Automata, Power Series, and Coinduction: Taking Input Derivatives Seriously (Extended Abstract)

J.J.M.M. Rutten

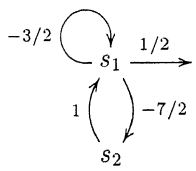
CWI, P.O. Box 94079, 1090 GB Amsterdam
janr@cwi.nl, http://www.cwi.nl/~janr

Formal power series are functions $\sigma : A^* \to k$ from the set of words over some alphabet A to some semiring k. Examples include formal languages ($k = \{0, 1\}$) and power series in classical analysis ($k = \mathbb{R}$, viewing the elements of A as variables). Because of their relevance to many different scientific areas, both in mathematics and computer science, a large body of literature on power series exists. Most approaches to the subject are essentially algebraic. The aim of this paper is to show that it is worthwhile to view power series from a dual perspective, called coalgebra (see [Rut96] for a general account). In summary, this amounts to supplying the set of all power series with a deterministic automaton structure, which has the universal property of being final. Finality then forms the basis for both definitions and proofs by *coinduction*, which is the coalgebraic counterpart of induction.

Coinductive *definitions* of operators on formal power series take the shape of what we have called *behavioural differential equations*, since they are formulated in terms of (a generalization of) Brzozowski's [Brz64] notion of input *derivative*: the input derivative σ_a of a series σ can intuitively be understood as the specification of the *behaviour* of σ after the input a has been accepted. For instance, the following behavioural differential equation defines the input derivative $(\sigma \parallel \tau)_a$ of the so-called shuffle product of σ and τ in terms of the input derivatives of σ and τ: $(\sigma \parallel \tau)_a = (\sigma_a \parallel \tau) + (\sigma \parallel \tau_a)$. It will be shown that these equations (one for each $a \in A$), together with an *initial condition*, determine a unique solution, which is taken as the formal definition of the shuffle product. Coinductive definitions allow easy proofs by the coinduction *proof* principle, which says that two series are equal whenever they are related by a bisimulation relation (which is the coalgebraic counterpart of a congruence relation). For coinductively defined operators, the construction of such bisimulations often is immediate from the defining differential equations.

The reader familiar with formal power series will know how to give a more elementary definition (cf. [BR88, p.20]) of the shuffle product mentioned as an example above, and therefore would call the differential equation a *property*. Our motivation for taking this and similar such differential equations as a *definition*, is three-fold: Firstly, the form of such equations will allow easy proofs by coinduction of many properties of the operators they define. In Section 4, a number of laws for the familiar operators on formal power series, will be shown to have easy proofs by coinduction. For instance, it takes a two-line proof to show that $\langle \sigma \parallel (\tau \parallel \rho), (\sigma \parallel \tau) \parallel \rho \rangle$ is contained in a bisimulation relation, implying that $\parallel$ is associative. Secondly, the approach can be easily generalized

to define new operators. This will be illustrated, in Section 5, by the definition of a new operator σ_{-1}, called shuffle inverse, because it satisfies $\sigma \parallel \sigma_{-1} = 1$ (to be proved by coinduction). It is unclear to us how this operator could be defined without the use of coinduction. Furthermore, many classical differential equations for analytic functions on $\mathbb{R}$ appear as particular instances of behavioural equations. Thirdly, we shall show, in Section 8, how from behavioural differential equations defining operators on power series, nondeterministic automata (with multiplicities in k) can be derived that implement these operators. The construction of these automata is again dictated by the shape of the differential equations, and is syntactic in the sense that their state space consists of expressions. The automata thus obtained are finite for rational power series (giving a new proof of the well-known fact that rational implies recognizable), and, in fact, generally very small. To give a flavour of this, the following two state automaton derives from the defining differential equation of the inverse operator σ^{-1} (with $\sigma \times \sigma^{-1} = 1$) on power series:



computes $(2 + 3X + 7X^2)^{-1} = \frac{1}{2} - \frac{3}{4}X - \frac{5}{8}X^2 + \cdots$

And so we see the following general scheme emerging: (a) (rational) behaviour is *specified* by differential equations, which often can be solved in a canonical way, giving rise to (b) (finite) nondeterministic automata that (efficiently) *implement* the specified behaviour. We see (a) and (b) as the two main contributions of our work.

Related work: The perspective of the present paper is essentially coalgebraic, and generalizes [Rut98], which deals with languages and regular expressions. The notion of input derivative of formal power series, generalizes Brzozowski's original definition for regular expressions [Brz64, Con71]. Its relation with *function derivatives* f' of functions f on $\mathbb{R}$ will be explained by invoking an example from [PE98], where a coinductive treatment of analytic functions in terms of their Taylor expansions is given. Our present theory generalizes the settings of [Rut98]: $k = \mathbb{B}$ and A is arbitrary, and [PE98]: $k = \mathbb{R}$ and $A = \{X\}$, since we are dealing with formal power series in many non-commutative variables (A is arbitrary) over any semiring (k is arbitrary). Although it is well known that rational series can be finitely represented (see [BR88], which has been our main reference on formal power series), also the syntactic construction of k-nondeterministic automata from their defining differential equations is to the best of our knowledge new.

Acknowledgements: I am grateful to Maurice Nivat, who offered me the opportunity to present a preliminary version of this paper at the Univ. of Paris VII.

Note: For a slightly more 'extended' abstract of the present ideas, amongst others containing some examples we had to leave out here, see: Technical Report SEN-R9901, CWI, 1999, which is available via `ftp.cwi.nl` or `www.cwi.nl`.

1 Preliminaries

We briefly recall the definitions of semiring and formal power series, and give a coalgebraic presentation of the notion of deterministic automaton.

Semirings: A semiring is something like a ring without subtraction. More formally, a semiring $k = \langle k, +, \times, 0, 1\rangle$ consists of a set k together with two binary operations $+$ and $\times$ (sum and product) and two constants 0 and 1, such that $(k, +, 0)$ is a commutative monoid with 0 as identity; $(k, \times, 1)$ is a monoid with 1 as identity; product is distributive with respect to sum; and $0x = x0 = 0$, all $x \in k$ (writing xy for $x \times y$). The following semirings will occur in examples in the paper: the Boolean semiring $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$, the reals $\mathbb{R} = (\mathbb{R}, +, \times, 0, 1)$, and the max-plus semiring $\mathbb{R}_{\max} = ([-\infty, \infty), \max, +, -\infty, 0)$. Note that both $\mathbb{B}$ and $\mathbb{R}_{\max}$ are *idempotent* semirings in that they satisfy $x + x = x$.

Words: Let A be a possibly infinite set and let A^* be the set of all finite words over A. Prefixing a word w in A^* with a letter a in A is denoted by aw. Concatenation of words w and w' is denoted by ww'. Let ε denote the empty word.

Formal power series: A *formal power series* is a function $\sigma : A^* \to k$. The set of all series is denoted by $k\langle\langle A\rangle\rangle$. A series σ assigns to each finite word $w \in A^*$ a *coefficient* $\sigma(w)$ in k, which may be interpreted as the *multiplicity* with which the word w occurs in σ. These multiplicities may have different interpretations, depending on the semiring k. If $k = \mathbb{B}$ then $\sigma(w)$ is either 1 or 0, indicating whether or not w belongs to σ, which in this case simply is a set of words. In other cases, the elements of A are best viewed as (formal non-commutative) *variables*. A basic but important example is $A = \{X\}$ and $k = \mathbb{R}$, when one gets the usual power series. As usual, k and A can be considered as subsets of $k\langle\langle A\rangle\rangle$, by taking $x \in k$ as the function $x : A^* \to k$ with $x(\varepsilon) = x$, and 0 everywhere else; similarly, $a \in A$ is identified with $a : A^* \to k$, defined by $a(a) = 1$, and 0 otherwise.

Deterministic automata: Let A be a possibly infinite set and let k be a semiring. A *deterministic automaton* (or Moore machine) with inputs in A and outputs in k is a pair $S = (S, \langle o_S, t_S\rangle)$ consisting of a set S of *states*, and a pair of functions: an *output function* $o_S : S \to k$, and a *transition function* $t_S : S \to S^A$. Here S^A is the set of all functions from A to S. The transition function t_S assigns to a state s a function $t_S(s) : A \to S$, which specifies the state $t_S(s)(a)$ that is reached after an input symbol a has been consumed. We shall sometimes write $s \xrightarrow{x}$ for $o_S(s) = x$ and $s \xrightarrow{a} s'$ for $t_S(s)(a) = s'$. Also we shall simply write o and t whenever the automaton S is clear from the context. A *homomorphism* between automata $S = (S, \langle o, t\rangle)$ and $S' = (S', \langle o', t'\rangle)$ is any function $f : S \to S'$ such that for all s in S, $o(s) = o'(f(s))$ and, for all a in A, $f(t(s)(a)) = t'(f(s))(a)$. A subset $i : S' \subseteq S$ of an automaton S is a *subautomaton* if i is a homomorphism. For a state s in S, $\langle s\rangle$ denotes the subautomaton *generated* by s. Homomorphisms map subautomata to subautomata. A relation $R \subseteq S \times S'$ is a *bisimulation* between two automata S and S' if, for all s in S, s' in S', and a in A: if $s\, R\, s'$ then $o(s) = o'(s')$ and $t(s)(a)\ R\ t'(s')(a)$. If there exists a bisimulation (between S and itself) containing $s, s' \in S$, then we write

$s \sim s'$ (s and s' are *bisimilar*). Bisimilarity itself is a bisimulation relation and an equivalence relation.

2 The automaton of formal power series

The set $k\langle\langle A\rangle\rangle$ of formal power series is turned into a deterministic automaton with inputs in A and outputs in k, having the universal property of being *final* and satisfying a principle of *coinduction*.

For an input a in A, the *input derivative* σ_a (or $\partial\sigma/\partial a$ or $a^{-1}\sigma$) of a series $\sigma : A^* \to k$ is defined by $\sigma_a(w) = \sigma(aw)$, for $w \in A^*$. The *constant part* (or output) of a series σ is defined by $\sigma(\varepsilon)$. These notions determine an automaton structure $k\langle\langle A\rangle\rangle = (k\langle\langle A\rangle\rangle, \langle o_k, t_k\rangle)$, defined, for $\sigma \in k\langle\langle A\rangle\rangle$ and $a \in A$, by $o_k(\sigma) = \sigma(\varepsilon)$ and $t_k(\sigma)(a) = \sigma_a$.

Theorem 1. *The automaton $k\langle\langle A\rangle\rangle$ satisfies the principle of (1)* coinduction*: for all series σ and τ in $k\langle\langle A\rangle\rangle$, if $\sigma \sim \tau$ then $\sigma = \tau$. Moreover, $k\langle\langle A\rangle\rangle$ is (2)* final*: for any automaton S there exists a* unique *homomorphism $l : S \to k\langle\langle A\rangle\rangle$; it satisfies: for $s, s' \in S$, $s \sim s'$ iff $l(s) = l(s')$.*

The series $l(s)$ is called the *behaviour* of the state s of the automaton S, and is defined as the function that assigns to any word w in A^* the output of the state that is reached from s after reading w; that is, for $w = a_0 \cdots a_{n-1}$,

$$l(s)(a_0 \cdots a_{n-1}) = o(s_n), \quad \text{where} \quad s = s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \cdots \xrightarrow{a_{n-1}} s_n.$$

We say that s *represents* the series $l(s)$, and also that $l(s)$ is the series *accepted* by the state s. Since it is easily shown that l is a homomorphism, this proves the existence half of Part (2). Uniqueness follows from Part (1), which is easily proved by induction on the length of words $w \in A^*$. The proof of this theorem also follows from general coalgebraic reasoning (see, e.g., [Rut96]). Note that it does not depend on the semiring structure of k.

Coinduction serves as a *proof* principle: in order to show $\sigma = \tau$, it is sufficient to establish the existence of a bisimulation relation R with $\sigma \; R \; \tau$. The proof principle will be illustrated in some detail in Section 4. Finality will be used as a *coinductive definition* principle (for instance, in Section 3).

The relation between derivatives f' of functions f on $\mathbb{R}$, and input derivatives σ_a of formal power series σ is explained by the following example on analytic functions, taken from [PE98]. Let $k = \mathbb{R}$ and $A = \{X\}$. Thus $\mathbb{R}\langle\langle X\rangle\rangle = \mathbb{R}^{\{X\}^*} \cong \mathbb{R}^\omega$, where $\omega = \{0, 1, \ldots\}$ is the set of natural numbers. In other words, formal power series are now infinite sequences, also called streams, of real numbers. Consider the set $\mathcal{A}$ of functions that are analytic in 0. For analytic functions, the n-th derivative $f^{(n)}(0)$ exists, for all $n \geq 0$. Following [PE98], $\mathcal{A}$ can be turned into an automaton by defining $o_{\mathcal{A}} : \mathcal{A} \to \mathbb{R}$ and $t_{\mathcal{A}} : \mathcal{A} \to \mathcal{A}$ (identifying $\mathcal{A}^{\{X\}} \cong \mathcal{A}$) by $o_{\mathcal{A}}(f) = f(0)$ and $t_{\mathcal{A}}(f) = f'$. Because $\mathbb{R}\langle\langle X\rangle\rangle$ is a final automaton, there exists a unique homomorphism $l : \mathcal{A} \to \mathbb{R}\langle\langle X\rangle\rangle$, which maps a function f to the series of its *Taylor* coefficients: $l(f) = (f(0), f'(0), f''(0), \ldots)$. Because l is a homomorphism, $l(f)_X = l(f')$. In words, the input derivative of the Taylor series of f is equal to the Taylor series of the derivative of f.

3 Behavioural differential equations

A number of operators on formal power series will be defined by *coinduction*. Similar to the way one can define, for instance, a function of exponentiation $exp : \mathbb{R} \to \mathbb{R}$ by specifying a differential equation and initial condition: $exp' = exp$ and $exp(0) = 1$, coinductive definitions of (elements of and) operators on $k\langle\langle A\rangle\rangle$ amount to the specification of *behavioural differential equations*. It will be a consequence of the finality of the automaton $k\langle\langle A\rangle\rangle$ that the systems of differential equations we shall use, have unique solutions.

Plunging into the matter, the aim of this section is to prove the following theorem.

Theorem 2. *There are unique functions* $+$, $\times$, $(-)^*$, $\|$, *and* $(-)^{-1}$, *called* sum, product, star, shuffle product, *and* inverse, *satisfying the following behavioural differential equations: For all* $\sigma, \tau \in k\langle\langle A\rangle\rangle$ *and* $a \in A$,

differential equation	*initial condition*
$(\sigma + \tau)_a = \sigma_a + \tau_a$	$(\sigma + \tau)(\varepsilon) = \sigma(\varepsilon) + \tau(\varepsilon)$
$(\sigma\tau)_a = \sigma_a \tau + \sigma(\varepsilon)\tau_a$	$(\sigma\tau)(\varepsilon) = \sigma(\varepsilon)\tau(\varepsilon)$
$(\sigma^*)_a = \sigma_a \sigma^*$	$(\sigma^*)(\varepsilon) = 1$
$(\sigma \parallel \tau)_a = (\sigma_a \parallel \tau) + (\sigma \parallel \tau_a)$	$(\sigma \parallel \tau)(\varepsilon) = \sigma(\varepsilon)\tau(\varepsilon)$
$(\sigma^{-1})_a = -\sigma(\varepsilon)^{-1}\sigma_a \sigma^*$	$(\sigma^{-1})(\varepsilon) = \sigma(\varepsilon)^{-1}$

Note that we have written $\sigma\tau$ for $\sigma \times \tau$, and that we use the same symbols for sum and product on $k\langle\langle A\rangle\rangle$ as for sum and product on k, respectively. Also note that in the expression $\sigma(\varepsilon)\tau_a = \sigma(\varepsilon) \times \tau_a$ above, we are interpreting $\sigma(\varepsilon)$, which is an element of k, as an element of $k\langle\langle A\rangle\rangle$, following the convention described in Section 1. Observe that in the definition of the initial conditions, the operators of the semiring structure of k are used. Finally note that the latter equation assumes k to be a *ring* rather than a semiring, since it uses subtraction. It is moreover *partial* since it only applies to such σ for which $\sigma(\varepsilon)$ is invertible in k. Whenever we shall write $-\sigma(\varepsilon)^{-1}$, both conditions will silently be assumed to apply. If either of these conditions does not hold then we put $(\sigma^{-1})_a = 0$ and $(\sigma^{-1})(\varepsilon) = 0$ (simply in order to keep all functions total).

Proof of Theorem 2: Let the set $\mathcal{E}$ of *expressions* be given by the following syntax: $E ::= \underline{\sigma} \mid E + F \mid EF \mid E^* \mid (E \parallel F) \mid E^{-1}$, where we write EF rather than $E \times F$, and where for every series σ in $k\langle\langle A\rangle\rangle$ a *symbol* $\underline{\sigma}$ is included in $\mathcal{E}$. The set $\mathcal{E}$ is next supplied with an automaton structure $(\mathcal{E}, \langle o_{\mathcal{E}}, t_{\mathcal{E}}\rangle)$. The functions $o_{\mathcal{E}} : \mathcal{E} \to k$ and $t_{\mathcal{E}} : \mathcal{E} \to \mathcal{E}^A$ are defined by induction on the structure of expressions, using the automaton structure of $k\langle\langle A\rangle\rangle$ for the symbols $\underline{\sigma}$, and following the structure of the differential equations in Theorem 2 for the operators: $o_{\mathcal{E}}(\underline{\sigma}) = \sigma(\varepsilon)$, $o_{\mathcal{E}}(E + F) = o_{\mathcal{E}}(E) + o_{\mathcal{E}}(F)$, $o_{\mathcal{E}}(EF) = o_{\mathcal{E}}(E \parallel F) = o_{\mathcal{E}}(E)o_{\mathcal{E}}(F)$, $o_{\mathcal{E}}(E^*) = 1$, $o_{\mathcal{E}}(E^{-1}) = o_{\mathcal{E}}(E)^{-1}$. (The latter expression should be interpreted as 0 if the inverse in k does not exist.) Writing E_a for $t_{\mathcal{E}}(E)(a)$, the function $t_{\mathcal{E}} : \mathcal{E} \to \mathcal{E}^A$ is given by the following clauses: $(\underline{\sigma})_a = \underline{\sigma_a}$, $(E + F)_a = E_a + F_a$, $(EF)_a = E_a F + o_{\mathcal{E}}(E)F_a$, $(E^*)_a = E_a E^*$, $(E \parallel F)_a = (E_a \parallel F) + (E \parallel F_a)$, $(E^{-1})_a = -o_{\mathcal{E}}(E)^{-1} E_a E^*$. (Read $\underline{0}$ for the last expression whenever $-o_{\mathcal{E}}(E)^{-1}$

is undefined.) Because $\mathcal{E}$ now has been turned into an automaton $(\mathcal{E}, \langle o_\mathcal{E}, t_\mathcal{E} \rangle)$, and because $k\langle\langle A \rangle\rangle$ is a final automaton, there exists, by Theorem 1, a unique homomorphism $l : \mathcal{E} \to k\langle\langle A \rangle\rangle$, which assigns to each expression E the formal power series $l(E)$ it represents. It can be used to define the operators on $k\langle\langle A \rangle\rangle$ that we are looking for: $\sigma + \tau = l(\underline{\sigma} + \underline{\tau})$, $\sigma\tau = l(\underline{\sigma}\,\underline{\tau})$, $\sigma^* = l(\underline{\sigma}^*)$, $\sigma \parallel \tau = l(\underline{\sigma} \parallel \underline{\tau})$, $\sigma^{-1} = l(\underline{\sigma}^{-1})$. (Note that the symbols for the operators on $k\langle\langle A \rangle\rangle$ are the same as the syntactic operators. The type will always be clear from the context.) One can show that $l(\underline{\sigma}) = \sigma$ and that l is compositional, e.g., $l(EF) = l(E)l(F)$, using the principle of coinduction (Theorem 1) and the fact that bisimilarity on $\mathcal{E}$ is a congruence relation (e.g., if $E \sim E'$ and $F \sim F'$ then $EF \sim E'F'$). For instance, the first statement follows by coinduction from the fact that $\{\langle l(\underline{\sigma}), \sigma \rangle \mid \sigma \in k\langle\langle A \rangle\rangle\}$ is a bisimulation relation on $k\langle\langle A \rangle\rangle$. One can now readily prove that the operators we have defined are solutions of their defining differential equations, using the fact that l is a compositional homomorphism. Uniqueness of these solutions follows from the uniqueness of l. □

Either by coinduction or, alternatively, using the uniqueness part of Theorem 2, one can prove that the above coinductive definitions of the operators on $k\langle\langle A \rangle\rangle$ coincide with the usual ones. For instance, $(\sigma^*)(w) = \sum_{n \geq 0} \sigma^n(w)$, if $\sigma(\varepsilon) = 0$.

Given the correspondence between derivative and input derivative, mentioned at the end of Section 2, one can easily show that the Taylor series of the *product* of analytic functions equals the *shuffle* product of their corresponding Taylor series: $l(fg) = l(f) \parallel l(g)$, where $(fg)(x) = f(x)g(x)$, as usual. (A proof by coinduction is easy, using the fact that $(fg)' = f'g + fg'$, which is of the same shape as the defining differential equation for $\parallel$.) This fact will be used in the definition of the shuffle inverse in Section 5. The correspondence between derivative and input derivative also shows that many classical differential equations, such as the example of *exp* mentioned at the beginning of this section, have a unique corresponding behavioural differential equation. For *exp*, this is the equation $(e)_X = e$ (with initial condition $e(\varepsilon) = 1$), which determines a unique series $e = (1, 1, 1, \ldots)$ in $\mathbb{R}\langle\langle X \rangle\rangle$, that is, the Taylor series of *exp*.

4 Proofs by coinduction

The use of coinduction is illustrated by proving some of the following familiar laws:

(1) $1 + \sigma\sigma^* = \sigma^*$, if $\sigma(\varepsilon) = 0$	(6) $\sigma = \sigma(\varepsilon) + \sum_{a \in A} a\sigma_a$
(2) $\sigma = \tau\sigma + \rho \Rightarrow \sigma = \tau^*\rho$, if $\tau(\varepsilon) = 0$	(7) $\sigma \parallel (\tau + \rho) = (\sigma \parallel \tau) + (\sigma \parallel \rho)$
(3) $(\sigma + \tau)^* = \sigma^*(\tau\sigma^*)^*$, if $\tau(\varepsilon) = 0$	(8) $\sigma \parallel (\tau \parallel \rho) = (\sigma \parallel \tau) \parallel \rho$
(4) $(\sigma + \tau)^* = (\sigma^*\tau)^*\sigma^*$, if $\tau(\varepsilon) = 0$	(9) $\sigma\sigma^{-1} = 1$
(5) $\sigma^* = (1 + \sigma)^*$	(10) $\sigma^{-1}\sigma = 1$

Coinductive proofs of equalities such as $\sigma_0 = \tau_0$ always proceed in the same way, by defining, in stages, a bisimulation relation R containing $\langle \sigma_0, \tau_0 \rangle$. The first pair to be included in R is $\langle \sigma_0, \tau_0 \rangle$. Next the following step is repeated until it does not yield any new pairs: the a-derivatives of the pairs $\langle \sigma, \tau \rangle$ already in

R are computed—for the operators, these are precisely given by the defining differential equations—and the resulting pairs $\langle \sigma_a, \tau_a \rangle$ are added to R. When adding a pair $\langle \sigma, \tau \rangle$ to R, at any stage of its construction, we should check that the constant parts are equal: $\sigma(\varepsilon) = \tau(\varepsilon)$. If this does not hold, the procedure aborts, and we conclude $\sigma \neq \tau$. Otherwise, the relation R that is thus obtained is by construction a bisimulation, and $\sigma_0 = \tau_0$ follows by coinduction. For instance, (1) follows by coinduction from the fact that

$$R_1 = \{\langle 1 + \sigma\sigma^*,\ \sigma^* \rangle \mid \sigma \in k\langle\langle A \rangle\rangle,\ \sigma(\varepsilon) = 0\} \cup \{\langle \sigma, \sigma \rangle \mid \sigma \in k\langle\langle A \rangle\rangle\}$$

is a bisimulation relation on $k\langle\langle A \rangle\rangle$: if $\sigma(\varepsilon) = 0$ then $(1 + \sigma\sigma^*)(\varepsilon) = 1 = \sigma^*(\varepsilon)$; moreover, $\langle (1 + \sigma\sigma^*)_a, (\sigma^*)_a \rangle = \langle 0 + \sigma_a\sigma^* + 0\sigma_a\sigma^*,\ \sigma_a\sigma^* \rangle = \langle \sigma_a\sigma^*,\ \sigma_a\sigma^* \rangle$, which is in R_1 again. Similarly,

$$R_2 = \{\langle \alpha\sigma + \beta,\ \alpha\tau^*\rho + \beta \rangle \mid \alpha, \beta \in k\langle\langle A \rangle\rangle\}$$

is a bisimulation relation, for σ, τ, ρ with $\sigma = \tau\sigma + \rho$ and $\tau(\varepsilon) = 0$, implying (2) by coinduction. All the other laws are proved similarly. To mention one last example, let R_8 be the smallest relation on $k\langle\langle A \rangle\rangle$ such that $\langle \sigma \parallel (\tau \parallel \rho),\ (\sigma \parallel \tau) \parallel \rho \rangle \in R_8$, for all $\sigma, \tau, \rho \in k\langle\langle A \rangle\rangle$, and such that $\langle \sigma_1, \tau_1 \rangle, \langle \sigma_2, \tau_2 \rangle \in R_8$ implies $\langle \sigma_1 + \sigma_2,\ \tau_1 + \tau_2 \rangle \in R_8$. It is straightforward to prove that R_8 is a bisimulation (using (7)), whence (8) by coinduction. Note that for none of the proofs above, additional structure had to be introduced. Notably, there is no need of turning $k\langle\langle A \rangle\rangle$ into a topological semiring, which is what is usually done (see, for instance, [BR88, Lm 4.1, p.5]).

5 Shuffle inverse

The correspondence between the product of two functions on the reals and the *shuffle* product of their corresponding Taylor series (mentioned at the end of Section 3), suggests the following definition of an operator that acts as a quotient with respect to the shuffle product. Recalling the familiar quotient law for derivatives: $(f^{-1})' = -f'(1/f^2) = -f'(ff)^{-1}$, consider the following behavioural differential equation: $(\sigma_{-1})_a = -\sigma_a \parallel (\sigma \parallel \sigma)_{-1}$ with initial condition $(\sigma_{-1})(\varepsilon) = \sigma(\varepsilon)^{-1}$. Note that we write σ_{-1} rather than σ^{-1}, since the latter notation is used, in Section 3, for the inverse with respect to multiplication. Further note that k is assumed to be a ring and that the above equation only applies to such σ for which $\sigma(\varepsilon)$ is invertible in k. The above equation has a unique solution, which can be proved along the same lines as Theorem 2. Assuming that k is a ring, the following equalities hold for all $\sigma \in k\langle\langle A \rangle\rangle$ for which $\sigma(\varepsilon)$ is invertible in k, showing that the shuffle inverse behaves as intended: $\sigma \parallel \sigma_{-1} = 1$ and $(\sigma_{-1})_{-1} = \sigma$. This can be readily proved by coinduction. It is not immediately obvious how this operator could be defined without coinduction. For now, we are satisfied with the fact that it has been possible to define it at all. Its use for the theory of power series is to be studied further.

6 Rational series

We recall the notion of rational series, and illustrate the need of nondeterministic automata with multiplicities in k in order to obtain finite representations for them. Let the set $\mathcal{R}$ of *regular expressions* be given by the following syntax: $E ::= \ x \in k \mid a \in A \mid E + F \mid EF \mid E^*$. Note that, for convenience, we write x and a rather than $\underline{x}$ and $\underline{a}$ and that, under the embedding of k and A in $k\langle\langle A \rangle\rangle$, $\mathcal{R}$ is a subset of the set of expressions $\mathcal{E}$, introduced in (the proof of) Theorem 2. A series σ is called *rational* if there exists a regular expression E with $\sigma = l(E)$, where $l : \mathcal{E} \to k\langle\langle A \rangle\rangle$ is the unique homomorphism of Theorem 2. Because l is compositional, a series is rational iff it is contained in the smallest subset of $k\langle\langle A \rangle\rangle$ that contains k and A (viewed as subsets of $k\langle\langle A \rangle\rangle$) and that is closed under the operators of sum, product, and star. In order to see whether a series σ is rational or not, it is sufficient to look at the subautomaton $\langle\sigma\rangle$ of $k\langle\langle A \rangle\rangle$ that it generates. This subautomaton is generally infinite: for instance, $\langle (xa)^* \rangle = \{x^n(xa)^* \mid n \geq 0\}$. This example is typical in the sense that the generated subautomaton of a rational series is characterized by the property that it is *finitely generated*. We shall not prove this in the present paper, but in Section 8, we shall see another, truly finitary characterization of rational power series (from which this property easily follows). There it will be shown that a rational series is recognized by a finite *nondeterministic* automaton with multiplicities in the semiring k.

7 Nondeterministic automata

In order to give a truly finite representation of rational series, this section introduces (a coalgebraic formulation of) nondeterministic automata and gives a coinductive definition of their behaviour. In Section 8, *finite* nondeterministic automata for rational series will be constructed.

A *k-nondeterministic automaton* (*nd-automaton* for short, also called k-transducer) with inputs in A and outputs in k is a pair $S = (S, \langle o, t \rangle)$ consisting of a set S of *states*, and a pair of functions: an *output function* $o : S \to k$, and a *nondeterministic transition function* $t : S \to k(S)^A$. Here $k(S)^A$ is the set of all functions from A to $k(S)$, which at its turn is defined by

$$k(S) = \{\phi : S \to k \mid supp(\phi) \text{ is finite } \}$$

where $supp(\phi) = \{s \in S \mid \phi(s) \neq 0\}$ is the *support* of ϕ. The observation function o assigns to each state s in S a multiplicity $o(s)$ in k. The transition function t assigns to a state s in S a function $t(s) : A \to k(S)$, which specifies for any a in A a function $t(s)(a) \in k(S)$. Such a function can be viewed as a kind of nondeterministic or distributed state, and specifies for any state s' in S a multiplicity $t(s)(a)(s')$ in k with which the a-transition from s to s' occurs. We shall sometimes write $s \xrightarrow{a|x} s'$ for $t(s)(a)(s') = x$ and $s \xrightarrow{x}$ for $o(s) = x$.

The *behaviour* of a state in a nd-automaton, which is again a formal power series, is defined coinductively. To this end, we shall first associate with every

nondeterministic automaton $\langle o, t\rangle : S \to k \times k(S)^A$ a corresponding *deterministic* automaton. The set of states of the new automaton is given by the set $k(S)$ (of distributed states) mentioned above. Next the set $k(S)$ is turned into a deterministic automaton $(k(S), \langle \hat{o}, \hat{t}\rangle)$, by defining an observation function $\hat{o} : k(S) \to k$ and a deterministic transition function $\hat{t} : k(S) \to k(S)^A$, as follows:

$$\hat{o}(\phi) = \sum_{s \in S} o(s)\phi(s), \quad \hat{t}(\phi)(a)(s) = \sum_{s' \in S} \phi(s')\,(t(s')(a)(s))$$

Note that both these sums exist because ϕ in $k(S)$ has finite support. The behaviour of a nd-automaton S can now be defined in terms of the, by Theorem 1, unique homomorphism $\lambda : k(S) \to k\langle\langle A\rangle\rangle$, which assigns to each configuration ϕ in $k(S)$ the formal power series $\lambda(\phi)$ it represents. Because of the existence of the obvious inclusion $\{\cdot\} : S \to k(S)$, with $\{s\}(s') = 1$ if $s = s'$, and $= 0$ otherwise, we have obtained a function $\lambda \circ \{\cdot\} : S \to k\langle\langle A\rangle\rangle$, which is the coinductive definition of the behaviour of $(S, \langle o, t\rangle)$ that we were after.

The term 'multiplicity' used above may have many different interpretations, depending on the semiring k. If $k = \mathbb{B}$ then $\mathbb{B}$-nondeterministic automata are precisely the classical nondeterministic automata, with $o : S \to \mathbb{B}$ specifying which states are terminal (accepting), and where for a state $s \in S$ and input letter $a \in A$, $t(s)(a) \in \mathbb{B}(S) \cong \mathcal{P}_f(S)$ gives the (finite) set of possible next states. The construction of a deterministic automaton above amounts in this case exactly to the familiar power set construction.

8 Recognizability

A formal power series $\sigma \in k\langle\langle A\rangle\rangle$ is *recognizable* if there exists a *finite* nd-automaton S and a state $s \in S$ such that $\lambda(\{s\}) = \sigma$ (with λ as defined in Section 7). The pair (S, s) is then called a *finite representation* of σ. In this section, we construct a finite representation for any *rational* series $l(E)$, with E a regular expression, thus giving a new proof of the well-known fact that any rational series is recognizable (cf. [BR88]). The representation is *syntactic* in the sense that its state space consists of (regular) expressions.

To this end, the entire set $\mathcal{R}$ of regular expressions is turned into an (infinite) nondeterministic automaton $(\mathcal{R}, \langle o_{\mathcal{R}}, t_{\mathcal{R}}\rangle)$, such that the behaviour of E is precisely given by $l(E)$. As we shall see, the subautomaton $\mathcal{R}_E \subseteq \mathcal{R}$ generated by E is finite, giving a finite representation $(\mathcal{R}_E, E)$ of $l(E)$.

The observation function $o_{\mathcal{R}} : \mathcal{R} \to k$ is defined by $o_{\mathcal{R}}(E) = o_{\mathcal{E}}(E)$, where $o_{\mathcal{E}}$ is the observation function for expressions, defined in Section 3. The nondeterministic transition function $t_{\mathcal{R}} : \mathcal{R} \to k(\mathcal{R})^A$ is defined by induction on the structure of regular expressions, following the shape of the behavioural differential equations of Theorem 2. We mention a few typical cases:

$$t_{\mathcal{R}}(E + F)(a)(G) = t_{\mathcal{R}}(E)(a)(G) + t_{\mathcal{R}}(F)(a)(G)$$

$$t_{\mathcal{R}}(EF)(a)(G) = \begin{cases} t_{\mathcal{R}}(E)(a)(E') + o_{\mathcal{R}}(E)\, t_{\mathcal{R}}(F)(a)(G) & \text{if } G = E'F \\ o_{\mathcal{R}}(E)\, t_{\mathcal{R}}(F)(a)(G) & \text{otherwise} \end{cases}$$

Applying now the definitions from Section 7, we can prove by coinduction that $\lambda(\{E\}) = l(E)$, for any regular expression $E \in \mathcal{R}$. Because the subautomaton $\mathcal{R}_E$ of $\mathcal{R}$ generated by E is finite, we have:

Theorem 3. *For a regular expression E in $\mathcal{R}$, $(\mathcal{R}_E, E)$ is a finite representation of $l(E)$ (hence any rational series is recognizable).* □

Finite representations for the shuffle product and the inverse (with respect to multiplication) of rational expressions can be obtained in a similar fashion, by extending the above approach, leading to nd-automata such as the two state automaton depicted in the introduction.

9 Discussion

We briefly mention some of the work that remains to be done: (1) General formats for behavioural differential equations, ensuring the existence of a unique solution, should be determined. (2) The effectiveness of the coinduction proof principle is to be further investigated, as well as the minimization of finite representations, to which it is closely related (cf. [Rut98]). (3) The example of the shuffle inverse should be further investigated. (4) Many more examples of specifications by behavioural differential equations and the corresponding implementations are to be studied, amongst others involving tropical and idempotent semirings as described in [Gun98]. (5) Applying the universal coalgebraic definition of bisimulation directly to nd-automata (and not only to deterministic automata as we have done here) will yield notions of equivalence that have an interest in their own right. For instance, taking $k = \mathbb{R}$ gives a notion of bisimulation that is (under conditions) probabilistic bisimulation. (6) The use of nd-automata for the representation of Taylor series of analytic functions, such as the trigonometric functions, yields surprising results and deserves further study.

References

[BR88] J. Berstel and C. Reutenauer. *Rational series and their languages*, volume 12 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1988.

[Brz64] J.A. Brzozowski. Derivatives of regular expressions. *Journal of the ACM*, 11(4):481–494, 1964.

[Con71] J.H. Conway. *Regular algebra and finite machines*. Chapman and Hall, 1971.

[Gun98] J. Gunawardena. *Idempotency*. Pub. of the Newton Institute. CUP, 1998.

[PE98] D. Pavlović and M. Escardó. Calculus in coinductive form. In LICS'98.

[Rut96] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. Report CS-R9652, CWI, 1996. To appear in Theoretical Computer Science.

[Rut98] J.J.M.M. Rutten. Automata and coinduction (an exercise in coalgebra). Report SEN-R9803, CWI, 1998. Also in the proceedings of CONCUR '98, LNCS 1466, 1998, pp. 194-218.