A Coalgebraic View of ε-Transitions

Alexandra Silva[⋆] and Bram Westerbaan[1]

ICIS, Radboud University Nijmegen

Abstract. In automata theory, a machine transitions from one state to the next when it reads an input symbol. It is common to also allow an automaton to transition without input, via an ε-transition. These ε-transitions are convenient, e.g., when one defines the composition of automata. However, they are not necessary, and can be eliminated. Such ε-elimination procedures have been studied separately for different types of automata, including non-deterministic and weighted automata.
It has been noted by Hasuo that it is possible to give a coalgebraic account of ε-elimination for some automata using trace semantics (as defined by Hasuo, Jacobs and Sokolova).
In this paper, we give a detailed description of the ε-elimination procedure via trace semantics (missing in the literature). We apply this framework to several types of automata, and explore its boundary.
In particular, we show that is possible (by careful choice of a monad) to define an ε-removal procedure for all weighted automata *over the positive reals* (and certain other semirings). Our definition extends the recent proposals by Sakarovitch and Lombardy for these semirings.

1 Introduction

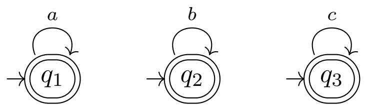
Automata are among the most basic structures in Computer Science. They have applications in a wide range of areas, including parsing, speech processing, and image recognition/generation software. Despite their simplicity, much research is still devoted to the semantics of automata and of related constructions.

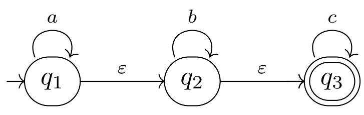
Coalgebra is a mathematical framework to study dynamical systems, of which automata are prime examples. Deterministic automata were the first automata to be studied as coalgebras in the seminal paper by Rutten [15]. Subsequently, various other types of automata and constructions were studied coalgebraically. This view has unified and generalized existing results and algorithms for different types of automata [17,1,2,18,3].

In this paper, we give a coalgebraic account of another concrete construction for automata: the elimination of ε-transitions. For this we use the abstract machinery of trace semantics. The advantage of this combination is two-fold. On the one hand, the concrete examples that the various types of automata provide clarify and ground the abstract notion of trace. On the other hand, trace semantics provides us with a uniform and intuitive definition for ε-elimination for many types of automata.

[⋆] Also affiliated with Centrum Wiskunde & Informatica (Amsterdam, The Netherlands) and HASLab / INESC TEC, Universidade do Minho (Braga, Portugal).

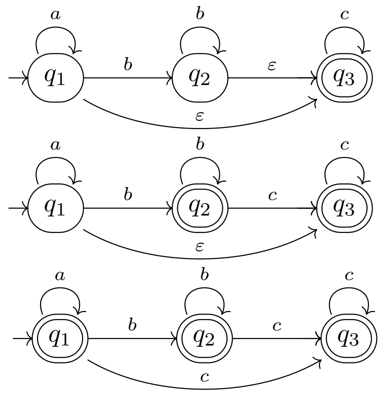
ε-Transitions are often useful at an intermediate stage. To illustrate this, let us show how to construct a non-deterministic automaton (without ε-transitions) that recognizes the language $a^*b^*c^*$. Note that it is easy to find automata recognizing the languages a^*, b^* and c^* (above, respectively).





If we compose these automata using ε-transitions, we obtain an automaton, on the left, that recognizes $a^*b^*c^*$. To obtain an automaton *without* ε-transitions that recognizes $a^*b^*c^*$ we incrementally eliminate the ε-transitions, as displayed below.

Hasuo and others [7,9] noted that result of the iterative process seen above can be captured using trace semantics in a Kleisli category, approach which we will discuss in more detail in Section 2.



In this paper, we take inspiration from [7,9] and we give an elaborate treatment of ε-elimination procedures using trace semantics. We extend their theory to include a class of weighted automata.

Though the process of ε-elimination for non-deterministic automata is classical and well-understood, for weighted automata things are less clear-cut, as witnessed by recent research [14,12]. The construction presented in this paper brings new results in comparison with the research presented in the aforementioned papers.

From a coalgebraic perspective, the challenge behind ε-elimination comes from the fact that many notions and definitions, such as bisimilarity for a functor, are given in a step-wise fashion. That is, the behavior of a certain system is fully determined by looking one step ahead at each time. This phenomenon, of having to deal with *multi-step* behavior, poses problems when having to model internal actions, such as ε-transitions, of a system. This is also present in concurrency theory, where internal actions (τ-steps) are discarded when defining weak bisimilarity. The theory presented in this paper might give a direction to improve the existing coalgebraic accounts of weak bisimilarity [20,4], which are not yet satisfactory.

The paper is organized as follows. In Section 2, we discuss the concrete construction for non-deterministic automata, we discuss how this paves the way to a coalgebraic account, and we introduce the idiosyncrasies behind the analogous construction for weighted automata. In Section 3, we present the general framework to formalize elimination of ε-transitions. In Section 4, we show how to model weighted automata in order to fit the framework. In Section 5, we discuss directions for future work.All proofs are omitted in the present article. A full version, containing all proofs and extra material can be found in [19].

2 Motivation

In this section we describe the existing ε-elimination procedures for weighted and non-deterministic automata more thoroughly. We also recall some of the basic notions concerning automata. We present the material in a manner that is suited to the purposes of this paper. For instance, we represent these automata as coalgebras and make no mention of initial states.

2.1 Non-Deterministic Automata

We represent a **non-deterministic automaton** (NDA) with states X over the alphabet A as a map $\alpha\colon X \longrightarrow \wp(A \times X + 1)$, where $\wp$ is the powerset functor. Given $q, r \in X$ and $a \in A$, we write, omitting the coproduct injections,

$$\begin{array}{lclr} q{\downarrow_\alpha} & \iff & * \in \alpha(q) & q \text{ is a final state} \\ q \xrightarrow{a}_\alpha r & \iff & (a, r) \in \alpha(q) & q \text{ has an } a\text{-transition to } r \end{array}$$

Let us recall the usual (language) semantics of α, i.e., which words a state $q \in X$ of α *accepts*. Let $w \equiv a_1 a_2 \cdots a_n$ be a word over A, and let $q \in X$. We say that q **accepts** w if there are $q_1, \ldots, q_n \in X$ such that

$$q \xrightarrow{a_1}_\alpha q_1 \xrightarrow{a_2}_\alpha \cdots \xrightarrow{a_n}_\alpha q_n \quad \text{and} \quad q_n{\downarrow_\alpha}. \tag{1}$$

So the semantics of α is captured by the map $[\![-]\!]_\alpha\colon X \longrightarrow \wp(A^*)$ given by

$$w \in [\![q]\!]_\alpha \qquad \iff \qquad q \text{ accepts } w,$$

where $q \in X$ and $w \in A^*$. So we will simply say that $[\![-]\!]_\alpha$ *is* the semantics of α.

ε-Transitions An NDA **with ε-transitions** (ε-NDA) with states X over an alphabet A is simply an NDA with states X over the alphabet $A + \{\varepsilon\}$,

$$\alpha\colon X \longrightarrow \wp((A + \{\varepsilon\}) \times X + 1),$$

but with a different semantics, which we define next.

Given a word $\tilde{w}$ over $A + \{\varepsilon\}$, let $\tilde{w}\backslash\varepsilon$ be the word on A one obtains by removing all the letters "ε" from $\tilde{w}$.

Let $w \in A^*$, and let $q \in X$. We say q **accepts** w (in the ε-NDA α) if there is $\tilde{w} \in (A + \{\varepsilon\})^*$ such that $w = \tilde{w}\backslash\varepsilon$ and q accepts $\tilde{w}$ in α seen as an NDA, as in (1).

Hence the semantics of α is the map $[\![-]\!]^\varepsilon_\alpha\colon X \longrightarrow \wp(A^*)$ given by, for $q \in X$,

$$[\![q]\!]^\varepsilon_\alpha \;=\; \big\{\, \tilde{w}\backslash\varepsilon\colon\; \tilde{w} \in [\![q]\!]_\alpha \,\big\}.$$

Or, more abstractly, $[\![-]\!]^\varepsilon_\alpha = \wp(-\backslash\varepsilon) \circ [\![-]\!]_\alpha$.

ε-Elimination for Non-Deterministic Automata Let α be an ε-NDA with states X and over an alphabet A. We construct an NDA $\alpha^{\#}\colon X \to \wp(A \times X + 1)$ which has the same semantics as α, in the sense that $[\![-]\!]_{\alpha^{\#}} = [\![-]\!]^{\varepsilon}_{\alpha}$. Since $\alpha^{\#}$ will have no ε-transitions we say "we have eliminated the ε-transitions".

The NDA $\alpha^{\#}$ is defined as follows. A state $q \in X$ has a transition in $\alpha^{\#}$ labelled by $a \in A$ to a state r if either this transition was already there in α or after a number of ε-transitions, starting from q, it is possible to make an a-transition to r. Formally:

$$q{\downarrow}_{\alpha^{\#}} \quad\iff\quad \left[\begin{array}{l} q \xrightarrow{\varepsilon}_{\alpha} q_1 \xrightarrow{\varepsilon}_{\alpha} \cdots \xrightarrow{\varepsilon}_{\alpha} q_n \quad \text{and} \quad q_n{\downarrow}_{\alpha} \\ \quad \text{for some } n \in \mathbb{N} \text{ and } q_1,\ldots,q_n \in X \end{array}\right.$$

$$q \xrightarrow{a}_{\alpha^{\#}} r \quad\iff\quad \left[\begin{array}{l} q \xrightarrow{\varepsilon}_{\alpha} q_1 \xrightarrow{\varepsilon}_{\alpha} \cdots \xrightarrow{\varepsilon}_{\alpha} q_n \quad \text{and} \quad q_n \xrightarrow{a}_{\alpha} r \\ \quad \text{for some } n \in \mathbb{N} \text{ and } q_1,\ldots,q_n \in X \end{array}\right.$$

Let $w \in A^*$ and $q \in X$. We leave it to the reader to verify that q accepts w in the ε-NDA α if and only if q accepts w in the NDA $\alpha^{\#}$, i.e., $[\![q]\!]_{\alpha^{\#}} = [\![q]\!]^{\varepsilon}_{\alpha}$. Hence, the following diagram commutes.

$$\begin{tikzcd}
X \arrow[r, "{[\![-]\!]_{\alpha}}"] \arrow[dr, "{[\![-]\!]_{\alpha^{\#}}}"'] & \wp((A+\{\varepsilon\})^*) \arrow[d, "\wp(-\backslash\varepsilon)"] \\
 & \wp(A^*)
\end{tikzcd}$$

Note that $[\![-]\!]_{\alpha}$ is the semantics of α considered as an NDA.

Coalgebraic Formulation We want to find an abstract definition of $\alpha^{\#}$ so that it can be instantiated for other types of automata. To this end it turns out to be fruitful to consider the following variant of $\alpha^{\#}$. Let

$$\mathrm{tr}_{\alpha}\colon X \longrightarrow \wp(\,\mathbb{N} \times (A \times X + 1)\,)$$

be the map given by: for all $q, r \in X$, and $a \in A$, and $n \in \mathbb{N}$:

$$(n,(a,r)) \in \mathrm{tr}_{\alpha}(q) \quad\iff\quad \left[\begin{array}{l} q \xrightarrow{\varepsilon}_{\alpha} q_1 \xrightarrow{\varepsilon}_{\alpha} \cdots \xrightarrow{\varepsilon}_{\alpha} q_n \quad \text{and} \quad q_n \xrightarrow{a}_{\alpha} r \\ \quad \text{for some } q_1,\ldots,q_n \in X \end{array}\right.$$

$$(n,*) \in \mathrm{tr}_{\alpha}(q) \quad\iff\quad \left[\begin{array}{l} q \xrightarrow{\varepsilon}_{\alpha} q_1 \xrightarrow{\varepsilon}_{\alpha} \cdots \xrightarrow{\varepsilon}_{\alpha} q_n \quad \text{and} \quad q_n{\downarrow}_{\alpha} \\ \quad \text{for some } q_1,\ldots,q_n \in X \end{array}\right.$$

The map tr_{α} contains more information than $\alpha^{\#}$. For example, $\alpha^{\#}$ tells us if a state $q \in X$ is final by whether $* \in \alpha^{\#}(q)$. The map tr_{α} tells us more, namely whether a final state can be reached from the state q using exactly n ε-transitions by whether $(n,*) \in \mathrm{tr}_{\alpha}(q)$.

Note that we can recover $\alpha^{\#}$ from the map tr_{α}; we have

$$b \in \alpha^{\#}(q) \quad\iff\quad \exists n \in \mathbb{N} \quad (n,b) \in \mathrm{tr}_{\alpha}(q), \tag{2}$$

for all $q \in X$ and $b \in B$, where $B := A \times X + 1$. More categorically, we can formulate Statement (2) as:

$$\begin{tikzcd} X \arrow[r, "\mathrm{tr}_\alpha"] \arrow[rr, bend right, "\alpha^\#"'] & \wp(\,\mathbb{N}\cdot B\,) \arrow[r, "\wp(\nabla)"] & \wp(B) \end{tikzcd} \qquad \text{commutes.}$$

Here, $\mathbb{N}\cdot B$ is the countable coproduct and $\nabla\colon \mathbb{N}\cdot B \to B$ is the *codiagonal* given by $\nabla(n,b) = b$ for all $(n,b) \in \mathbb{N}\cdot B$.

We are interested in tr_α because it satisfies a recursive relation, namely

$$\begin{aligned} (0,b) \,\in\, \mathrm{tr}_\alpha(q) \quad &\iff \quad b \,\in\, \alpha(q) \\ (n+1,b) \,\in\, \mathrm{tr}_\alpha(q) \quad &\iff \quad \exists r \in X\big[\ (\varepsilon,r) \,\in\, \alpha(q) \ \wedge\ (n,b) \,\in\, \mathrm{tr}_\alpha(r)\ \big], \end{aligned} \tag{3}$$

where $q \in X$ and $n \in \mathbb{N}$ and $b \in B$.

The recursive relation (3) can be cast in an abstract form, and this allows us to define tr_α (and hence $\alpha^\#$) for different types of automata at once.

For this we will use the Kleisli category $\mathcal{K}\ell(\wp)$ of the monad $\wp$. Recall that a map $f\colon V \to \wp(W)$ is a morphism from V to W in $\mathcal{K}\ell(\wp)$, which we will write as $f\colon\ V \longrightarrow\!\!\!\!\!\circ\ \ W$.

Indeed, we will see that the map $\mathrm{tr}_\alpha\colon X \to \wp(\mathbb{N}\cdot B)$ is the unique morphism such that the following diagram commutes, in $\mathcal{K}\ell(\wp)$.

$$\begin{tikzcd} X \arrow[r, "\mathrm{tr}_\alpha"] \arrow[d, "\alpha'"'] & \mathbb{N}\cdot B \arrow[d, "\xi"] \\ X + B \arrow[r, "\mathrm{tr}_\alpha + 1"'] & \mathbb{N}\cdot B + B \end{tikzcd} ,$$

where $\alpha'\colon X \longrightarrow \wp(X+B)$ is the composition of the following maps

$$X \xrightarrow{\ \alpha\ } \wp(\,(A+\{\varepsilon\})\times X + 1\,) \xrightarrow{\ \cong\ } \wp(\,X + (A\times X+1)\,)\ , \tag{4}$$

and $\xi\colon \mathbb{N}\cdot B \longrightarrow \wp(\mathbb{N}\cdot B + B)$ is given by $\xi(0,b) = \{b\}$, and $\xi(n+1,b) = \{(n,b)\}$, for all $b \in B$ and $n \in \mathbb{N}$.

We can formulate this more coalgebraically, as follows. Let F be the functor on $\mathcal{K}\ell(\wp)$ given by $F = - + B$. Then we can regard α' as an F-coalgebra,

$$\begin{tikzcd} X \arrow[r, "\alpha'"] & FX \end{tikzcd} = X + B$$

The final F-coalgebra is ξ, and tr_α is the unique homomorphism from α' to ξ.

Such a unique homomorphism tr_α into the final coalgebra in a Kleisli category is called a *trace map* by Hasuo, Jacobs, and Sokolova [8].

We will use the observations above to study ε-elimination in a more general setting later on. But let us first consider the class of weighted automata.

2.2 Weighted Automata

Let S be a semiring (such as $\mathbb{R}$). A weighted automaton is similar to a non-deterministic automaton, but each transition and state carries a weight, $s \in S$. Depending on the semiring, one may think of the weight of the transition between two states q and r as the distance of the transition from q to r, or as the probability that α transitions from q to r. For more information on weighted automata, see [5].

We represent a **weighted automaton** over the semiring S with states X over an alphabet A by a map $\alpha\colon X \longrightarrow \mathscr{M}(A \times X + 1)$, where $\mathscr{M}$ is the *multiset monad* over S. Recall that

$$\mathscr{M}(X) \;=\; \big\{\varphi \mid \varphi\colon X \to S,\ \operatorname{supp}\varphi \text{ is finite }\big\}.$$

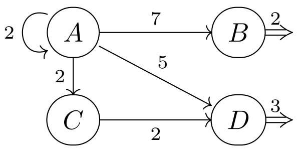
Given $q, r \in X$ and $a \in A$ and $s \in S$, we write

$$\begin{array}{lcl r}
q{\downarrow}^{s}_{\alpha} & \iff & s \;=\; \alpha(q)(*) & q \text{ outputs weight } s \\
q \xrightarrow{a|s}_{\alpha} r & \iff & s \;=\; \alpha(q)(a, r) & \begin{array}{r} q \text{ has an } a\text{-transition to } r \\ \text{with weight } s \end{array}
\end{array}$$

The subscript α will be omitted whenever it is clear from the context.

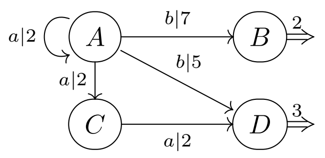
Let $a \in A$ be given. Note that formally there is an a-transition between any pair of states with *some* weight. We will typically only depict transitions with non-zero weight.

Semantics We explain the semantics of weighted automata by an example. Consider the following variation on a directed graph that represents a maze.



Suppose we stand at vertex A, and want to find the shortest path to exit the maze (via one of the exits, $\Rightarrow$). It is $A \to C \to D \Rightarrow$ with length 7.

Let us increase the complexity of the maze by adding some labels.



(5)

Again, we stand at A and want to find the shortest path to one of the exits, but this time we are only allowed to move along an ab-labelled path. That is, to exit the maze, we are only allowed to first move along an edge labelled by a, and then along an edge labelled by b, and then along $\Rightarrow$. Now the shortest path is $A \to A \to D \Rightarrow$ with length 10.

The maze in (5) can be represented by a weighted automaton α with states $X := \{A, B, C, D\}$ and alphabet $A := \{a, b\}$ over a semiring[1] on $\mathbb{R} \cup \{+\infty\}$ in a straightforward manner. When there is no c-labelled edge from one vertex to another we use a c-transition of weight $+\infty$, e.g., $\alpha(B)(a, D) = +\infty$. We interpret the symbol " $\overset{s}{\Rightarrow}$ " at a vertex q to mean that q outputs s.

Note that we can express the length of the shortest ab-labelled path from A to an exit using α as follows.

$$\min_{q_1 \in X} \min_{q_2 \in X} \Big[\ \alpha(A)(\kappa_\ell(a, q_1)) \ +\ \alpha(q_1)(\kappa_\ell(b, q_2)) \ +\ \alpha(q_2)(\kappa_r(*))\ \Big]$$

Note that "+" and "min" form a semiring $\mathscr{T}_{\min}$ on $\mathbb{R} \cup \{+\infty\}$, called the *tropical semiring.* Confusingly, "+" is the *multiplication* of $\mathscr{T}_{\min}$ while "min" is the *addition.* Hence the *zero* of $\mathscr{T}_{\min}$ is $+\infty$ and the *one* is 0.

Observe that if we change the operations "+" and "min" (that is, if we change the semiring on $\mathbb{R} \cup \{+\infty\}$) we get different semantics $[\![-]\!]_\alpha$. For instance, if we take "+" and "max" instead, $[\![q]\!]_\alpha(w)$ will be the the length of the *longest* w-labelled path from q to an exit.

We now give the general definition of semantics for weighted automata. Let S be a semiring. Let $\alpha\colon X \longrightarrow \mathscr{M}(A \times X + 1)$ be a weighted automaton over S. Then the **semantics** of α is the map $[\![-]\!]_\alpha\colon X \longrightarrow S^{A^*}$, given by, for $q_1 \in X$, and a word $w = a_1 \cdots a_n \in A^*$,

$$[\![q_1]\!]_\alpha(w) \ :=\ \sum_{q_2 \in X} \cdots \sum_{q_{n+1} \in X} \Big(\prod_{i=1}^{n} \alpha(q_i)(a_i, q_{i+1})\Big) \cdot \alpha(q_{n+1})(*). \qquad (6)$$

So a state in the weighted automaton α recognizes functions in S^{A^*}. These functions are usually referred to as *formal power series (over S).*

Non-deterministic automata are a special case of weighted automata. Indeed, the reader can verify that if we take S to be the Boolean semiring then weighted automata over S correspond exactly to NDAs.

ε-Transitions A **weighted automaton with ε-transitions** α over a semiring S with states X and alphabet A is simply a weighted automaton over S with states X and alphabet $A + \{\varepsilon\}$.

To explain the semantics of α, we first consider the tropical case $S = \mathscr{T}_{\min}$. Following the earlier discussion of the semantics of ordinary weighted automata over $\mathscr{T}_{\min}$ and shortest paths, it seems natural to define the semantics of α to be the map $[\![-]\!]^\varepsilon_\alpha\colon X \longrightarrow S^{A^*}$ given by, for $q \in X$ and $w \in A^*$,

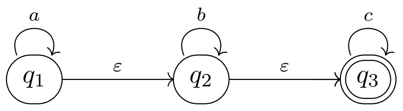
$$[\![q]\!]^\varepsilon_\alpha(w) \ =\ \min\ \big\{\ [\![q]\!]_\alpha(\tilde{w})\colon\ \tilde{w} \in (A + \{\varepsilon\})^* \text{ and } \tilde{w}\backslash\varepsilon = w\ \big\}. \qquad (7)$$

In the maze analogy, $[\![q]\!]^\varepsilon_\alpha(w)$ is the length of a shortest w-labelled path from q to an exit when ε-moves are not counted.

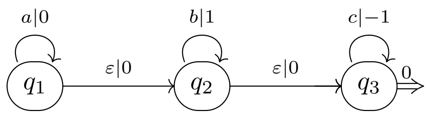
However, note that Equation (7) is not a sound definition for all α since the minimum might not exist. We will return to this problem shortly.

[1] The appropriate semiring structure on $\mathbb{R} \cup \{+\infty\}$ will become clear later on.

But first, we will further illustrate the semantics of ε-transitions. Recall that state q_1 in the following NDA accepts the language denoted by $a^*b^*c^*$:



Instead of talking about acceptance we now want to assign to each word in the language $a^*b^*c^*$ the difference between the number of b's and c's occurring in the word. In order to do that, we modify the above automaton into a weighted automaton over the tropical semiring $\mathscr{T}_{\min}$.



Note that for $w \in \{a,b,c\}^*$ the weight $[\![q_1]\!]^{\varepsilon}_{\alpha}(w)$ is precisely the number of b's occuring in w minus the number of c's occuring in w.

Inspired by Equation (7) we would like to define the semantics of a weighted automaton α with ε-transitions over any semiring S with states X and alphabet A to be the map $[\![-]\!]^{\varepsilon}_{\alpha} \colon X \longrightarrow S^{A^*}$ given by, for $q \in X$ and $w \in A^*$,

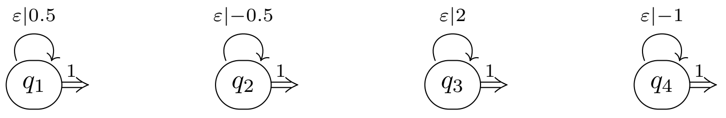
$$[\![q]\!]^{\varepsilon}_{\alpha}(w) \;=\; \sum \big\{\, [\![q]\!]_{\alpha}(\tilde{w}) \colon\; \tilde{w} \in (A + \{\varepsilon\})^* \text{ and } \tilde{w}\backslash\varepsilon = w \,\big\}. \tag{8}$$

However, without further information this is only a valid definition if the set

$$\big\{\, [\![q]\!]_{\alpha}(\tilde{w}) \colon \tilde{w} \in (A + \{\varepsilon\})^* \text{ and } \tilde{w}\backslash\varepsilon = w \,\big\}. \tag{9}$$

is finite. Otherwise, we do not know how we should interpret the symbol "$\sum$".

The problem is quite subtle. For example, consider the following weighted automata with ε-transitions over $\mathbb{R}$ (with the normal "+" and "$\cdot$").



Writing $\Box$ for the empty word, one sees using Equation (8), that

$$\begin{aligned}
[\![q_1]\!]^{\varepsilon}(\Box) \;&=\; 1 + 0.5 + (0.5)^2 + \cdots \;=\; 2,\\
[\![q_2]\!]^{\varepsilon}(\Box) \;&=\; 1 - 0.5 + (0.5)^2 - \cdots \;=\; {}^2\!/\!{}_3,
\end{aligned}$$

and in a daring mood we can compute,

$$[\![q_3]\!]^{\varepsilon}(\Box) \;=\; 1 + 2 + 4 + 8 + \cdots \;=\; +\infty,$$

but what should we make of the following?

$$[\![q_4]\!]^{\varepsilon}(\Box) \;=\; 1 - 1 + 1 - 1 + \cdots$$

To give proper meaning to weighted automata with ε-transitions it seems necessary to require that the semiring is equipped with a notion of summation for some sequences, and we must restrict ourselves to a class of weighted automata with ε-transitions for which the set in Expression (9) is summable.

Possibly due to this problem, the formal semantics of weighted automata with ε-transitions has not yet been settled in the literature.

In a recent proposal by Lombardy and Sakarovitch [12], semantics is given to a certain class of '*valid*' weighted automata with ε-transitions over *topological* semirings using a sophisticated ε-elimination *algorithm*. The automata with states q_1 and q_2 are valid, and the other two are not valid.

In this paper, the abstract view on automata gives rise to semantics to *all* weighted automata over certain semirings, namely *positive partial σ-semirings*. The semiring $[0, +\infty)$ is such a semiring, while $\mathbb{R}$ is not. So the general theory yields semantics for the automata with states q_1 and q_3, but not for the automata with states q_2 and q_4.

We will return to the example of weighted automata is Section 4.

3 Generalised ε-Elimination

Let us now turn to ε-elimination in a more general setting.

3.1 Automata in General

Setting 1 *Let* $\mathbf{C}$ *be a category, and assume that* $\mathbf{C}$ *has all finite limits and all countable colimits. Let* F *be a functor on* $\mathbf{C}$*, and let* T *be a monad on* $\mathbf{C}$*, with Kleisli category* $\mathcal{K}\ell(T)$*.*

In this setting, we abstractly define an automaton, parametrized by a functor F and a monad T, as follows.

Definition 2 *Let* X *be an object from* $\mathbf{C}$*. An* ***automaton*** *of type* T, F *with states* X *is a morphism* $\alpha\colon X \longrightarrow TFX$*.*

In other words, an automaton of type T, F is a morphism in $\mathcal{K}\ell(T)$ of the form

$$\alpha\colon X \multimap FX.$$

Examples 3 *Let* $\mathbf{C} = \mathbf{Sets}$*, and* $F = A \times - + 1$ *for some object* A *of* $\mathbf{C}$*.*

(i) Let $T = \wp$ *be the powerset monad. Then the automata of type* T, F *are non-deterministic automata with alphabet* A*.*
(ii) Let S *be a semiring. Let* $T := \mathscr{M}$ *be the multiset monad over* S*. Then the automata of type* T, F *are weighted automata with alphabet* A *over* S*.*

Example 4 *Let* $\mathbf{C} = \mathbf{Meas}$*. Let* $F = A \times - + 1$*. Let* $T = \mathscr{G}$ *be the sub-probability monad (see [11]). Then the automata of type* T, F *are sub-probabalistic automata.*

3.2 Semantics of Automata

Setting 5 *All conditions from Setting 1 and, in addition, assume that F is lifted to a functor $\overline{F}$ on $\mathcal{K}\ell(T)$, via a distributive law $\lambda\colon FT \longrightarrow TF$, and that $\mathcal{K}\ell(T)$ has a final $\overline{F}$-coalgebra, $\omega\colon \Omega \multimap F\Omega$.*

Note that the $\overline{F}$-coalgebras in $\mathcal{K}\ell(T)$ are precisely the automata (of type T, F). The final $\overline{F}$-coalgebra is what we will use in order to abstractly define the semantics for F, T automata:

Definition 6 *Let $\alpha\colon X \multimap \overline{F}X$ in $\mathcal{K}\ell(T)$ be given. We call unique homomorphism into the final coalgebra $[\![-]\!]_\alpha\colon X \multimap \Omega$ the **semantics** of α.*

3.3 Trace and Iterate in General

Before we turn to the study of ε-elimination for these general automata, we present some theory on the assignment $\alpha \mapsto \alpha^{\#}$. The material is a slight simplification of the work by Hasuo in [7].

Setting 7 *Let $\mathbf{K}$ be a category that has all countable coproducts. Moreover, assume that for each object B from $\mathbf{K}$, there is a final $-+B$-coalgebra,*

$$\xi_B\colon N_B \longrightarrow N_B + B.$$

This setting is equivalent to require that the functor $-+B$ is iteratable [13]. In the sequel we instantiate $\mathbf{K}$ to the Kleisli category of a given monad.

Recall that since ξ_B is final, there is a unique homomorphism from each $-+B$-coalgebra to ξ_B. We call this homomorphism **trace**.

Definition 8 *Let $\beta\colon X \to X + B$ be a morphism in $\mathbf{K}$. The **trace** of β is the unique morphism $\mathrm{tr}_\beta\colon X \to N_B$ such that the following diagram commutes.*

$$\begin{tikzcd}
X \arrow[r, "\mathrm{tr}_\beta"] \arrow[d, "\beta"'] & N_B \arrow[d, "\xi_B"] \\
X + B \arrow[r, "\mathrm{tr}_\beta + B"] & N_B + B
\end{tikzcd}$$

Setting 9 *Let $\mathbf{K}$ be a category that has all countable coproducts. Let B be an object from $\mathbf{K}$. Denote the initial $-+B$-algebra by $\iota_B\colon \mathbb{N}\cdot B + B \longrightarrow \mathbb{N}\cdot B$. Assume also that $\xi_B := \iota_B^{-1}$ is the final $-+B$-coalgebra. So we have*

$$\xi_B\colon \mathbb{N}\cdot B \longrightarrow \mathbb{N}\cdot B + B.$$

Before we define the **iterate** operator, we need two additional definitions.

Definition 10 *Let $g\colon A \to B$ be a morphism in $\mathbf{K}$. Let $\mathbb{N}\cdot g\colon \mathbb{N}\cdot A \longrightarrow \mathbb{N}\cdot B$ be given by, for all $n \in \mathbb{N}$, $(\mathbb{N}\cdot g) \circ \kappa_n = \kappa_n \circ g$. Equivalently, $\mathbb{N}\cdot g$ is the unique morphism such that*

$$\begin{tikzcd}
\mathbb{N}\cdot A \arrow[r, "\mathbb{N}\cdot g"] \arrow[d, "\xi_A"] & \mathbb{N}\cdot B \arrow[d, "\xi_B"] \\
\mathbb{N}\cdot A + A \arrow[r, "\mathbb{N}\cdot g + g"] & \mathbb{N}\cdot B + B
\end{tikzcd}$$

commutes.

Definition 11 *Let B be an object of $\mathbf{K}$. The* ***codiagonal*** *is the morphism $\nabla_B \colon \mathbb{N} \cdot B \longrightarrow B$ given by $\nabla_B \circ \kappa_n \;=\; \mathrm{id}_B$, where $n \in \mathbb{N}$, and $\kappa_n \colon B \longrightarrow \mathbb{N} \cdot B$ is the n-th coprojection.*

Definition 12 *Let X and A be objects from $\mathbf{K}$, and let $\alpha\colon X \longrightarrow X + A$ be a morphism. Then the* ***iterate*** *of α is the morphism $\alpha^{\#}\colon X \longrightarrow A$ given by $\alpha^{\#} := \nabla_A \circ \mathrm{tr}_\alpha$.*

Proposition 13 *Suppose we have a commuting diagram in $\mathbf{K}$ of the form*

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\alpha"] & Y \arrow[d, "\beta"] \\
X + A \arrow[r, "f+g"'] & Y + B
\end{tikzcd}$$

where $g\colon A \to B$. Then the following square commutes.

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\alpha^{\#}"] & Y \arrow[d, "\beta^{\#}"] \\
A \arrow[r, "g"] & B
\end{tikzcd}$$

3.4 ε-Elimination in General

First, we define what an abstract automaton with ε-transitions is. (Since our general automata do not explicitly contain an alphabet this is not immediately clear.) Recall that in the case of non-deterministic automata, an automaton with ε-transitions is a map $\alpha\colon X \longrightarrow \wp(\,(A + \{\varepsilon\}) \times X + 1\,)$, and this map gives rise to a second map,

$$\alpha'\colon X \longrightarrow \wp(\,X \;+\; (A \times X + 1)\,).$$

We base our definition on the second map, α', instead of α.

Definition 14 *Let X be an object from $\mathbf{C}$. An* ***ε-automaton*** *of type T, F with states X is a morphism $\alpha\colon X \longrightarrow T(\,X + FX\,)$. In other words, α is an automaton of type T, F_ε, where F_ε is the functor with*

$$F_\varepsilon X \;=\; X + FX.$$

To provide the semantics of ε-automata, we need some assumptions.

Setting 15 *In addition to the assumptions in Setting 5, we assume that $\mathcal{K}\ell(T)$ has a final $\overline{F}_\varepsilon$-coalgebra $\omega_\varepsilon\colon \Omega_\varepsilon \;\multimap\; \Omega_\varepsilon + F\Omega_\varepsilon$. Here, $\overline{F}_\varepsilon$ is the lifting of F_ε to $\mathcal{K}\ell(T)$, via the distributive law λ_ε given by $(\lambda_\varepsilon)_X \;=\; [T\kappa_\ell, T\kappa_r \circ \lambda_X]$, where X is an object from $\mathbf{C}$. Moreover, let B be an object from $\mathbf{C}$. We denote the initial $- + B$-algebra in $\mathcal{K}\ell(T)$ by $\iota_B\colon \mathbb{N} \cdot B + B \;\multimap\; \mathbb{N} \cdot B$. Assume that $\xi_B := \iota_B^{-1}$ is the* final *$- + B$-coalgebra in $\mathcal{K}\ell(T)$.*

We need a last definition, before providing semantics to ε-automata.

Definition 16 *Let $-\backslash\varepsilon$ be the unique morphism in* $\mathbf{C}$ *such that*

$$\begin{tikzcd}
\Omega_\varepsilon \arrow[r, "\cdot\backslash\varepsilon"] \arrow[d, "\omega_\varepsilon^\#"'] & \Omega \arrow[d, "\omega"] \\
F\Omega_\varepsilon \arrow[r, "F(\cdot\backslash\varepsilon)"'] & F\Omega
\end{tikzcd}$$

commutes. That is, $\cdot\backslash\varepsilon$ is the semantics of the automaton $\omega_\varepsilon^\#$, $\cdot\backslash\varepsilon \;=\; [\![-]\!]_{\omega_\varepsilon^\#}$.

Definition 17 *Let $\alpha\colon X \longrightarrow\!\!\circ\!\!\longrightarrow X + FX$ be an ε-automaton of type T, F. The **semantics** of α is the map $[\![-]\!]_\alpha^\varepsilon\colon X \longrightarrow\!\!\circ\!\!\longrightarrow \Omega$ such that*

$$\begin{tikzcd}
X \arrow[r, "{[\![-]\!]_\alpha^\varepsilon}"] \arrow[dr, "{[\![-]\!]_\alpha}"'] & \Omega \\
& \Omega_\varepsilon \arrow[u, "-\backslash\varepsilon"']
\end{tikzcd}$$

commutes, where $[\![-]\!]_\alpha$ is the semantics of α seen as automaton of type T, F_ε.

We can now present one of the main results of this paper, showing that (language) semantics is preserved by the abstract ε-elimination procedure.

Theorem 18 (ε-Elimination) *Let X be from* $\mathbf{C}$. *Let $\alpha\colon X \longrightarrow\!\!\circ\!\!\longrightarrow X + FX$ be an ε-automaton of type T, F. Then the iterate $\alpha^\#\colon X \longrightarrow\!\!\circ\!\!\longrightarrow FX$ is an automaton of type T, F with the same semantics as α. That is,*

$$[\![-]\!]_{\alpha^\#} \;=\; [\![-]\!]_\alpha^\varepsilon.$$

4 Weighted Automata and the $\mathscr{M}$ Monad

We now briefly return to the case of the weighted automata. Due to space constraints, we leave most details to the reader. Recall that a weighted automaton over a semiring S with states X and alphabet A is a map $\alpha\colon X \longrightarrow \mathscr{M}FX$, where $F = A \times - + 1$. So α is an automaton of type $\mathscr{M}, F$.

Unfortunately, the type $\mathscr{M}, F$ does not fit our general framework for automata (see Setting 15), since the inverse

$$\iota_B^{-1}\colon \mathbb{N}\cdot B \longrightarrow\!\!\circ\!\!\longrightarrow \mathbb{N}\cdot B + B$$

of the initial $- + B$-algebra ι_B in $\mathcal{K}\ell(\mathscr{M})$ is not the final $- + B$-coalgebra.

Indeed, this follows from the following example.

Example 19 *Let $B := \{b\}$ and let $\alpha\colon \{*\} \longrightarrow\!\!\circ\!\!\longrightarrow \{*\} + B$ be given by*

$$\alpha(*)(\kappa_\ell(*)) = \alpha(*)(\kappa_r(b)) = 1.$$

Suppose $\tau\colon \{*\} \longrightarrow\!\!\circ\!\!\longrightarrow \mathbb{N}\cdot B$ is a homomorphism from α to ι_B^{-1}. Then $\operatorname{supp}\tau(*)$ *is finite by definition of $\mathscr{M}$. However, the reader can verify that $\tau(*)(n, b) = 1$ for all $n \in \mathbb{N}$. So we see that* $\operatorname{supp}\tau(*)$ *must be infinite as well. No such τ exists. Hence, ι_B^{-1} is not the final $- + B$-coalgebra in $\mathcal{K}\ell(\mathscr{M})$.*

In order to study weighted automata in the general framework, we use

$$\mathscr{M}X \ := \ \{ \ \varphi\colon X \to S \mid \operatorname{supp}\varphi \text{ is at most countable } \}$$

instead of $\mathscr{M}X$. To turn $\mathscr{M}$ in a monad we need to assume that S is equipped with a notion of countable sums. For more details, see [19].

Note that an automaton of type $\mathscr{M}, F$ represents a weighted automaton that is allowed to have infinitely many (proper) transitions from a given state, while an automaton of type $\mathscr{M}, F$ is a weighted automaton with only finitely many transitions from a given state.

Fortunately, the automata of type $\mathscr{M}, F$ do fit nicely in our framework. That is, Setting 15 applies to them.

Proposition 1. *Given a set B, the inverse $\iota_B^{-1}\colon \mathbb{N}\cdot B \longrightarrow\!\!\circ\; \mathbb{N}\cdot B + B$ of the initial $-+B$-algebra ι_B in $\mathcal{K}\ell(\mathscr{M})$ is the final $-+B$-coalgebra. Similarly, the inverse $\xi\colon A^* \longrightarrow\!\!\circ\; A\times A^* + 1$ of the initial $\overline{F}$-algebra is the final $\overline{F}$-coalgebra in $\mathcal{K}\ell(\mathscr{M})$.*

Moreover, given a set A, and $\alpha\colon X \longrightarrow\!\!\circ\; X + A$ in $\mathcal{K}\ell(\mathscr{M})$, the iterate $\alpha^{\#}$ of α is given by, for all $q_0 \in X$ and $a \in A$,

$$\alpha^{\#}(q_0)(a) \ = \sum_{n\in\mathbb{N}}\sum_{q_1\in X}\cdots\sum_{q_{n+1}\in X} \Big(\prod_{i=1}^{n} \alpha(q_i)(q_{i+1})\Big) \cdot \alpha(a).$$

So we see that the abstract theory gives the expected results: the semantics $[\![-]\!]_\alpha$ of an automaton of type $\mathscr{M}, F$ turns out to be precisely the same as the semantics that we discussed before (see Equation (6)).

4.1 Valid Semirings

There is, however, a catch. The monad $\mathscr{M}$ is only defined over a *σ-semiring*, that is, a semiring S equipped with a summation operation that assigns to each family $(x_i)_{i\in I}$ of elements of S a sum $\sum_{i\in I} x_i$.

Usually, a semiring S is only equipped with a sum for some families of elements, which are then called *summable*. This idea is formalised in the notion *partial σ-semiring.* An example is the semiring of non-negative reals, $[0,\infty)$, equipped with a sum for all absolutely summable sequences. There are many examples of such partial σ-semirings. In fact, any semiring S is a partial σ-semiring in which only the *finite* families are summable.

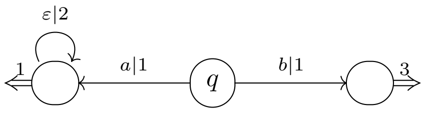
It is often possible to extend a partial σ-semiring S to a σ-semiring by adding one element $*$ to S and declaring that the sum of a family of element $(x_i)_{i\in I}$ of $S\cup\{*\}$ is the sum in S when $(x_i)_{i\in I}$ was summable in S and otherwise $*$.

Indeed, the above construction is possible if the partial σ-semiring has the following property: for all $a, b \in S$, $a + b = 0 \implies a = 0$ and $b = 0$. We call such semirings **positive** (using the terminology Gumm introduced for monoids [6]). In fact, any σ-semiring must be positive. So we see that only the positive partial

σ-semirings can be extended to a σ-semiring. A typical example of a semiring that is not positive is $\mathbb{R}$.

Let S be a positive partial σ-semiring. Then S can be extended to a σ-semiring $S \cup \{*\}$, and hence the abstract framework for automata is applicable to weighted automata over the semiring $S \cup \{*\}$.

The object $*$ acts as an "undefined" element. Consider the following weighted automaton α with ε-transitions over the semiring $\mathbb{R}$ with alphabet $\{a, b\}$.



Let us compute the semantics of α with Equation (8). We see that $[\![q]\!]^{\varepsilon}_{\alpha}(b) = 3$, but there is a difficulty when computing $[\![q]\!]^{\varepsilon}_{\alpha}(a) = 1 + 2 + 4 + \cdots$. However, if we consider α as a weighted automaton over the semiring $S \cup \{*\}$, then we simply get $[\![q]\!]^{\varepsilon}_{\alpha}(a) = *$, while still $[\![q]\!]^{\varepsilon}_{\alpha}(b) = 3$.

All in all, the abstract framework applies to, and hence given us semantics, ε-elimination, and so on, for *all* weighted automata over *positive* semirings (possibly equipped with a partial summation).

5 Discussion

We have presented a framework where ε-elimination can be thought of in an abstract manner. The framework yields procedures for non-deterministic automata and, notably, for weighted automata. What we presented here can be seen as the beginning of a larger quest to understand multi-step behavior, which is still a challenge coalgebraically. There are several directions we would like to explore further and which we discuss briefly next.

Kleisli versus Eilenberg–Moore Recovering coalgebraic definitions of language equivalence has been done in two different settings. The one we used in this paper, based on Kleisli categories, and the one presented in [18,17,10], based on Eilenberg-Moore categories and a generalized powerset construction. The definition of iterate is natural in Kleisli and hence we have taken the first approach. We want to explore if it is possible to define similar notions in the Eilenberg-Moore setting and enlarge the examples the framework covers. For instance, the generalized powerset construction works for every weighted automaton, without having to resort to changes in the monad.

Weak Bisimilarity ε-transitions are in some sense similar to τ-transitions in labelled transition systems (LTS). However, there are some subtleties to be tackled, before fully exploring the present framework to study weak bisimilarity. In particular, consider the example of the processes $\mathtt{a} + \mathtt{b}$ and $\tau.\mathtt{a} + \mathtt{b}$, which are not weakly bisimilar. Naively applying the framework above would erroneously identify them and extra care needs to be taken in order to avoid this. A more detailed account on the applications to weak bisimilarity can be found in [19].

Acknowledgments. We are grateful to Tomasz Brengos, Bart Jacobs, Herman Geuvers, Ichiro Hasuo, Jan Rutten, Pawel Sobocinski and the anonymous referees for useful comments and pointers to the literature. The work of Bram Westerbaan is funded by the NWO project *Practical conduction*. The work of Alexandra Silva is partially funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundao para a Cincia e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-020537 and SFRH/BPD/71956/2010.

References

1. F. Bonchi, M. Bonsangue, M. Boreale, J. Rutten, and A. Silva. A coalgebraic perspective on linear weighted automata. *Inf. Comput.*, 211:77–105, 2012.
2. F. Bonchi, M. Bonsangue, J. Rutten, and A. Silva. Brzozowski's algorithm (co)algebraically. In *Logic and Program Semantics*, vol. 7230 of *LNCS*, pp. 12–23. Springer, 2012.
3. F. Bonchi and D. Pous. Checking NFA equivalence with bisimulations up to congruence. In *POPL*, pp. 457–468. ACM, 2013.
4. T. Brengos. Weak bisimulations for coalgebras over ordered functors. In *IFIP TCS*, vol. 7604 of *LNCS*, pp. 87–103. Springer, 2012.
5. M. Droste, W. Kuich, and W. Vogler. *Handbook of Weighted Automata*. Springer-Verlag, 2009.
6. H. Peter Gumm, Tobias Schröder. Monoid-labeled transition systems. *ENTCS* 44(1):185–204, 2001.
7. I. Hasuo, B. Jacobs, and A. Sokolova. Generic Forward and Backward Simulations. (Partly in Japanese) Proceedings of *JSSST Annual Meeting*, 2006.
8. Ichiro Hasuo, Bart Jacobs, and Ana Sokolova. Generic Trace Semantics via Coinduction. *Logical Methods in Computer Science* 3(4) (2007).
9. B. Jacobs. From coalgebraic to monoidal traces. *ENTCS*, 264(2):125–140, 2010.
10. B. Jacobs, A. Silva, and A. Sokolova. Trace semantics via determinization. In *CMCS*, volume 7399 of *LNCS*, pp. 109–129. Springer, 2012.
11. H. Kerstan and B. König. Coalgebraic trace semantics for probabilistic transition systems based on measure theory. *CONCUR 2012*, pp. 410–424, 2012.
12. S. Lombardy and J. Sakarovitch. The Removal of Weighted ε-Transitions. In *CIAA*, vol. 7381 of *LNCS*, pp. 345–352, Springer, 2012.
13. Stefan Milius. On Iteratable Endofunctors. In *CTCS*, vol. 69 of *ENTCS*, pp. 287–304, Elsevier, 2002.
14. M. Mohri. Generic ε-removal algorithm for weighted automata. In *CIAA*, vol. 2088 of *LNCS*, pp. 230–242. Springer, 2000.
15. Jan Rutten. Automata and coinduction (an exercise in coalgebra). In *CONCUR*, vol. 1466 of *LNCS*, pp. 194–218. Springer, 1998.
16. Davide Sangiorgi. An introduction to bisimulation and coinduction. Cambridge University Press, 2012.
17. A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Generalizing the powerset construction, coalgebraically. In *FSTTCS*, volume 8 of *LIPIcs*, pp. 272–283, 2010.
18. A. Silva, F. Bonchi, M. Bonsangue, and J. Rutten. Generalizing determinization from automata to coalgebras. *LMCS*, 9(1), 2013.

19. A. Silva and B. Westerbaan. A Coalgebraic View of ε-Transitions. Extended abstract, with proofs. `http://alexandrasilva.org/files/epsilon-extended.pdf`.
20. A. Sokolova, E. de Vink, and H. Woracek. Coalgebraic weak bisimulation for action-type systems. *Sci. Ann. Comp. Sci.*, 19:93–144, 2009.