A Completeness Result for Finite λ-bisimulations*

Joost Winter

Faculty of Mathematics, Informatics, and Mechanics, University of Warsaw,
Warsaw, Poland

Abstract. We show that finite λ-bisimulations (closely related to bisimulations up to context) are sound and complete for finitely generated λ-bialgebras for distributive laws λ of a monad T on **Set** over an endofunctor F on **Set**, such that F preserves weak pullbacks and finitely generated T-algebras are closed under taking kernel pairs. This result is used to infer the decidability of weighted language equivalence when the underlying semiring is a subsemiring of an effectively presentable Noetherian semiring. These results are closely connected to [ÉM10] and [BMS13], concerned with respectively the decidability and axiomatization of weighted language equivalence w.r.t. Noetherian semirings.

1 Introduction

The notion of bisimulation, originating from the world of process algebra, plays an important role in the field of universal coalgebra: a survey of important results can be found, for example, in [Rut00]. Bisimulation up to techniques, generalizing ordinary bisimulations, have been first considered coalgebraically in [Len99]; later, extensions were given in, for example [PS11], [RBR13], [Pou13], and [RBB$^+$13]. The soundness of various notions of coalgebraic bisimulation up to has been extensively studied; in [BP13], moreover, a completeness result for finite bisimulations up to context (in the setting of NFAs) is presented, together with an efficient algorithm for deciding equivalence. As far as the author is aware, this is so far the only result of this type present in the literature.

Structures that have both an algebraic and coalgebraic structure can often be described as λ-*bialgebras* using *distributive laws*. Introductions to this framework can be found in e.g. [Bar04], [Jac06], and [Kli11]. This framework has been used to formulate the *generalized powerset construction*, considered in [SBBR10], [JSS12], and [SBBR13], providing a category-theoretical generalization of the classical powerset construction.

Weighted automata, introduced in [Sch61], have been extensively studied: surveys can be found in e.g. [Eil76] or [BR11]. An important notion here is that of a *simulation* between automata, which can be used to prove equivalence of weighted automata, studied in for example [BLS06] and [ÉM10]. In [ÉM10], it is shown that weighted language equivalence is decidable over semirings that

* Work supported by the Warsaw Center of Mathematics and Computer Science, and by the Polish National Science Centre (NCN) grant 2012/07/E/ST6/03026.

A. Pitts (Ed.): FOSSACS 2015, LNCS 9034, pp. 117–132, 2015.
DOI: 10.1007/978-3-662-46678-0_8

are Noetherian and effectively presentable, using the notions of simulation and proper semirings. In the Appendix of this paper, we show how these notions relate to the results in this paper, and how some of the results from [ÉM10] can be derived from the main result in this paper.

Co- and bialgebraic treatments to weighted automata, instantiating the framework of λ-bialgebras, are found in e.g. [BBB$^+$12], [BMS13], and [JSS12]. In [BMS13], an (abstract) sound and complete axiomatization is presented for monads and endofunctors satisfying the same conditions as those required for Proposition 7, and subsequently instantiated to a concrete axiomatization for weighted languages over Noetherian semirings. The methods used differ substantially from those used in this paper, but the obtained results are closely related.

After presenting the required preliminaries from the existing literature, in this paper we show that finite λ-bisimulations are, in certain cases, complete already (λ-bisimulations in general are complete whenever the behaviour functor preserves weak pullbacks). From this we derive the decidability of weighted language equivalence over subsemirings of semirings that are Noetherian and effectively presentable. Finally, in an appendix we discuss the relationship between some parts of the coalgebraic and classical, respectively, approaches to weighted automata.

Hence, one of the aims of this paper can be stated as bringing closer together, on a general level, the classical and coalgebraic approaches to weighted automata, and, in particular, relating the results from [BMS13] to those from [ÉM10].

2 Preliminaries

We will, in this section, present the preliminary material required for presenting the main result in the next section. We assume familiar the basic notions of category theory (which can be found in e.g. [Awo10] or [Mac71]), as well as the notions of monoids, semirings, and (left and right) semimodules over a semiring (which can be found in e.g. [BR11]). All of the material presented in this section can be found in existing literature.

Given a category $\mathbf{C}$ and a monad T on $\mathbf{C}$, $\mathbf{C}^T$ denotes the category of Eilenberg-Moore algebras for T. We moreover adopt the convention of using the term *S-module* to refer to *left S-semimodules.*

Some of the results in this paper require the axiom of choice (which can be formulated categorically by stating that (in **Set**) *every epi splits*, i.e. has a right inverse): these results are labelled with the marker (AC).

2.1 Algebras and Congruences

In this subsection, we present the notions of a *finitely generated algebra* and of a *kernel pair*, on a relatively concrete level, sufficient for obtaining the main results later in the paper.[1] Next, we give the definition of a *congruence*, and present a

[1] These are related to *finitely presentable algebras*, which are extensively studied in [AR94]: however, for the results in this paper, this notion is not needed.

result on the existence of coequalizers in $\mathbf{Set}^T$, required for our main result in the next section.

In the (concrete) case where T is a monad on $\mathbf{Set}$, a T-algebra (X, α_X) is called *finitely generated*[2] whenever there is a finite set Y together with a function $i : Y \to X$ such that the unique T-algebra morphism $i^* : (T(Y), \mu_Y) \to (X, \alpha_X)$ extending i is a regular epimorphism.[3] The condition of the epi i^* being regular directly implies that the mapping $U(i^*) : T(Y) \twoheadrightarrow X$ obtained by applying the forgetful functor is an epi in $\mathbf{Set}$, that is, a surjective function. We can moreover, without problems, assume that i itself is an injective function, i.e. a mono, and hence that Y can be regarded as a subset of X.

Given a morphism $f : X \to Y$ in a category with pullbacks, the *kernel pair* is the pullback of f with itself. Because the forgetful functor $U : \mathbf{Set}^T \to \mathbf{Set}$ creates all limits, the carrier of a kernel pair in $\mathbf{Set}^T$ can be described as the set

$$\{(x, y) \,|\, x, y \in UX \wedge Uf(x) = Uf(y)\}$$

and moreover, its algebra structure is compatible with the product algebra $X \times X$, i.e. the kernel pair is a subalgebra of $X \times X$.

Given an endofunctor T on $\mathbf{Set}$, and algebras (X, α_X) and (Y, α_Y) for this functor, a *congruence* between these algebras is a relation $R \subseteq X \times Y$ such that there is a unique T-algebra structure α_R on R making the following diagram commute:

$$\begin{tikzcd}
TX \arrow[d, "\alpha_X"'] & TR \arrow[l, "T\pi_1"'] \arrow[r, "T\pi_2"] \arrow[d, "\alpha_R"'] & TY \arrow[d, "\alpha_Y"] \\
X & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & FY
\end{tikzcd}$$

We furthermore will need the following result establishing the existence of coequalizers in $\mathbf{Set}^T$:

Proposition 1. *(AC) For any monad T on $\mathbf{Set}$, coequalizers exist in $\mathbf{Set}^T$ and are preserved by the forgetful functor.*

Proof. Established in the proof of [BW06, Proposition 9.3.4]. □

2.2 Universal Coalgebra

We will, in this section, consider some elementary and required results from the theory of universal coalgebra. For a more comprehensive reference to the theory, where the results below can also be found, we refer to [Rut00].

Given an endofunctor F on a category $\mathbf{C}$, a F-coalgebra consists of an object X in $\mathbf{C}$, together with a mapping $\delta : X \to TX$. Given two F-coalgebras (X, γ)

[2] This is known to correspond to the more general categorical definition of a finitely generated algebra; see e.g. the remark on `http://ncatlab.org/nlab/show/finitely+generated+object` under ‘Definition in concrete categories’.

[3] An morphism is a *regular epimorphism* iff it is the coequalizer of a parallel pair of morphisms, see e.g. [Bor94]

and (Y, δ), a morphism between these coalgebras consists of a morphism $f : X \to Y$ such that the following diagram commutes:

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\gamma"'] & Y \arrow[d, "\delta"] \\
FX \arrow[r, "Ff"] & FY
\end{tikzcd}$$

F-coalgebras and their morphisms form a category, and a terminal object in this category is called a *final coalgebra*. Given a F-coalgebra (X, δ_X), we let $[\![-]\!]_X$ denote the unique mapping into the final coalgebra whenever F has a final coalgebra.

Given two F-coalgebras (X, δ_X) and (Y, δ_Y), a *F-bisimulation* between X and Y is a relation $R \subseteq X \times Y$ such that there is some (not necessarily unique) F-coalgebra structure δ_R on R making the following diagram commute:

$$\begin{tikzcd}
X \arrow[d, "\delta_X"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\delta_R"'] & Y \arrow[d, "\delta_Y"] \\
FX & FR \arrow[l, "F\pi_1"'] \arrow[r, "F\pi_2"] & FY
\end{tikzcd}$$

In general, a largest bisimulation between two F-coalgebras always exists, and is denoted by $\sim_{X,Y}$. (We omit the subscripts when no confusion can arise.) Elements $x \in X$ and $y \in Y$ are called *bisimilar* whenever $x \sim_{X,Y} y$, and *behaviourally equivalent* whenever there is some F-coalgebra morphism f such that $f(x) = f(y)$. Whenever a final F-coalgebra exists, the latter condition is equivalent to $[\![x]\!]_X = [\![y]\!]_Y$.

In general, if two elements $x \in X$ and $y \in Y$ in F-coalgebras (X, δ_X) and (Y, δ_Y) are bisimilar, it follows that x and y are behaviourally equivalent (one may refer to this condition as the *soundness* of bisimulation). Under the condition that the functor F preserves *weak pullbacks* (a weak pullback is defined in the same way as a pullback, but without the uniqueness condition), the converse (which may be called the *completeness* of bisimulation) also holds.

2.3 λ-bialgebras

In this section, we will present, on an abstract level, the relevant material from the theory of λ-bialgebras, and the closely related *generalized powerset construction*. Comprehensive introductions to the material presented here can be found in e.g. [Bar04], [Jac06], [Kli11], [SBBR10], and [JSS12]. We will be concerned, in particular, with λ-bialgebras for a distributive law of a monad (T, μ, η) over an endofunctor F, without assuming any additional structure (e.g. that of a copointed functor or comonad) on the behaviour functor F.

Given a monad (T, μ, η) and an endofunctor F on any category $\mathbf{C}$, a *distributive law* of the monad T over F is a natural transformation

$$\lambda : TF \Rightarrow FT$$

such that the two diagrams of natural transformations

$$\begin{tikzcd}
F \arrow[r, Rightarrow, "\eta F"] \arrow[dr, Rightarrow, "F\eta"'] & TF \arrow[d, Rightarrow, "\lambda"] \\
& FT
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
TTF \arrow[rr, Rightarrow, "\mu F"] \arrow[d, Rightarrow, "T\lambda"'] & & TF \arrow[d, Rightarrow, "\lambda"] \\
TFT \arrow[r, Rightarrow, "\lambda T"] & FTT \arrow[r, Rightarrow, "F\mu"] & FT
\end{tikzcd}$$

commute.

Furthermore, given a distributive law $\lambda : TF \Rightarrow FT$, *$\lambda$-bialgebra* (X, α, γ) consists of a coalgebra (X, γ) for the functor F together with an algebra (X, α) for the monad T, such that the diagram

$$\begin{tikzcd}
TX \arrow[r, "\alpha"] \arrow[d, "T\gamma"'] & X \arrow[r, "\gamma"] & FX \\
TFX \arrow[rr, "\lambda_X"] & & FTX \arrow[u, "F\alpha"']
\end{tikzcd}$$

commutes. Morphisms of λ-algebras are mappings that simultaneously are F-coalgebra morphisms and T-algebra morphisms.

The two following, elementary, lemmata can be found in e.g. [Bar04]:

Lemma 2. *Given a distributive law λ of a monad (T, μ, η) over an endofunctor F and a FT-coalgebra (X, δ), $(TX, \mu_X, \hat{\delta})$ is a λ-bialgebra, with $\hat{\delta}$ given as:*

$$\hat{\delta} = F\mu_X \circ \lambda_{TX} \circ T\delta.$$

Proof. This is the first part of [Bar04, Lemma 4.3.3]. □

Lemma 3. *Given a distributive law λ of a monad (T, μ, η) over an endofunctor F, a λ-bialgebra (Q, α, γ) and an FT-coalgebra (X, δ), if $f : X \to Q$ makes the diagram*

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\delta"'] & Q \arrow[d, "\gamma"] \\
FTX \arrow[r, "Ff^*"] & FQ
\end{tikzcd}$$

commute (where $f^* : TX \to Q$ is obtained by applying the forgetful functor to the unique T-algebra morphism from (TX, μ_X) to (Q, α) extending f), then f^* is a morphism of λ-bialgebras between $(TX, \mu_X, \hat{\delta})$ and (Q, α, γ).

Proof. See e.g. [Bar04, Lemma 4.3.4]. □

Given two λ-bialgebras (X, α_X, δ_X) and (Y, α_Y, δ_Y), a *λ-bisimulation* between X and Y is a relation $R \subseteq X \times Y$ such that there is some (not necessarily unique) FT-coalgebra structure on R making the following diagram commute:

$$\begin{tikzcd}
X \arrow[d, "\delta_X"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\delta_R"'] & Y \arrow[d, "\delta_Y"] \\
FX & FTR \arrow[l, "F\pi_1{}^*"'] \arrow[r, "F\pi_2{}^*"] & FY
\end{tikzcd}$$

The following proposition establishes the soundness of λ-bisimulations:

Proposition 4. *Given two λ-bialgebras (X, α_X, δ_X) and (Y, α_Y, δ_Y), every λ-bisimulation $R \subseteq X \times Y$ is contained in a bisimulation S (for the functor F).*

Proof. See [Bar04, Corollary 4.3.5]. □

Moreover, the greatest bisimulation on two λ-bialgebras (X, α_X, δ_X) and (Y, α_Y, δ_Y) is a congruence:

Proposition 5. *Given two λ-bialgebras (X, α_X, δ_X) and (Y, α_Y, δ_Y), the relation $\sim_{X,Y}$ is a congruence, and its algebra structure is an Eilenberg-Moore algebra.*

Proof. See [Bar04, Corollary 3.4.22] and [Bar04, Corollary 3.4.23]. □

Any final F-coalgebra can be uniquely extended to a final λ-bialgebra:

Proposition 6. *Given a distributive law $\lambda : TF \Rightarrow FT$ for a functor F that has a final coalgebra (Ω, δ_Ω), there is a unique λ-bialgebra compatible with this final coalgebra, which is a final λ-bialgebra.*

Proof. See [Bar04, Corollary 3.4.19] and the following remark. □

In this case, we can combine the extension from Lemma 2 with the unique mapping into the final F-coalgebra, obtaining the diagram

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[d, "\delta"'] & TX \arrow[r, "\llbracket - \rrbracket"] \arrow[dl, "\ulcorner\delta"'] & \Omega \arrow[d, "\omega"] \\
FTX \arrow[rr, "F\llbracket - \rrbracket"] & & F\Omega
\end{tikzcd}$$

This construction, called the *generalized powerset construction*, is extensively studied in [SBBR10], [SBBR13], and [JSS12].

We finish this section by noting that there is a close relationship between the notion of a λ-bisimulation, and that of a *bisimulation up to context* (see e.g. [RBB$^+$13] for a comprehensive treatment of this notion). Given two λ-bialgebras (X, α_X, δ_X) and (Y, α_Y, δ_Y), a relation $R \subseteq X \times Y$ is called a bisimulation up to context whenever there is some δ_R making the diagram

$$\begin{tikzcd}
X \arrow[d, "\delta_X"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\delta_R"'] & Y \arrow[d, "\delta_Y"] \\
FX & Fc(R) \arrow[l, "F\pi_1"'] \arrow[r, "F\pi_2"] & FY
\end{tikzcd}$$

commute, where $c(R) = \langle \alpha_X \circ T\pi_1, \alpha_Y \circ T\pi_2 \rangle(TR) \subseteq X \times Y$. We note that there is a surjection $e : TR \twoheadrightarrow c(R)$, and thus, if δ_R is a witness to R being a λ-bisimulation, it directly follows that $Fe \circ \delta_R$ is a bisimulation up to context. If we assume the axiom of choice, the converse also holds: it now follows that there is some $f : c(R) \to TR$ such that $e \circ f = 1_{c(R)}$, and if δ_R witnesses that R is a bisimulation up to context, then $Ff \circ \delta_R$ witnesses that R is a λ-bisimulation.

2.4 Weighted Automata, Bialgebraically

We now briefly present the bialgebraic approach to weighted automata (over arbitrary semirings S). More comprehensive treatments of this bialgebraic approach can be found in e.g. [BMS13], [BBB+12], and [JSS12].

Here the monad T is instantiated as $\mathrm{Lin}_S(-)$, where $\mathrm{Lin}_S(X)$ is the set

$$\{f : X \to S \,|\, f \text{ has finite support}\}$$

regarded as representing finite (left) S-linear combinations of elements of X, for any semiring S, and the monadic structure can be specified by

$$\eta_X(x)(y) = \textbf{if } x = y \textbf{ then } 1 \textbf{ else } 0$$

and

$$\mu_X(f)(x) = \sum_{g \in \mathrm{supp}(f)} f(g) \cdot g(x).$$

The category of algebras for this monad is isomorphic to the category of S-modules and (left) S-linear mappings.

Furthermore, the behaviour functor is instantiated as $S \times -^A$. A coalgebra for this functor (or for the functor $S \times T(-)^A$ where T is some monad) is usually represented as a pair of mappings $(o, \delta) : X \to S \times X^A$, with $\delta(x)(a)$ usually represented as x_a (or, in the case of a single alphabet symbol, x'), and called the *a-derivative* of x, and with $o(x)$ referred to as the *output* of x. This notation allows us to conveniently represent these coalgebras as *systems of behavioural differential equations.*

There exists a final coalgebra for this functor, with its carrier given by the set

$$S\langle\langle A\rangle\rangle = (A^* \to S),$$

of formal power series in noncommuting variables, and the coalgebraic structure given by, for any $\sigma \in S\langle\langle A\rangle\rangle$, $o(\sigma) = \sigma(1)$ (with 1 denoting the empty word), and $\sigma_a(w) = \sigma(aw)$.

The distributive law

$$\lambda : \mathrm{Lin}_S(S \times -^A) \Rightarrow S \times \mathrm{Lin}_S(-)^A$$

can be given componentwise by

$$\lambda_X\left(\sum_{i=1}^{n} s_i(o_i, d_i)\right) = \left(\sum_{i=1}^{n} s_i o_i, a \mapsto \sum_{i=1}^{n} s_i d_i(a)\right).$$

We call a λ-bialgebra for this distributive law an *S-linear automaton.* The final bialgebra for this distributive law can be given by adding a pointwise S-module structure to $S\langle\langle A\rangle\rangle$. We regard coalgebras for the functor $S \times \mathrm{Lin}_S(-)^A$ as S-weighted automata, which can be extended into S-linear automata using Lemma 2. The formal power series accepted by a S-weighted automaton is then given by the unique mapping of this S-linear automaton into the final S-linear automaton.

The notion of λ-bisimulation here instantiates to the notion of *bisimulation up to linear combinations*. This condition can be concretely expressed as follows: given S-linear automata (X, o_X, δ_X) and (Y, o_Y, δ_Y), a relation $R \subseteq X \times Y$ is a bisimulation up to linear combinations whenever, for all $(x, y) \in R$, $o_X(x) = o_Y(y)$, and for every alphabet symbol $a \in A$ there is a $n \in \mathbb{N}$, together with elements $x_0, \ldots, x_{n-1} \in X$, $y_0, \ldots, y_{n-1} \in Y$, and scalars $s_0, \ldots, s_{n-1} \in S$, such that for each $i \leq n$, $(x_i, y_i) \in R$, and furthermore, $x_a = \sum_{i=1}^{n} s_i x_i$ and $y_a = \sum_{i=1}^{n} s_i y_i$. The latter condition can conveniently be represented using the following notation:

$$x_a = \sum_{i=1}^{n} s_i x_i \;\; \Sigma R \;\; \sum_{i=1}^{n} s_i y_i = y_a$$

3 Main Result

We now are able to state the main result, which can be seen as a *completeness* result for finite λ-bisimulations for distributive laws satisfying the required conditions, similarly to how Proposition 4 can be seen as stating the *soundness* of λ-bisimulations in general.

We first note that, given a λ-bialgebra (X, α, δ), the bisimilarity relation $\sim$ has both a F-coalgebra structure (by the definition of bisimulations), as well as that of an algebra for the monad T (by Proposition 5). Moreover, as a result of Proposition 1, the set $X/\sim$ has the structure of an algebra for the monad T, such that the function $h : X \to X/\sim$ sending each $x \in X$ to its equivalence class w.r.t. $\sim$ is a T-algebra morphism.

Proposition 7. *(AC) Assume:*

1. *T is a monad on* **Set** *such that finitely generated T-algebras are closed under taking kernel pairs.*
2. *F is an endofunctor on* **Set** *that preserves weak pullbacks.*
3. *λ is a distributive law $TF \Rightarrow FT$.*
4. *(X, α_X, δ_X) is a finitely generated λ-bialgebra.*

Then, given two states $x, y \in X$, x and y are behaviourally equivalent if and only if there is a finite λ-bisimulation $R \subseteq X \times X$ with $(x, y) \in R$.

Proof. If such a λ-bisimulation R exists, it immediately follows that R is contained in some bisimulation, and hence, that x and y are behaviourally equivalent.

Conversely, assume that there are $x, y \in X$ that are behaviourally equivalent. Because F preserves weak pullbacks, it directly follows that $x \sim y$. We now start by taking the kernel pair of the morphism $h : X \to X/\sim$. This kernel pair can be given by the set

$$\sim = \{(x, y) \mid Uh(x) = Uh(y)\}$$

with an algebra structure $\alpha_\sim : T(\sim) \to \sim$ such that $(\sim, \alpha_\sim)$ is a subalgebra of the product algebra $(X, \alpha_X) \times (X, \alpha_X)$. Because (X, α_X) is finitely generated, it follows from the first assumption that $(\sim, \alpha_\sim)$ again is finitely generated.

Simultaneously, $\sim$ is the greatest bisimulation on X, i.e., there is some (not necessarily unique) $\delta_\sim$ making the diagram

$$\begin{tikzcd}
X \arrow[d, "\delta_X"'] & \sim \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "\delta_\sim"'] & X \arrow[d, "\delta_X"] \\
FX & F(\sim) \arrow[l, "F\pi_1"'] \arrow[r, "F\pi_2"] & FX
\end{tikzcd}$$

commute. (Note that, although $\sim$ is both the carrier of an algebra for the monad T and a F-coalgebra, we have not established that $\sim$ is a λ-bialgebra.)

Because $\sim$ is finitely generated, there is some finite $R \subseteq \sim$ (let i denote the inclusion of R into $\sim$) such that the extension $i^* : T(R) \to \sim$ is a regular epimorphism in $\mathbf{Set}^T$, and hence an epi in $\mathbf{Set}$.

Because the epi i^* splits by the axiom of choice, it has a right inverse j, and we can now construct δ_R as $Fj \circ \delta_\sim \circ i$ to make the diagram

$$\begin{tikzcd}
R \arrow[r, "i"] \arrow[d, "\delta_R"'] & \sim \arrow[d, "\delta_\sim"] \\
FTR \arrow[r, "F(i^*)"] & F(\sim)
\end{tikzcd}$$

commute. We can furthermore assume that $(x, y) \in R$, simply by adding this single element to the finite set of generators.

We can now conclude that the diagram

$$\begin{tikzcd}
X \arrow[d, "\delta_X"'] & R \arrow[l, "\pi_1 \circ i"'] \arrow[r, "\pi_2 \circ i"] \arrow[d, "\delta_R"'] & X \arrow[d, "\delta_X"] \\
FX & FTR \arrow[l, "F(\pi_1 \circ i^*)"'] \arrow[r, "F(\pi_2 \circ i^*)"] & FX
\end{tikzcd} \tag{1}$$

again commutes.

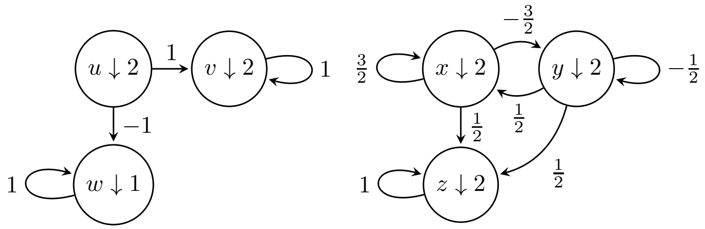
As i^* and π_1 are both T-algebra morphisms, it now also follows that $\pi_1 \circ i^*$ is an T-algebra morphism extending $\pi_1 \circ i$. Because (TR, μ_R) is a free algebra, it now follows that $\pi_1 \circ i^* = (\pi_1 \circ i)^*$ and $\pi_2 \circ i^* = (\pi_2 \circ i)^*$. Making these substitutions in Diagram (1), we can conclude that R is a λ-bisimulation. □

Remark. If a final coalgebra for the functor F exists, there exists a unique λ-bialgebra structure on this final coalgebra, and hence it is possible to replace the morphism h used in the proof with the unique morphism $[\![-]\!]$ into the final λ-bialgebra, now yielding $\sim$ as the kernel pair of the morphism $[\![-]\!]$. The reliance on the axiom of choice can then be relaxed to the condition that, for finite sets X and arbitrary Y, every epi from TX to Y splits. In particular, this condition is satisfied by the monad $\mathrm{Lin}_S(-)$, as $\mathrm{Lin}_S(X)$ is countable whenever X is finite. As a consequence, in the next section, the decidability result can be established without reliance on the axiom of choice.

4 An Example

In this section, we will present an example illustrating how to prove the equivalence of states in an automaton using finite λ-bisimulation (or concretely, bisimulation up to linear combinations). The example given is a direct adaptation of one of the examples given in [BMS13].

Consider the following $\mathbb{Q}$-weighted automaton (over a singleton alphabet)



which corresponds to the following system of behavioural differential equations:

$$\begin{array}{rlrl}
o(u) = 2 & u' = v - w & o(x) = 2 & x' = \frac{3}{2}x - \frac{3}{2}y + \frac{1}{2}z \\
o(v) = 2 & v' = v & o(y) = 2 & y' = \frac{1}{2}x - \frac{1}{2}y + \frac{1}{2}z \\
o(w) = 1 & w' = w & o(z) = 2 & z' = z
\end{array}$$

Next, consider the following relation:

$$R = \{(u, x), (v, z), (\tfrac{1}{2}v - w, \tfrac{3}{2}x - \tfrac{3}{2}y)\}$$

A proof that R is a bisimulation up to linear combinations is given by

$$\begin{array}{rcl}
u' = \frac{1}{2}v + (\frac{1}{2}v - w) & \Sigma R & \frac{1}{2}z + (\frac{3}{2}x - \frac{3}{2}y) = x' \\
v' = v & \Sigma R & z = z' \\
(\frac{1}{2}v - w)' = \frac{1}{2}v - w & \Sigma R & \frac{3}{2}x - \frac{3}{2}y = (\frac{3}{2}x - \frac{3}{2}y)'
\end{array}$$

or alternatively by assigning the following weighted automaton structure to R:

$$\begin{array}{rr}
o(u, x) = 2 & (u, x)' = \frac{1}{2}(v, z) + (\frac{1}{2} - w, \frac{3}{2}x - \frac{3}{2}y) \\
o(v, z) = 2 & (v, z)' = (v, z) \\
o(\frac{1}{2}v - w, \frac{3}{2}x - \frac{3}{2}y) = 0 & (\frac{1}{2}v - w, \frac{3}{2}x - \frac{3}{2}y)' = (\frac{1}{2}v - w, \frac{3}{2}x - \frac{3}{2}y)
\end{array}$$

5 Decidability of Weighted Language Equivalence

Following [ÉM10], we call a semiring *Noetherian* whenever any submodule of a finitely generated S-module is again finitely generated. Using the result from Section 3, we can now directly derive a decidability result for equivalence of (states in) weighted automata over Noetherian semirings. We start by noting that, if S is a Noetherian semiring, and X is a finitely generated S-module, Y

is an arbitrary S-module, and $f : X \to Y$ is a S-linear mapping, then the kernel pair of f is a sub-S-module of the finitely generated S-module $X \times X$, and hence again finitely generated. Hence, the monad $\mathrm{Lin}_S(-)$ satisfies the first condition of Proposition 7 whenever S is a Noetherian semiring.

We moreover call, following the definition in [ÉM10], a semiring *effectively presentable*, whenever its carrier can be represented as a recursive subset of $\mathbb{N}$ such that the operations $+$ and $\cdot$ are recursive functions. This condition by itself is enough to establish the semidecidability of non-equivalence.

The results in this section are closely related to the decidability results from [ÉM10]. In the proof of semidecidability of equivalence, the crucial difference is relying on Proposition 7 instead of on a concrete result establishing properness.

The semidecidability of non-behavioural equivalence holds in general for effectively presentable semirings:

Proposition 8. *Given any effectively presentable semiring S, non-behavioural equivalence of states in finitely generated S-linear automata is semidecidable.*

Proof. (See also [ÉM10, Lemma 5.1].) If states x, y in a finitely generated S-linear automaton (X, o, δ) are not equivalent, there is some word $w \in A^*$ such that $o(x_w) \neq o(y_w)$. We can enumerate all words $w \in A^*$ and, because S is effectively presentable, can check for each word whether $o(x_w) = o(y_w)$. If x and y are not equivalent, eventually some word w witnessing this will be found. □

Moreover, if S additionally is a subsemiring of a Noetherian semiring, we can also derive semidecidability of behavioural equivalence (and hence, in combination with the preceding result, decidability) using Proposition 7.

Proposition 9. *Given any semiring S that is a subsemiring of an effectively presentable Noetherian semiring S', behavioural equivalence of states in free finitely generated S-linear automata is semidecidable.*

Proof. We start by noting that we can see any free finitely generated S-linear automaton as a free finitely generated S'-linear automaton (X, o, δ). Because S' is effectively presentable, it is countable, and the set of tuples

$$(R \in \mathcal{P}_\omega(\mathrm{Lin}_S(X) \times \mathrm{Lin}_S(X)), \delta_R : R \to S \times \mathrm{Lin}_S(R))$$

again is countable, giving an enumeration of its elements.

For each element of this set, we can check whether $(x, y) \in R$ and whether (R, δ_R) makes Diagram (1) commute. If $[\![x]\!]_X = [\![y]\!]_X$, a suitable candidate will eventually be found as a result of Proposition 7, so the process will terminate. □

Corollary 10. *Given any semiring S that is a subsemiring of an effectively presentable Noetherian semiring S', behavioural equivalence of states in free finitely generated S-linear automata is decidable.*

6 Further Directions

The results in this paper give rise to several possible directions for future work. One possibility is looking for extensions of the main result to distributive laws of a monad over a functor with additional structure, e.g. that of a copointed functor or a comonad.

As a final observation, we note that it is also possible to use the main result to conclude that certain monads do *not* have the property that finitely generated algebras are closed under taking kernel pairs. A first example follows the approach in [ÉM10], where it is shown that the tropical semiring $\mathbb{T}$ is not Noetherian: likewise, we can show that finitely generated algebras for the monad $\mathrm{Lin}_{\mathbb{T}}(-)$ are not closed under taking kernel pairs, as this would imply decidability and it is known that equivalence of $\mathbb{T}$-weighted automata is not decidable.

A second example of such a negative result can be given by the monad $\mathcal{P}_\omega((- + A)^*)$: because the context-free languages can be characterized using a distributive law of this monad over the functor $2 \times -^A$ ([BHKR13], [WBR13]), and because equivalence of context-free languages is not decidable, it follows that algebras for the monad $\mathcal{P}_\omega((- + A)^*)$ are not closed under taking kernel pairs. (This result can be contrasted to the results in [Cau90] and [CHS95], which establish the decidability of bisimilarity for context-free processes. However, note that bisimulation over determinized systems is equal to language equivalence, which corresponds to the process-algebraic notion of trace equivalence.) A more detailed study of this type of results is left as future work.

Acknowledgements. For helpful suggestions and comments, the author would like to thank Bartek Klin (in particular for the observation that the existence of a final coalgebra is not required for the main result) and Jurriaan Rot (in particular for sharing his observations about the relationship between λ-bisimulation and bisimulation up to context, which have been incorporated into the paper). The author would furthermore like to thank the anonymous reviewers for their constructive comments on the first version of the paper.

References

[AR94] Adámek, J., Rosicky, J.: Locally Presentable and Accessible Categories. Cambridge University Press (1994)

[Awo10] Awodey, S.: Category Theory. Oxford University Press (2010)

[Bar04] Bartels, F.: On Generalized Coinduction and Probabilistic Specification Formats. PhD thesis, Vrije Universiteit Amsterdam (2004)

[BBB$^+$12] Bonchi, F., Bonsangue, M.M., Boreale, M., Rutten, J., Silva, A.: A coalgebraic perspective on linear weighted automata. Information and Computation 211, 77–105 (2012)

[BHKR13] Bonsangue, M.M., Hansen, H.H., Kurz, A., Rot, J.: Presenting distributive laws. In: Heckel, R., Milius, S. (eds.) CALCO 2013. LNCS, vol. 8089, pp. 95–109. Springer, Heidelberg (2013)

[BLS06] Béal, M.-P., Lombardy, S., Sakarovitch, J.: Conjugacy and equivalence of weighted automata and functional transducers. In: Grigoriev, D., Harrison, J., Hirsch, E.A. (eds.) CSR 2006. LNCS, vol. 3967, pp. 58–69. Springer, Heidelberg (2006)

[BMS13] Bonsangue, M.M., Milius, S., Silva, A.: Sound and complete axiomatizations of coalgebraic language equivalence. ACM Transactions on Computational Logic 14(1), 7 (2013)

[Bor94] Borceux, F.: Handbook of Categorical Algebra. Basic category theory, vol. 1. Cambridge University Press (1994)

[BP13] Bonchi, F., Pous, D.: Checking NFA equivalence with bisimulations up to congruence. In: POPL, pp. 457–468. ACM (2013)

[BR11] Berstel, J., Reutenauer, C.: Noncommutative Rational Series with Applications. Cambridge University Press (2011)

[BW06] Barr, M., Wells, C.: Toposes, triples and theories. Reprints in Theory and Applications of Categories 12, 1–287 (2006)

[Cau90] Caucal, D.: Graphes canoniques de graphes algébriques. ITA 24, 339–352 (1990)

[CHS95] Christensen, S., Hüttel, H., Stirling, C.: Bisimulation equivalence is decidable for all context-free processes. Information and Computation 121(2), 143–148 (1995)

[Eil76] Eilenberg, S.: Automata, Languages, and Machines. Academic Press, Inc. (1976)

[ÉM10] Ésik, Z., Maletti, A.: Simulation vs. equivalence. In: Arabnia, H.R., Gravvanis, G.A., Solo, A.M.G. (eds.) FCS, pp. 119–124. CSREA Press (2010)

[HM13] Heckel, R., Milius, S. (eds.): CALCO 2013. LNCS, vol. 8089. Springer, Heidelberg (2013)

[Jac06] Jacobs, B.: A bialgebraic review of deterministic automata, regular expressions and languages. In: Futatsugi, K., Jouannaud, J.-P., Meseguer, J. (eds.)Goguen Festschr 2006. LNCS, vol. 4060, pp. 375–404. Springer, Heidelberg (2006)

[JSS12] Jacobs, B., Silva, A., Sokolova, A.: Trace semantics via determinization. In: Pattinson, D., Schröder, L. (eds.) CMCS 2012. LNCS, vol. 7399, pp. 109–129. Springer, Heidelberg (2012)

[Kli11] Klin, B.: Bialgebras for structural operational semantics: An introduction. Theoretical Computer Science 412(38), 5043–5069 (2011)

[Len99] Lenisa, M.: From set-theoretic coinduction to coalgebraic coinduction: some results, some problems. Electronic Notes in Theoretical Computer Science 19, 2–22 (1999)

[Mac71] MacLane, S.: Categories for the Working Mathematician. Graduate Texts in Mathematics, vol. 5. Springer, New York (1971)

[Pou13] Pous, D.: Coalgebraic up-to techniques. In: Heckel, R., Milius, S. (eds.) CALCO 2013. LNCS, vol. 8089, pp. 34–35. Springer, Heidelberg (2013)

[PS11] Pous, D., Sangiorgi, D.: Enhancements of the coinductive proof method. In: Advanced Topics in Bisimulation and Coinduction. Cambridge University Press (2011)

[RBB$^+$13] Rot, J., Bonchi, F., Bonsangue, M.M., Pous, D., Rutten, J., Silva, A.: Enhanced coalgebraic bisimulation. Submitted to Mathematical Structures in Computer Science (2013)

[RBR13] Rot, J., Bonsangue, M., Rutten, J.: Coalgebraic bisimulation-up-to. In: van Emde Boas, P., Groen, F.C.A., Italiano, G.F., Nawrocki, J., Sack, H.

(eds.) SOFSEM 2013. LNCS, vol. 7741, pp. 369–381. Springer, Heidelberg (2013)

[Rut00] Rutten, J.: Universal coalgebra: A theory of systems. Theoretical Computer Science 249(1), 3–80 (2000)

[SBBR10] Silva, A., Bonchi, F., Bonsangue, M.M., Rutten, J.: Generalizing the powerset construction, coalgebraically. In: Lodaya, K., Mahajan, M. (eds.) FSTTCS. LIPIcs, vol. 8, pp. 272–283. Schloss Dagstuhl—Leibniz-Zentrum für Informatik (2010)

[SBBR13] Silva, A., Bonchi, F., Bonsangue, M., Rutten, J.: Generalizing determinization from automata to coalgebras. Logical Methods in Computer Science 9(1) (2013)

[Sch61] Schützenberger, M.-P.: On the definition of a family of automata. Information and Control 4(2-3), 245–270 (1961)

[WBR13] Winter, J., Bonsangue, M.M., Rutten, J.J.M.M.: Coalgebraic characterizations of context-free languages. Logical Methods in Computer Science 9(3:14) (2013)

A Simulations and Bialgebra Homomorphisms

This appendix is meant to elucidate the relation between the notion of a *simulation*[4], which has been an important tool in the classical theory of weighted automata for proving the equivalence between automata, and the bialgebraic notion of a *homomorphism* between S-linear automata, which plays a similar role in the co- and bialgebraic approach.

To make the correspondence between the two approaches somewhat more straightforward, we give a presentation of classical weighted automata that is symmetric to the traditional one: i.e. in terms of left-linear mappings and matrix multiplication on the left, rather than in terms of right-linear mappings.

A.1 Weighted Automata

In the classical presentation, a (finite) *weighted automaton* of dimension $n \geq 1$ over a finite alphabet A and a semiring S is a triple $\mathcal{A} = (\alpha, M_{a \in A}, \beta)$ where

- $\alpha \in S^{n \times 1}$ is a vector of length n, the initial vector;
- for every $a \in A$, $M_a \in S^{n \times n}$ is the transition matrix for the alphabet symbol a; and
- $\beta \in S^{1 \times n}$ is a vector of length n, the final vector.

The correspondence with the coalgebraic view on automata is now given as follows: we note that we can view every $n \times n$ matrix as a left-linear mapping[5] $\mathrm{Lin}_S(n) \to \mathrm{Lin}_S(n)$ (corresponding to the extension $\hat{\delta}(-, a)$), uniquely determined by a function $n \to \mathrm{Lin}_S(n)$ (corresponding to $\delta(-, a)$), and the final vector can be seen as a left-linear mapping from the left S-module $\mathrm{Lin}_S(n)$ to S itself, seen as a left S-module, again uniquely determined by a function $n \to S$.

[4] Unrelated to simulations as defined in process algebra.

[5] Note that $\mathrm{Lin}_S(n)$ can simply be seen as S^n here.

Ignoring the initial vector α, a traditional weighted automaton can then be seen as a coalgebra as follows:

$$\mathrm{Lin}_S(n) \xrightarrow{(\beta, M)} S \times \mathrm{Lin}_S(n)^A$$

The initial vector α, furthermore, simply is an element of $\mathrm{Lin}_S(n)$, and taking the word derivative of α to a word $w = a_1 \dots a_n$ corresponds to the sequence of matrix multiplications

$$M_{a_n} \dots M_{a_1} \alpha.$$

Finally, the formal power series $\mathcal{L}(\mathcal{A})$ *accepted* by a weighted automaton $\mathcal{A} = (\alpha, M_{a \in A}, \beta)$ can be specified by

$$\mathcal{L}(\mathcal{A})(a_1 \dots a_k) = \beta M_{a_k} \dots M_{a_1} \alpha$$

and by the above construction, it is seen to be equal to the power series $[\![\alpha_w]\!]$, where $[\![-]\!]$ is the usual notion of final coalgebra semantics for the functor $S \times -^A$, with respect to the coalgebra $(\mathrm{Lin}_S(n), \beta, M)$.

A.2 Simulations and Homomorphisms

Given two weighted automata $\mathcal{A} = (\alpha, M, \beta)$ (of dimension m) and $\mathcal{B} = (\gamma, N, \delta)$ (of dimension n), a matrix $Z \in S^{n \times m}$ is called a *simulation* from $\mathcal{A}$ to $\mathcal{B}$ whenever the following equations hold (in the case of the second equation, for all $a \in A$):

$$Z\alpha = \gamma \qquad ZM_a = N_a Z \qquad \beta = \delta Z$$

This definition corresponds to the one given in [ÉM10], with the modification that Z now represents a left-linear mapping, rather than a right-linear mapping.

A basic fact about simulations is that a simulation between weighted automata $\mathcal{A} = (\alpha, M, \beta)$ and $\mathcal{B} = (\gamma, N, \delta)$ in all cases implies equivalence of the weighted languages accepted by these automata, i.e. $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$. We will now turn to the connection between this notion of a simulation, and the notion of a homomorphism of coalgebras, from which this equivalence directly follows.

As $S^{n \times m}$ matrices are in bijective correspondence with left-linear mappings from $\mathrm{Lin}_S(m)$ to $\mathrm{Lin}_S(n)$, it directly follows that the second and third condition are equivalent to the condition that the following diagram commutes:

$$\begin{tikzcd}
\mathrm{Lin}_S(m) \arrow[r, "Z"] \arrow[d, "(\beta, M)"'] & \mathrm{Lin}_S(n) \arrow[d, "(\delta, N)"] \\
S \times \mathrm{Lin}_S(m)^A \arrow[r, "1_S \times Z^A"] & S \times \mathrm{Lin}_S(n)^A
\end{tikzcd}$$

Hence, for finite weighted automata (X, o_X, δ_X) and (Y, o_Y, δ_Y), the classical notion of a simulation between these automata corresponds to the coalgebraic notion of a homomorphism h from the extended automaton $(\mathrm{Lin}_S(X), \hat{o}_X, \hat{\delta}_X)$ to the extended automaton $(\mathrm{Lin}_S(Y), \hat{o}_Y, \hat{\delta}_Y)$, together with two elements $x \in \mathrm{Lin}_S(X)$ and $y \in \mathrm{Lin}_S(Y)$ such that $h(x) = y$.

A.3 Proper Semirings

In [ÉM10], a semiring S is called *proper* whenever, if two automata $\mathcal{A} = (\alpha, M, \beta)$ and $\mathcal{B} = (\gamma, N, \delta)$ are equivalent, i.e. $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$, there is a finite sequence of automata $\mathcal{A}_1, \ldots, \mathcal{A}_k$ for some k with $\mathcal{A}_1 = \mathcal{A}$ and $\mathcal{A}_k = \mathcal{B}$, such that for each i with $1 \leq i < k$ there either is a simulation from $\mathcal{A}_i$ to $\mathcal{A}_{i+1}$ or there is a simulation from $\mathcal{A}_{i+1}$ to $\mathcal{A}_i$.

Using the results from Section 3, we can now directly conclude that every Noetherian semiring is proper, as follows: assume we have two S-weighted automata (X, o_X, δ_X) and (Y, o_Y, δ_Y) and elements $x \in \mathrm{Lin}_S(X)$ and $y \in \mathrm{Lin}_S(Y)$ such that $[\![x]\!]_X = [\![y]\!]_Y$ w.r.t. the linear extensions of these automata.

We can first construct a weighted automaton $(X + Y, o_{X+Y}, \delta_{X+Y})$, and it is easy to see that this gives homomorphisms

$$\mathrm{Lin}_S(\kappa_1) : \mathrm{Lin}_S(X) \to \mathrm{Lin}_S(X + Y)$$
$$\mathrm{Lin}_S(\kappa_2) : \mathrm{Lin}_S(Y) \to \mathrm{Lin}_S(X + Y)$$

where $\kappa_1 : X \to X + Y$ and $\kappa_2 : Y \to X + Y$ denote the injections of the coproduct. Instantiating $\mathrm{Lin}_S(X + Y)$ for X in Diagram (1) and Proposition 7, we can now conclude that there are homomorphisms

$$(\pi_1 \circ i)^* : \mathrm{Lin}_S(R) \to \mathrm{Lin}_S(X + Y)$$
$$(\pi_2 \circ i)^* : \mathrm{Lin}_S(R) \to \mathrm{Lin}_S(X + Y)$$

and that $(\kappa_1(x), \kappa_2(y)) \in R$.

It now follows that this gives a chain of simulations of automata, as a result of (a) $\mathrm{Lin}_S(\kappa_1)$ mapping x to $\kappa_1(x)$; (b) $(\pi_1 \circ i)^*$ mapping $(\kappa_1(x), \kappa_2(y))$ to $\kappa_1(x)$; (c) $(\pi_2 \circ i)^*$ mapping $(\kappa_1(x), \kappa_2(y))$ to $\kappa_2(y)$; and (d) $\mathrm{Lin}_S(\kappa_2)$ mapping y to $\kappa_2(y)$.

Hence, we can now conclude that S satisfies the conditions of being proper. This fact is also observed in [ÉM10], using a somewhat different argument, using the properties of Noetherian semirings, rather than the property of monads where kernel pairs of finitely generated objects are finitely generated again.