Available online at [www.sciencedirect.com](http://www.sciencedirect.com/)

[Electronic Notes in Theoretical Computer Science 325 (2016) 253–276](http://dx.doi.org/10.1016/j.entcs.2016.09.042)

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

Coalgebraic Minimization of Automata by Initiality and Finality

Jurriaan Rot[1](#_bookmark0)*,*[2](#_bookmark0)

*Universit´e de Lyon, ENS de Lyon, CNRS, UCB Lyon 1, LIP*

**Abstract**

Deterministic automata can be minimized by partition refinement (Moore’s algorithm, Hopcroft’s algo- rithm) or by reversal and determinization (Brzozowski’s algorithm). In the coalgebraic perspective, the first approach can be phrased in terms of a minimization construction along the final sequence of a func- tor, whereas a crucial part of the second approach is based on a reachability construction along the initial

sequence of another functor. We employ this coalgebraic perspective to establish a precise relationship between the two approaches to minimization, and show how they can be combined. Part of these results are extended to an approach for language equivalence of a general class of systems with branching, such as non-deterministic automata.

*Keywords:* minimization, automata, coalgebra

# Introduction

The problem of minimizing deterministic automata has been studied since the early days of automata theory, and a number of different approaches have been proposed. Probably the most well-known family of algorithms, which includes Hopcroft’s [[11](#_bookmark37)] and Moore’s algorithm [[19](#_bookmark45)] as well as typical textbook constructions [[12](#_bookmark38)], is based on a stepwise refinement of a partition of states. Another approach, due to Brzo- zowski [[7](#_bookmark33)], is based on determinization and reversal. That approach appears (and is usually considered) to be fundamentally different than partition refinement [[3](#_bookmark27),[24](#_bookmark50)]. To the best of our knowledge, a connection was only established in the work of Champarnaud et al [[8](#_bookmark34)] (and further extended in [[9](#_bookmark35)]), who explicitly showed how the partition of states that are language equivalent is obtained from the reversed determinized automaton that appears in Brzozowski’s algorithm.

1 This work was performed within the framework of the LABEX MILYON (ANR-10-LABX-0070) of Uni- versit´e de Lyon, within the program “Investissements d’Avenir” (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR), and was supported by project ANR 12IS02001 PACE.

2 Email: [jurriaan.rot@ens-lyon.fr](mailto:jurriaan.rot@ens-lyon.fr)

<http://dx.doi.org/10.1016/j.entcs.2016.09.042>

1571-0661/© 2016 The Author(s). Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Partition refinement can be phrased abstractly as an inductive computation along the final sequence of a functor, generalizing from automata to coalgebras [[1](#_bookmark28)]. Starting with [[6](#_bookmark32)], Brzozowski’s algorithm has also received significant attention from a coalgebraic perspective, as an elegant instance of duality between algebra and coalgebra [[5](#_bookmark31)], in several different formulations [[5](#_bookmark31),[4](#_bookmark30),[15](#_bookmark41)].

In this paper we employ the coalgebraic perspective on partition refinement and Brzozowski’s algorithm to understand and establish their relationship. First, we dualize the construction of [[1](#_bookmark28)] and combine it with a variant of the Brzozowski construction from [[15](#_bookmark41)], to obtain a minimization construction based on a stepwise computation of reachability along an initial sequence. We then show how the *i*-th step of this reachability construction yields the *i*-th partition of states in partition refinement by a simple factorization, thus establishing a fundamental connection between the two minimization constructions. Based on this result, we define a minimization construction that combines partition refinement with the computation of reachability. In our motivating example of deterministic automata, we retrieve the combined minimization construction due to Champarnaud et al [[8](#_bookmark34)].

Our Brzozowski construction is based on [[15](#_bookmark41)], where it is formulated for systems with branching, such as non-deterministic, alternating and tree automata. In the last part of the paper, we consider such branching systems, and show how the reachability computation yields an abstract procedure for language equivalence.

*Outline.* In Section [2](#_bookmark1) we describe the ideas of partition refinement and Brzozowski’s algorithm and their connection, for deterministic automata. Section [3](#_bookmark6) contains pre- liminaries, Section [4](#_bookmark8) recalls coalgebraic partition refinement, and Section [5](#_bookmark12) intro- duces the dual reachability construction. Section [6](#_bookmark15) introduces the abstract Brzo- zowski construction, and Section [7](#_bookmark18) establishes the connection with partition refine- ment. Section [8](#_bookmark23) concerns branching systems. Proofs can be found in the appendix.

*Acknowledgments.* The author is grateful to Filippo Bonchi, Matias Lee, Damien Pous and Alexandra Silva for comments, suggestions and discussions.

# Minimization of deterministic automata

We fix an alphabet *A*, denote the set of words over *A* by *A∗* and the empty word by *ε*. A *deterministic automaton* is a triple (*X, o, f* ) consisting of a set of states *X*, a transition function *f* : *X → XA* and an output function *o* : *X →* 2, where 2 = *{*0*,* 1*}* is a two-element set. Note that the state space *X* is not required to be finite, and there is no initial state. The semantics of an automaton is a function *l* : *X →* 2*A∗* mapping each state to the language it accepts, inductively defined by *ε ∈ l*(*x*) iff *o*(*x*)=1 and *aw ∈ l*(*x*) iff *w ∈ l*(*f* (*x*)(*a*)), for any letter *a ∈ A* and word *w ∈ A∗*.

Our aim is to *minimize* deterministic automata: given an automaton (*X, o, f* ) we search the automaton with the least number of states that accepts the same lan- guages as those accepted by the states in *X*. Formulated slightly more abstractly, the aim is to find a factorization of the semantics *l* : *X →* 2*A∗* as a surjective func- tion *e* : *X → E* followed by an injective function *m* : *E →* 2*A∗* . Such a factorization uniquely turns the set *E* into a (minimal) automaton accepting all languages of

states in *X*. We describe the ideas underlying two standard approaches to min- imization, based respectively on representing *E* as a quotient of states, and on representing the image of *X* along *l* by taking a quotient of words.

* 1. *Minimization by equivalence of states*

Let (*X, o, f* ) be a deterministic automaton, with language semantics *l* : *X →* 2*A∗* . Consider the equivalence relation *≡⊆ X × X* defined as the kernel of *l*, i.e., *x ≡ y* iff *l*(*x*) = *l*(*y*). Two states are related by *≡* precisely if they are language equiva- lent. Once we computed the relation *≡*, the minimization of our automaton can be obtained as the quotient of states w.r.t. *≡*.

The relation *≡* can be *approximated* by defining a family of equivalence relations

*≡n ⊆ X × X* indexed by natural numbers, called Moore equivalences [[3](#_bookmark27)], as follows: *x ≡n y* iff *∀w ∈ A∗* with *|w| < n*: (*w ∈ l*(*x*) iff *w ∈ l*(*y*)), where *|w|* is the length of a word *w*. In words, *≡n* is language equivalence for words with length below *n*.

The point is that we can characterize *≡n* by induction, setting *≡*0 = *X × X* and

*x ≡n*+1 *y* iff *o*(*x*)= *o*(*y*) and *∀a ∈ A* : *f* (*x*)(*a*) *≡n f* (*y*)(*a*) *.*

If *X* is finite, then this inductive computation will eventually stabilize, at which point we have computed the relation *≡* and, hence, a minimal automaton (e.g., [[12](#_bookmark38)]). (The usual presentation is slightly different, starting from the relation that dis- tinguishes between accepting and non-accepting states, and leaving the condition *o*(*x*)= *o*(*y*) out. We prefer the above variation to match the theory in Section [4](#_bookmark8).)

Phrasing the above inductive characterization in terms of partitions of *X* yields a construction based on stepwise refinement of partitions. Moore’s minimization algorithm [[19](#_bookmark45)], for instance, is an implementation of this construction, whereas Hopcroft’s minimization algorithm [[11](#_bookmark37)] is a more advanced (and efficient) version of partition refinement. We refer to [[3](#_bookmark27)] for a detailed analysis of these algorithms.

* 1. *Minimization by equivalence of words*

We define an equivalence relation *≈ ⊆ A∗ × A∗* by *w ≈ v* iff *∀x ∈ X* : (*w ∈ l*(*x*) iff *v ∈ l*(*x*)). This relation is dual to *≡*, in the sense that it is the kernel of the *transpose lb* : *A∗ →* 2*X* of the language semantics *l*. Two words are related by *≈* if there is no state in the automaton that accepts one but not the other.

Given an equivalence class [*w*] in the quotient *A∗/≈*, a state *x ∈ X* either accepts all words in [*w*], or none. Hence, the language of every *x ∈ X* arises as a union

S*{*[*w*] *| w ∈ l*(*x*)*}* of equivalence classes in *A∗/≈*. The set *{{*[*w*] *| w ∈ l*(*x*)*}| x ∈ X}*

is isomorphic to the set of languages accepted by the automaton (the image of *X*

along *l*), which is (the state space of) a minimal automaton.

But how are these equivalence classes of words computed and represented? The crux is that there is an isomorphism between the quotient *A∗/≈* and the set *R* =

*{{x ∈ X | w ∈ l*(*x*)*} | w ∈ A∗}*, that is, every equivalence class of words is represented as the set of states accepting these words. The set *R* has an inductive

characterization, as the limit of:

*R*0 = *∅ Ri*+1 = *{{x ∈ X | f* (*x*)(*a*) *∈ S}| a ∈ A, S ∈ Ri}∪ {{x ∈ X | o*(*x*)= 1*}}*

If the state space *X* is finite, then this sequence stabilizes after a finite number of steps, at which point we computed *R* and, hence, the partition of *A∗*. The language of a state *x ∈ X* is then represented by the set *{S ∈ R | x ∈ S}*, and (the state space of) our minimal automaton is obtained by taking *{{S ∈ R | x ∈ S}| x ∈ X}*. Similar to the case of *≡i*, the above presentation of the sets *Ri* is chosen to match the abstract theory of Section [6](#_bookmark15).

The inductive computation of *Ri*’s corresponds to the *reachable* (sets of) states in the automaton with state space 2*X* obtained from (*X, o, f* ) by reversing transi- tions, turning the set of final states into the initial state and determinizing. This computation is at the heart of *Brzozowski’s minimization algorithm* [[7](#_bookmark33)]. That algo- rithm minimizes a deterministic automaton (with initial and final states) by doing the following twice: reverse and determinize the automaton, and take the part that is reachable from the new initial state.

Brzozowski’s algorithm is usually explained differently, based on the fact that the reverse of an automaton recognizes the reverse language (e.g., [[22](#_bookmark48),[3](#_bookmark27),[5](#_bookmark31)]). We prefer the above explanation in terms of equivalence classes, because it explains the construction directly in terms of the original automaton, and highlights a tight correspondence between Brzozowski’s construction and partition refinement.

Indeed, for each *i* we have:

*x ≡i y* iff (*∀S ∈ Ri* : *x ∈ S* iff *y ∈ S*) (1) which means that *≡i* can be obtained directly from *Ri* and, as shown in [[8](#_bookmark34)], that *≡*

can be obtained from *R*. In terms of partitions, writing *Ei* for the quotient of *X* by

*≡i*, the above equation ([1](#_bookmark4)) shows how to compute *Ei* from *Ri* by *splitting* the set *X* according to the sets in *Ri*: informally, *Ei* is obtained by starting with the trivial partition *Ei* = *{X}* and then, for each *S ∈ Ri*, replacing each *Q ∈ Ei* by *Q\S* and *Q ∩ S* if both are nonempty. It is not difficult to see that to compute *Ei*+1 from *Ei*, one only needs to compute *Ri*+1 from *Ri* and split all the equivalence classes in *Ei* according to the new sets (splitters) in *Ri*+1. This is the basis of an algorithm, proposed in [[8](#_bookmark34)], that combines partition refinement with Brzozowski’s algorithm.

**Example 2.1** Consider the following deterministic automaton over the alphabet

*{a, b, c}*, where the only accepting state is *x*.

*x* , *a,b,c* /*z ¸*

*a,b,c*

,*,,¸* ,*¸*

*a b c*

*b,c * *u ,*¸*,a,c*

*v* ¸*, w*

*a,b*

We compute the quotients *Ei* of *X* by *≡i*, and the sets *Ri*’s as explained above:

*E*0 = *{{x, u, v, w, z}} R*0 = *∅*

*E*1 = *{{x}, {u, v, w, z}} R*1 = *{{x}}*

*E*2 = *{{x}, {u}, {v}, {w}, {z}} R*2 = *{{x}, {u}, {v}, {w}}*

*R*3 = *{{x}, {u}, {v}, {w}, {u, v}, ∅}*

*R*4 = *{{x}, {u}, {v}, {w}, {u, v}, ∅, {u, w}, {v, w}}*

Each *Ei* is computed from *Ri* by only identifying those states that appear in the same sets in *Ri*, or, more efficiently, by splitting the partitions in *Ei−*1 according to the newly added sets in *Ri*. For instance, we obtain *E*2 from *E*1 and *R*2 by splitting

*{x}* and *{u, v, w, z}* by *{u}, {v}, {w}*, in particular by splitting *{u, v, w, z}* by *{u}*,

yielding *{u}, {v, w, z}*; then *{v, w, z}* by *{v}* yielding *{v}, {w, z}*; and finally *{w, z}* by *{w}* (notice that the order of splitting does not matter). Observe that we can compute *Ei* from *Ri*, but not vice versa. And the sequence of *Ei*’s may stabilize earlier than the sequence of *Ri*’s.

# Preliminaries

For the remainder of this paper, we assume familiarity with basic notions of category theory. Given a category C,a *coalgebra* for a functor *B* : C *→* C is a pair (*X, c*) where *X* is an object in C and *c* is a morphism *c* : *X → BX*. A coalgebra homomorphism from (*X, c*) to (*Y, d*) is a C-morphism *h* : *X → Y* such that *Fh ◦ c* = *d ◦ h*. The category of coalgebras for a functor *B* is denoted by coalg(*B*). A coalgebra (*Z, ζ*) is called *ﬁnal* if it is a final object in coalg(*B*), i.e., for every coalgebra (*X, c*) there exists a unique coalgebra morphism from (*X, c*) to (*Z, ζ*).

For our running example, consider the functor *B* : Set *→* Set defined by *BX* = 2 *× XA*, where *A* is a fixed set. A *B*-coalgebra *⟨o, f⟩* : *X →* 2 *× XA* is a determin- istic automaton (with no initial state), as in Section [2](#_bookmark1). The functor *B* has a final coalgebra, given by the set of languages over *A*. The unique morphism from any automaton to this final coalgebra maps each state to the language it accepts [[21](#_bookmark47)].

An *algebra* for a functor *L* : D *→* D is defined dually to a coalgebra, i.e., it is a pair (*X, a*) where *a* : *LX → X*, and an algebra morphism from (*X, a*) to (*Y, b*) is a morphism *h* : *X → Y* such that *h ◦ a* = *b ◦ Lh*. The category of *L*-algebras is denoted by alg(*L*). An algebra is called *initial* if it is an initial object in alg(*L*).

As an example, consider the functor *L* : Set *→* Set defined by *LX* = *A × X* + 1, where *A* is a fixed set and 1 = *{∗}* a singleton. An *L*-algebra consists of a set *X* and a map [*g, ι*]: *A × X* +1 *→ X*. We interpret *L*-algebras as deterministic automata with initial state *ι*(*∗*) and transition function *g* (but no final states). This functor *L* has an initial algebra, given by the set of words *A∗* with the empty word *ε* as initial state and (*a, w*) *∈ A × A∗* mapped to the concatenation *aw*. Given an *L*-algebra (deterministic automaton), the unique morphism from *A∗* maps a word *w* to the

state that is reached after processing *w* from the initial state, reading the letters from right to left.

*Contravariant adjunctions.* We will consider functors *F* : Cop *→* D, *G* : Dop *→* C that form an adjunction *F* op *E G*, i.e., such that there is a natural bijection C(*X, GY* ) *∼*= D(*Y, FX*). We denote both sides of this bijection by (*−*)*b*, and for a morphism *f* in either of the two homsets we call *fb* the *transpose* of *f* . An adjunction as above has two units *η* : Id *⇒ GF* and *ι* : Id *⇒ F G*. For a morphism *f* : *X → GY* the transpose is given by *fb* = *Ff ◦ ιY* and, for *g* : *Y → FX*, by *gb* = *Gg ◦ ηX* . The standard example is C = D = Set with *F* = *G* = 2*−* the contravariant powerset functor.

To avoid too much of the (*−*)op notation, we treat *F* and *G* as *contravariant functors* between C and D, meaning that they reverse the direction of arrows, and refer to an adjunction as above as a *contravariant adjunction*. This should not lead to confusion, as all the adjunctions considered in this paper are contravariant.

*Factorization systems.* Let C be a category, and *E, M* classes of morphisms in C. The pair (*E, M*) is called a *factorization system* if (a) both *E* and *M* are closed under

isomorphisms, (b) for every morphism *f* in

C there is an (*E, M*)*-factorization*: a pair

*A*  *e* /*B¸*

*A*  *e* /*B¸*

of morphisms *e ∈ E* , *m ∈M* s.t. *m ◦ e* = *f* , *g*

and (c) for every commutative square as on

*ƒ g d*

*s*

*ƒ* (2)

the left-hand side of ([2](#_bookmark7)), with *e ∈ E* and

*C* /*m¸* /*D¸*

*C* /*m¸* /*D¸*

*m ∈ M*, there is a unique diagonal *d* making the right-hand side commute [[2](#_bookmark29)].

Both *E* and *M* are closed under composition of morphisms. Further, (*E, M*)- factorizations are unique up to isomorphism [[2](#_bookmark29)]. We denote morphisms in *E* by arrows of the form *A* /*B¸*and morphisms in *M* by arrows of the form *C* /*¸*/*D¸*. If *E* is the class of epimorphisms and *M* the class of monomorphisms then we speak of an (epi,mono)-factorization system. A standard example is the (epi,mono)- factorization system of the category Set of sets and functions.

Given a functor *F* : C *→* C on a category C with a factorization system (*E, M*), if *F* preserves morphisms in *M* then the factorization system lifts to coalg(*F* ) [[18](#_bookmark44),[1](#_bookmark28)]. If *F* preserves morphisms in *E* then the factorization system lifts to alg(*F* ). A category C is called *wellpowered* if, for every object *X*, there is (up to isomorphism) only a set of monomorphisms with codomain *X*. It is called *cowellpowered* if every object *X* has (up to isomorphism) only a set of epimorphisms with domain *X*.

# Minimization

In this section we recall from [[1](#_bookmark28)] the notion of minimization, and an associated ab- stract partition refinement procedure. Throughout this section, let C be a complete category with an (*E, M*)-factorization system, and *B* : C *→* C a functor.

**Definition 4.1** A *minimization* of a *B*-coalgebra (*X, c*) isa *B*-coalgebra (*E, ϵ*) with a coalgebra morphism *e* : (*X, c*) *→* (*E, ϵ*) with *e ∈ E* such that for every coalgebra morphism *e∗* : (*X, c*) *→* (*Y, d*) with *e∗ ∈ E* there is a unique coalgebra morphism *h* : (*Y, d*) *→* (*E, ϵ*) with *h ◦ e∗* = *e*.

A *minimization* of a *B*-coalgebra (*X, c*) isa *B*-coalgebra (*E, ϵ*) with a coalgebra morphism *e* : (*X, c*) *→* (*E, ϵ*) with *e ∈ E* such that for every coalgebra morphism *e∗* : (*X, c*) *→* (*Y, d*) with *e∗ ∈ E* there is a unique coalgebra morphism *h* : (*Y, d*) *→* (*E, ϵ*) with *h ◦ e∗* = *e*. If a minimization exists then it is unique up to isomorphism, therefore we often speak about *the* minimization. If *B* has a final coalgebra (*Z, ζ*) and *B* preserves *M*-morphisms, then the minimization of (*X, c*) is equivalently given by (*E, M*)-factorization (in coalg(*B*)) of the unique coalgebra morphism to (*Z, ζ*):

*X*  *e* /*E¸*/*¸m* /*Z¸*

*c є ζ*

/

*BX Be*

*BE*

*¸* /*¸*

*Bm*

/*B¸Z*

The procedure from [[1](#_bookmark28)] for computing a minimization is based on the final sequence. We denote the poset category of ordinal numbers by Ord.

**Definition 4.2** The *ﬁnal sequence W* : Ordop *→* C of *B* is the unique sequence defined by *W*0 = 1 (the final object of C), *Wi*+1 = *BWi* and *Wj* = lim*i<j Wi* for a limit ordinal *j*, whose connecting morphisms *wj,i* : *Wj → Wi* (with *i ≤ j*) satisfy *wi,i* = id, *wj*+1*,i*+1 = *Bwj,i* and if *j* is a limit ordinal then (*wj,i*)*i<j* is a limit cone.

Any coalgebra *c* : *X → BX* defines a unique cone (*ci* : *X → Wi*)*i∈*Ord satisfying *ci*+1 = *Bci ◦ c*. We use the notation *ci* throughout this paper to refer to elements of the above cone, for a coalgebra (*X, c*).

**Definition 4.3** For any coalgebra *c* : *X → BX* and ordinal *i*, we define the *i-* *minimization* to be the *E* -morphism *ei* : *X → Ei* of an (*E, M*)-factorization of *ci*.

The *Ei*’s form an ordinal indexed chain, with connecting morphisms *ej,i* : *Ej →*

*Ei* (for *i ≤ j*) arising by diagonalization (so that *ei* = *ei*+1*,i ◦ ei*+1 for all *i*).

The following theorem collects what we need to know about (*i*-)minimizations. The first two items concern the existence of minimizations, and the third is a tech- nique for computing *i*-minimizations.

**Theorem 4.4** *[*[*1*](#_bookmark28)*] Let c* : *X → BX be a coalgebra.*

1. *Suppose that E consists of epimorphisms, and suppose that the i-minimization ei* : *X → Ei of* (*X, c*) *is a coalgebra morphism from* (*X, c*) *to a B-coalgebra* (*Ei, ϵ*)*. Then* (*Ei, ϵ*) *is the minimization of* (*X, c*)*.*
2. *In addition to the above assumptions, suppose* C *is cowellpowered, and B pre-* *serves morphisms in M. Then the minimization of any B-coalgebra exists, with carrier Ei for some ordinal number i.*
3. *Suppose B preserves morphisms in M, and ei* : *X → Ei is the i-minimization of* (*X, c*)*. Then the E-morphism of an* (*E, M*)*-factorization of Bei ◦ c* : *X → BEi is the* (*i* + 1)*-minimization of* (*X, c*)*.*

**Example 4.5** Consider the Set functor *BX* = 2 *× XA*, whose coalgebras are de- terministic automata, with the factorization system given by epis and monos. For

an ordinal *i*, the set *Wi* in the final sequence of *B* consists of all languages over *A* where all words have length below *i*. Given a *B*-coalgebra (*X, c*), the function *ci* : *X → Wi* maps a state *x* to the set of words of length below *i* accepted by *x*. Its kernel is the relation *≡i* given in Section [2.1](#_bookmark2). Thus *Ei* is the quotient of states by *≡i*. The inductive computation of *ei* in Theorem [4.4](#_bookmark10)(iii) underlies partition refinement algorithms for deterministic automata. For details and more examples, see [[1](#_bookmark28)].

# Reachability

We define the notion of reachable part of an algebra, and a procedure to compute it. The definitions and results are dual to those of the previous section, but since they play an important role in the remainder of this paper we spell out some of the details, and state the dual of Theorem [4.4](#_bookmark10). Throughout this section, let D be a cocomplete category with an (*E, M*)-factorization system and *L* : D *→* D a functor. The *reachable part* of an *L*-algebra (*X, a*) is an *L*-algebra (*R, Q*) with a morphism *m* : (*R, Q*) *→* (*X, a*) with *m ∈ M*, satisfying the expected property dual to that of a minimization. If *L* has an initial algebra (*A, α*) and *L* preserves *E* -morphisms, then the reachable part of (*X, a*) is equivalently given by (*E, M*)-factorization (in alg(*L*))

of the unique algebra morphism from (*A, α*) to (*X, a*).

The *initial sequence V* : Ord *→* D of *L* is the unique sequence defined by *V*0 = 0 (the initial object of D)*, Vi*+1 = *LVi* and *Vj* = colim*i<jVi* for a limit ordinal *j*, whose connecting morphisms *v**i,j* : *Vi → Vj* (with *i ≤ j*) satisfy *vi,i* = id, *vi*+1*,j*+1 = *Lvi,j* and if *j* is a limit ordinal then (*vi,j*)*i<j* is a colimit cocone.

Any algebra *a* : *LX → X* defines a unique cocone (*ai* : *Vi → X*)*i∈*Ord satisying *ai*+1 = *a ◦ Lai*. We define the *i-reachable part* to be the *M*-morphism *mi* : *Ri → X* of an (*E, M*)-factorization of *ai*. The *Ri*’s form an ordinal indexed chain, with connecting morphisms *ri,j* : *Ri → Rj* (for *i ≤ j*) arising by diagonalization (so that *mi* = *mi*+1 *◦ ri,i*+1 for all *i*).

**Theorem 5.1** *Let a* : *LX → X be an algebra.*

1. *Suppose that M consists of monomorphisms, and suppose that the i-reachable part mi* : *Ri → X of* (*X, a*) *is an algebra morphism from an L-algebra* (*Ri, Q*) *to* (*X, a*)*. Then* (*Ri, Q*) *is the reachable part of* (*X, a*)*.*
2. *In addition to the above assumptions, suppose* D *is wellpowered, and L pre-* *serves morphisms in E. Then the reachable part of any L-algebra exists, with carrier Ri for some ordinal number i.*
3. *Suppose L preserves morphisms in E, and mi* : *Ri → X is the i-reachable part of* (*X, a*)*. Then the M-morphism of an* (*E, M*)*-factorization of a◦Lmi* : *LRi → X is the* (*i* + 1)*-reachable part of* (*X, a*)*.*

**Example 5.2** Let *L* be the Set endofunctor defined by *LX* = *A × X* + 1. As explained in Section [3](#_bookmark6), an algebra [*g, ι*]: *A×X* +1 *→ X* is a deterministic automaton with initial state *ι*(*∗*), transition function *g* and no final states. A set *Vi* in the initial sequence of *L* is the set of words of length below *i*, and the function [*g, ι*]*i* : *Vi → X* maps *w ∈ Vi* to the state that is reached after processing *w* from right to left:

[*g, ι*]*i*(*ε*)= *ι*(*∗*) and, for *a ∈ A* and *w ∈ Vi−*1, [*g, ι*]*i*(*aw*)= *g*(*a,* [*g, ι*]*i*(*w*)).

The *i*-reachable part *mi* : *Ri → X* is concretely presented by letting *Ri* be the set of states reachable from words of length below *i*, and *mi* the inclusion map. For *i* = 0 we have *V*0 = *∅*, hence *R*0 = *∅*. The computation of *mi*+1 : *Ri*+1 *→ X* from *mi* in Theorem [5.1](#_bookmark13) amounts to taking the image of *LRi* along [*g, ι*] *◦ Lmi*, i.e., *Ri*+1 = *{g*(*a, mi*(*x*)) *| a ∈ A, x ∈ Ri}∪ {ι*(*∗*)*}*. The reachable part of (*X,* [*g, ι*]) consists of all states that are reachable from some word in *A∗*, starting from the initial state.

# Minimization via reachability

We formulate the minimization construction sketched in Section [2.2](#_bookmark3) in terms of (co)algebras. The instantiation to deterministic automata is presented in Exam- ple [6.2](#_bookmark17), which can be read without necessarily understanding the abstract construc- tion. For the abstract construction, we assume:

(A1) categories C and D, both with (epi,mono)-factorization systems; (A2) a functor *B* : C *→* C that preserves epis;

(A3) a functor *L* : D *→* D that preserves monos;

(A4) a (contravariant) adjunction between functors *F* : Cop *→* D and *G* : Dop *→* C; (A5) a natural isomorphism *ρ* : *BG ⇒ GL*;

(A6) the existence of an initial *L*-algebra.

By (A1) . . . (A3), both C and D have (epi,mono)-factorization systems that extend to coalg(*B*) and alg(*L*) respectively. The contravariant adjunction of (A4) lifts, using the isomorphism in (A5), to a (contravariant) adjunction between *F* : coalg(*B*)op *→* alg(*L*) and *G* : alg(*L*)op *→* coalg(*B*) (see [[10](#_bookmark36)], and also [[13](#_bookmark39),[15](#_bookmark41)]).

**Theorem 6.1** *Assume (A1) . . . (A6) from the beginning of this section, and let* (*X, c*) *be a B-coalgebra. Let m* : (*R, Q*) *→ F* (*X, c*) *be the reachable part of F* (*X, c*)*. Take an (epi,mono)-factorization (in* coalg(*B*)*) of the adjoint transpose mb of m:*

*mb*

(*X, c*) /(*E¸, ϵ*) /*¸* /z*G¸* (*R, Q*) (3)

*Then* (*E, ϵ*) *is the minimization of* (*X, c*)*.*

The functor *F* is defined on objects by *F* (*X, c*)= (*FX, Fc◦ρb* ), where *ρb* : *LF ⇒*

*FB* is the *mate* of *ρ*, and *G* by *G*(*X, a*)= (*GX, ρ−*1 *◦Ga*). See [[14](#_bookmark40),[15](#_bookmark41)] for details. We

*X*

*X*

often abbreviate *Fc ◦ ρb* by *Fc*, and in particular we write ((*Fc*)*i* : *Vi → FX*)*i∈*Ord

*X*

for the cocone over the initial sequence of *L* induced by *F* (*X, c*).

The construction in Theorem [6.1](#_bookmark16) is based on [[15](#_bookmark41)], which in turn is based on techniques from coalgebraic modal logic. Indeed, a natural transformation *ρ* of the above form (without the assumption that it is an iso) is by now a standard way of defining the semantics of coalgebraic modal logic, see, e.g., [[14](#_bookmark40),[17](#_bookmark43)].

The minimization construction of [[15](#_bookmark41)] concerns a more general class of coalge-

bras, that may involve branching. As explained in Section [8](#_bookmark23), the factorization of *mb* yielding a minimal automaton can not be formulated in that setting. The con- struction is also connected to the one in [[4](#_bookmark30)], which however assumes a duality rather than a contravariant adjunction (making the factorization of *mb* unneccesary, since it is automatically an epi because of the duality). That construction rules out our example of deterministic automata below.

**Example 6.2** We apply the construction of Theorem [6.1](#_bookmark16) to deterministic automata over an alphabet *A*. The ingredients (A1) . . . (A6) of the beginning of this section are as follows: C = D = Set, *F* = *G* = 2*−* (the contravariant powerset functor), *B* : Set *→* Set is given by *BX* = 2 *× XA*, *L* : Set *→* Set is given by *LX* = *A× X* + 1. The required isomorphism *ρ* : *BG ⇒ GL* is 2 *×* (2*−*)*A ∼*= 2*A×−*+1. Recall that *L* has an initial algebra, given by the set of words *A∗*.

Let *⟨o, f⟩* : *X →* 2*×XA* be a *B*-coalgebra. The first step of the construction is to compute *F* (*X,⟨o, f⟩*)= (2*X,* 2*⟨o,ƒ⟩ ◦ρb* ), which we denote by [*g, ι*]: *A×*2*X* +1 *→* 2*X* . Intuitively, (2*X,* [*g, ι*]) is obtained by reversing and determinizing the automaton (*X,⟨o, f⟩*), where reversal comes from the application 2*⟨o,ƒ⟩* of the contravariant powerset functor. By computing the mate *ρb* of *ρ*, we obtain (see [[15](#_bookmark41),[23](#_bookmark49)] for details):

*X*

*g*(*a, S*)= *{x ∈ X | f* (*x*)(*a*) *∈ S}* and *ι*(*∗*)= *{x ∈ X | o*(*x*)= 1*} .*

The reachable part *R ⊆* 2*X* (technically, an inclusion map *m* : *R →* 2*X* ) consists of all reachable (sets of) states in (2*X,* [*g, ι*]). By Theorem [5.1](#_bookmark13), *R* can be obtained by computing *i*-reachable parts by induction on *i*, according to (see Example [5.2](#_bookmark14)):

*Ri*+1 = *{{x ∈ X | f* (*x*)(*a*) *∈ S}| a ∈ A, S ∈ Ri}∪ {{x ∈ X | o*(*x*)= 1*}}* and *R*0 = *∅*. We thus retrieve the reachable sets as constructed in Section [2.2](#_bookmark3).

Following Theorem [6.1](#_bookmark16), we compute an (epi,mono)-factorization of the transpose *mb* of *m*, and obtain a coalgebra (*E, ϵ*) which is the minimization of (*X,⟨o, f⟩*). The transpose *mb* : *X →* 2*R* is given by *mb*(*x*) = *{S ∈ R | x ∈ S}*. Concretely, the factorization *E* can be defined as the image of *X* along *mb*. But observe that we can also define *e* : *X → E* (and, implicitly, *E*) by *e*(*x*)= *{y | ∀S ∈ R* : *x ∈ S* iff *y ∈ S}*. Then *E* is the quotient of *X* by language equivalence, see Section [2.2](#_bookmark3).

# Relating minimization and reachability

We have seen how minimization can be computed either by a stepwise computation along the final sequence, or by a stepwise computation along an initial sequence followed by a factorization. Next we show that, when both approaches apply, there is a strong correspondence: the arrows from the initial sequence and those into the final sequences are each others adjoint transpose, up to isomorphism (Theo- rem [7.2](#_bookmark19)). Based on this correspondence, we derive an abstract method to compute the *i*-th partition from the *i*-th reachability step (Corollary [7.3](#_bookmark21)), generalizing the computation of *≡i* (or *Ei*) from *Ri* in Section [2.2](#_bookmark3).

Throughout this section we assume (A1) . . . (A5) from the beginning of Section [6](#_bookmark15), i.e., categories C and D with (epi,mono)-factorization systems, functors *B* : C *→* C preserving monos and *L* : D *→* D preserving epis, a contravariant adjunction between *F* and *G* and finally a natural iso *ρ* : *BG ⇒ GL*. We further assume that C is complete and D is cocomplete.

**Lemma 7.1** *Let W* : Ordop *→* C *be the ﬁnal sequence of B, and V* : Ord *→* D *the initial sequence of L. There is a natural isomorphism κ* : *W ⇒ GV* op : Ordop *→* C *satisfying κi*+1 = *ρVί ◦ Bκi for all ordinals i.*

The following is the heart of the matter, relating the cone (*ci* : *X → Wi*)*i∈*Ord over the final sequence of *B* induced by (*X, c*) to the cocone ((*Fc*)*i* : *Vi → FX*)*i∈*Ord over the initial sequence of *L* induced by *F* (*X, c*).

**Theorem 7.2** *Let* (*X, c*) *be a B-coalgebra. For any ordinal i, the following diagram commutes:*

*X*  *cί* /*W¸i*

(*F c*)*b*

*ί*

*κί*

z

*GVi*

**Corollary 7.3** *Let* (*X, c*) *be a B-coalgebra. Let mb* : *X → GRi be the transpose of the i-reachable part of F* (*X, c*)*. Then the epic morphism ei* : *X → Ei of an (epi,mono)-factorization of mb is the i-minimization of* (*X, c*)*. Further, if mi* : *Ri → FX is the reachable part of F* (*X, c*)*, then ei is the minimization of* (*X, c*)*.*

*i*

*i*

**Example 7.4** In Section [2.2](#_bookmark3) we have seen how the *i*-th partition of the states of a deterministic automaton can be obtained from the sets of states reachable in the reversed determinized automaton in less than *i* steps, by interpreting the reachable sets as splitters. This result is a special case (and, indeed, we derived it from) Corollary [7.3](#_bookmark21). To see this, let (*X, c*) be a deterministic automaton, recall from Example [4.5](#_bookmark11) that the *i*-th partition is the *i*-minimization of (*X, c*), and recall from Example [6.2](#_bookmark17) that the sets of states reachable in the reversed determinized au- tomaton are given by the *i*-reachable part *mi* of *F* (*X, c*). The “splitting” operation corresponds to a specific factorization of *mb*, similar to the last part of Example [6.2](#_bookmark17).

*i*

One may wonder whether there is a converse, i.e., if we can obtain the *i*-reachable part of *F* (*X, c*) from the *i*-minimization of (*X, c*). Example [2.1](#_bookmark5) shows that this is not the case: partition refinement for deterministic automata may stop earlier than the computation of reachable sets in the reversed determinized automaton.

Under the assumptions of Theorem [5.1](#_bookmark13), the reachable part of any *L*-algebra arises as one of the *i*-reachable parts, hence Corollary [7.3](#_bookmark21) shows that, in that case, Theorem [6.1](#_bookmark16) holds even if *L* does not have an initial algebra.

We can also use Corollary [7.3](#_bookmark21) to *combine* the minimization procedure based on *i*-minimizations and the one based on *i*-reachable parts. The possibility of com- puting the *i*-minimization from the *i*-reachable part suggests a procedure where we inductively compute *i*-reachable parts as in Theorem [5.1](#_bookmark13), compute *i*-minimizations along the way and terminate when the *i*-minimization is a minimization.

In this procedure, when computing the (*i* + 1)-minimization from the (*i* + 1)- reachable part, one would like to use the *i*-minimization as well. Concretely, for deterministic automata, given the partition *Ei* computed from the splitters *Ri* (Sec- tion [2.2](#_bookmark3)), and the new set of splitters *Ri*+1, we want to compute *Ei*+1 by splitting the partition in *Ei* according to the new splitters, i.e., those appearing in *Ri*+1 but not in *Ri*. Abstractly, one can compute *Ei*+1 from *Ei*, *Ri* and *Ri*+1 as follows.

**Lemma 7.5** *Suppose* C *has pullbacks. Let ei* : *X → Ei be the i-minimization of a coalgebra c* : *X → BX, and let ri,i*+1 : *Ri → Ri*+1 *be the arrow (see Section* [*5*](#_bookmark12)*) from the i-reachable part mi* : *Ri → FX to the* (*i* + 1)*-reachable part mi*+1 : *Ri*+1 *→ FX of F* (*X, c*)*. By Corollary* [*7.3*](#_bookmark21)*, mb* = *m∗ ◦ei for some mono m∗. Let P be the pullback*

*i*

*of m∗ and Gri,i*+1*:*

*b ί*+1

*m*

*X*

*h*

z /z

*P*

*eί*

z

*G¸Ri*+1

*Grί,ί*+1

*Ei* /*¸∗* /*G¸Ri*

*m*

*There is a unique mediating morphism h as above. The epic part of an (epi,mono)- factorization of h is the* (*i* + 1)*-minimization of* (*X, c*)*.*

To understand the above construction, consider the case of deterministic au- tomata, with *Ei* presented as a partition and *Ri* as a set of splitters, as above. The pullback *P* can be presented by *P* = *{*(*Q, C*) *∈ Ei ×* 2*Rί*+1 *| C ⊆ Ri*+1*\Ri}* (see the appendix). The function *h* : *X → P* maps *x* to the pair (*ei*(*x*)*, {S ∈ Ri*+1*\Ri | x ∈ X}*), i.e., the pair consisting of the equivalence class of *x* in *Ei* and the set of all “new” splitters, appearing in *Ri*+1 but not in *Ri*, containing *x*. The factorization of *h* can be presented by mapping each *x* to *{y ∈ ei*(*x*) *| ∀S ∈ Ri*+1*\Ri* : *x ∈ S* iff *y ∈ S}*, yielding the partition obtained by splitting *Ei* according to all the new splitters.

For deterministic automata, the inductive computation of *i*-reachable parts, and *i*-minimizations from them using Corollary [7.3](#_bookmark21) and Lemma [7.5](#_bookmark22) closely resembles the construction presented in [[8](#_bookmark34), Algorithm 1] and the end of Section [2.2](#_bookmark3). However, the algorithm in [[8](#_bookmark34)] terminates only when the reachable part *R* has been found, whereas using Corollary [7.3](#_bookmark21) we can terminate once the *i*-minimization is a minimization. This may occur before the *i*-reachable part is the reachable part (Example [2.1](#_bookmark5)).

# Branching systems

In the previous sections, we studied minimization of *B*-coalgebras, with determin- istic automata as the main example. Next, we investigate the case of systems involving branching, such as non-deterministic or alternating automata. Here, we do not focus on finding minimal non-deterministic automata: it is well-known that they are not unique, and it is in fact much less obvious how to even define the notion of minimization. Instead, we show how to compute language equivalence inductively based on reachability.

*Language semantics.* We are interested in coalgebras for a composite functor of the form *BT* or *TB*, where *B* models the observations that are to be recorded in traces, and *T* is the type of branching. For instance, taking *BX* = 2 *×XA* as before and *T* = *P* the (covariant) powerset functor, *BT* -coalgebras are non-deterministic automata; and with *T* = *PP*, *BT* -coalgebras are a form of alternating automata. Taking *B* to be a polynomial functor and *T* = *P* one obtains tree automata as *TB*- coalgebras, and for a certain choice of *T* one obtains weighted tree automata [[15](#_bookmark41)]. Because of space limitations, we focus on *BT* -coalgebras in this section, and only treat the example of non-deterministic automata.

The final semantics of *BT* -coalgebras such as those in the above examples (which exists, for instance, when we restrict *T* to the finite powerset functor) does, in general, not coincide with the expected language semantics. We recall the ap- proach of [[15](#_bookmark41)] to define language semantics based on initial algebras rather than final coalgebras. To this end, assume functors *B, T* : C *→* C, a functor *L* : D *→* D with an initial algebra and, as before (Section [6](#_bookmark15)), a contravariant adjunction be- tween *F* and *G*. To define language semantics, we assume a natural transforma- tion *ρ* : *BG ⇒ GL* (not necessarily an isomorphism) and a natural transforma- tion *α* : *TG ⇒ G*. This induces a functor *Fα* : coalg(*BT* ) *→* alg(*L*) defined by

*Fα*(*X, c*) = (*FX, Fc ◦ ρb ◦ Lαb*), see [[15](#_bookmark41)] for details and explanation. Given a

*TX*

coalgebra *c* : *X → BTX*, one then computes the unique map *s* : (*A, α*) *→ Fα*(*X, c*) from the initial *L*-algebra, and defines the *(language) semantics* of (*X, c*) to be the transpose *sb* : *X → GA* of *s*. We define the *language quotient* of (*X, c*) as the epic part of an (epi,mono)-factorization of the language semantics *sb*.

**Theorem 8.1** *Suppose that* C *and* D *have (epi,mono)-factorization systems. Let c* : *X → BTX be a coalgebra, and let m* : (*R, Q*) *→ Fα*(*X, c*) *be the reachable part of Fα*(*X, c*)*. Then the epic part of an (epi,mono)-factorization (in* C*) of the transpose mb* : *X → GR is the language quotient of* (*X, c*)*.*

**Example 8.2** Let *F* = *G* = 2*−* be the contravariant powerset adjunction, let

*BX* = 2 *× XA* and *LX* = *A× X* + 1. A non-deterministic automaton is a coalgebra

*⟨o, f⟩* : *X →* 2 *×* (*PX*)*A* for the composite functor *BP*. Define the components of

*α* : *P*2*− ⇒* 2*−* by union, and let *ρ* be the isomorphism from Example [6.2](#_bookmark17).

We denote the algebra *Fα*(*X,⟨o, f⟩*) by [*g, ι*]: *A ×* 2*X* +1 *→* 2*X* . It is given by

*g*(*a, S*)= *{x | ∃y ∈ f* (*x*)(*a*) s.t. *y ∈ S} ι*(*∗*)= *{x | o*(*x*)= 1*}*

(cf. Example [6.2](#_bookmark17)). Hence, the unique algebra morphism *s* : *A∗ →* 2*X* satisfies *s*(*ε*)=

*{x | o*(*x*) = 1*}* and *s*(*aw*) = *{x | ∃y ∈ f* (*x*)(*a*) s.t. *y ∈ s*(*w*)*}*. The transpose *sb* is thus the usual semantics of non-deterministic automata [[15](#_bookmark41)].

The reachable part *R ⊆* 2*X* consists of all reachable (sets of) states in (2*X,* [*g, ι*]). By Theorem [5.1](#_bookmark13), *R* can be obtained by computing *i*-reachable parts by induction on *i*, according to (see Example [5.2](#_bookmark14)) *R*0 = *∅* and:

*Ri*+1 = *{{x | ∃y ∈ f* (*x*)(*a*) s.t. *y ∈ S}| a ∈ A, S ∈ Ri}∪ {{x ∈ X | o*(*x*)= 1*}} .*

The function *mb* : *X →* 2*R* maps every state *x* to those sets in *R* that contain it, and, like in Example [6.2](#_bookmark17), we may define the epic part of a factorization of *mb* by *e*(*x*) = *{y ∈ X | ∀S ∈ R* : *x ∈ S* iff *y ∈ S}*. Then *e* maps every state *x* to the equivalence class of states that accept the same language.

It was shown in [[15](#_bookmark41)] that, in the context of Theorem [8.1](#_bookmark24), if the natural trans- formation *ρ* is an isomorphism, then *GR* is a *B*-coalgebra, whose unique morphism *h* to the final coalgebra is mono, and such that *sb* = *h ◦ mb*. This means that the construction yields a *B*-coalgebra whose states are behaviourally equivalent if and only if they are equal, and whose final semantics represents the language semantics of the original automaton. Instances where the conditions of the construction are met include non-deterministic, alternating and weighted automata, see [[15](#_bookmark41)]. Here, our characterization of reachable sets shows how to compute the factorization of the morphism from the initial algebra.

The construction from [[15](#_bookmark41)] mentioned above is reminiscent of Brzozowski’s min- imization algorithm, but it does not generalize the construction for *B*-coalgebras in Theorem [6.1](#_bookmark16). The latter is based on another (epi,mono)-factorization in coalg(*B*). In the example of non-deterministic automata, the construction from [[15](#_bookmark41)] yields a deterministic automaton, which is not minimal in any reasonable sense: it may contain states that are not reachable from any state in the image of *X* along *mb*. Note that the reachable states can not be obtained in general by taking the image of the state space *X* along *mb*, since the minimal deterministic automaton may have more states than the non-deterministic one that we start with. Instead, one should construct the *least subautomaton* containing this image. In Set, this is easy to define (e.g., [[21](#_bookmark47)]), but at the abstract level it seems less clear.

*Language equivalence: a dual view.* We briefly consider a construction for branching systems that is not unlike the minimization construction of Section [4](#_bookmark8). To this end, suppose C is complete and D is cocomplete, and let *V* be the initial sequence of *L* : D *→* D. Given *c* : *X → BTX*, there is a unique cone (*c*¯*i* : *X → GVi*)*i∈*Ord over *GV* op satisfying the following:

*c*¯*i*+1

= ( *X*  *c* /*B¸T X*  *BTc*¯*ί* /*B¸T GVi*

*BαVί* /*B¸GV*

*ρVί* /*G¸LV* ) *.*

Call the epic morphism of an (epi,mono)-factorization of *c*¯*i* the *i-language quotient* of (*X, c*). (If *ρ* : *BG ⇒ GL* is an isomorphism, then the above cone can equivalently be defined over the final sequence.)

*i*

*i*

*i*

*i*

**Theorem 8.3** *Let c* : *X → BTX be a coalgebra, and* ((*Fαc*)*i* : *Vi → FX*)*i∈*Ord *the cocone over the initial sequence of L induced by Fα*(*X, c*)*. For any i, we have c*¯*i* = (*Fαc*)*b. Further, let mi* : *Ri → FX be the i-reachable part of Fα*(*X, c*)*. Then the epic morphism of an (epi,mono)-factorization of the transpose mb* : *X → GRi*

*is the i-language quotient of* (*X, c*)*.*

The crucial property for the minimization construction in Theorem [4.4](#_bookmark10) is that the (*i* + 1)-minimization can be computed from the *i*-minimization. This approach does not seem to work for *i*-language quotients, since *α* and *ρ* are not (componentwise)

mono in general. Indeed, for non-deterministic automata, *Ei* is the quotient of states by language equivalence of words with length below *i*, and it is unclear how one could obtain *Ei*+1 only from *Ei* and the automaton under consideration.

In the previous section (Lemma [7.5](#_bookmark22)), we have seen how the (*i* + 1)-minimization can be obtained given the (*i* + 1)-reachable part and the *i*-minimization. A sim- ilar approach could be taken here, generating a sequence of *i*-language quotients. However, it is not clear whether this is of much use. The problem is that, in the current context, it may be the case that the *i*-language quotient is isomorphic to the (*i* + 1)-language quotient, but not to the *j*-language quotient for some *j > i* + 1. Hence, we can not use such an isomorphism as a termination condition.

**Example 8.4** Consider the following non-deterministic automaton, where the only accepting state is *u*.

*w* *v*

*c*

*a*

*u*,*,¸*¸,*,,¸*

*b*

*a,b*

*c*

*x*,*,*  *y* ¸*,,c*

*a*

*a*

*z*,*,*

*a*

The *i*-reachable sets *Ri*, as computed in Example [8.2](#_bookmark25), and the *i*-language quotients

*Ei*, which we compute from the *Ri*’s (Theorem [8.3](#_bookmark26)), are:

*E*0 = *{{u, x, y, z, w, v}} R*0 = *∅*

*E*1 = *{{u}, {x, y, z, w, v}} R*1 = *{{u}}*

*E*2 = *{{u}, {w, v}, {z}, {x}, {y}} R*2 = *{{u}, {w, v}, {x, z}, {y, z}}*

*E*3 = *{{u}, {w, v}, {z}, {x}, {y}} R*3 = *{{u}, {w, v}, {x, z}, {y, z}, ∅, {z}}*

*E*4 = *{{u}, {w}, {v}, {z}, {x}, {y}} R*4 = *{{u}, {w, v}, {x, z}, {y, z}, ∅, {z}, {v}}*

Notice that *E*3 = *E*2, but *E*4 */*= *E*3. Indeed, all states except *w* and *v* are distin- guished by the empty word or a word of length 1, whereas it requires a word of length 3 to distinguish *w* and *v*.

# Future work

We established a connection between partition refinement and Brzozowski’s min- imization construction, based on an abstract coalgebraic perspective. Our inter- est was to understand deterministic automata, which is hence the one example we cover in detail. The necessary assumptions of our results are also satisfied by Moore automata (and stream systems), and potentially other examples (e.g., [[23](#_bookmark49)]). In particular, it would be interesting to use the dualities of [[20](#_bookmark46)] and our results on branching systems to develop generic constructions for canonical branching systems. In this context, the connection to weak factorization systems as used in [[1](#_bookmark28)] and the

approach of [[16](#_bookmark42)] also remain to be understood. Further, the interaction between minimization and coalgebraic determinization constructions is left open.

# References

1. Ad´amek, J., F. Bonchi, M. Hu¨lsbusch, B. K¨onig, S. Milius and A. Silva, *A coalgebraic perspective on minimization and determinization*, in: *Procs. FOSSACS 2012*, LNCS **7213**, 2012, pp. 58–73.
2. Ad´amek, J., H. Herrlich and G. E. Strecker, “Abstract and Concrete Categories - The Joy of Cats,” Dover Publications, 2009.
3. Berstel, J., L. Boasson, O. Carton and I. Fagnot, *Minimization of automata*, CoRR **abs/1010.5318**

(2010), to appear in the Handbook of Automata.

1. Bezhanishvili, N., C. Kupke and P. Panangaden, *Minimization via duality*, in: *Procs. WoLLIC 2012*, LNCS **7456**, 2012, pp. 191–205.
2. Bonchi, F., M. M. Bonsangue, H. H. Hansen, P. Panangaden, J. J. M. M. Rutten and A. Silva, *Algebra-* *coalgebra duality in Brzozowski’s minimization algorithm*, ACM Trans. Comput. Log. **15** (2014), p. 3.
3. Bonchi, F., M. M. Bonsangue, J. J. M. M. Rutten and A. Silva, *Brzozowski’s algorithm (co)algebraically*, in: *Logic and Program Semantics - Essays Dedicated to Dexter Kozen on the Occasion of His 60th* *Birthday*, 2012, pp. 12–23.
4. Brzozowski, J., *Canonical regular expressions and minimal state graphs for definite events.*, Mathematical Theory of Automata **12** (1962), pp. 529–561.
5. Champarnaud, J., A. Khorsi and T. Parantho¨en, *Split and join for minimizing: Brzozowski’s algorithm*, in: M. Bal´ık and M. Sim´anek, editors, *Proceedings of the Prague Stringology Conference 2002, Prague, Czech Republic, September 23-24, 2002* (2002), pp. 96–104.
6. Garc´ıa, P., D. L´opez and M. V´azquez de Parga, *DFA minimization: Double reversal versus split minimization algorithms*, Theor. Comput. Sci. **583** (2015), pp. 78–85.
7. Hermida, C. and B. Jacobs, *Structural induction and coinduction in a fibrational setting*, Inf. and Comp. **145** (1997), pp. 107–152.
8. Hopcroft, J. E., *An n log n algorithm for minimizing states in a finite automaton*, in: *Theory of Machines and Computations*, 1971, pp. 189–196.
9. Hopcroft, J. E. and J. D. Ullman, “Introduction to Automata Theory, Languages and Computation,” Addison-Wesley, 1979.
10. Jacobs, B., A. Silva and A. Sokolova, *Trace semantics via determinization*, J. Comput. Syst. Sci. **81**

(2015), pp. 859–879.

1. Klin, B., *Coalgebraic modal logic beyond sets*, ENTCS **173** (2007), pp. 177–201.
2. Klin, B. and J. Rot, *Coalgebraic trace semantics via forgetful logics*, in: *FoSSaCS 2015. Proceedings*, 2015, pp. 151–166.
3. K¨onig, B. and S. Ku¨pper, *Generic partition refinement algorithms for coalgebras and an instantiation* *to weighted automata*, in: *Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014, Rome, Italy, September 1-3, 2014. Proceedings*, 2014, pp. 311–325.
4. Kupke, C. and D. Pattinson, *Coalgebraic semantics of modal logics: An overview*, Theor. Comput. Sci.

**412** (2011), pp. 5070–5094.

1. Kurz, A., “Logics for coalgebras and applications to computer science,” Ph.D. thesis, Ludwigs- Maximilians-Universit¨at Mu¨nchen (2000).
2. Moore, E. F., *Gedanken-experiments on sequential machines*, Automata studies **34** (1956), pp. 129–153.
3. Myers, R. S. R., J. Ad´amek, S. Milius and H. Urbat, *Coalgebraic constructions of canonical nondeterministic automata*, Theor. Comput. Sci. **604** (2015), pp. 81–101.
4. Rutten, J. J. M. M., *Universal coalgebra: a theory of systems*, Theor. Comput. Sci. **249** (2000), pp. 3–80.
5. Sakarovitch, J., “Elements of Automata Theory,” Cambridge University Press, 2009.
6. Salamanca, J., M. Bonsangue and J. Rot, *Duality of equations and coequations via contravariant adjunctions*, in: *CMCS*, 2016, to appear.
7. Watson, B. W., “Taxonomies and Toolkits of Regular Language Algorithms,” Ph.D. thesis, Eindhoven University of Technology (1995).

# Proofs of Section [4](#_bookmark8)

Theorem [4.4](#_bookmark10) is proved in [[1](#_bookmark28)] (with item (i) inlined in the proof of item (ii)). Because of this presentation difference, and for convenience, we recall the proof here. First, we need the following technicality, see [[1](#_bookmark28)].

**Lemma A.1** *Let h* : (*X, c*) *→* (*Y, d*) *be a coalgebra homomorphism. Then ci* = *di◦h*

*for all ordinals i.*

**Theorem 4.4** *[*[*1*](#_bookmark28)*] Let c* : *X → BX be a coalgebra.*

* 1. *Suppose that E consists of epimorphisms, and suppose that the i-minimization ei* : *X → Ei of* (*X, c*) *is a coalgebra morphism from* (*X, c*) *to a B-coalgebra* (*Ei, ϵ*)*. Then* (*Ei, ϵ*) *is the minimization of* (*X, c*)*.*
  2. *In addition to the above assumptions, suppose* C *is cowellpowered, and B pre- serves morphisms in M. Then the minimization of any B-coalgebra exists, with carrier Ei for some ordinal number i.*
  3. *Suppose B preserves morphisms in M, and ei* : *X → Ei is the i-minimization of* (*X, c*)*. Then the E-morphism of an* (*E, M*)*-factorization of Bei ◦ c* : *X → BEi is the* (*i* + 1)*-minimization of* (*X, c*)*.*

**Proof.**

1. By assumption, *ei* is a coalgebra homomorphism from (*X, c*) to (*Ei, ϵ*). Let *h* : (*X, c*) *→* (*Y, d*) be a coalgebra morphism with *h ∈ E* . By Lemma [A.1](#_bookmark51), the upper right triangle in the diagram on the left-hand side commutes:

*X*  *h* /*Y ¸ X*  *h* /*Y ¸*

*eί cί d ί e*

*ί*

*e*

*d*

*∗*

*ί*

z , *7*

*Ei* /*¸mί* /*W¸i Ei* /*¸mί* /*W¸i*

By Definition [4.3](#_bookmark9), the *i*-minimization of (*X, c*) is an (*E, M*)-factorization of *ci* with *E* -morphism *ei*; we denote the *M*-morphism by *mi*, hence the lower left triangle commutes by definition. As a consequence of commutativity of the square, we obtain a unique diagonal *e∗* making the diagram on the right-hand side commute. It only remains to be shown that *e∗* is a coalgebra morphism. This follows since *h* is epic and both *e∗ ◦ h* = *ei* and *h* are coalgebra mor- phisms [[21](#_bookmark47), Lemma 2.4].

1. First, for any given *i*, let *ei* : *X → Ei* be the *i*-minimization of (*X, c*), with corresponding *M*-morphism *mi* (i.e., such that *ci* = *mi ◦ ei*). The outside of

the following diagram commutes:

*X eί*+1 /*E¸*

*i*+1

*c*

*єί m*

*BX*

*Beί*

*ί*+1

*BE* /*¸* /*B¸W*

*i* *i*

*Bmί*

where *Bmi* is in *M* by assumption. Thus, we obtain a diagonal *ϵ*.

As explained in [[1](#_bookmark28)], since C is cowellpowered and the *ei*’s form a chain of epimorphisms with domain *X*, there is an *i* such that the arrow *ei*+1*,i* : *Ei*+1 *→ Ei* is an isomorphism. We denote its inverse by *ι* : *Ei → Ei*+1, then *ι◦ei* = *ei*+1 (since *ei* = *ei*+1*,i ◦ ei*+1, see Section [4](#_bookmark8)). We obtain a coalgebra on *Ei* turning *ei* into a coalgebra morphism:

*X*  *eί* /*Ei¸*

*eί*+1

*ι*

z

*c Ei*+1

*єί*

/

*BX Beί B¸Ei*

By (i), the coalgebra on *Ei* is the minimization of (*X, c*).

1. Let *mi* be the *M*-morphism such that *ci* = *mi ◦ ei*. Then *ci*+1 is the upper path in the diagram below.

*X*  *c* /*B¸X*  *Beί* /˛*B¸ Ei* /*¸Bmί* /*B¸Wi* = *Wi*+1

*eί*+1

z ˛

*E*

*i*+1

(A.1)

Notice that *Bmi* is in *M*, since *B* preserves *M*-morphisms by assumption. Let *ei*+1 : *X → Ei*+1 be the *E* -morphism of an (*E, M*)-factorization of *Bei ◦ c*. We obtain an (*E, M*)-factorization of *ci*+1, since *M*-morphisms are closed under composition. Thus *ei*+1 is the (*i* + 1)-minimization of (*X, c*).

*2*

# Proofs of Section [5](#_bookmark12)

**Theorem 5.1** *Let a* : *LX → X be an algebra.*

* 1. *Suppose that M consists of monomorphisms, and suppose that the i-reachable part mi* : *Ri → X of* (*X, a*) *is an algebra morphism from an L-algebra* (*Ri, Q*) *to* (*X, a*)*. Then* (*Ri, Q*) *is the reachable part of* (*X, a*)*.*
  2. *In addition to the above assumptions, suppose* D *is wellpowered, and L pre- serves morphisms in E. Then the reachable part of any L-algebra exists, with carrier Ri for some ordinal number i.*
  3. *Suppose L preserves morphisms in E, and mi* : *Ri → X is the i-reachable part of* (*X, a*)*. Then the M-morphism of an* (*E, M*)*-factorization of a◦Lmi* : *LRi → X is the* (*i* + 1)*-reachable part of* (*X, a*)*.*

**Proof.** This follows directly by duality and Theorem [4.4](#_bookmark10): the factorization system (*E, M*) on D yields the factorization system (*M, E* ) on Dop, *L*-algebras in D are *L*op-algebras in Dop and (*i*-)reachable parts in D are (*i*-)minimizations in Dop.

For item (iii), it may be helpful to see a direct proof. Let *ei* be the *E* -morphism such that *mi ◦ ei* = *ai*. Consider the following diagram, where the horizontal path is *ai*+1 : *Vi*+1 *→ X*, and *mi*+1 : *Ri*+1 *→ X* is the *M*-morphism of an (*E, M*)- factorization of *a ◦ Lmi*:

*Vi*+1 = *LVi*  *Leί* /*L¸R¸i*  *Lmί* /*LX¸*  *a* /\_*X¸¸* (B.1)

z

*R*

*i*+1

\_*¸mί*+1

The morphism *Lei* is in *E* , since *L* preserves *E* -morphisms by assumption. Since *E* -morphisms compose, this yields a factorization of *ai*+1, so that *mi*+1 is the (*i* + 1)- reachable part of (*X, a*). *2*

# Proofs of Section [6](#_bookmark15)

**Theorem 6.1** *Assume (A1) . . . (A6) from the beginning of this section, and let* (*X, c*) *be a B-coalgebra. Let m* : (*R, Q*) *→ F* (*X, c*) *be the reachable part of F* (*X, c*)*. Take an (epi,mono)-factorization (in* coalg(*B*)*) of the adjoint transpose mb of m:*

*mb*

(*X, c*) /(*E¸, ϵ*) /*¸* /z*G¸* (*R, Q*) (3)

*Then* (*E, ϵ*) *is the minimization of* (*X, c*)*.*

**Proof.** Since *L* has an initial algebra (*A, α*), *m* is the monic part of an (epi,mono)- factorization *m◦e* : (*A, α*) *→ F* (*X, c*). Because *G* is a right adjoint, it maps colimits to limits, hence *G*(*A, α*) is a final coalgebra. Further, because *G* is a right adjoint and *e* is an epi, *Ge* is a mono (into the final coalgebra). Take an (epi,mono)-factorization of *mb*, and compose with *Ge*:

*mb*

(*X, c*) /(*E¸, ϵ*) /*¸* /z*G¸* (*R, Q*) /*¸Ge* /*G¸*(*A, α*)

Since monos are closed under composition, we have an (epi,mono)-factorization of the (unique) coalgebra morphism from (*X, c*) to the final *B*-coalgebra, i.e., (*E, ϵ*) is the minimization of (*X, c*). *2*

# Proofs of Section [7](#_bookmark18)

**Lemma 7.1** *Let W* : Ordop *→* C *be the ﬁnal sequence of B, and V* : Ord *→* D *the initial sequence of L. There is a natural isomorphism κ* : *W ⇒ GV* op : Ordop *→* C *satisfying κi*+1 = *ρVί ◦ Bκi for all ordinals i.*

**Proof.** We define *κi* by transfinite induction. The successor step is given by the statement of the lemma. For a limit ordinal *j*, suppose we have an isomorphism *κi* : *Wi ⇒ GVi* for all *i < j*. Since *G* is a right (contravariant) adjoint it maps colimits to limits, hence *GVj* = *G*(colim*i<jVi*) = lim*i<j GVi*. The aim is thus to find an isomorphism *κj* : lim*i<j Wi →* lim*i<j GVi*. By the inductive hypothesis we obtain cones

(*κi ◦ wj,i* : lim *Wi → GVi*)*i<j* and (*κ−*1 *◦ Gvi,j* : lim *GVi → Wi*)*i<j .*

*i<j*

*i i<j*

By the universal property of lim*i<j GVi* and lim*i<j Wi*, we then obtain morphisms

*κj* : lim*i<j Wi →* lim*i<j GVi* and *κ−*1 : lim*i<j GVi →* lim*i<j Wi*.

*j*

*κj*

lim*i<j Wi* ¸

*κ−*1

*wj,ί j*

lim*i<j GVi*

*Gvί,j*

*Wi* ¸

*κί*

*GVi*

z

*−*1

*κ*

*ί*

The naturality squares as above are satisfied for each ordinal *i* with *i ≤ j*, and it is not difficult to prove that *κj* and *κ−*1 are indeed each others inverse. *2*

*j*

For the proof of Theorem [7.2](#_bookmark19) (and Theorem [8.3](#_bookmark26)), we will use the following result, which assumes functors *B, L*, a contravariant adjunction between *F* an *G* (as in Section [6](#_bookmark15)) and a natural transformation *ρ* : *BG ⇒ GL*. Here *ρ* is not assumed to be an isomorphism; in this setting, the lifting *F* : coalg(*B*)op *→* alg(*L*) of *F* is defined (as in Section [6](#_bookmark15)), but, in general it does not have a right adjoint. As before, we denote the mate of *ρ* by *ρb* : *LF ⇒ FB*.

**Lemma D.1** *Let c* : *X → BX be a coalgebra, and* ((*Fc*)*i* : *Vi → FX*)*i∈*Ord *the cocone over the initial sequence of L induced by F* (*X, c*)*. There is a unique cone* (*cρ* : *X → GVi*)*i∈*Ord *over GV* op *such that*

*i*

*ρV*

*ρ*  *c* /*¸*

*Bcρ* /*¸* /*¸*

*ci*+1

= ( *X*

*BX*  *ί BGVi*  *ί GLVi* ) *.*

*For every i ∈* Ord*, we have cρ* = (*Fc*)*b.*

*i* *i*

**Proof.** Let (*X, c*) be a coalgebra. Uniqueness of the cone follows from the fact that when *j* is a limit ordinal, then *GVj* = *G*colim*i<jVi* = lim*i<j GVi*, where the latter equality holds since *G* is a right adjoint.

We show that

* 1. ((*Fc*)*b* : *X → GVi*)*i∈*Ord is a cone over *GV* op;
  2. for all *i*, we have (*Fc*)*b*

*i*

*i*+1

= *ρVί*

* *B*(*Fc*)*b ◦ c*.

Since (*cρ* : *X → Wi*)*i∈*Ord is the unique cone with the property *cρ*

*i*

= *ρV*

* *Bcρ ◦ c*,

*i*

it then follows that *cρ* = (*Fc*)*b* for all *i*.

*i*+1 *ί* *i*

*i* *i*

* 1. Let *i, j* be ordinals with *i ≤ j*. Since ((*Fc*)*i* : *Vi → FX*)*i∈*Ord is a cone over the initial sequence *V* , the triangle on the left-hand side commutes:

*Vj*

*vί,j*

(*F c*)*b*

*X GV*,*i ,*

,*,* (*F c*)

*j*

(*F c*)*b*

*j*

*ί* /*¸*

*Gvί,j*

zz

*Vi*  /*F¸X*

(*F c*)*ί*

*GVj*

As a consequence, the triangle on the right-hand side commutes.

*X*

* 1. By definition of *F* we have *F* (*X, c*) = *Fc ◦ ρb* , and by definition of the cone ((*Fc*)*i* : *Vi → FX*)*i∈*Ord induced by *F* (*X, c*), the following commutes for any *i*:

*LVi*

*L*(*F c*)*ί*

(*F c*)*ί*+1

*b X*

/*LF¸X*  /*F¸BX*

*ρ*

*Fc*

z*˛*

/*F¸X*

(D.1)

Consider the following diagram.

*ί*+1

(*F c*)*b*

*X*

*ηX* /*G¸F X*  *GF c* /*G¸F B*,*X,*

*GρX*

*b*

/*¸*

*GLF X*

,*,*

*GL*(*F c*)

*ί*

/*¸*

*GLV*

z

,*i,*

*c ηBX*

*ρFX*

*ρVί*

(D.2)

z*B X*  *BηX* /*B¸GF X BG*(*F c*)*ί* /*B¸GV*

¸*,i*

*B*(*F c*)*b*

*ί*

The upper crescent commutes by ([D.1](#_bookmark53)) and the definition of the adjoint trans- pose, and the lower crescent commutes as well by definition of the transpose. The left triangle and right square commute by naturality. The middle square is a standard property relating *ρ* and its mate, see, e.g., (the full version of) [[15](#_bookmark41)]. Commutativity of the outside of the diagram is the desired property.

*2*

**Theorem 7.2** *Let* (*X, c*) *be a B-coalgebra. For any ordinal i, the following diagram commutes:*

*X*  *cί* /*W¸i*

(*F c*)*b*

*ί*

*κί*

z

*GVi*

**Proof.** Let *κ* : *W ⇒ GV* op be the isomorphism from Lemma [7.1](#_bookmark20). Consider the cone (*ci* : *X → Wi*)*i∈*Ord induced by (*X, c*). By naturality of *κ*, this yields a cone (*κi ◦ ci* : *X → GVi*)*i∈*Ord over *GV* op. Given an ordinal *i*, consider the following diagram:

*X cί*+1 /*B¸W*

*i*

*c*

/*¸*

*κί*+1 /*G¸LV*

*ρVί*

,*i,*

/*¸*

*BX Bcί*

*BWi Bκί*

*BGVi*

The left square commutes by definition of (*ci*)*i∈*Ord and the right square commutes by Lemma [7.1](#_bookmark20). We have shown that (*κi ◦ ci* : *X → GVi*)*i∈*Ord is a cone over *GV* op, satisfying *κi*+1 *◦ ci*+1 = *ρVί ◦ B*(*κi ◦ ci*) *◦ c*. By Lemma [D.1](#_bookmark52), we obtain *κi ◦ ci* = (*Fc*)*i* for all *i*. *2*

**Corollary 7.3** *Let* (*X, c*) *be a B-coalgebra. Let mb* : *X → GRi be the transpose of the i-reachable part of F* (*X, c*)*. Then the epic morphism ei* : *X → Ei of an (epi,mono)-factorization of mb is the i-minimization of* (*X, c*)*. Further, if mi* : *Ri → FX is the reachable part of F* (*X, c*)*, then ei is the minimization of* (*X, c*)*.*

*i*

*i*

**Proof.** The arrow *mi* : *Ri → FX* is the *i*-reachable part of *F* (*X, c*), thus it is part of a factorization *mi ◦ e∗* = (*Fc*)*i* for some epi *e∗* : *Vi → Ri*. Consider the diagram:

*X*  *cί* /*W¸i*

,

(*F c*)*b*

*ί*

*mb*

*κ*

*ί*

*ί*

*i* /*¸* /*G¸Ri* /*¸ ∗*

z

z*\_*

*Ge*

/*G¸Vi*

*,*

*−*1

*E*

where the lower left triangle is an (epi,mono)-factorization of *mb*. The middle triangle commutes by construction of *mi*, and the upper right by Theorem [7.2](#_bookmark19). The arrow *Ge∗* is mono, since *e∗* is epi and *G* is a right adjoint. Hence we obtained an (epi,mono)-factorization of *ci*, so the epic part is the *i*-minimization of (*X, c*).

*i*

For the second part of the statement, suppose *mi* is the reachable part of *F* (*X, c*), meaning in particular that there is an algebra (*Ri, Q*) on *Ri* turning *mi* : (*Ri, Q*) *→ F* (*X, c*) into an algebra morphism. Then the adjoint transpose *mb* (in the lifted adjunction between *F* and *G*) is a coalgebra morphism *mb* : (*X, c*) *→ G*(*Ri, Q*). Consider the epic part of a factorization *ei* : (*X, c*) *→* (*Ei, ϵ*) of this coalgebra mor- phism *mb*. The underlying map *ei* : *X → Ei* is the epic part of the factorization of *mb* (in C), hence, by the first part of the corollary, it is the *i*-minimization. Since *ei* is a coalgebra morphism, by Theorem [4.4](#_bookmark10) it is the minimization of (*X, c*). *2*

*i*

*i*

*i*

*i*

**Lemma 7.5** *Suppose* C *has pullbacks. Let ei* : *X → Ei be the i-minimization of a coalgebra c* : *X → BX, and let ri,i*+1 : *Ri → Ri*+1 *be the arrow (see Section* [*5*](#_bookmark12)*) from the i-reachable part mi* : *Ri → FX to the* (*i* + 1)*-reachable part mi*+1 : *Ri*+1 *→ FX of F* (*X, c*)*. By Corollary* [*7.3*](#_bookmark21)*, mb* = *m∗ ◦ei for some mono m∗. Let P be the pullback*

*i*

*of m∗ and Gri,i*+1*:*

*b ί*+1

*m*

*X*

*h*

z /z

*P*

*eί*

z

*G¸Ri*+1

*Grί,ί*+1

*Ei* /*¸∗* /*G¸Ri*

*m*

*There is a unique mediating morphism h as above. The epic part of an (epi,mono)- factorization of h is the* (*i* + 1)*-minimization of* (*X, c*)*.*

**Proof.** The arrow *ri,i*+1 satisfies *mi*+1 *◦ ri,i*+1 = *mi*, so *Gri,i*+1 *◦ mb* = *mb*, and

*i*+1

*i*

since *mb* = *m∗ ◦ ei*, we get *m∗ ◦ ei* = *Gri,i*+1 *◦ mb*

. Hence, the unique morphism

*i i*+1

*h* : *X → P* arises by the universal property of the pullback.

Pullbacks are stable under monomorphisms: since *m∗* is a mono, the arrow

*P → GRi*+1 is a mono as well. Hence, an (epi,mono)-factorization of *h* yields, by

composition, an (epi,mono)-factorization of *mb*

*i*+1

. The epic part of such a factor-

ization is the (*i* + 1)-minimization of (*X, c*), by Corollary [7.3](#_bookmark21). *2*

Below Lemma [7.5](#_bookmark22), we gave a concrete presentation of the pullback, for the case of deterministic automata. Here we fill in missing details. By assumption, *Ri, Ri*+1 are presented as subsets of 2*X* , i.e., *mi* : *Ri →* 2*X* and *mi*+1 : *Ri*+1 *→* 2*X* are inclusion maps. Since *mi*+1 *◦ ri,i*+1 = *mi*, we have *Ri ⊆ Ri*+1, witnessed by the inclusion map *ri,i*+1 : *Ri → Ri*+1. Hence *Gri,i*+1 = 2*rί,ί*+1 : 2*Rί*+1 *→* 2*Rί* is given by 2*rί,ί*+1 (*C*) =

*{S ∈ Ri | ri,i*+1(*S*) *∈ C}* = *C ∩ Ri*. Further, we have *mb*(*x*) = *{S ∈ Ri | x ∈ S}*, and *m∗* : *Ei →* 2*Rί* is given by *m∗*(*Q*)= *{S ∈ Ri | Q ⊆ S}*.

*i*

We start from a standard description of the pullback of *m∗* and 2*rί,ί*+1 in Set in the derivation below, as the set of pairs that are equated by *m∗* and 2*rί,ί*+1 (together with the projection maps to *Ei* and 2*Rί*+1 ).

*{*(*Q, C*) *∈ Ei ×* 2*Rί*+1 *| m∗*(*Q*)= 2*rί,ί*+1 (*C*)*}*

= *{*(*Q, C*) *∈ Ei ×* 2*Rί*+1 *| {S ∈ Ri | Q ⊆ S}* = *C ∩ Ri}*

= *{*(*Q, C*) *∈ Ei ×* 2*Rί*+1 *| ∀S ∈ Ri* : *S ∈ C* iff *Q ⊆ S}*

*∼*= *{*(*Q, C*) *∈ Ei ×* 2*Rί*+1 *| C ⊆ Ri*+1*\Ri} .*

The latter set is the characterization of the pullback *P* given in Section [7](#_bookmark18). The isomorphism, up-down is given by (*Q, C*) *'→* (*Q, {S ∈ C | S ∈ Ri*+1*\Ri}*), and down-up by (*Q, C*) *'→* (*Q, {S ∈ Ri | Q ⊆ S}∪ C*) = (*Q, m∗*(*Q*) *∪ C*). It is easy to check that these maps indeed form each others inverse. By the isomorphism, the maps *p*1 : *P → Ei* and *p*2 : *P →* 2*Rί*+1 of the pullback are given by *p*1(*Q, C*) = *Q* and *p*2(*Q, C*)= *m∗*(*Q*) *∪C*. The map *h* : *X → P* given in Section [7](#_bookmark18) trivially satisfies

*p*1 *◦ h* = *ei*. Further,

*p*2 *◦ h*(*x*)= *p*2(*ei*(*x*)*, {S ∈ Ri*+1*\Ri | x ∈ X}*)

= *m∗*(*ei*(*x*)) *∪ {S ∈ Ri*+1*\Ri | x ∈ X}*

= *mb*(*x*) *∪ {S ∈ Ri*+1*\Ri | x ∈ X}*

*i*

= *{S ∈ Ri | x ∈ X}∪ {S ∈ Ri*+1*\Ri | x ∈ X}*

*b i*+1

= *m*

(*x*)

which means that *h* is indeed the mediating map.

# Proofs of Section [8](#_bookmark23)

**Theorem 8.1** *Suppose that* C *and* D *have (epi,mono)-factorization systems. Let c* : *X → BTX be a coalgebra, and let m* : (*R, Q*) *→ Fα*(*X, c*) *be the reachable part of Fα*(*X, c*)*. Then the epic part of an (epi,mono)-factorization (in* C*) of the transpose mb* : *X → GR is the language quotient of* (*X, c*)*.*

**Proof.** Let (*A, α*) be the initial algebra (which exists by assumption) and let *s* : (*A, α*) *→ Fα*(*X, c*) be the unique algebra homomorphism. The reachable part *m* : (*R, Q*) *→ Fα*(*X, c*) is the monic morphism of an (epi,mono) factorization *m◦e* = *s* (in alg(*L*)). We get *sb* = *Ge◦mb*, and since *e* is epi and *G* is a right (contravariant) adjoint, *Ge* is mono. Thus, an (epi,mono)-factorization of *mb* yields an (epi,mono)- factorization of *Ge ◦ mb* = *sb*, by composition. Its epic part is, by definition, the language quotient of (*X, c*). *2*

**Theorem 8.3** *Let c* : *X → BTX be a coalgebra, and* ((*Fαc*)*i* : *Vi → FX*)*i∈*Ord *the cocone over the initial sequence of L induced by Fα*(*X, c*)*. For any i, we have c*¯*i* = (*Fαc*)*b. Further, let mi* : *Ri → FX be the i-reachable part of Fα*(*X, c*)*. Then the epic morphism of an (epi,mono)-factorization of the transpose mb* : *X → GRi*

*i*

*i*

*is the i-language quotient of* (*X, c*)*.*

**Proof.** The natural transformations *ρ* : *BG ⇒ GL* and *α* : *TG ⇒ G* compose to a natural transformation *ρ ◦ Bα* : *BTG ⇒ GL*. Now, the cone (*c*¯*i*)*i∈*Ord is the same as the cone (*cρ◦Bα*)*i∈*Ord of Lemma [D.1](#_bookmark52) (instantiating *B* and *ρ* in the lemma respectively to *BT* and *ρ ◦ Bα*; then the functor *F* in the lemma coincides with *Fα* from Section [8](#_bookmark23)). By the lemma, we obtain *c*¯*i* = (*Fαc*)*b* for all *i*.

*i*

*i*

For the second part of the statement, let *mi* : *Ri → FX* be the *i*-reachable part of *F* (*X, c*), and let *ei* be the epi such that *mi ◦ ei* = (*Fαc*)*i*. Then *c*¯*i* = (*Fαc*)*b* = *Gei ◦ mb*, and since *ei* is epi and *G* is a right (contravariant) adjoint, *Gei* is mono. Thus, an (epi,mono)-factorization of *mb* yields an (epi,mono)-factorization of *c*¯*i*, by composition. Its epic part is, by definition, the *i*-language quotient of (*X, c*). *2*

*i*

*i*

*i*