in the dual modal algebra. The examples in [16] include partially observable DFAs, linear weighted automata viewed as coalgebras over

finite-dimensional vector spaces, and belief automata which are coalgebras on compact Hausdorff spaces.

In [20], Brzozowski's double-reversal minimisation algorithm for deterministic finite automata (with both initial and final states) was described categorically and its correctness explained via the duality between reachability and observability, whose origins trace back to seminal work of Kalman in control theory [45]. Kalman's work was extended to automata theory in a collection of papers by Arbib and Manes [6–12]. The work of [20] is closely related to these and includes generalisations of Brzozowski's algorithm to Moore and weighted automata over commutative semirings.

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 with a comparison between the approaches in [16] and [20] (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 [16] and [20]: 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 place 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 [65] 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).

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 [60]. For general categorical notions, see e.g. [3]. We assume familiarity with classic automata such as (non)deterministic finite automata, and Moore automata.

Categories are denoted by $\mathcal{C}, \mathcal{D}, \ldots$, objects of categories by $X, Y, Z, \ldots$, and arrows/morphisms of categories by $f, g, h, \ldots$. We denote by 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 $\pi_i \colon X_1 \times X_2 \to X_i$, $i = 1, 2$. Similarly, their coproduct (if it exists) is written $X + Y$ with coprojection maps $\mathsf{in}_i \colon X_i \to X_1 + X + 2$. In Set, $X \times Y$ and $X + Y$ 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 $\mathcal{P} \colon \mathsf{Set} \to \mathsf{Set}$ sends a set X to its powerset $\mathcal{P}(X)$ and a function $f \colon X \to Y$ to the direct-image map $\mathcal{P}(f) \colon \mathcal{P}(X) \to \mathcal{P}(Y)$. The contravariant powerset functor $\mathcal{Q} \colon \mathsf{Set} \to \mathsf{Set}^{\mathsf{op}}$ also sends a set X to its powerset, now denoted $\mathcal{Q}(X)$, and a function $f \colon X \to Y$ to its inverse-image map $\mathcal{Q}(f) \colon \mathcal{P}(Y) \to \mathcal{P}(X)$.

2.1 Coalgebras, Algebras and Monads

Given an endofunctor $F \colon \mathcal{C} \to \mathcal{C}$, an *$F$-coalgebra* is a pair $(X, \gamma \colon X \to FX)$, where X is a $\mathcal{C}$-object and $\gamma \colon X \to FX$ 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 γ specifies the dynamics. An *F-coalgebra morphism* from an F-coalgebra (X, γ) to an F-coalgebra (Y, δ) is a $\mathcal{C}$-arrow $h \colon X \to Y$ that preserves the coalgebra structure, *i.e.*, $\delta \circ h = Fh \circ \gamma$. F-coalgebras and F-coalgebra morphisms form a category denoted by $\mathsf{Coalg}_{\mathcal{C}}(F)$. A *final F-coalgebra* is a final object in $\mathsf{Coalg}_{\mathcal{C}}(F)$, i.e., an F-coalgebra (Ω, ω) is final if for all T-coalgebras (X, γ) there is a unique F-coalgebra morphism $h \colon (X, c) \to (\Omega, \omega)$.

An F-algebra is the dual concept of an F-coalgebra. An *F-algebra* is a pair (X, α), where X is a $\mathcal{C}$-object and $\alpha \colon FX \to X$ 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, α) to an F-algebra (Y, β) is a $\mathcal{C}$-arrow $h \colon X \to Y$ that preserves the algebra structure, *i.e.*, $h \circ \alpha = \beta \circ Fh$. F-algebras and F-algebra morphisms form a category denoted by $\mathsf{Alg}_{\mathcal{C}}(F)$. An *initial F-algebra* is an initial object (A, α) in $\mathsf{Alg}_{\mathcal{C}}(F)$, i.e., for all F-algebras (X, β) there is a unique F-algebra morphism $h \colon (A, \alpha) \to (X, \beta)$.

A *monad* (on $\mathcal{C}$) is a triple (T, η, μ) consisting of a functor $T \colon \mathcal{C} \to \mathcal{C}$ and two natural transformations $\eta \colon Id \to T$ (the unit) and $\mu \colon TT \to T$ (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, α) 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 $\mathsf{EM}(T)$. In particular, for every X in $\mathcal{C}$, (TX, μ_X) is the free Eilenberg-Moore T-algebra on X, i. e., for every (A, α) in $\mathsf{EM}(T)$ and every $\mathcal{C}$-arrow $f \colon X \to A$ there is a unique T-algebra morphism (called the *free extension of f*) $f^\sharp \colon (TX, \mu_X) \to (A, \alpha)$ such that $f^\sharp \circ \eta_X = f$. Notice also that we have $f^\sharp = \alpha \circ Tf$.

2.2 Determinisation

Let (T, η, μ) be a monad on Set and $F\colon \mathsf{Set} \to \mathsf{Set}$ a functor given by $FX = B \times X^A$ where A is a set and B is the carrier of an Eilenberg-Moore T-algebra (B, β). Then FT-coalgebras can be seen as automata with input alphabet A, output in B and branching structure given by T. For example, nondeterministic automata are $F\mathcal{P}$-coalgebras where $FX = 2 \times X^A$ and $\beta = \vee\colon \mathcal{P}2 \to 2$ is the join (or max). Such FT-coalgebras can be "determinised" using a generalisation of the classic powerset construction [67], and the result can be seen as an F-coalgebra in the category $\mathsf{EM}(T)$. We follow [15, 41] in explaining this general construction. As shown in [41], there is a so-called distributive law $\lambda\colon TF \Rightarrow FT$ of the monad (T, η, μ) over the functor F given by

$$\lambda_X\colon T(B \times X^A) \xrightarrow{\langle T\pi_1, T\pi_2\rangle} TB \times T(X^A) \xrightarrow{\beta \times \mathsf{st}} B \times (TX)^A \tag{2.1}$$

where $\mathsf{st}\colon T \circ (-)^A \Rightarrow (-)^A \circ T$ is the strength natural transformation that exists for all monads on Set. Such a distributive law λ corresponds to a lifting of $F\colon \mathsf{Set} \to \mathsf{Set}$ to a functor $F_\lambda\colon \mathsf{EM}(T) \to \mathsf{EM}(T)$ [44], and it induces a functor $(-)^\sharp\colon \mathsf{Coalg}_{\mathsf{Set}}(FT) \to \mathsf{Coalg}_{\mathsf{EM}(T)}(F_\lambda)$ which sends an FT-coalgebra $\gamma = \langle o, t\rangle\colon X \to B \times (TX)^A$ to its determinisation $\gamma^\sharp = F\mu_X \circ \lambda_{TX} \circ T\gamma$, that is,

$$\gamma^\sharp = TX \xrightarrow{T\gamma} T(B \times (TX)^A) \xrightarrow{\lambda_{TX}} B \times (TTX)^A \xrightarrow{B \times (\mu_X)^A} B \times (TX)^A \tag{2.2}$$

Another perspective is that λ induces an Eilenberg-Moore T-algebra structure α on FTX, and $\gamma^\sharp\colon (TX, \mu_S) \to (FTX, \alpha)$ is the free extension of γ induced by α. This also justifies our use of the notation $(-)^\sharp$.

The determinisation $\gamma^\sharp$ can be seen as a Moore automaton in $\mathsf{EM}(T)$. 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

3.1 Unifying Previous Approaches

One aim of this paper is to find a unifying perspective on the automata minimisation constructions in [16] and [20]. We therefore start by summarising the two papers, and discuss the differences and similarities.

First we establish some terminology regarding key notions. 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. These notions can be generalised to other types of automata using that automata are in a sense both algebras and coalgebras as we will explain in Section 3.2. We will call an algebra *reachable* if it has no proper subalgebras, and a coalgebra is *observable* if it has no proper quotients. A (generalised) automaton is then *minimal*

if its algebra part is reachable and its coalgebra part is observable. Note that in the literature, observable coalgebras are usually called minimal coalgebras.

In [16], (generalised) Moore automata (without initial state) are modelled as coalgebras for a functor $F = B \times (-)^{\Sigma}$ on base categories of algebras or topological spaces. The main observation used in [16] 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 can be seen as a generalisation of the Jonsson-Tarski duality known from modal logic, which in turn arises from Stone duality. The algebras in [16] 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.

The main contribution of [16] can then be formulated as follows: Letting γ be a coalgebra, the minimal subalgebra of its dual modal algebra α consists of the predicates over γ that are definable in the modal logic. Hence to make γ observable, compute the subalgebra of definable subsets (which is reachable by construction), and dualise to obtain an observable coalgebra. Although, this is not stated in [16], for classic automata the computation of definable subsets corresponds to the partition refinement algorithm.

The minimisation-via-duality approach of [16] 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 [20], Brzozowski's double-reversal minimisation algorithm [26] for classic automata was described categorically. The Brzozowski algorithm works as follows. Starting with a classic, finite (possibly nondeterministic) automaton accepting a language $\mathcal{L}$, reverse the transitions, swap initial and final states, and make the result deterministic using the subset construction. This reversed automaton accepts the reversed language $rev(\mathcal{L})$. Take the reachable part of the reversed automaton. Now, do all of this again. The result will be a reachable and observable (i.e., minimal) classic deterministic automaton accepting $\mathcal{L}$. The correctness of the algorithm was explained in [20] via the duality between reachability and observability known from control theory (cf. [6, 9, 45]. This duality arises from a dual adjunction between algebras and coalgebras, and therefore only works in one direction, so to speak, namely, a reachable algebra dualises to an observable coalgebra, but not vice versa. This, however, is sufficient to formalise Brzozowski's algorithm in terms of a dual adjunction between categories of automata (with both initial and final states).

Generalisations of the Brzozowski algorithm were then formulated in [20] for Moore automata (over Set) and weighted automata, which include nondeterministic and linear weighted automata as instances. More precisely, weighted automata were first determinised into Moore automata over semimodules, and after the reverse-determinise step, the semimodule structure is forgotten in order to take the reachable part. Example 8.3

in [20] illustrates that one generally wants to take a subsemimodule that spans the reachable part, but this was not fully formalised. One aim of the present paper is to make this part precise.

We summarise the main differences and similarities. In [16], the minimisation-via-duality approach produces from a coalgebra (with structured state space), an observable coalgebra of the same type. In [20], the Brzozowski-based approach starts with a Set-based automaton that possibly has branching structure specified by a monad T. This automaton is determinised to yield a Moore automaton over the category $\mathsf{EM}(T)$ of Eilenberg-Moore algebras for T, and the result is a reachable and observable (i.e., minimal) Moore automaton over $\mathsf{EM}(T)$. If the automaton has no branching, we just proceed with Brzozowki over Set. In Appendix A.1, we give a small example illustrating the difference between the two minimisation constructs on a concrete DFA. In [16], the perspective is based on modal logic. Language semantics and reachability of automata is not an explicit part of the story, although it is implicitly present via trace logic, however the connection to reachability (in the usual set-theoretic sense) is not made. In [20], the perspective is language-based. No link is made to modal logic.

In the remainder of this section, we present a categorical picture that unifies both approaches. 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 [20,39,46,58] – precise connections are detailed throughout the sections and in Section 6.

3.2 Automata, Algebras and Coalgebras

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

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

Definition 3.1 *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 Σ-indexed set of transition morphisms $\{t_a\colon X \to X \mid a \in \Sigma\}$, an initialisation morphism $i\colon I \to X$, and an output morphism $f\colon X \to B$. 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 $h\colon X_1 \to X_2$ 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 $\mathsf{Aut}_{\mathcal{C}}^{I,B}$.*

A classic deterministic automaton is then easily seen to be a Set-automaton with output in 2 and initialisation in 1.

A central observation in [20] 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

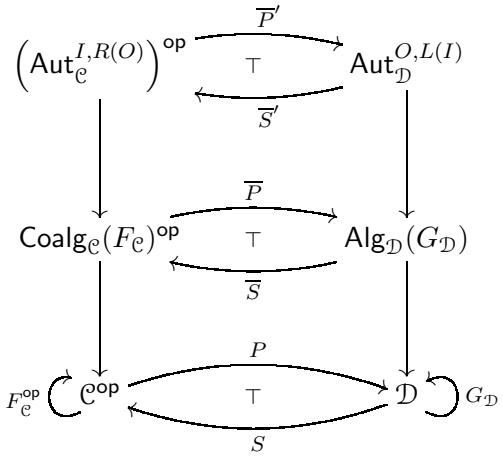
products and coproducts, the transition morphisms $\{t_a \colon X \to X \mid a \in \Sigma\}$ correspond uniquely to morphisms of the following type:

$$\frac{\langle t_a \rangle_{a \in \Sigma} \colon X \to X^\Sigma}{[t_a]_{a \in \Sigma} \colon \Sigma \cdot X \to X} \tag{3.1}$$

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 $\langle f, \langle t_a \rangle_{a \in \Sigma} \rangle \colon X \to B \times X^\Sigma$ with intialisation $i \colon I \to X$. Or equivalently, a G-algebra $[i, [t_a]_{a \in \Sigma}] \colon GX \to X$ with output $f \colon X \to B$.

3.3 Dual Adjunctions of Coalgebras, Algebras and Automata

The categorical picture that unifies the work in [16] and [20] is sketched in the diagram (3.2) 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 [16] 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 [20], which uses automata with initial states. In the remainder of the section, we will explain the details of how this picture comes about.



(3.2)

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

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 op, and treat P and S as contravariant functors. The units of the dual adjunction will be denoted $\eta \colon Id \Rightarrow PS$ and $\varepsilon \colon Id \Rightarrow SP$. The natural isomorphism of Hom-sets $\theta_{X,Y} \colon \mathcal{C}(X, SY) \to \mathcal{D}(Y, PX)$, will sometimes be written in both directions simply as $f \mapsto f^\flat$. For $f \colon X \to SY$, its adjoint is $f^\flat = Pf \circ \eta_Y$, and for $g \colon Y \to PX$, its adjoint is $g^\flat = Sg \circ \varepsilon_X$.

In all our examples, $\mathcal{C}$ and $\mathcal{D}$ are concrete categories, and the dual adjunction arises from homming into a dualising object Δ (cf. [57]), 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 $f\colon Y \to \Delta^X$ we have $f^\flat(x)(y) = f(y)(x)$. Moreover, the units are given by evaluation. E.g. $\eta_X\colon X \to \Delta^{\Delta^X}$ is defined by $\eta_X(x)(f) = f(x)$.

Example 3.2 *A central example is the self-dual adjunction of* Set *given by the* contravariant powerset functor $\mathcal{Q} = \mathsf{Set}(-, 2)$ *which maps a set* X *to its powerset* 2^X *and a function* $f : X \to Y$ *to its inverse image map* $f^{-1}\colon 2^Y \to 2^X$. *The functor* $\mathcal{Q}$ *is dually self-adjoint with* $\mathcal{Q}^{\mathsf{op}} \dashv \mathcal{Q}$, *and the isomorphism of Hom-sets is given by taking exponential transposes, i.e., for* $f\colon X \to 2^Y$ *we have* $f^\flat\colon Y \to 2^X$.

Dual adjunctions are also called *logical connections* as they form the basis of semantics for coalgebraic modal logics [22, 42, 47]. 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 a predicate A to the theories of A. The logic given by $G_\mathcal{D}$ can be interpreted over $F_\mathcal{C}$-coalgebra by providing a so-called one-step modal semantics in the form of a natural transformation $\varrho\colon G_\mathcal{D}P \Rightarrow PF_\mathcal{C}$, or equivalently via its mate $\xi\colon F_\mathcal{C}S \Rightarrow SG_\mathcal{D}$. The pair $(G_\mathcal{D}, \varrho)$ is referred to as a logic. By assuming that the initial $G_\mathcal{D}$-algebra (A_0, α_0) exists, and viewing its elements as formulas, the semantics of formulas in a $F_\mathcal{C}$-coalgebra is (C, γ) is obtained by initiality: $s^{G_\mathcal{D}}\colon (A_0, \alpha_0) \to P(\gamma) \circ \varrho_C$, i.e., as an underlying $\mathcal{D}$-map, it has type $s^{G_\mathcal{D}}\colon A_0 \to P(C)$. Alternatively, the semantics can be specified by the theory map $th^{G_\mathcal{D}}\colon C \to S(A_0)$ which is defined as the adjoint of $s^{G_\mathcal{D}}$. We refer to [22, 42, 47] for a more detailed introduction to coalgebraic modal logic via dual ajdunctions.

3.3.2 Dual adjunction between coalgebras and algebras

We lift the base dual adjunction to coalgebras and algebras using some some basic results from [39, 46]. We assume that $\mathcal{C}$ has products, $\mathcal{D}$ has coproducts, and that we have functors $F_\mathcal{C}$ and $G_\mathcal{D}$ as given above, i.e.,

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

We know from [39, Cor. 2.15] (see also [46, Thm. 2.5]), that the dual base adjunction $S \dashv P$ lifts to a dual adjunction $\overline{S} \dashv \overline{P}$ between $\mathsf{Coalg}_\mathcal{C}(F_\mathcal{C}) = \mathsf{Alg}_{\mathcal{C}^{\mathsf{op}}}(F_\mathcal{C}^{\mathsf{op}})$ and $\mathsf{Alg}_\mathcal{D}(G_\mathcal{D})$ if there is a natural isomorphism $\xi\colon F_\mathcal{C}S \xRightarrow{\sim} SG_\mathcal{D}$. We have for all $D \in \mathcal{D}$,

$$SG_\mathcal{D}(D) = S(O + \Sigma \cdot D) \cong S(O) \times S(D)^\Sigma = F_\mathcal{C}S(D) \tag{3.3}$$

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

$$\varrho = PF_\mathcal{C}\varepsilon \circ P\xi P \circ \eta G_\mathcal{D}P \tag{3.4}$$

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

$$\begin{array}{rcl} \overline{P}(\gamma) & = & P\gamma \circ \varrho_C\colon G_{\mathcal{D}}PC \to PC, \quad \overline{P}(f) = P(f) \\ \overline{S}(\alpha) & = & \xi_D \circ S\alpha\colon SD \to F_{\mathcal{C}}SD, \quad \overline{S}(g) = S(g) \end{array} \tag{3.5}$$

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

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

The natural transformation $\varrho\colon G_{\mathcal{D}}P \Rightarrow PF_{\mathcal{C}}$ 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 Δ in which case Δ is a domain of truth-values, i.e., the logic is Δ-valued, and when $\mathcal{D}$ is category of algebras with operations given by a signature Sgn. The algebra functor $G_{\mathcal{D}} = O + \Sigma \cdot (-)$ then corresponds to a modal language $\mathbb{L}(G_{\mathcal{D}})$ that has atomic propositions from O, 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:

$$\varphi ::= q \in O \mid [a]\varphi, a \in \Sigma \mid \sigma(\Psi), \sigma \in Sgn$$

where Ψ is a set of formulas of cardinality matching the arity of the operation σ.

For our specific choice of functors $F_{\mathcal{C}}$ and $G_{\mathcal{D}}$, and when the adjunction arises from a dualising object Δ, we can compute the concrete definition of ϱ from (3.4) (see Appendix A.2) and we get the following Δ-valued modal semantics of the language $\mathbb{L}(G_{\mathcal{D}})$:

$$\begin{array}{rcll} [\![q]\!](x) & = & j(q), & \text{where } \gamma(x) = \langle j\colon \Delta^O, d\colon X^{\Sigma}\rangle \\ [\![[a]\varphi]\!](x) & = & [\![\varphi]\!](d(a)), & \text{where } \gamma(x) = \langle j\colon \Delta^O, d\colon X^{\Sigma}\rangle \\ [\![\sigma(\Psi)]\!](x) & = & \sigma(\{[\![\psi]\!](x) \mid \psi \in \Psi\}) & \end{array}$$

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

Example 3.3 *We consider the case of classic deterministic automata. Here $\mathcal{C} = \mathcal{D} = \mathsf{Set}$, $F_{\mathsf{Set}} = 2 \times (-)^{\Sigma}$ and $G_{\mathsf{Set}} = 1 + \Sigma \cdot (-)$, and the self-dual adjunction of Set is given by the contravariant powerset functor $\mathcal{Q} = \mathsf{Set}(-, 2)$ (Example 3.2). The logic we obtain is* trace logic *[47], but here interpreted over DFAs rather than labelled transition systems as in [47]. The initial G_{Set}-algebra is Σ^*, the set of finite words over Σ, and these are the formulas $\mathbb{L}(G_{\mathcal{D}})$, since $\mathcal{D} = \mathsf{Set}$ means that there are no propositional connectives. The natural transformation ϱ has type $\varrho_X\colon 1 + \Sigma \cdot 2^X \to 2^{2 \times X^{\Sigma}}$, and is given concretely here together with the induced semantics, where we write $x \Vdash \varphi$ iff $[\![\varphi]\!](x) = 1$:*

$$\begin{array}{rcl} \varrho_X(*) & = & \{(b, d) \in 2 \times X^{\Sigma} \mid b = 1\} \\ \varrho_X(a, U) & = & \{(b, d) \in 2 \times X^{\Sigma} \mid d(a) \in U\} \end{array} \quad \left\| \quad \begin{array}{lcl} x \Vdash * & \iff & x \text{ is accepting} \\ x \Vdash [a]\varphi & \iff & x \xrightarrow{a} y \text{ and } y \Vdash \varphi \end{array} \right.$$

3.3.3 Dual adjunction between automata

In order to obtain the upper adjunction in (3.2) (which formalises Brzozowski), 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 $\overline{S} \dashv \overline{P}$ obtained by defining how an initialisation map $I \to C$ for an $F_{\mathcal{C}}$-coalgebra γ is turned into an observation map $PC \to PI$ for the $G_{\mathcal{D}}$-algebra $\overline{P}(\gamma)$, and vice versa for $\overline{S}$.

Theorem 3.4 *Under the assumptions of section 3.3.2, the dual adjunction $\overline{S} \dashv \overline{P}$ between $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$ and $\mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$ lifts to a dual adjunction $\overline{S}' \dashv \overline{P}'$ between $\mathsf{Aut}_{\mathcal{C}}^{I,SO}$ and $\mathsf{Aut}_{\mathcal{D}}^{O,PI}$ by defining $\overline{P}'$ and $\overline{R}'$ as follows for all $\gamma\colon C \to F_{\mathcal{C}}C$ and $\alpha\colon G_{\mathcal{D}}D \to D$:*

$$\begin{array}{rcll} \overline{P}'(\gamma, i\colon I \to C) & = & (\overline{P}(\gamma)\colon G_{\mathcal{D}}PC \to PC, P(i)\colon PC \to PI), & \overline{P}'(f) = P(f) \\ \overline{S}(\alpha, j\colon D \to PI) & = & (\overline{R}(\alpha)\colon SD \to F_{\mathcal{C}}SD, j^\flat\colon I \to SD), & \overline{S}'(g) = S(g) \end{array}$$

Proof. This is a minor generalisation of Prop. 9.1 in [20]. It suffices to show that for all $\mathcal{C}$-arrows $i\colon I \to C$, and all $\mathcal{D}$-arrows $g\colon D \to PI$ and $h\colon D \to PX$: $g = Pi \circ h$ 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 ε. Conversely, if $g^\flat = h^\flat \circ i$, then $g = Pg^\flat \circ \eta_D = Pi \circ Ph^\flat \circ \eta_D = Pi \circ h$. QED

It is straightforward to verify that for our choice of $F_{\mathcal{C}}$, the final $F_{\mathcal{C}}$-coalgebra exists, and we usually view it as having carrier $S(O)^{\Sigma^*}$, hence for $\gamma\colon C \to F_{\mathcal{C}}(C)$, the final morphism $!_\gamma\colon C \to S(O)^{\Sigma^*}$ assigns to each state in C what can be seen as an $S(O)$-weighted language. For $\mathcal{X} = \langle \gamma, i \rangle \in \mathsf{Aut}_{\mathcal{C}}^{I,S(O)}$, 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 an Σ^*-indexed family of $\mathcal{C}$-morphisms $\langle\!|\mathcal{X}|\!\rangle_w\colon I \to SO$ defined for all $w = a_1 \cdots a_k \in \Sigma^*$ by

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

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

$$\langle\!|\mathcal{X}|\!\rangle_w^\flat \;=\; P(I) \xleftarrow{i} P(X) \xleftarrow{Pt_{a_1}} \cdots \xleftarrow{Pt_{a_k}} P(X) \xleftarrow{f^\flat} O$$

Hence $\langle\!|\mathcal{X}|\!\rangle_w^\flat = \langle\!|\overline{P}'(\mathcal{X})|\!\rangle_{w^R}$ where $w^R = a_k \cdots a_1$ is the reversal of w. Similarly, we find that for all $\mathcal{Y} \in \mathsf{Aut}_{\mathcal{D}}^{O,P(I)}$, $\langle\!|\mathcal{Y}|\!\rangle_w^\flat = \langle\!|\overline{S}'(\mathcal{Y})|\!\rangle_{w^R}$. In the case of classic DFAs from Example 3.3 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.

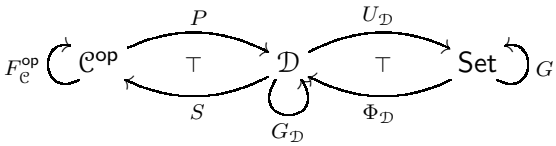
3.4 Language Semantics and Trace Logic

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 [16], 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 Set with adjunction $\Phi_{\mathcal{D}} : \mathcal{D} \rightleftarrows \mathsf{Set} : U_{\mathcal{D}}$ This adjoint situation allows us to relate the Set-based language semantics to the final $F_{\mathcal{C}}$-coalgebra semantics as we will show now.

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

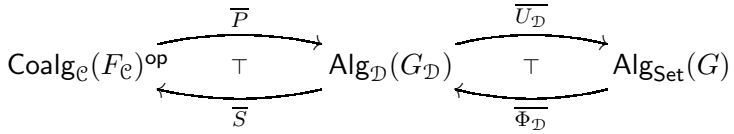


(3.6)

Lemma 3.5 *Assume we have the situation in* (3.6)*, and that $F_{\mathcal{C}}$, $G_{\mathcal{D}}$, G are defined by:*

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

Then (3.6) *lifts to*



(3.7)

Proof. The dual adjunction on the left lifts because of a special case of (3.3). For similar reasons, the adjunction on the right lifts, because there is a natural isomorphism $\kappa : \Phi_{\mathcal{D}}G \xRightarrow{\sim} G_{\mathcal{D}}\Phi_{\mathcal{D}}$ that can be obtained as follows

$$\kappa\colon \Phi_{\mathcal{D}}GX = \Phi_{\mathcal{D}}(\Omega + \Sigma \cdot X) \cong \Phi_{\mathcal{D}}(\Omega) + \Sigma \cdot \Phi_{\mathcal{D}}(X) = G_{\mathcal{D}}\Phi_{\mathcal{D}}(X), \tag{3.8}$$

since $\Phi_{\mathcal{D}}$ (being a left adjoint) preserves colimits. By [39, Thm. 2.14], $\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}$ lifts to an adjunction between $\overline{\Phi_{\mathcal{D}}} \dashv \overline{U_{\mathcal{D}}}$ between $\mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$ and $\mathsf{Alg}_{\mathsf{Set}}(G)$ where the functor $\overline{\Phi_{\mathcal{D}}}$ maps a G-algebra (X, α) 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

$$\xi^{\mathsf{trc}} \coloneqq S\kappa \circ \xi\Phi_{\mathcal{D}} : F_{\mathcal{C}}S\Phi_{\mathcal{D}} \xRightarrow{\sim} S\Phi_{\mathcal{D}}G \tag{3.9}$$

where $\xi\colon F_{\mathcal{C}}S \xRightarrow{\sim} SG_{\mathcal{D}}$ from (3.3) is the mate of the modal logic $(G_{\mathcal{D}}, \varrho)$. Hence by [39, Thm. 2.14,Cor. 2.15] (see also [46, Thm. 2.5]), the adjunction $S\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}P$ lifts to one between $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})^{\mathsf{op}}$ and $\mathsf{Alg}_{\mathsf{Set}}(G)$. QED

Letting $\varrho^{\mathsf{trc}}\colon GU_{\mathcal{D}}P \Rightarrow U_{\mathcal{D}}PF_{\mathcal{C}}$ be the mate of ξ^{trc} from (3.9), then $(G, \varrho^{\mathsf{trc}})$ is a modal logic for $F_{\mathcal{C}}$-coalgebras. Since its formulas are the elements of the intial G-algebra of traces, we refer to $(G, \varrho^{\mathsf{trc}})$ as a trace logic.

Lemma 3.6 *The theory maps th^G and $th^{G_{\mathcal{D}}}$ of the logics $(G, \varrho^{\mathsf{trc}})$ and $(G_{\mathcal{D}}, \varrho)$ coincide.*

Proof. Due to the adjunctions in (3.6), the intial 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. A more detailed argument is given in Appendix A.3. QED

Since the mates ξ and ξ^{trc} are both natural isomorphisms, it follows from [42, 47] (and $\mathcal{C}$ having a suitable factorisation system, cf. Theorem 3.10) that the full modal logic $(G_{\mathcal{D}}, \varrho)$ and trace logic $(G, \varrho^{\mathsf{trc}})$ 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 epxressive power to $\mathbb{L}(G) = \Sigma^*\Omega$. In summary, we arrive at the following proposition.

Proposition 3.7 *With the above assumptions, the trace logic $(G, \varrho^{\mathsf{trc}})$ and the full logic $(G_{\mathcal{D}}, \varrho)$ are equally expressive over $F_{\mathcal{C}}$-coalgebras, meaning that for all $F_{\mathcal{C}}$-coalgebras $\gamma\colon C \to F_{\mathcal{C}}(C)$, 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 $(G, \varrho^{\mathsf{trc}})$ 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 $\sigma\colon S\Phi_{\mathcal{D}}(\Omega)^{\Sigma^*} \xrightarrow{\sim} S\,\Phi_{\mathcal{D}}(\Sigma^*\Omega)$ which links the language semantics in the automata/coalgebraic sense with trace logic semantics given by initiality.

Proposition 3.8 *For all $F_{\mathcal{C}}$ coalgebras γ, 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 $\overline{S}\,\overline{\Phi_{\mathcal{D}}}(\Sigma^*\Omega)$, (and with the theory map $th^{G_{\mathcal{D}}}$) via the isomorphism σ.*

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 $\tau\colon F'_{\mathcal{C}} \Rightarrow F_{\mathcal{C}}$ which is abstract mono, i.e., all components are mono. Assume that $\mathcal{C}$ has factorisation system (E, M) with $E \subseteq Epi$ and $M \subseteq Mono$. Defining $\xi' = \xi^{\mathsf{trc}} \circ \tau_S$, then $\xi'\colon F'_{\mathcal{C}}S\Phi_{\mathcal{D}} \Rightarrow S\Phi_{\mathcal{D}}G$ defines semantics of trace formulas over $F'_{\mathcal{C}}$-coalgebras which is essentially the same as the semantics over $F_{\mathcal{C}}$-coalgebras. Since τ is abstract mono and ξ^{trc} is a natural iso, it follows that ξ' is abstract mono, and hence the asscoiated logic is expressive [42, 47].*

3.5 Reachability and Observability

A main point emphasised in [20] is that reachability is an algebraic concept, and observability is a coalgebraic concept, and both concepts apply to automata as they are both coalgebras and algebras. We will call an algebra *reachable* if it has no proper subalgebras, and a coalgebra is *observable* if it has no proper quotients.

Both [16] and [20] use that a reachable algebra dualises to an observable coalgebra, only the perspectives differ. Note that in [16], observable coalgebras are referred to as minimal automata. In [16], the reachable part of a $G_{\mathcal{D}}$-algebra is defined as its least subalgebra, and its existence was ensured by assuming that $\mathcal{C}$ is wellpowered. In [20], automata were generally considered as automata over 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 Appendix A.4, we show that the two reachability notions in [16] and [20] coincide when conditions for both are satisfied.

In [16], the least $G_{\mathcal{D}}$-algebra of the dual $G_{\mathcal{D}}$-algebra $\overline{P}(\gamma)$ was characterised as the subalgebra α_{Def} of $\mathbb{L}(G_{\mathcal{D}})$-definable subsets of C (or more abstractly Δ-valued predicates on C). It was observed that α_{Def} is generated by subsets definable by trace formulas on (C,γ). The general statement of this fact is Proposition 3.7. Hence to compute α_{Def}, it suffices to compute trace logic definable subsets. In the case of classic DFA, these subsets are precisely the reachable states (in the usual transition-sense) of $\overline{P}(\gamma)$.

The general setup described in Lemma 3.5 is most closely related to that of [20], as we have an initial G-algebra and an initial $G_{\mathcal{D}}$-algebra. The latter is mapped by $\overline{S}$ to a final $F_{\mathcal{C}}$-coalgebra since the dual adjoint functors turn colimits into limits. For this reason, $\overline{S}$ maps epis to monos, but monos are not necessarily mapped to epis. In particular, we cannot argue that a least subalgebra of $\overline{P}(\gamma)$ is mapped by $\overline{S}$ to a largest quotient of γ. But using factorisation and existence of an initial $G_{\mathcal{D}}$-algebra, we obtain that the reachable part of $\overline{P}(\gamma)$ is mapped by $\overline{S}$ to an observable coalgebra.

Proposition 3.9 *Under the assumptions of Lemma 3.5, and assuming further that $\mathcal{D}$ has a factorisation system (E,M) such that $E \subseteq Epi$ and $M \subseteq Mono$, we then have:*

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

$$\overline{\Phi_{\mathcal{D}}}(\Sigma^*\Omega,\alpha) \overset{e}{\twoheadrightarrow} reach(D,\delta) \overset{m}{\hookrightarrow} (D,\delta).$$

Then $\overline{S}(reach(D,\delta))$ is an observable $F_{\mathcal{C}}$-coalgebra.

Proof. The epimorphism $e\colon \overline{\Phi_{\mathcal{D}}}(\Sigma^*\Omega,\alpha) \twoheadrightarrow reach(D,\delta)$ is mapped by $\overline{S}$ to a monomorphism

$$\overline{S}(e)\colon \overline{S}(reach(D,\delta)) \hookrightarrow \overline{S\Phi_{\mathcal{D}}}(\Sigma^*\Omega,\alpha).$$

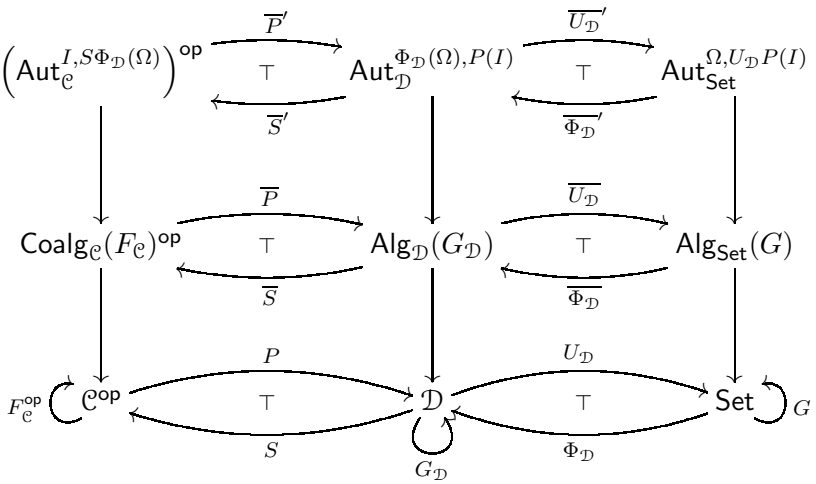
Since $\overline{S\Phi_{\mathcal{D}}}(\Sigma^*\Omega,\alpha)$ is a final $F_{\mathcal{C}}$-coalgebra, we can conclude that $\overline{S}(reach(D,\delta))$ is an observable $F_{\mathcal{C}}$-coalgebra. QED

The above proposition thus tells us how to obtain an observable $F_{\mathcal{C}}$-coalgebra, and hence also an observable $\mathcal{C}$-automaton, by taking the reachable part on the dual side.

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. Brzozowski's algorithm produces a minimal $\mathcal{C}$-automaton by 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 that also $\mathcal{C}$ has an suitable factorisation system.

3.6 Abstract minimisation algorithms

We now put everything together into one diagram with which we can describe both approaches from [16] and [20] including the role of trace logic.



(3.10)

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

Theorem 3.10 *Let $\mathcal{C}, \mathcal{D}$ be concrete categories, both having products and coproducts, and both having factorisation systems (E, M) such that $E \subseteq Epi$ and $M \subseteq Mono$. Let Ω be a finite set (of observations), and I an (initialisation) object in $\mathcal{C}$, and assume that we have the adjoint situation between $\mathcal{C}$, $\mathcal{D}$, Set and functors as described at the bottom level of* (3.10)*. Then the lower adjunctions lift to the upper two levels in* (3.10) *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 γ, compute $\overline{S}(reach(\overline{P}(\gamma)))$ which will be an observable $F_{\mathcal{C}}$-coalgebra.*

Algo2 *Given a $\mathcal{C}$-automaton (γ, i), compute $reach(\overline{S}'(reach(\overline{P}'(\gamma, i))))$, 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. Furthermore, we note that the one could consider an algorithm that uses the horizontally composed adjunction in (3.10), i.e., compute $\overline{S\Phi_{\mathcal{D}}}(\mathit{reach}(U_{\mathcal{D}}\overline{P}(\gamma)))$. Although, the result will be an observable coalgebra, this is, however, not a good choice in general, because the reachable part is now computed over Set, and this may yield an infinite coalgebra/automaton whereas it might have been finitely generated as a coalgebra/automaton over $\mathcal{D}$. An example where this happens is found in Example 8.3 of [20].

Although [16] does not describe a concrete algorithm, the implicit abstract algorithm is essentially **Algo1**, since the conceptual emphasis is placed on computing the least $G_{\mathcal{D}}$-subalgebra of $\overline{P}(\gamma)$ as the subalgebra of (G_D, ϱ)-definable subsets/predicates. The characterisation of this $G_{\mathcal{D}}$-subalgebra as being freely generated by the least G-subalgebra of $\overline{U_{\mathcal{D}}P}(\gamma)$ (i.e. the reachable part in the usual set-theoretic sense) can be viewed as an optimisation: to determine the reachable part of a given $G_{\mathcal{D}}$-algebra it suffices to compute the part that can be "reached"/defined via trace formulas. This is the information contained in the right-hand side of the diagram.

In comparison, Brzozowski's algorithm and its generalisation to weighted automata in Section 4.2 are instances of **Algo2** as they use initial states. Classic Brzozowski is the case $\mathcal{C} = \mathcal{D} = \mathsf{Set}$, $G_{\mathcal{D}} = G$, and $\Omega = I = 1$. The set-based algorithm for weighted automata in [20] is neither of the above algorithms, but it can be described as constructing $\mathit{reach}(\overline{U_{\mathcal{D}}}'\overline{P}'(\gamma, i))$, and then dualise back (without going through $\mathbb{S}\mathsf{Mod}$) to get a Set-based Moore automaton. As mentioned above, this may result in the reachable part of the reversed automaton being infinite.

In the case where the dual adjunction is a full duality, 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 the dual adjunction is not a full duality, the transformation of the initial state goes via the adjunction, and factorisation on the dual side, and this is what Theorem 3.4 formalises.

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 Set meaning that they are equivalent to an Eilenberg-Moore category $\mathsf{EM}(T)$ for a Set-monad T. For such a category $\mathsf{EM}(T)$, we know that it is complete, cocomplete and exact [24, Thm 4.3.5]. W.r.t factorisation systems, $(\mathit{Epi}, \mathit{Mono})$ is generally not a factorisation system for $\mathsf{EM}(T)$, rather $(\mathit{RegEpi}, \mathit{Mono})$ is. Using that regular epis are the surjective morphisms, and monos are the injective morphisms, one can prove that in $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$ and $\mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}})$ the surjective and injective morphisms form a factorisation system. We refer to Lemma A.2.

4 Revisiting Examples

4.1 Deterministic Kripke Models

A central example from [16] 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.

Definition 4.1 *We define* deterministic Kripke models *to be quintuples* $\mathcal{S} = (S, \Sigma, \Omega, \gamma : S \to 2^\Omega, \delta : S \to S^\Sigma)$ *where* S *is a finite set of* states, Σ *is a finite set of* actions, Ω *is a finite set of* observations, δ *is a* transition function *and* γ *is an* observation function. *A function* $f : S_1 \to S_2$ *is a morphism between Kripke models* $(S_1, \Sigma, \Omega, \gamma_1, \delta_1)$ *and* $(S_2, \Sigma, \Omega, \gamma_2, \delta_2)$ *if for all* $s \in S_1$ *and all* $a \in \Sigma$ *we have* $\gamma_1(s) = \gamma_2(s)$ *and* $f(\delta_1(s)(a)) = \delta_2(f(s))(a)$. *We let* DKM *denote 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.

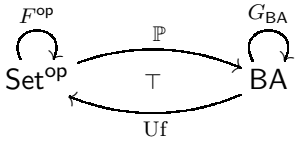
Definition 4.2 *The category* BAO *of (deterministic)* Boolean algebras with operators *(BAOs) has as objects Boolean algebras* B *with the usual operators* $\wedge$ *and* $\neg$ *with a greatest element* $\top$ *and least element* $\bot$ *together with unary operators* $(a) : B \to B$, *for each action* $a \in \Sigma$, *such that* (a) *is a Boolean homomorphism. For each observation* $\omega \in \Omega$, *we also have constants* $\underline{\omega}$. *We denote an object of* BAO *by*

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

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. [19, 36]).

Fact 4.3 *There is a dual adjunction between* Set *and* BA *as depicted in Figure 1 given by the contravariant functor* $\mathbb{P}$ *that maps a set to its Boolean algebra of subsets and the functor* $\mathrm{Uf} := \mathrm{Hom}(-, \mathbf{2})$, *ie., the contravariant functor the maps a Boolean algebra to its collection of ultrafilters.This adjunction restricts to a dual equivalence between the category* FSet *of finite sets and the category* FBA *of finite Boolean algebras.*



$F(X) = 2^{\Omega} \times X^{\Sigma}$ $\quad$ $G_{\mathsf{BA}}(X) = \Phi_{\mathsf{BA}}(\Omega) + \Sigma \cdot X$
$I = 1$ $\quad$ $\mathrm{Uf}(\Phi_{\mathsf{BA}}(\Omega)) \cong 2^{\Omega}$

Figure 1: Functors and base adjunction for deterministic Kripke frames

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 4.4 *We have the following equivalences:*

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

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_{BA}-algebras and BAOs. As a consequence of the proposition we obtain the following duality results by applying our general framework.

Proposition 4.5 *The dual adjunction* $\mathrm{Uf} \dashv \mathbb{P}$ *lifts to a dual adjunction between* DKM *and* BAO *and to an adjunction between* $\mathsf{Aut}^{1,2^{\Omega}}_{\mathsf{Set}}$ *and* $\mathsf{Aut}^{2,F\Phi_{\mathsf{BA}}(\Omega)}_{\mathsf{BA}}$. *If we start with the dual equivalence* $\mathsf{FSet} \cong \mathsf{FBA}$, *both liftings are dual equivalences as well.*

Proof. For the dual adjunction between DKM and BAO recall from Proposition 4.4 that both categories are equivalent to categories of F-coalgebras and G_{BA}-algebras for certain functors F and G_{BA}, respectively. Furthermore, we have $\mathrm{Uf}(\Phi_{\mathsf{BA}}(\Omega)) \cong 2^{\Omega}$, which follows from the well-known fact that the set of homomorphisms of type $\Phi_{\mathsf{BA}}(\Omega) \to \mathbf{2}$ (i.e., ultrafilters) is in one-one correspondence with the set of functions of type $\Omega \to 2$. Therefore the functors F and G_{BA} have the shape required by our general lifting result from Section 3.3.2 and we obtain functors $\overline{\mathbb{P}} : \mathsf{Coalg}(F)^{\mathsf{op}} \to \mathsf{Alg}(G_{\mathsf{BA}})$ and $\overline{Uf} : \mathsf{Alg}(G_{\mathsf{BA}}) \to \mathsf{Coalg}(F)^{\mathsf{op}}$ with $\overline{Uf} \dashv \overline{\mathbb{P}}$.

To extend the adjunction $\overline{Uf} \dashv \overline{\mathbb{P}}$ between $\mathsf{Coalg}(F)$ and $\mathsf{Alg}(G_{\mathsf{BA}})$ further to a dual adjunction $\overline{Uf}' \dashv \overline{\mathbb{P}}'$ between $\mathsf{Aut}^{1,2^{\Omega}}_{\mathsf{Set}}$ and $\mathsf{Aut}^{2,\Phi_{\mathsf{BA}}(\Omega)}_{\mathsf{BA}^{\mathsf{op}}}$ - the latter is a slight extension of the former by adding a initial state to deterministic Kripke models and by viewing BAOs as some kind of automata with acceptance predicate - 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 Set and BA with FSet and FBA, respectively, is a matter of routine checking. QED

This shows, in particular, that we get a duality between finite deterministic Kripke models and FBAO s. This is the key for obtaining a minimal realization via logical theories.

Definition 4.6 *Consider the language* $\mathbb{L}(G_{\mathsf{BA}})$*:*

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

with semantics defined as below (A.1) *on page 39. For a given automaton* $\mathcal{S} = (S, \gamma, \delta)$ *we say that a subset* U *of* S *is* definable by $\mathbb{L}(G_{\mathsf{BA}})$ *if* $U = [\![\varphi]\!]$*for some* $\varphi \in \mathbb{L}(G_{\mathsf{BA}})$ *where we identify the predicate* $[\![\varphi]\!] : S \to 2$ *with the set* $\{s \in S \mid [\![\varphi]\!](s) = 1\}$ *. We let* $\mathrm{Def}(\mathcal{S}) = (\mathrm{Def}(S), \{(a)_{\mathcal{S}}\}_{a\in\Sigma}, \{[\![\hat{\omega}]\!]\}_{\omega\in\Omega})$ *be the BAO based on the definable subsets of* $\mathcal{S}$*, where* $(a)_{\mathcal{S}}([\![\varphi]\!]) = [\![[a]\varphi]\!]$*.*

In other words, the modal logic is the modal logic with (deterministic) Σ-indexed modalities and atomic propositions from Ω. For a given deterministic Kripke model, the algebra of definable subsets yields the reachable part (=zero generated subalgebra) of the dual automaton (=algebra).

Proposition 4.7 *Let* $\mathcal{S} = (S, \gamma, \delta) \in \mathsf{DKM}$ *be a deterministic Kripke model,* $s_I \in S$ *an initial state and let* $\mathrm{Def}(\mathcal{S}) \in \mathsf{BAO}$ *the dual algebra of definable subset. Then* $\mathrm{Def}(\mathcal{S}) \in \mathsf{BAO}$ *is isomorphic to the reachable part reach*$(\overline{\mathbb{P}}(\mathcal{S}))$ *of* $\overline{\mathbb{P}}(\mathcal{S}) \in \mathsf{BAO}$*.*

Proof. The result follows from the fact that the unique morphism $i_{\mathrm{Def}}(\mathcal{S})$ from the initial G_{BA}-algebra $\mathcal{I}$ to $\mathrm{Def}(\mathcal{S})$ together with the embedding $m : \mathrm{Def}(\mathcal{S}) \to \overline{\mathbb{P}}(\mathcal{S})$ is the image-factorisation of $i_{\overline{\mathbb{P}}(\mathcal{S})} : \mathcal{I} \to \overline{\mathbb{P}}(\mathcal{S})$, which is the (E, M)-factorisation obtained from the factorisation system of surjective and injective Boolean homomorphisms (cf. Lemma A.2). QED

Definition 4.8 *We call a formula* φ *of the form* $[a_1] \ldots [a_n]\omega$ *for some* $n \in \mathbb{N}$ *and* $a_i \in \Sigma$ *for* $i \in \{1, \ldots, n\}$ *a trace formula. For a DKM* $\mathcal{S}$ *we denote by* $\mathrm{Def}^*(\mathcal{S})$ *the collection of trace-definable subsets of* $\mathcal{S}$*, i.e, the collection of subsets definable by a trace formula.*

Proposition 4.9 *For all DKMs* $\mathcal{S}$ *we have*

$$\overline{\mathrm{Uf}}(\mathrm{reach}(\overline{\mathbb{P}}(\mathcal{S}))) \cong (\overline{\mathrm{Uf}}(\mathrm{Def}(\mathcal{S})) \cong (\overline{\mathrm{Uf}}(\overline{\Phi}_{\mathsf{BA}}(\mathrm{Def}^*(\mathcal{S}))).$$

Proof. By the results in Section 3.4, we have $\Phi_{\mathsf{BA}}\Sigma^*\Omega$ is isomorphic to the Lindenbaum algebra of $\mathbb{L}(G_{\mathsf{BA}})$. Therefore the reachable part of $\overline{\mathbb{P}}(\mathcal{S})$ is obtained as the image of $\Phi_{\mathsf{BA}}\Sigma^*\Omega$ under the initial morphism from $\Phi_{\mathsf{BA}}\Sigma^*\Omega$ to $\overline{\mathbb{P}}(\mathcal{S})$ which can be easily checked to be $\Phi_{\mathsf{BA}}(\mathrm{Def}^*(\mathcal{S}))$. QED

We finish with a key observation from [16] that allows to compute quotients of finite DKMs via duality.

Corollary 4.10 *Given a finite DKM* $\mathcal{S}$*, the quotient of* $\mathcal{S}$ *modulo bisimulation is isomorphic to* $(\overline{\mathrm{Uf}}(\overline{\Phi}_{\mathsf{BA}}(\mathrm{Def}^*(\mathcal{S}))))$*.*

Proof. By Prop 3.9 we have that $(\overline{\mathrm{Uf}}(\mathrm{Def}(\mathcal{S}))$ and thus $(\overline{\mathrm{Uf}}(\overline{\Phi}_{\mathsf{BA}}(\mathrm{Def}^*(\mathcal{S})))$ is observable. As FSet and FBA are dually *equivalent*, we get that $(\overline{\mathrm{Uf}}(\mathrm{Def}(\mathcal{S}))$ and thus $(\overline{\mathrm{Uf}}(\overline{\Phi}_{\mathsf{BA}}(\mathrm{Def}^*(\mathcal{S})))$ are the maximal quotient of $\mathcal{S}$. QED

4.2 Weighted Automata

4.2.1 Semirings and semimodules

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:

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

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 $\mathbb{S} \times M \to M$ 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:

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

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 $h \colon M_1 \to M_2$ 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 a $\mathbb{S}$-semimodule M_1 to M_2 is denoted by $\mathbb{S}\mathsf{Mod}(M_1, M_2)$.

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

$$\begin{array}{rcl} \mathcal{V}_{\mathbb{S}}(X) & = & \{\, \varphi \colon X \to \mathbb{S} \mid \varphi \text{ has finite support} \,\}, \\ \mathcal{V}_{\mathbb{S}}(h(\varphi)) & = & \big(y \mapsto \textstyle\sum_{x \in h^{-1}(y)} \varphi(x)\big), \end{array}$$

where a function $\varphi \colon X \to \mathbb{S}$ 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:

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

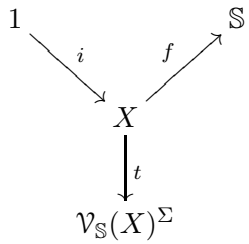
We sometimes write the elements of $\mathcal{V}_{\mathbb{S}}(X)$ as formal sums $s_1 x_1 + \ldots + s_n x_n$ with $s_i \in \mathbb{S}$ and $x_i \in X$. $\mathcal{V}_{\mathbb{S}}(X)$ is a monad and the category of Eilenberg-Moore algebras is $\mathbb{S}\mathsf{Mod}$, the category of $\mathbb{S}$-semimodules and $\mathbb{S}$-linear maps. As usual, free $\mathbb{S}$-semimodules enjoy the following universal property: for every function $h \colon X \to M$ from a set X to a

semimodule M, there exists a unique linear map $h^\sharp \colon \mathcal{V}_\mathbb{S}(X) \to M$ that is called the *linear extension* of h.

We can define for an $\mathbb{S}$-semimodule M its *dual space* $M^\star$ to be the set $\mathbb{S}\mathsf{Mod}(M, \mathbb{S})$ 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 $\mathbb{S} \cong V(1)$ and that $\mathbb{S}^\star = \mathbb{S}\mathsf{Mod}(\mathbb{S}, \mathbb{S}) \cong \mathbb{S}$.

4.2.2 Weighted automata and weighted languages

A *weighted automaton* with finite input alphabet Σ and weights over a semiring $\mathbb{S}$ is given by a set of states X, a function $t \colon X \to \mathcal{V}_\mathbb{S}(X)^\Sigma$ (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 $f \colon X \to \mathbb{S}$ associating an output weight with every state, and an initial state function $i \colon 1 \to \mathcal{V}_\mathbb{S}(X)$. Diagrammatically:

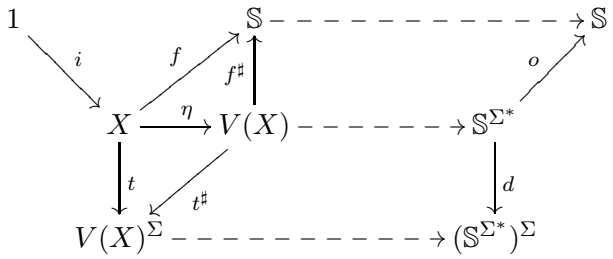


The function $t \colon X \to \mathcal{V}_\mathbb{S}(X)^\Sigma$ can be inductively extended to words $w \in \Sigma^*$:

$$\begin{aligned} t(x)(\varepsilon) &= 1.x \\ t(x)(aw) &= v_1 t(x_1)(w) + \cdots + v_n t(x_n)(w), \text{ where } t(x)(a) = v_1 x_1 + \cdots + v_n x_n \end{aligned}$$

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 $L(\mathcal{X}) \colon \Sigma^* \to \mathbb{S}$ that maps $w \in \Sigma^*$ to $f(t(i)(w)) \in \mathbb{S}$. More concretely, the value $L(\mathcal{X})(w)$, for $w = a_1 a_2 \cdots a_n$, is the sum of all $v_1 \cdot \ldots \cdot v_n \cdot f(x_{n+1})$ over all paths $p_w = x_1 \xrightarrow{a_1, v_1} \ldots \xrightarrow{a_n, v_n} x_{n+1}$ labelled by w. The value of $L(\mathcal{X})(w)$ can be easily computed using the usual matrix representation of linear maps: the initial state function i is then a column vector, the final state function f is a row vector, and the transition relation t can be represented as a Σ-indexed collection of $X \times X$-matrices t_a where $t_a(y, x) = t(x)(a)(y)$ for all $x, y \in X$. $L(\mathcal{X})(w)$ is then obtained by the following matrix multiplication $f \times t_{a_n} \times \ldots \times t_{a_0} \times i$.

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 $\langle f, t \rangle : X \to \mathbb{S} \times (\mathcal{V}_\mathbb{S} X)^\Sigma$ into a Moore automaton over $\mathbb{S}\mathsf{Mod}$.

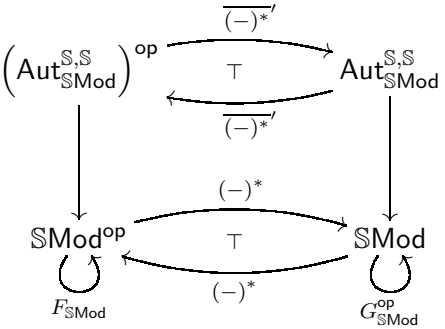


The unique map into final Moore automaton of weighted languages gives precisely the language semantics concretely given above.

4.2.3 Brozowski for Weighted Automata

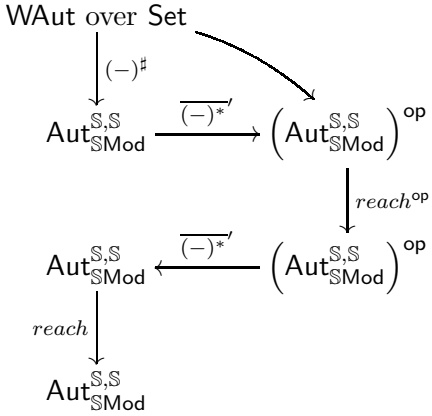
There is self-dual adjunction of $\mathbb{S}\mathsf{Mod}$ obtained by taking dual space: $(-)^* = \mathbb{S}\mathsf{Mod}(-, \mathbb{S})$. A special case is the self-dual adjunction of vector spaces in case $\mathbb{S}$ is a field, which restricts to a duality between finite-dimensional vector spaces. This duality was used in [16] to obtain observable Moore automata over vector spaces.

We lift the base adjunction to one between Moore automata in $\mathbb{S}\mathsf{Mod}$ using Theorem 3.4. Let $\mathcal{C} = \mathcal{D} = \mathbb{S}\mathsf{Mod} = \mathsf{EM}(\mathcal{V}_\mathbb{S})$ and $F_{\mathbb{S}\mathsf{Mod}}(X) = \mathbb{S} \times X^\Sigma$ and $G_{\mathbb{S}\mathsf{Mod}}(X) = \mathbb{S} + \Sigma \cdot X$. Since $\mathbb{S}^* \cong \mathbb{S}$, the conditions for Theorem 3.4 hold, and the adjunction lifts, as illustrated here:



We can now give the Brzozowski algorithm for weighted automata by instantiating **Algo2** of Theorem 3.10 for the determinised automaton. Start with a weighted automaton in Set, determinise it into a Moore automaton in $\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}}$ (to have a canonical representative of the accepted language), reverse and determinise, take the reachable part (w.r.t $G_{\mathbb{S}\mathsf{Mod}}$-structure over $\mathbb{S}\mathsf{Mod}$), reverse and determinise, take the reachable part again. Diagramatically, **Algo2** is (putting $^{\mathsf{op}}$ on the right-hand side to start and

end in $\mathsf{Aut}^{\mathbb{S},\mathbb{S}}_{\mathbb{S}\mathsf{Mod}}$):



At this point we have built a minimal Moore automaton accepting the same language as the weighted automaton 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 in Set, with as state space the generators of the state space of the minimal Moore automaton resulting after applying our Brzozowski algorithm. 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 is actually a weighted automaton in Set. Fortunately, we know from a result of Tan [71] 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 ([71, Theorem 4.3]). Furthermore, because N is free, it follows that it is also finitely generated and of rank smaller than that of M [([71, 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.

4.3 Topological Automata via Gelfand Duality

A very popular model heavily used in reforcement 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 on S. When seeking to minimize this using duality [16], 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 *sub*distributions on X endowed with the relative topology when viewed as a subset of $[0,1]^X$. This is a compact Hausdorff space.

Definition 4.11 *A **compact Hausdorff automaton** is a* 5*-tuple*

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

where S is a compact Hausdorff space, Σ is a finite set of actions or inputs, Ω is a finite set of observations, t is a continuous *transition function and f is a continuous observation function.*

We recall a few basic facts about C^*-algebras, and refer to [13, 18, 43, 61] 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 real numbers.

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 $(x^*)^*$, 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 $\mathsf{CUC^*Alg}$ the category of unital, commutative, real-valued C^*-algebras and their homomorphisms.

In [56] it was shown [1] that U has a left adjoint $M\colon \mathsf{Set} \to \mathsf{CUC^*Alg}$ given by

$$M(X) = C([0,1]^X) = \{f\colon [0,1]^X \to \mathbb{R} \mid f \text{ continuous}\} \tag{4.1}$$

$$M(g\colon X \to Y) = f(v \circ g) \quad \text{where } f \in C([0,1]^X), v \in [0,1]^Y. \tag{4.2}$$

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

We denote by KHaus the category of compact Hausdorff spaces and continuous maps. Given a compact Hausdorff space X, the hom-set $C(X) = \mathrm{Hom}_{\mathsf{KHaus}}(X, \mathbb{R})$ becomes a commutative, unital, real-valued C^*-algebra by defining operations pointwise. In particular, the unit is the constantly 1 map, and for $f \in C(X)$, and the norm is $\|f\| = \sup\{|f(x)| \mid x \in X\}$; recall that for a compact space and a continuous function

[1] Strictly speaking, she showed it for complex-valued C^*-algebras, but the result also holds for real-valued ones.

the supremum is attained. For a morphism $g\colon X \to Y$ in KHaus, defining $C(g)(h) = h \circ g$ makes $C(-)$ a functor from KHaus to $\mathsf{CUC}^*\mathsf{Alg}^{\mathsf{op}}$.

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

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

$$\mathsf{KHaus} \underset{\mathit{Spec}}{\overset{C}{\cong}} \mathsf{CUC}^*\mathsf{Alg}^{\mathsf{op}} \tag{4.3}$$

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 isomomorphisms. The unit $\eta_A\colon A \to C(\mathit{Spec}(A))$ is known as the *Gelfand transform*, and is given by $\eta_A(x)(\Phi) = \Phi(x)$. For all $A \in \mathsf{CUC}^*\mathsf{Alg}$, ϵ_A is an isometric isomorphism in $\mathsf{CUC}^*\mathsf{Alg}$.

We will lift the base dual adjunction between KHaus and $\mathsf{CUC}^*\mathsf{Alg}$ to an adjunction between the category CHA of compact Hausdorff automata and CAO of $\mathsf{CUC}^*\mathsf{Alg}$-automata. These are obtained from F-coalgebras and G-algebras, respectively, where

$$\begin{array}{ll} F\colon \mathsf{KHaus} \to \mathsf{KHaus}, & F(X) = \mathcal{D}_{\leq 1}(\Omega) \times X^{\Sigma} \\ G\colon \mathsf{CUC}^*\mathsf{Alg} \to \mathsf{CUC}^*\mathsf{Alg}, & G(A) = M(\Omega)/J + \Sigma \cdot A \end{array} \tag{4.4}$$

where $\mathcal{D}_{\leq 1}(\Omega)$ is the set of discrete, subdistributions on the set Ω of observations equipped with the relative topology viewed as a subset $\mathcal{D}_{\leq 1}(\Omega) \subseteq [0,1]^{\Omega}$. This makes $\mathcal{D}_{\leq 1}(\Omega)$ a compact Hausdorff space. Recall from (4.1) that M is the left adjoint of the unit interval functor U. Finally, $J \subseteq M(\Omega)$ is an ideal of the $\mathsf{CUC}^*\mathsf{Alg}$-algebra $M(\Omega)$ which we describe in a moment. Note that $\mathsf{CUC}^*\mathsf{Alg}$ has coproducts. This can readily be seen from the fact that KHaus has products and using Gelfand duality.

In order to lift the base dual adjunction $\mathit{Spec} \dashv C$ to a dual adjunction between $\mathsf{Coalg}(F)$ and $\mathsf{Alg}(G)$ 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 $M(Y) \in \mathsf{CUC}^*\mathsf{Alg}$ defined by $[0,1]^Y$. For each $y \in Y$, we have a projection map $\pi_y \in M(Y) \to \mathbb{R}$ given by $\pi_y(v) = v(y)$. Let $\pi = \sum_{y \in Y} \pi_y$. Then $\pi\colon [0,1]^Y \to \mathbb{R}$ 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:

$$\begin{array}{rcl} \pi \preceq 1 & \iff & \pi \vee 1 = 1 \\ & \iff & \frac{1}{2}(\pi + 1) + \frac{1}{2}|\pi + 1| = 1 \\ & \iff & |1 - \pi| = 1 - \pi \end{array}$$

Definition 4.12 *We define the ideal J of $M(Y)$ as the principal ideal generated by the element $(|\pi^-| - \pi^-)$ where $\pi^- := 1 - \pi$. That is,*

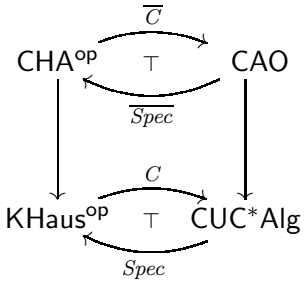
$$J = \{m \in M(Y) \mid \exists k \in M(Y) : m = k(|\pi^-| - \pi^-)\}.$$

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 $\equiv_J$.

The rather technical proof of the following isomorphism lemma is in Appendix A.6.

Lemma 4.13 *For any set Y, $\mathcal{D}_{\leq 1}(Y) \cong Spec(M(Y)/J)$ in* KHaus.

From Lemma 4.13 and section 3.3.2, it follows that the base dual adjunction lifts to one between F-coalgebras and G-algebras.



The abstract algorithm **Algo1** applies since KHaus and $\mathsf{CUC}^*\mathsf{Alg}$ are monadic over Set (cf. Section 3.6) In particular, KHaus is the Eilenberg-Moore category of the ultrafilter monad [54]. In order to show that the associated trace logic is expressive we need an extra argument, since the functor F defined in (4.4) does not have the shape required by Lemma 3.5 and Theorem 3.10. However, we can apply Remark 2 after observing the following. Let $F' := Spec(M(\Omega)) \times (-)^{\Sigma}$. Then the associated natural isomorphism $\xi^{\mathsf{trc}'} \colon F'SpecM \Rightarrow SpecMG$ specifies semantics of trace logic over F'-coalgebras. To obtain a suitable $\tau \colon F \Rightarrow F'$ note that quotienting with J in $\mathsf{CUC}^*\mathsf{Alg}$ yields an epi $e : M(O) \twoheadrightarrow M(O)/J$ from which we get a mono in KHaus $Spec(e) : Spec(M(O)/J) \rightarrowtail Spec(M(O))$. Pre-composing $Spec(e)$ with the isomorphism $h : SubD(O) \xrightarrow{\sim} Spec(M(O)/J)$ given by Lemma 4.13 and defining $\tau := (Spec(e) \circ h) \times id$, it follows that $\tau \colon F \Rightarrow F'$ has all components mono in KHaus. It now follows that trace logic is also expressive for F-coalgebras, i.e., for compact Hausdorff automata.

Remark 3 *In order to view Gelfand duality* (4.3) *as a concrete dualty 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 [25, 28, 29, 49, 53] as a finite-state analog of alternating Turing machines. Let Σ be a fixed finite *input alphabet*. An *alternating finite automaton* (AFA) over Σ is a tuple $\mathcal{A} = (X, \iota, \delta, F)$, where

- X is a finite set of *states*,
- $F \subseteq X$ are the *final states*,
- $\delta : \Sigma \to X \to 2^X \to 2$ is the *transition function*, and
- $\iota : 2^X \to 2$ 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 δasb, 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 $\iota c = 1$, otherwise it rejects.

Alternating automata accept all and only regular sets. It was shown in [50] 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 *complete atomic Boolean algebras* (CABA) and *discrete spaces* (Set), which underlie *powerset Boolean algebras*.

5.1 CABA, EM(N), and $\mathsf{Set}^{\mathsf{op}}$

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 morphisms of CBA 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. The morphisms of CABA are just the morphisms of CBA.

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:

$$\neg \bigvee a = \bigwedge \{\neg x \mid x \in a\} \qquad (\bigvee a) \wedge x = \bigvee \{y \wedge x \mid y \in a\}$$
$$\neg \bigwedge a = \bigvee \{\neg x \mid x \in a\} \qquad (\bigwedge a) \vee x = \bigwedge \{y \vee b \mid y \in a\}$$

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 [37, 38, 55, 66] for further information on the theory of CBAs and CABAs.

5.1.2 EM(N)

The self-dual adjunction $\mathcal{Q}^{\mathsf{op}} \dashv \mathcal{Q}$ of the contravariant powerset functor (Example 3.2) gives rise to a Set-monad $N = \mathcal{Q} \circ \mathcal{Q}^{\mathsf{op}}$, where for X a set and $f : X \to Y$ a set function,

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

The unit and multiplication are

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

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

5.1.3 Equivalence of CABA, Set$^{\mathsf{op}}$, 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$^{\mathsf{op}}$, as observed in [72].

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

$$\mathsf{Set}^{\mathsf{op}} \xrightarrow{J} \mathsf{EM}(N) \xrightarrow{D} \mathsf{CABA} \xrightarrow{At} \mathsf{Set}^{\mathsf{op}}. \tag{5.1}$$

Here J is the Eilenberg-Moore comparison functor [3,52]. Concretely, J sends a set X to the CABA 2^X and a function $f : X \to Y$ to its inverse image map. That is, $J = \mathsf{Set}(-, 2)$ with Boolean structure. The functor At takes a CABA to its set of atoms and a CABA morphism $f : A \to B$ to $At\, f : At\, B \to At\, A$, 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 $At \cong \mathsf{CABA}(-, 2)$. In other words, the equivalence given by J and At is a concrete duality with dualising object 2.

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

$$D\alpha : TX \to X \qquad\qquad D\alpha = \alpha \circ (\tau N \circ T\eta)_X, \tag{5.2}$$

where $T\eta_X : TX \to TNX$ substitutes $\eta_X(x)$ for $x \in X$ in a term and $\tau_{NX} : TNX \to NX$ 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:

$$\begin{array}{llll} \bigvee_n x_n = \alpha(\bigcup_n \eta_X(x_n)) & & \bigwedge_n x_n = \alpha(\bigcap_n \eta_X(x_n)) & \\ \neg x = \alpha(\sim \eta_X(x)) & 0 = \alpha(\emptyset) & 1 = \alpha(2^X). & \end{array} \tag{5.3}$$

The action of D on morphisms is the identity.

The natural transformation $\tau N \circ T\eta : T \to N$ in (5.2) relating CABA terms and double powerset is invertible up to CABA equivalence. Consider the natural transformation

$$\upsilon : N \to T \qquad\qquad \upsilon_X(A) = \bigvee_{a \in A} \Big(\bigwedge_{x \in a} x \wedge \bigwedge_{x \notin a} \neg x \Big), \quad A \in 2^{2^X}.$$

It can be shown that

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

By the latter we mean that for any term $t \in TX$, $\upsilon_X(\tau_{NX}(T\eta_X(t))) \equiv t$ 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 $\mathcal{A} = (X, \delta, f, \iota)$ be an AFA with states X and components

$$\iota : 1 \to 2^{2^X} \qquad\qquad \delta_a : X \to 2^{2^X},\ a \in \Sigma \qquad\qquad f : X \to 2$$

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

[2] 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.

The language accepted by $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) \triangleq \{w \in \Sigma^* \mid \iota(\delta'_w(F)) = 1\}$, where

$$\delta'_w : 2^X \to 2^X \qquad \delta'_\varepsilon(A) = A \qquad \delta'_{aw}(A)(s) = \delta_a(s)(\delta'_w(A)).$$

As constructed in [50], the associated DFA for the reverse language is $\mathcal{A}'$ with states 2^X and components

$$f^\flat : 1 \to 2^X \qquad \delta^\flat_a : 2^X \to 2^X,\ a \in \Sigma \qquad \iota^\flat : 2^X \to 2.$$

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

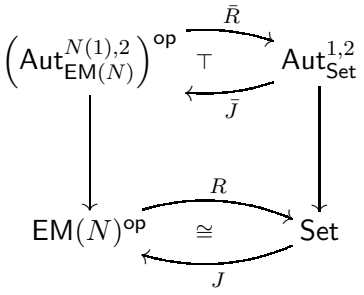
$$\delta^\flat_\varepsilon = id_{2^X} \qquad \delta^\flat_{wa} = \delta^\flat_a \circ \delta^\flat_w.$$

The combinatorial construction of [50] amounts to recurrying the components of the automata. Denoting the reverse of a string w by w^R and using the fact that $\delta^\flat_a = \delta'_a$, it can be shown inductively that $\delta^\flat_w = \delta'_{w^R}$, therefore the language accepted by $\mathcal{A}'$ is the reverse of the language accepted by $\mathcal{A}$:

$$\mathcal{L}(\mathcal{A}') = \{w \mid \iota^\flat(\delta^\flat_w(f^\flat)) = 1\} = \{w \mid \iota(\delta'_{w^R}(f)) = 1\} = \{w \mid w^R \in \mathcal{L}(\mathcal{A})\}.$$

5.3 Alternating automata as $\mathsf{EM}(N)$-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 $\mathsf{EM}(N)$ and $\mathsf{Set}^{\mathsf{op}}$ 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 op on $\mathsf{EM}(N)$), and consider R and J as contravariant functors.



(5.4)

More precisely, we show that $\mathcal{A}' = \bar{R}(\det \mathcal{A})$, where $\det \mathcal{A}$ is the deterministic automaton over $\mathsf{EM}(N)$ 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 (5.1)).

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

$$\iota : 1 \to 2^{2^X} \qquad \delta_a : X \to 2^{2^X},\ a \in \Sigma \qquad f : X \to 2$$

over Set, we have a deterministic automaton $\det \mathcal{A}$ with

$$\iota^\sharp : 2^{2^1} \to 2^{2^X} \qquad \delta_a^\sharp : 2^{2^X} \to 2^{2^X},\ a \in \Sigma \qquad f^\sharp : 2^{2^X} \to 2$$

over $\mathsf{EM}(N)$, using the CABA structure on 2. In $\det \mathcal{A}$, 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 $\mathsf{EM}(N)$-algebras (NX, μ_X) and $(N\emptyset, \mu_\emptyset)$ on generators X and $\emptyset$, respectively.

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

A dual automaton in $\mathsf{Aut}^{1,2}_{\mathsf{Set}}$ (with states X) is a coalgebra for $F = 2 \times (-)^\Sigma$ together with an initial state $j : 1 \to X$, or equivalently an algebra for $G = 1 + \Sigma \times (-)$ with output $f : X \to 2$. It is easy to check that the conditions for Theorem 3.4 hold. First note that $I = 2^{2^1}$ and $O = 1$. We then easily verify that $F_{\mathsf{EM}(N)} \cong J(1) \times (-)^\Sigma$ 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 (3.5) and Theorem 3.4. We describe the reversal functor $\bar{R}$ a bit more concretely as a contravariant functor from $\mathsf{Aut}^{N(1),2}_{\mathsf{EM}(N)}$ to $\mathsf{Aut}^{1,2}_{\mathsf{Set}}$. The base adjunction of (5.4) gives us a bijection of homsets:

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

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

$$\iota : 1 \to (A, \alpha) \qquad \delta_a : (A, \alpha) \to (A, \alpha) \qquad f : (A, \alpha) \to 2$$

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

$$\theta f : 1 \to R(A, \alpha) \qquad R(\delta_a) : R(A, \alpha) \to R(A, \alpha) \qquad R\iota^\sharp : R(A, \alpha) \to 2. \tag{5.5}$$

Applying $\bar{R}$ to $\det \mathcal{A}$, which is

$$\iota : 1 \to 2^{2^X} \qquad \delta_a^\sharp : 2^{2^X} \to 2^{2^X} \qquad f^\sharp : 2^{2^X} \to 2$$

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

$$\theta f^\sharp : 1 \to 2^X \qquad \bar{R}\delta_a^\sharp : 2^X \to 2^X \qquad R\iota^\sharp : 2^X \to 2.$$

Theorem 5.1 *For any alternating automaton*

$$\mathcal{A} = (X, \{\delta_a : X \to 2^{2^X} \mid a \in \Sigma\}, \iota : 1 \to 2^{2^X}, f : X \to 2),$$

$\mathcal{A}' \cong \bar{R}(\det \mathcal{A})$.

Proof. The state space of $\mathcal{A}'$ is 2^X and the state space of $\bar{R}(\det \mathcal{A})$ is the set of atoms of the CABA $D(2^{2^X}, \mu_X)$ which is the set $\{\{a\} \mid a \subseteq X\}$. The correspondence between the initial and final states is shown in Lemmas A.4 and A.5. The transition function δ is the tupling of maps $\delta_a : X \to 2^{2^X}$, i.e., $\delta = \langle \delta_a \rangle_{a \in \Sigma}$, and similarly for $\delta^\sharp$ and $\delta^\flat$. The result follows by applying Lemma A.5 to each δ_a and retupling. QED

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 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 3.10 hold (with $\mathcal{D} = \mathsf{Set}$ 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 $\mathsf{Aut}^{1,2}_{\mathsf{Set}}$ 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 $\mathsf{Aut}^{N(1),2}_{\mathsf{EM}(N)}$. 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 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 [16] and [20], 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 [16].

Our starting points are Brzozowski's algorithm [26] 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 [45] and then extended to automata theory in [9–11].

The duality between reachability and observability has been studied, e.g. in [17] 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 [33,34,59] which recently culminated into a general algebraic

and coalgebraic understanding of equations, coequations, Birkhoff's and Eilenbergtype correspondences [4, 5, 14, 62–64].

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 [23,46,47] in the context of coalgebraic modal logic, and in [16,48] 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 [20], 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 [16] and [20] under a common umbrella, but we hasten to remark that the concept of minimisation via logic presented in section 3.3 is already in [58]. At its core, [58] 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 esssentially what we call **Algo1**. The setting of [58] 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 [58] is to generalise the setup in [16] from dual equivalences to dual adjunctions. The central contribution in [58] 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 [58], 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 [30] languages and their acceptors are regarded as functors which provides a different perspective on minimisation in which Brzozowski 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 [32].

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] Jiří 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 E. G. Manes. Extensions of semilattices. *The American Mathematical Monthly*, 82(7):744–746, 1975.

[8] M. A. Arbib and E. G. Manes. Fuzzy machines in a category. *Bulletin of the Australian Mathematical Society*, 13(2):169–210, 1975.

[9] M.A. Arbib and H.P. Zeiger. On the relevance of abstract algebra to control theory. *Automatica*, 5:589–606, 1969.

[10] 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.

[11] Michael A. Arbib and Ernest G. Manes. Foundations of system theory: The Hankel matrix. *Journal of Computer and System Sciences*, 20:330–378, 1980.

[12] Michael A. Arbib and Ernest G. Manes. Machines in a category. *J. of Pure and Applied Algebra*, 19:9–20, 1980.

[13] William Arveson. *An Invitation to C^*-Algebras*, volume 39 of *Graduate Texts in Mathematics*. Springer-Verlag, 1976.

[14] 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.

[15] F. Bartels. *On Generalised Coinduction and Probabilistic Specification Formats*. PhD thesis, Vrije Universiteit Amsterdam, 2004.

[16] 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.

[17] 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.

[18] Bruce Blackadar. *Operator algebras: theory of C*-algebras and von Neumann algebras*, volume 122 of *encyclopedia of Mathematical Sciences*. Springer-Verlag, 2006.

[19] P. Blackburn, M. de Rijke, and Y. Venema. *Modal logic*. Cambridge University Press, Cambridge, 2001.

[20] 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.

[21] 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.

[22] M. M. Bonsangue and A. Kurz. Duality for logics of transition systems. In *FoSSaCS05*, 2005.

[23] Marcello M. Bonsangue and Alexander Kurz. Presenting functors by operations and equations. In Luca Aceto and Anna Ingólfsdóttir, 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.

[24] Francis Borceux. *Handbook of Categorical Algebra 2: Categories and Structure*. Cambridge University Press, 1994.

[25] J. A. Brzozowski and E. Leiss. On equations for regular languages, finite automata, and sequential networks. *Theoretical Computer Science*, 10:19–35, 1980.

[26] 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.

[27] A. Chagrov and M. Zakharyaschev. *Modal logic*, volume 35 of *Oxford Logic Guides*. The Clarendon Press, New York, 1997.

[28] Ashok Chandra, Dexter Kozen, and Larry Stockmeyer. Alternation. *J. Assoc. Comput. Mach.*, 28(1):114–133, 1981.

[29] Ashok K. Chandra and Larry J. Stockmeyer. Alternation. In *Proc. 17th Symp. Found. Comput. Sci.*, pages 98–108. IEEE, October 1976.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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 Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lect. Notes in Comp. Sci.*, pages 246–257. Springer, 2008.

[35] 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.

[36] Steven Givant and Paul Halmos. *Introduction to Boolean Algebras*. Undergraduate Texts in Mathematics. Springer-Verlag, 2009.

[37] Steven Givant and Paul Halmos. *Introduction to Boolean Algebras*. Springer, 2009.

[38] Paul R. Halmos. *Lectures on Boolean Algebras*. Springer, 1974.

[39] Claudio Hermida and Bart Jacobs. Structural induction and coinduction in a fibrational setting. *Information and Computation*, 145:107–152, 1998.

[40] Edward V. Huntington. Sets of independent postulates for the algebra of logic. *Trans. Amer. Math. Soc.*, 5(3):288–309, July 1904.

[41] 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 *LNCS*, pages 375–404. Springer, 2006.

[42] Bart Jacobs and Ana Sokolova. Exemplaric expressivity of modal logics. *J. Log. Comput.*, 20(5):1041–1068, 2010.

[43] P. T. Johnstone. *Stone spaces*. Cambridge University Press, Cambridge, 1982.

[44] P.T. Johnstone. Adjoint lifting theorems for categories of algebras. *Bulletin London Mathematical Society*, 7:294–297, 1975.

[45] R. Kalman. On the general theory of control systems. *IRE Transactions on Automatic Control*, 4(3):110–110, 1959.

[46] 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.

[47] 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.

[48] Bartek Klin and Jurriaan Rot. Coalgebraic trace semantics via forgetful logics. *Logical Methods in Computer Science*, 12(4), 2016.

[49] Dexter Kozen. On parallelism in Turing machines. In *Proc. 17th Symp. Found. Comput. Sci.*, pages 89–97. IEEE, October 1976.

[50] Dexter Kozen. *Theory of Computation*. Springer, New York, 2006.

[51] Alexander Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwigs-Maximilians-Universität Mïchen, 2000.

[52] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971.

[53] Ernst Leiss. Succinct representation of regular languages by Boolean automata. *Theoretical Computer Science*, 13:323–330, 1981.

[54] 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.

[55] J. Donald Monk and eds. R. Bonnet. *Handbook of Boolean Algebras*. North-Holland, 1989.

[56] Joan W. Negrepontis. Duality in analysis from the point of view of triples. *Journal of Algebra*, 19:228–253, 1971.

[57] Hans-E. Porst and Walter Tholen. Concrete dualities. In H. Herrlich and Hans-E. Porst, editors, *Category Theory at Work*. Heldermann Verlag, 1991.

[58] 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.

[59] F. Roumen. Canonical automata via duality. Unpublished note., 2011.

[60] J. J. M. M. Rutten. Universal coalgebra: A theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.

[61] Shoichiro Sakai. *C^*-Algebras and W^*-algebras*. Springer-Verlag, 1971.

[62] Julian Salamanca. Unveiling eilenberg-type correspondences: Birkhoff's theorem for (finite) algebras + duality. *CoRR*, abs/1702.02822, 2017.

[63] 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.

[64] 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.

[65] Marcel Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, 1961.

[66] Roman Sikorski. *Boolean Algebras*. Springer, 1966.

[67] 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.

[68] M. H. Stone. The theory of representations for Boolean algebras. *Trans. Amer. Math. Soc.*, 40(1):37–111, 1936.

[69] M. H. Stone. Topological representation of distributive lattices and Brouwerian logics. *Časopis Pešt. Mat. Fys.*, 67:1–25, 1937.

[70] Ross Street. The formal theory of monads. *J. Pure and Applied Algebra*, 2:149–168, 1972.

[71] Yi-Jia Tan. Free sets and free subsemimodules in a semimodule. *Linear Algebra and its Applications*, 496:527–548, 2016.

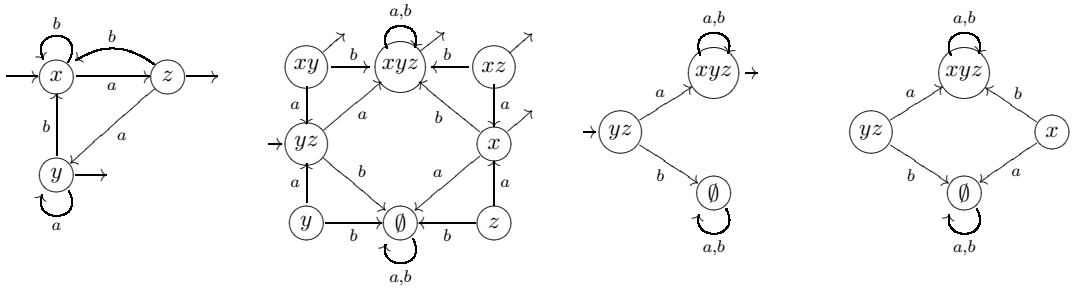
[72] Paul Taylor. Subspaces in abstract Stone duality. *Theory and Applications of Categories*, 10(13):300–366, 2002.

Appendix

A.1 Example: PODFA-style minimisation vs Brzozowski

Classic DFAs are PODFAs with a single observation, hence they can be minimised using the duality approach in [16] using the duality between finite sets and finite Boolean algebras. $\mathbb{P} : \mathsf{FinSet}^{\mathsf{op}} \rightleftarrows \mathsf{FinBA} : \mathrm{Uf}$ or using Brozowski's algorithm via the self-dual adjunction, cf. [20]. $\mathcal{Q} : \mathsf{Set}^{\mathsf{op}} \rightleftarrows \mathsf{Set} : \mathcal{Q}$ Consider the DFA $\mathcal{X}$ below left from (11) in [20]. The DFA $\mathcal{X}$ accepts the language $(a+b)^*a$. The result $\mathcal{X}'$ after the first reverse-determinise step in Brozowski's algorithm is shown on the right. Disregarding initial and final states, $\mathcal{X}'$ is also the modal algebra obtained from $\mathcal{X}$ We then take reachable parts to get the automaton $\mathcal{Y}$ and the subalgebra $\mathcal{A}$ of definable subsets of the modal language with a single proposition letter p which is true precisely at accepting states y, z of γ: $[\![p]\!] = \{yz\}$.

Start: $\mathcal{X}$ $\quad$ $\mathcal{X}' = det(rev(\mathcal{X}))$ $\quad$ $reach(\mathcal{X}')$ $\quad$ $\mathcal{A}$



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

Result of Brzozowski: $\quad$ Result of minimisation-via-duality

Accepts $rev(rev(L)) = (a+b)^*a$ $\quad$ State x accepts $L = (a+b)^*a$

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

A.2 Details of ϱ

For our specific choice of functors $F_{\mathcal{C}}$ and $G_{\mathcal{D}}$, we compute the concrete definition of ϱ from (3.4) when the adjunction arises from a dualising object.

$$
\begin{array}{lcccccc}
\varrho_X\colon & & & & & & \\
O+\Sigma\cdot PX & \overset{\eta_{GPX}}{\longrightarrow} & PS(O+\Sigma\cdot PX) & \overset{\sim}{\longrightarrow} & P(SO\times(SPX)^{\Sigma}) & \overset{PF_{\mathcal{C}}\varepsilon_X}{\longrightarrow} & P(SO\times X^{\Sigma}) \\
O+\Sigma\cdot\Delta^X & \overset{\eta_{G(\Delta^X)}}{\longrightarrow} & \Delta^{\Delta^{(O+\Sigma\cdot\Delta^X)}} & \overset{\sim}{\longrightarrow} & \Delta^{(\Delta^O\times(\Delta^{\Delta^X})^{\Sigma})} & \overset{\Delta^{F_{\mathcal{C}}\varepsilon_X}}{\longrightarrow} & \Delta^{\Delta^O\times X^{\Sigma}} \\
q & \mapsto & \eta(q) & \mapsto & \eta(q) & \mapsto & \lambda(j,d).j(q) \\
(a,p) & \mapsto & \eta(a,p) & \mapsto & \eta(a,p) & \mapsto & \lambda(j,d).p(d(a))
\end{array}
$$

Here we used that the units evaluate, and that

$$
\begin{array}{lccc}
\Delta^{F_{\mathcal{C}}\varepsilon_X}\colon & \Delta^{(\Delta^O\times(\Delta^{\Delta^X})^{\Sigma})} & \longrightarrow & \Delta^{\Delta^O\times X^{\Sigma}} \\
& h & \mapsto & \lambda(j,d).h(j,\lambda a.\lambda g.g(d(a)))
\end{array}
$$

In short,

$$
\begin{array}{rrcl}
\varrho_X\colon\colon & O+\Sigma\cdot\Delta^X & \longrightarrow & \Delta^{\Delta^O\times X^{\Sigma}} \\
& \varrho(q)(j,d) & = & j(q) \\
& \varrho(a,p)(j,d) & = & p(d(a))
\end{array}
\tag{A.1}
$$

A.3 Theory maps of $(G,\varrho^{\mathsf{trc}})$ and $(G_{\mathcal{D}},\varrho)$ coincide

Proof of Lemma 3.6
Due to the natural isomorphisms of Hom-sets given by the adjunctions in (3.6), we have the following correspondence for all $F_{\mathcal{C}}$-coalgebras γ and all G-algebras α:

$$
\begin{array}{ll}
\gamma \to \overline{S}\,\overline{\Phi_{\mathcal{D}}}(\alpha) & \mathsf{Coalg}_{\mathcal{D}}(F_{\mathcal{C}}) \\
\hline\hline
\overline{\Phi_{\mathcal{D}}}(\alpha) \to \overline{P}(\gamma) & \mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}}) \\
\hline\hline
\alpha \to \overline{U_{\mathcal{D}}}\,\overline{P}(\gamma) & \mathsf{Alg}_{\mathsf{Set}}(G)
\end{array}
\tag{A.2}
$$

In particular, since $(\Sigma^*\Omega,\alpha)$ is an initial G-algebra, it follows that $\overline{\Phi_{\mathcal{D}}}(\Sigma^*\Omega,\alpha)$ is an initial $G_{\mathcal{D}}$-algebra, since left adjoints preserve initial objects.

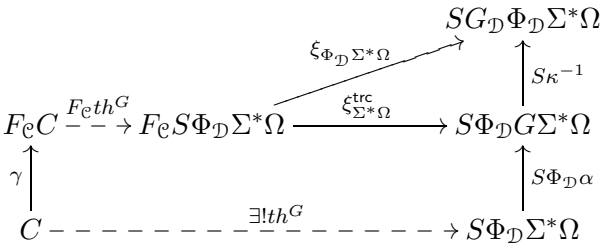
Furthermore, since contravariant adjoint functors turn colimit into limits, $\overline{S}\,\overline{\Phi_{\mathcal{D}}}(\Sigma^*\Omega,\alpha)$ is a final $F_{\mathcal{C}}$-coalgebra.

The semantic maps of the two logics are the unique morphisms from the initial algebras, and we denote them $s^G\colon \Sigma^*\Omega \to U_{\mathcal{D}}P(C)$ and $s^{G_{\mathcal{D}}}\colon \Phi_{\mathcal{D}}(\Sigma^*\Omega) \to P(C)$. The correspondence in (A.2) says that $s^{G_{\mathcal{D}}}$ is fully determined by s^G. When $\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}$ is a free-forgetful adjunction (as in most of our examples), this tells us that formulas that are contained in both logics have the same semantics.

By definition, th^G is the $(S\Phi_{\mathcal{D}} \dashv U_{\mathcal{D}}P)$-adjoint of s^G and $th^{G_{\mathcal{D}}}$ is the $(S \dashv P)$-adjoint of $s^{G_{\mathcal{D}}}$. From (A.2) we see that $th^G = th^{G_{\mathcal{D}}}$, and both are the unique map into the final $F_{\mathcal{C}}$-coalgebra.

An alternative argument using the mates ξ and ξ^{trc} is as follows.

Consider the following diagram:

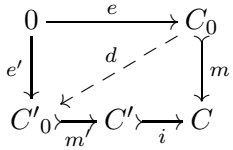


where α denotes the initial G-algebra. By the universal property of the theory map (cf. e.g. (2.5) in [48]) we have that th^G is the unique map that makes the above square commute. The upper triangle commutes by definition of ξ^{trc}. As $\overline{\Phi_{\mathcal{D}}}$ maps initial G-algebra to initial $G_{\mathcal{D}}$-algebra we also have that $(\Phi_{\mathcal{D}}(\Sigma^*\Omega), \Phi_{\mathcal{D}}\alpha \circ \kappa^{-1})$ is the initial $G_{\mathcal{D}}$-algebra. Therefore the diagram shows that th^G also satisfies the universal property of $th^{G_{\mathcal{D}}}$ and thus $th^G = th^{G_{\mathcal{D}}}$ as claimed.

A.4 Coincidence of reachability notons

Lemma A.1 *Let* A *be a wellpowered category with initial object* 0_{A} *and factorisation system* (E, M) *such that* $M \subseteq Mono(\mathsf{A})$*. For all objects* $A \in \mathsf{A}$*, the least subobject* A_0 *of* A *is obtained by* (E, M)*-factorisation of the unique morphism from* 0_{A} *to* A*.*

Proof. Let $0_{\mathsf{A}} \xrightarrow{e} A_0 \xrightarrow{m} A$ be the (E, M)-factorisation of the initial map from 0_{A} to A. We show that A_0 is the least subobject of A. To this end, let $A' \overset{i}{\rightarrowtail} A$ be a subobject and let $0 \xrightarrow{e'} A'_0 \xrightarrow{m'} A'$ be the (E, M)-factorisation of the initial morphism for A'. By the diagonal fill-in property of (E, M) we get a unique morphism $d\colon A_0 \to A'_0$ such that $e' = d \circ e$ and $m = i \circ m' \circ d$ as shown here:



Take $h = m' \circ d\colon A_0 \to A'$. To see that h is unique, suppose $h'\colon A_0 \to A'$ is such that $m = i \circ h'$, then $i \circ h' = i \circ h$ and hence $h = h'$ since i is a mono. QED

A.5 Factorisation systems for coalgebras and algebras

Lemma A.2 *Assume that* $\mathcal{C} \cong \mathsf{EM}(T)$ *for a* Set*-monad* T*, and* $\mathcal{D} \cong \mathsf{EM}(T)$ *for a* Set*-monad* T*. We have:*

1. (E, M) *with* $E =$ *surjective* $F_{\mathcal{C}}$*-coalgebra morphisms and* $M =$ *injective* $F_{\mathcal{C}}$*-coalgebra morphisms is a factorisation system for* $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$*.*

2. (E, M) *with* $E =$ *surjective* $G_{\mathcal{D}}$*-algebra morphisms and* $M =$ *injective* $G_{\mathcal{D}}$*-algebra morphisms is a factorisation system for* $\mathsf{Alg}_{\mathcal{C}}(G_{\mathcal{D}})$.

3. (E, M) *with* $E =$ *surjective* G*-algebra morphisms and* $M =$ *injective* G*-algebra morphisms is a factorisation system for* $\mathsf{Alg}_{\mathsf{Set}}(G)$.

Proof. Item 1 follows from [51, Thm. 3.1.7]. We show that the conditions for [51, Thm. 3.1.7] hold. For all Set-monads T, $\mathsf{EM}(T)$ is a regular category (because it is exact) [24, Thm 4.3.5], and hence $(RegEpi, Mono)$ is a factorisation system. Furthermore, in $\mathsf{EM}(T)$, regular epis are the surjective homomorphisms (but epis need not be surjective) [3, 7.2] and RegMono = Mono = injective homomorphisms [3, 6.9,6.12]. It is easy to show that $F_{\mathcal{C}}$ defined as above preserves monos. Letting $V\colon \mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}}) \to \mathcal{C}$ be the forgetful functor that maps an $F_{\mathcal{C}}$-coalgebra (C, γ) to C, it then follows from [51, Thm. 3.1.7], that $(V^{-1}RegEpi, V^{-1}Mono)$ is a factorisation system for $\mathsf{Coalg}_{\mathcal{C}}(F_{\mathcal{C}})$. Note that $V^{-1}RegEpi$ and $V^{-1}Mono$ are the surjective and injective $F_{\mathcal{C}}$-coalgebra morphisms, respectively. It is straightforward to prove that the functors $G_{\mathcal{D}}$ and G preserve regular epis.

Items 2 and 3 can be proved similarly to [51, Thm. 3.1.7] using some dual arguments. Sketch: Using that $G_{\mathcal{D}}$ preserves regular epis, one can show that $V\colon \mathsf{Alg}_{\mathcal{D}}(G_{\mathcal{D}}) \to \mathcal{D}$ creates $(RegEpi, Mono)$-factorisations in $\mathcal{D}$ using the diagonal fill-in property (similar to [51, Prop. 1.3.3]). Using that regular epis in $\mathcal{D}$ are surjective, one can show that the diagonal fill-in obtained from $(RegEpi, Mono)$ in $\mathcal{D}$ is an G_D-algebra morphism (similar to [51, Thm. 3.1.7] and [60, Lem. 2.4]). It follows that $(V^{-1}RegEpi, V^{-1}Mono)$ is a factorisation system for $\mathsf{Alg}_{\mathcal{D}}(G_D)$. QED

A.6 Isomorphism lemma for topological automata

Proof of Lemma 4.13 Let $\alpha\colon \mathcal{D}_{\leq 1}(Y) \to Spec(M(Y)/J)$ be defined by $\alpha(\varphi)(m) = m(\varphi)$ where $\varphi \in \mathcal{D}_{\leq 1}(Y)$ and $m \in M(Y)$. Note that this is well defined. If $m \equiv_J m'$ then their difference lies in J which means that $m(\varphi) - m'(\varphi) = k(\varphi)(|\pi^-| - \pi^-)(\varphi)$. This condition is equivalent to $1 - \pi$ is positive and hence $|\pi^-|$ and π^- are equal and hence the second term is 0, whence $m(\varphi) = m'(\varphi)$. Note that the topology of $Spec(M(Y)/J)$ is generated by the taking as the closed sets, sets of maximal ideals that contain a fixed element φ of $M(Y)$. Any maximal ideal consists of the functions that vanish at a point y, call this m_y. So if we fix such an φ for it to be in a maximal ideal m_y, we have $\varphi(y) = 0$. This means that α^{-1} of a closed set is the set of subdistributions that assign 0 to a particular element y; this is a closed set so α is continuous.

Let $\beta\colon Spec(M(Y)/J) \to \mathcal{D}_{\leq 1}(Y)$ be defined by $\beta(\Phi)(y) = \Phi(\pi_y)$ where $\pi_y \in M(Y)/J$ projects onto y. We check that $\beta(\Phi)$ is a subdistribution:

$$\sum_{y \in Y} \beta(\Phi)(y) = \sum_{y \in Y} \Phi(\pi_y) = \Phi\left(\sum_{y \in Y} \pi_y\right) \preceq \Phi(1) = 1$$

The second and last identity hold because Φ is a $\mathsf{CUC}^*\mathsf{Alg}$-homomorphism (hence linear and unital); the inequality holds since we are in $M(Y)/J$ (which says that $\pi \preceq 1$) and Φ is monotone.

We now show that for $\varphi \in \mathcal{D}_{\leq 1}(Y)$, $\beta(\alpha(\varphi)) = \varphi$. Let $y \in Y$, we then have: $\beta(\alpha(\varphi))(y) = \alpha(\varphi)(\pi_y) = \pi_y(\varphi) = \varphi(y)$ We show that for $\Phi \in \mathit{Spec}(M(Y)/J)$, $\alpha(\beta(\Phi)) = \Phi$. Let $m \in M(Y)$, we then have $\alpha(\beta(\Phi))(m) = m(\beta(\Phi))$. By the Stone-Weierstrass theorem, the polynomials on the compact Hausdorff space $[0,1]^Y$ are dense in $C([0,1]^Y)$. Since $m\colon [0,1]^Y \to \mathbb{R}$ is continuous, it therefore suffices to show that for all polynomials p on $[0,1]^Y$ we have that $p(\beta(\Phi)) = \Phi(p)$.

Case $p = 1$*:* $p(\beta(\Phi)) = 1 = \Phi(1) = \Phi(p)$. *Case* $p = r \in \mathbb{R}$*:* $p(\beta(\Phi)) = r = \Phi(r) = \Phi(p)$. *Case* $p = \sum_{y\in Y} r_y\pi_y$*:* $\Phi(p) = \Phi(\sum_{y\in Y} r_y\pi_y) = \sum_{y\in Y} r_y\Phi(\pi_y) = \sum_{y\in Y} r_y\beta(\Phi)(y) = p(\beta(\Phi))$. Finally, let $p = \sum_{y\in Y} r_y\pi_y$ and $q = \sum_{y\in Y} s_y\pi_y$. Then $\Phi(pq) = \Phi(p)\Phi(q) = p(\beta(\Phi))q(\beta(\Phi)) = (pq)(\Phi)$. It follows that $p(\beta(\Phi)) = \Phi(p)$ holds for all polynomials p[3] and we have now shown that $\alpha\colon \mathcal{D}_{\leq 1}(Y) \to \mathit{Spec}(M(Y)/J)$ is a bijection with inverse β.

A.7 Alternating automata

These lemmas are needed for Theorem 5.1.

Lemma A.3 *Let* $d : Y \to 2^{2^X}$*. Then* $d^\sharp = d^{\flat^{-1}}$.

Proof. For all $a \subseteq X$ and all $A \in 2^{2^Y}$,

$$\begin{aligned} a \in d^\sharp(A) &\Leftrightarrow a \in \mu_X(Nd(A)) \Leftrightarrow \eta_{QX}(a) \in Nd(A) \Leftrightarrow \eta_{QX}(a) \in (d^{-1})^{-1}(A) \\ &\Leftrightarrow d^{-1}(\eta_{QX}(a)) \in A \Leftrightarrow \{x \mid d(x) \in \eta_{QX}(a)\} \in A \Leftrightarrow \{x \mid a \in d(x)\} \in A \\ &\Leftrightarrow d^\flat(a) \in A. \end{aligned}$$

QED

The following lemmas show that the transition structure of $\mathcal{A}'$ coincides with the transition structure of $\bar{R}(\det \mathcal{A})$.

Lemma A.4 *For all functions* $f : X \to 2$*,* $\theta(f^\sharp) = f^\flat$.

Proof. This is immediate from the fact that $\theta^{-1}(f^\flat) = f^{\flat^{-1}}$ and Lemma A.3. QED

Lemma A.5 *For all* $d : Y \to 2^{2^X}$ *and all* $a \subseteq X$*,* $R(d^\sharp)(\{a\}) = \{d^\flat(a)\}$*. Thus* $R(d^\sharp) = d^\flat$ *up to bijections relating atoms* $\{a\}$ *and their singleton elements* a*. In particular, for all functions* $i : 1 \to 2^{2^X}$*,* $R(i^\sharp) = i^\flat$ *up to these bijections.*

Proof. The atoms of $(2^{2^X}, \mu_X)$ are of the form $\{a\}$ for $a \subseteq X$, hence for $A \in 2^{2^Y}$,

$$\begin{aligned} R(d^\sharp)(\{a\}) &= \bigwedge\{A \in 2^{2^Y} \mid \{a\} \leq d^\sharp(A)\} \\ &= \bigcap\{A \in 2^{2^Y} \mid a \in d^\sharp(A)\} && \text{since } \textstyle\bigwedge \text{ is } \bigcap \text{ in } (2^{2^X}, \mu_X) \\ &= \bigcap\{A \in 2^{2^Y} \mid d^\flat(a) \in A\} && \text{Lemma A.3} \\ &= \{d^\flat(a)\} && \text{since } d^\flat(a) \in 2^Y. \end{aligned}$$

QED

[3] Clearly all polynomials can be expressed as sums of products of lower degree polynomials.