

## CORRIGENDUM TO OUR PAPER: HOW EXPRESSIONS CAN CODE FOR AUTOMATA

SYLVAIN LOMBARDY<sup>1</sup> AND JACQUES SAKAROVITCH<sup>2</sup>

**Abstract.** In a previous paper, we have described the construction of an automaton from a rational expression which has the property that the automaton built from an expression which is itself computed from a co-deterministic automaton by the state elimination method is co-deterministic. It turned out that the definition on which the construction is based was inappropriate, and thus the proof of the property was flawed. We give here the correct definition of the broken derived terms of an expression which allow to define the automaton and the detailed full proof of the property.

**Mathematics Subject Classification.** 68Q45, 68Q70.

### 1. THE PROBLEM, ITS PARTIAL SOLUTION, AND A FLAW

The conversion of finite automata into regular expressions and conversely of regular expressions to finite automata is an old, and fundamental, problem in automata theory. The algorithms that produce an expression which denotes the language accepted by a finite automaton (McNaughton-Yamada algorithm, state elimination method) are more or less ‘equivalent’ [9] and yield an expression whose length may be exponential in the size of the automaton. The algorithms that build a (non deterministic) automaton from an expression (Glushkov, Antimirov) output an automaton whose size is roughly (and at most for Antimirov) equal to the length of the expression. In a previous paper of ours [8], we have considered the possibility of finding a method that would be reversible, in the sense that

---

*Keywords and phrases.* Finite automata, regular expression, derivation of expressions, quotient of automata.

<sup>1</sup> IGM-LabInfo (UMR 8049), Université Paris-Est Marne-la-Vallée, 77454 Marne-la-Vallée Cedex 2, France; [lombardy@univ-mlv.fr](mailto:lombardy@univ-mlv.fr).

<sup>2</sup> LTCI (UMR 5141), CNRS/Télécom ParisTech, 46 rue Barrault, 75634 Paris Cedex 13, France; [sakarovitch@enst.fr](mailto:sakarovitch@enst.fr).

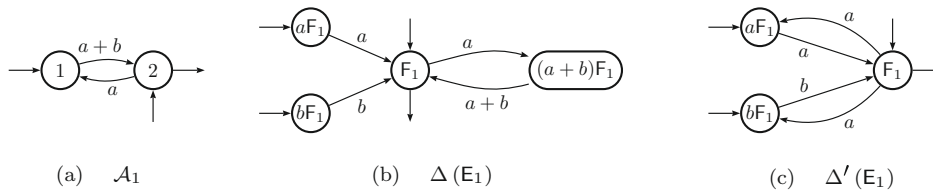


FIGURE 1. The old and new constructions of automata from an expression.

it recovers an automaton from an expression that has been computed from this automaton. More precisely we have stated the following problem:

*Find an algorithm  $\Omega$  which, given any rational expression  $E$  computed from an automaton  $\mathcal{A}$  – e.g.  $E = \Phi(\mathcal{A})$  where  $\Phi$  is the state elimination method –, computes an automaton  $\Omega(E)$  isomorphic to  $\mathcal{A}$ .*

We did not solve the problem completely in [8], but we have identified two constructions that are good candidates to be the core components of such an algorithm  $\Omega$ . The first one is a variant of Antimirov’s construction of an automaton, the second one, which plays a more ancillary role, is the operation of ‘co-quotient’. The flaw in the original paper is due to an inappropriate definition of the variant of Antimirov’s derivation that we used to build the automaton, there denoted by  $\Delta(E)$ , from an expression  $E$ . We describe now *another* variant of Antimirov’s derivation, which is based on what we called the *broken derived terms* in another paper of ours ([7]) and defined there just for the sake of the exploration. Once this new derivation is set up, the construction from the expression  $E$  of the automaton, that we denote by  $\Delta'(E)$ , goes then as in [8], and what was considered as the key statement can be stated again.

**Theorem 1.1.** *Let  $\mathcal{A}$  be a co-deterministic automaton and  $E = \Phi(\mathcal{A})$  a rational expression computed from  $\mathcal{A}$  by the state elimination method. Then, the broken derived term automaton  $\Delta'(E)$  of  $E$  is co-deterministic.*

A simple example will illustrate the transformation between the construction in our first paper [8] and the new one.

Let  $\mathcal{A}_1$  be the automaton of Figure 1a, and let

$$E_1 = \Phi(\mathcal{A}_1) = (a + b + 1) [a(a + b)]^*$$

be the expression computed from  $\mathcal{A}_1$  by the state elimination method (by suppressing first state 1 and then state 2). We write  $E_1 = (a + b + 1)F_1$  with  $F_1 = [a(a + b)]^*$ ; Figure 1b shows  $\Delta(E_1)$  and Figure 1c shows  $\Delta'(E_1)$ . The automaton  $\mathcal{A}_1$  is co-deterministic and so is  $\Delta'(E_1)$ , whereas  $\Delta(E_1)$  is not co-deterministic. Now, the minimal co-quotient of  $\Delta'(E_1)$ , denoted by  $\Upsilon(\Delta'(E_1))$  (all definitions are given in the next section), is isomorphic to  $\mathcal{A}_1$  – whereas  $\Upsilon(\Delta(E_1))$  is not, of

course. Hence,  $\mathcal{A}_1 = \Upsilon \left( \Delta' (\Phi(\mathcal{A}_1)) \right)$  and  $\Upsilon \circ \Delta'$  is a good candidate to make up the algorithm  $\Omega$ .

The remainder of this paper is devoted to the definitions (Sect. 2) and to the proof of Theorem 1.1 (Sects. 3 and 4). There must have been a flaw in the proof of the corresponding statement in [8]. It is not interesting to find it, but it is important to give a full (and correct) proof of Theorem 1.1 in order to keep valid all the work that has been done around it in [8].

## 2. THE (CORRECTED) DEFINITIONS AND STATEMENT

### 2.1. PRELIMINARIES

In the sequel,  $A$  is a finite alphabet and  $A^*$  the free monoid generated by  $A$ . The empty word, denoted by  $1_{A^*}$ , is the identity of  $A^*$ . We denote by  $A^+$  the set of non-empty words:  $A^+ = A^* \setminus \{1_{A^*}\}$ .

For any subset  $L$  of  $A^*$ , and any  $u$  in  $A^*$ , the (*left*) *quotient* of  $L$  by  $u$  is defined as  $u^{-1}L = \{v \in A^* \mid uv \in L\}$ . The quotient operation defines an *action* of  $A^*$  onto the power set of  $A^*$ , that is:

$$\forall u, v \in A^*, \forall L \subseteq A^* \quad (uv)^{-1}L = v^{-1} [u^{-1}L]. \quad (1)$$

#### 2.1.1. Basics I: rational expressions

The set of *rational expressions over*  $A$ , denoted by  $\text{RatE}(A)$ , is the set of well-formed formulas built inductively from the *constants* 0 and 1 and the letters  $a$  in  $A$  as *atomic formulas* and with two binary operators  $+$  and  $\cdot$  and one unary operator  $*$ : if  $E$  and  $F$  are rational expressions, so are  $(E + F)$ ,  $(E \cdot F)$ , and  $(E^*)$ . We often write simply *expression* instead of *rational expression*.

With every rational expression  $E$  is associated a language of  $A^*$  which is called the *language denoted by*  $E$  and we write<sup>1</sup> it  $|E|$ . The language  $|E|$  is inductively defined by  $|0| = \emptyset$ ,  $|1| = \{1_{A^*}\}$ ,  $|a| = \{a\}$  for every  $a$  in  $A$ ,  $|(E + F)| = |E| \cup |F|$ ,  $|(E \cdot F)| = |E||F|$ , and  $|(E^*)| = \{|E|\}^*$ . Two expressions are *equivalent* if they denote the same language.

In the sequel, any operator defined on expressions is implicitly extended additively to sets of expressions. For instance, we have:

$$\forall X \subseteq \text{RatE}(A) \quad |X| = \bigcup_{E \in X} |E|. \quad (2)$$

Using the classical precedence relation between operators: ' $* > \cdot > +$ ' allows to save parentheses in the writing of expressions – *e.g.*  $E + F \cdot G^*$  is an unambiguous

<sup>1</sup>The notation  $L(E)$  is probably more common. But  $|E|$  proves to be lighter in the computations performed in this paper. Note also that this latter notation is more appropriate when dealing with expressions with multiplicity, although it is not the case here.

writing for the expression  $(E + (F \cdot (G^*)))$  – but one should be aware that for instance  $(E \cdot (F \cdot G))$  and  $((E \cdot F) \cdot G)$  are two equivalent *but distinct* expressions.

Even if for the sake of our purpose and the correct definition of the derivation, we distinguish between expressions that seem to be so obviously equivalent, all the computations on expressions that will be defined below are performed modulo a set of seven identities, that we call *trivial identities*:

$$E + 0 \equiv 0 + E \equiv E, \quad E \cdot 0 \equiv 0 \cdot E \equiv 0, \quad E \cdot 1 \equiv 1 \cdot E \equiv E, \quad 0^* \equiv 1. \quad (\mathbf{T})$$

An expression is said to be *reduced* if it contains no subexpression which is a left-hand side of one of the above identities; in particular,  $0$  does not appear in a non-zero reduced expression. It is not necessary to set up a full theory of rational identities in order to understand that any expression  $H$  can be rewritten in an equivalent reduced expression  $H'$ , and that this  $H'$  is unique and independent from the way the rewriting is conducted.

**Definition 2.1.** Let  $E$  be a rational expression. The *constant term* of  $E$ , written  $c(E)$ , is the Boolean value defined as follows:

$$c(0) = 0, \quad c(1) = 1, \quad \forall a \in A \quad c(a) = 0, \\ c(F + G) = c(F) \vee c(G), \quad c(F \cdot G) = c(F) \wedge c(G), \quad c(F^*) = 1.$$

**Property 2.2.** The constant term of an expression  $E$  is equal to 1 if, and only if,  $1_{A^*}$  belongs to  $|E|$ .

### 2.1.2. Basics II: finite automata

We recall here classical and very general definitions and constructions on automata. We postpone the description of normalised automata, which is not less classical but which will specifically be used in the proof.

We denote by  $\mathcal{A} = \langle Q, A, E, I, T \rangle$  a (finite) automaton over  $A$  whose set of states, initial states and final states are  $Q$ ,  $I$  and  $T$  respectively and whose set of transitions is  $E \subseteq (Q \times A \times Q)$ . The *language (of  $A^*$ ) accepted (or recognised)* by  $\mathcal{A}$  is denoted by  $|\mathcal{A}|$  (*cf.* note 1). An automaton is *trim* if every state belongs to a successful computation.

The *transpose* of  $\mathcal{A}$  is the automaton  ${}^t\mathcal{A} = \langle Q, A, {}^tE, T, I \rangle$  whose transitions are the reverse of those of  $\mathcal{A}$ :  ${}^tE = \{(q, a, p) \in (Q \times A \times Q) \mid (p, a, q) \in E\}$ . This construction will be used to define ‘dual’ notions on automata.

The automaton  $\mathcal{A}$  is *deterministic* if  $I$  is a singleton (or empty), and if for every  $p$  in  $Q$  and every  $a$  in  $A$ , there exists at most one  $q$  in  $Q$  such that  $(p, a, q)$  is in  $E$ . The automaton  $\mathcal{A}$  is *co-deterministic* if  ${}^t\mathcal{A}$  is deterministic, that is, if  $T$  is a singleton (or empty), and if for every  $p$  in  $Q$  and every  $a$  in  $A$ , there exists at most one  $q$  in  $Q$  such that  $(q, a, p)$  is in  $E$ .

For each state  $q$  of  $\mathcal{A}$ , the *past of  $q$  (in  $\mathcal{A}$ )* is the set of labels of computation which go from an initial state of  $\mathcal{A}$  to  $q$ , and we write it  $\text{Past}_{\mathcal{A}}(q)$ ; the *future of  $q$  (in  $\mathcal{A}$ )* is the set of labels of computations that go from  $q$  to a final state of  $\mathcal{A}$  and

we write it  $\text{Fut}_{\mathcal{A}}(q)$ :

$$\text{Past}_{\mathcal{A}}(q) = \{w \in A^* \mid \exists i \in I \quad i \xrightarrow[\mathcal{A}]{w} q\}, \quad \text{Fut}_{\mathcal{A}}(q) = \{w \in A^* \mid \exists t \in T \quad q \xrightarrow[\mathcal{A}]{w} t\}.$$

**Property 2.3.** A trim automaton  $\mathcal{A}$  is *deterministic* if, and only if, the pasts of states are pairwise disjoint, and dually,  $\mathcal{A}$  is *co-deterministic* if, and only if, the futures of states are pairwise disjoint.

The following definitions and statements, that are, or should be, classical [7,9], are taken in order to state Proposition 2.7, which, combined with Theorem 1.1, makes of  $\Upsilon$  a component of an algorithm  $\Omega$ . The definition of morphism and of minimal quotients of automata is a generalisation of the classical minimisation of deterministic automata (similar to the definition of *simulation* among *transition systems*). These notions apply to any kind of automata and are more versatile. On the other hand, the minimal quotient of an automaton is not canonically attached to the recognized language anymore but depends on the automaton it is computed from.

**Definition 2.4.** Let  $\mathcal{A} = \langle Q, A, E, I, T \rangle$  and  $\mathcal{B} = \langle R, A, F, J, U \rangle$  be two automata; a map  $\psi: Q \rightarrow R$  is a *morphism (of automata)* if:

- (i)  $\psi(I) \subseteq J$ ,
- (ii)  $\psi(T) \subseteq U$ , and
- (iii) for every transition  $(p, a, q)$  in  $E$ ,  $(\psi(p), a, \psi(q))$  is a transition in  $F$ .

If  $\psi$  is such a morphism, we write  $\psi: \mathcal{A} \rightarrow \mathcal{B}$ , and it immediately follows that  $|\mathcal{A}| \subseteq |\mathcal{B}|$ .

**Definition 2.5.** The automaton  $\mathcal{B}$  is a *quotient* of  $\mathcal{A}$ , if there exists a morphism  $\psi: \mathcal{A} \rightarrow \mathcal{B}$  such that:

- (i)  $\psi(Q) = R$ ,
- (ii)  $\psi(I) = J$ ,
- (iii)  $\psi(T) = U$ , and
- (iv) for every transition  $(r, a, s)$  in  $F$  and every  $p$  in  $\psi^{-1}(r)$ , there exists a  $q$  in  $\psi^{-1}(s)$  such that  $(p, a, q)$  is a transition in  $E$ .

If  $\mathcal{B}$  is a quotient of  $\mathcal{A}$ , then  $|\mathcal{A}| = |\mathcal{B}|$ .

**Proposition 2.6.** *Every automaton  $\mathcal{A}$  has a minimal quotient  $\mathcal{C}$ , which is unique up to an isomorphism, and which is the quotient of any quotient  $\mathcal{B}$  of  $\mathcal{A}$ .*

The terminology does not make it clear, but the notion of quotient is directed (from the origin to the destination of the transitions). The automaton  $\mathcal{B}$  is a *co-quotient* of  $\mathcal{A}$  if  $\mathfrak{t}\mathcal{B}$  is a quotient of  $\mathfrak{t}\mathcal{A}$ . And the dual of Proposition 2.6 also holds.

**Proposition 2.7.** *Every automaton  $\mathcal{A}$  has a minimal co-quotient  $\mathcal{D}$ , which is unique up to an isomorphism, and which is the co-quotient of any co-quotient  $\mathcal{B}$  of  $\mathcal{A}$ .*

The minimal quotient or co-quotient of an automaton can be computed by a kind of Moore algorithm that consists in successive refinements of the trivial partition on the set of states. We denote the co-minimisation algorithm by  $\Upsilon$ . Let us note again that  $\Upsilon(\mathcal{A})$  is canonically attached to  $\mathcal{A}$  and not to the language accepted by  $\mathcal{A}$ , but for the case where  $\mathcal{A}$  is co-deterministic, in which case  $\Upsilon(\mathcal{A})$  is *the* minimal co-deterministic automaton of  $|\mathcal{A}|$ .

## 2.2. DERIVATION OF EXPRESSIONS

### 2.2.1. Antimirov derivation

Let us first recall the notion of *derivation* as defined by Antimirov.

**Definition 2.8.** [Antimirov [2]] Let  $E$  be a rational expression over  $A$  and  $a$  a letter in  $A$ . The  $\mathbb{B}$ -derivation<sup>2</sup> of  $E$  with respect to  $a$ , denoted  $\frac{\partial}{\partial a} E$ , is a set of rational expressions over  $A$ , recursively defined by

$$\begin{aligned} \frac{\partial}{\partial a} 0 &= \frac{\partial}{\partial a} 1 = \emptyset, \\ \forall b \in A \quad \frac{\partial}{\partial a} b &= \begin{cases} \{1\} & \text{if } b = a, \\ \emptyset & \text{otherwise,} \end{cases} \end{aligned}$$

$$\frac{\partial}{\partial a} (F + G) = \frac{\partial}{\partial a} F \cup \frac{\partial}{\partial a} G, \quad (3)$$

$$\frac{\partial}{\partial a} (F \cdot G) = \left[ \frac{\partial}{\partial a} F \right] \cdot G \cup c(F) \frac{\partial}{\partial a} G, \quad (4)$$

$$\frac{\partial}{\partial a} (F^*) = \left[ \frac{\partial}{\partial a} F \right] \cdot F^*. \quad (5)$$

Equation (4) should be understood as

$$\frac{\partial}{\partial a} (F \cdot G) = \begin{cases} \left[ \frac{\partial}{\partial a} F \right] \cdot G \cup \frac{\partial}{\partial a} G, & \text{if } c(F) = 1, \\ \left[ \frac{\partial}{\partial a} F \right] \cdot G, & \text{if } c(F) = 0. \end{cases}$$

That is, the product  $xX$  of a set  $X$  by a Boolean value  $x$  is  $xX = X$  if  $x = 1$  and  $xX = \emptyset$  if  $x = 0$ .

The induction implied by Equations (3)–(5) should be interpreted by extending derivation additively (as are always derivation operators) and by distributing (on the right) product over union as in (2); if  $X$  is a set of expressions, we have:

$$\frac{\partial}{\partial a} X = \bigcup_{E \in X} \frac{\partial}{\partial a} E, \quad X \cdot F = \bigcup_{E \in X} (E \cdot F).$$

<sup>2</sup>We call it ‘ $\mathbb{B}$ -derivation’ in order to avoid confusion with the ‘derivatives’ defined by Brzozowski [3]. In other papers and works, when dealing with expressions with multiplicity in a semiring  $\mathbb{K}$ , we called the result of the derivation a ‘ $\mathbb{K}$ -derivative’ (cf. [7,9]).

**Definition 2.9.** Let  $E$  be a rational expression over  $A$  and  $v$  a non empty word of  $A^*$ , i.e.  $v = ua$  with  $a$  in  $A$ . The  $\mathbb{B}$ -*derivation* of  $E$  with respect to  $v$ , denoted  $\frac{\partial}{\partial v} E$ , is the set of rational expressions over  $A$ , recursively defined by formulae (3)–(5) and by:

$$\forall u \in A^+, \forall a \in A \quad \frac{\partial}{\partial ua} E = \frac{\partial}{\partial a} \left( \frac{\partial}{\partial u} E \right). \quad (6)$$

We call *derived term* of  $E$ , either the expression  $E$  itself<sup>3</sup>, or any rational expression which belongs to a set  $\frac{\partial}{\partial v} E$  for some  $v$  in  $A^+$ . We write  $D(E)$  for the set of derived terms of  $E$ .

The derivation with respect to a word is consistent with the left quotient by a word, in the sense that the following holds.

$$\forall u \in A^+, \forall E \in \text{Rat}E(A) \quad \left| \frac{\partial}{\partial u} E \right| = u^{-1} |E|. \quad (7)$$

The definition of derived terms allows then the definition of an automaton, which was the original purpose.

**Definition 2.10.** [Antimirov [2]] The *derived term automaton* of a rational expression  $E$  is the finite automaton  $\mathcal{A}_E$  whose set of states is  $D(E)$  and whose transitions are defined by:

- (i) if  $K$  and  $K'$  are derived terms of  $E$  and  $a$  a letter of  $A$ , then  $(K, a, K')$  is a transition if, and only if,  $K'$  belongs to  $\frac{\partial}{\partial a} K$ ;
- (ii) the initial state is  $E$ ;
- (iii) a derived term  $K$  is final if, and only if,  $c(K) = 1$ .

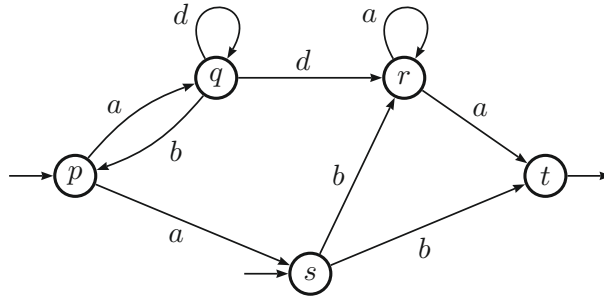
It is well-known that  $\mathcal{A}_E$  has at most  $\ell(E) + 1$  states and accepts the language denoted by  $E$  (cf. [2]). We also denote  $\mathcal{A}_E$  as  $A(E)$  in order to have a notation which is parallel to the one used for the broken derived term automaton  $\Delta'(E)$  (see below); this notation is used in the proof of Proposition 4.7.

### 2.2.2. Breaking derivation

The essence of the derivation (by a letter), as defined by Antimirov and repeated here, is to “break” the expression into pieces when the operator at the upper level of the expression is “+”. The modification of the derivation we consider now consists in supposing that this breaking happens *spontaneously*, as if it were a derivation with respect to the empty word, before the first derivation by a letter, and after every such derivation. To that end, we define a new operation on rational expressions which we denote by  $B()$  and which, roughly speaking, consists in

---

<sup>3</sup>We would rather not say that  $E$  is the result of a derivation by the empty word as the derivation of a sum should be the sum (here, the union) of the derivation of the summands. And this is what will happen with the breaking derivation that we define below.

FIGURE 2. The co-deterministic automaton  $\mathcal{A}_2$ .

decomposing an expression into a set of expressions whose left factor is not a sum<sup>4</sup>.

The definition of  $B()$  requires two further notations. If  $X$  is a *set of expressions*, then  $[X]_p$  is the same set possibly without the expression 1:  $[X]_p = X \setminus \{1\}$ , and  $\delta_X$  is the Boolean that takes the value 1 if the expression 1 belongs to  $X$  and 0 otherwise. For instance,  $\delta_{[X]_p} = 0$  and  $X = [X]_p \cup \delta_X \{1\}$  for any set  $X$ .

**Definition 2.11.** Let  $E$  be an expression. Then,  $B(E)$  is the set of expressions inductively defined as follows:

$$\begin{aligned} B(0) &= \{0\}, & B(1) &= \{1\}, & \forall a \in A & B(a) = \{a\}, \\ B(E + F) &= B(E) \cup B(F), & B(E \cdot F) &= [B(E)]_p \cdot F \cup \delta_{B(E)} B(F), & B(E^*) &= \{E^*\}. \end{aligned}$$

The set  $B(E)$  is called the set of *broken terms* of  $E$ .

As noted in (2), the breaking operator  $B$  is also additive over sets of rational expressions:

$$\forall X \subseteq \text{RatE}(A) \quad B(X) = \bigcup_{E \in X} B(E).$$

Moreover, the breaking operator is *idempotent*:  $B(B(E)) = B(E)$  for any expression  $E$ .

**Example 2.12.** Let  $\mathcal{A}_2$  be the co-deterministic automaton of Figure 2.

The expression  $E_2$  computed by the state elimination method with the ordering  $q \leq r \leq p \leq s \leq t$  is:

$$E_2 = (ad^*b)^*ad^*da^*a + (1 + (ad^*b)^*a)(b + ba^*a).$$

Then,  $B(E_2) = \{(ad^*b)^*ad^*da^*a, (ad^*b)^*a(b + ba^*a), b, ba^*a\}$ .

<sup>4</sup>Here lies the correction we make to our original paper. The definition of the derivation we apply to the expressions is modified in two ways: first, the breaking  $B()$  is more ‘complete’ as it ‘goes through’ sets of expressions which contain 1; second, we replace the derivation by the breaking derivation (in the original paper, the breaking  $B()$  was applied only once before any derivation).



**Definition 2.13.** The *breaking derivation* of a rational expression  $E$  over  $A$  with respect to a letter  $a$  in  $A$  is defined by:

$$\frac{\partial_b}{\partial a} E = B\left(\left[\frac{\partial}{\partial a} E\right]\right).$$

The breaking derivation with respect to words is then defined by induction on the length of words as in (6):

$$\forall u \in A^+, \forall a \in A \quad \frac{\partial_b}{\partial ua} E = \frac{\partial_b}{\partial a} \left( \frac{\partial_b}{\partial u} E \right).$$

We call *true broken derived term* of  $E$  any rational expression which belongs to a set  $\frac{\partial_b}{\partial u} E$  for some non-empty word  $u$  in  $A^+$ .

We write  $\text{TBD}(E)$  for the set of true broken derived terms of  $E$  and the set of *broken derived terms*,  $\text{BD}(E)$ , is defined by:

$$\text{BD}(E) = \text{TBD}(E) \cup B(E).$$

It is easy to check that for every rational expression  $E$  over  $A$

$$\forall a \in A \quad \frac{\partial}{\partial a} B(E) = \frac{\partial}{\partial a} E, \quad \text{and thus} \quad \forall u \in A^+ \quad \frac{\partial_b}{\partial u} E = B\left(\left[\frac{\partial}{\partial u} E\right]\right),$$

which in turn implies that the broken derived terms are obtained by ‘breaking’ the derived terms:

$$\text{BD}(E) = \bigcup_{K \in D(E)} B(K). \quad (8)$$

As for the derivation, one of the purposes of the breaking derivation, the one which is of interest here, is the definition of an automaton.

**Definition 2.14.** The *broken derived term automaton* of a rational expression  $E$  over  $A$  is the finite automaton  $\Delta'(E)$  whose set of states is  $\text{BD}(E)$  and whose transitions are defined by:

- (i) if  $K$  and  $K'$  are broken derived terms of  $E$  and  $a$  a letter of  $A$ , then  $(K, a, K')$  is a transition if, and only if,  $K'$  belongs to  $\frac{\partial_b}{\partial a} K$ ;
- (ii) the set of initial states is  $B(E)$ ;
- (iii) a broken derived term  $K$  is final if, and only if,  $c(K) = 1$ .

The automaton  $\Delta'(E)$  accepts the language  $|E|$  (cf. [7], Thm. 9)<sup>5</sup>. More precisely, the following holds.

**Property 2.15.** For every  $K$  in  $\text{BD}(E)$ , the future of  $K$  in  $\Delta'(E)$  is equal to  $|K|$ .

<sup>5</sup>We use that notation  $\Delta'()$  here, in this corrigendum, in order to have a different notation from the one used in [8]; it might be the case that in further publications, we shall use  $\Delta()$  which is more natural. The automaton  $\Delta'(E)$  has a number of states which is bounded in function of  $\ell(E)$  as for  $\mathcal{A}_E$ , but the bound is more tricky to establish (cf. [1]).

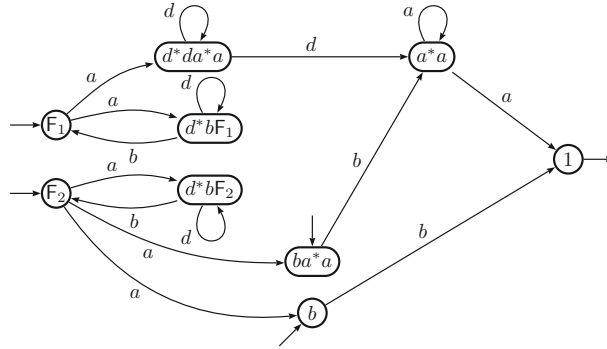


FIGURE 3. The broken derived term automaton of  $E_2$ .

**Example 2.16** (Ex. 2.12 cont.). We have computed  $B(E_2) = \{F_1, F_2, b, ba^*a\}$ , with  $F_1 = (ad^*b)^*ad^*da^*a$  and  $F_2 = (ad^*b)^*a(b + ba^*a)$ . The following breaking derivations give rise to the true broken derived terms of  $E_2$ .

$$\begin{aligned} \frac{\partial_b}{\partial a} F_1 &= \{d^*bF_1, d^*da^*a\} & \frac{\partial_b}{\partial a} F_2 &= \{d^*bF_2, b, ba^*a\} & \frac{\partial_b}{\partial d} d^*da^*a &= \{d^*da^*a, a^*a\} \\ \frac{\partial_b}{\partial d} d^*bF_1 &= d^*bF_1 & \frac{\partial_b}{\partial d} d^*da^*a &= \{d^*da^*a, a^*a\} & \frac{\partial_b}{\partial b} ba^*a &= a^*a. \end{aligned}$$

The broken derived term automaton of  $E_2$  is shown at Figure 3. Hopefully, it is co-deterministic and its minimal co-quotient is isomorphic to  $\mathcal{A}_2$ .

We are now ready to state the key result of this work.

**Theorem 1.1.** *Let  $\mathcal{A}$  be a co-deterministic automaton and  $E = \Phi(\mathcal{A})$  a rational expression computed from  $\mathcal{A}$  by the state elimination method. Then, the broken derived term automaton  $\Delta'(E)$  of  $E$  is co-deterministic.*

In [8], we have given a number of examples where the mapping  $\Upsilon \circ \Delta'$  applied to an expression that is computed from a co-minimal automaton  $\mathcal{A}$  yields  $\mathcal{A}$  itself. But the only case where we have been able to prove the property is when  $\mathcal{A}$  is co-deterministic. Moreover, this result gives the possibility of dealing with the general case by means of tagging. This is the reason why we consider Theorem 1.1 as the key statement in our search for an algorithm  $\Omega$  which would be the inverse of  $\Phi$ <sup>6</sup>.

The introduction of the breaking operation is necessary to deal with automata with several initial states. Nevertheless, the core of the proof of Theorem 1.1 is Proposition 4.7 which establishes that the derived term automaton of an expression computed from a *normalised* co-deterministic automaton is co-deterministic, that is, a property of Antimirov derivation (definition of normalised automata is recalled

<sup>6</sup>In [8], the corresponding statement is Theorem 3.5, and Theorem 1.1 is the corrected version of it.

below). A remarkable feature of this result is that its proof goes by induction on the number of states of the normalised automaton we start from. The main ingredients of this proof are first Lemma 3.1 that describes how the (Antimirov) derivation translates under a *continuous rational substitution* and then Proposition 3.9 which states that every (broken) derived term is contained in the *future* of some state of the automaton on which the expression has been computed. The transfer from the hypothesis of Proposition 4.7 (normalised co-deterministic automata) to the one of Theorem 1.1 (co-deterministic automata) is made possible by Lemma 3.8.

### 3. PRELIMINARY TO THE PROOF OF THEOREM 1.1

We first establish two properties of derivation that will be used in the sequel and whose scope is wider than the statement alone. For the ease of the proof, we prefer to work with *normalised* automata. We then show, in two steps, that we can make this assumption without loss of generality.

#### 3.1. A SUBSTITUTION LEMMA

Let  $A$  and  $B$  be two alphabets. We call *rational substitution* a map

$$\varphi: B \rightarrow \text{RatE}(A) \quad \text{which is extended to} \quad \varphi: \text{RatE}(B) \rightarrow \text{RatE}(A)$$

by replacing every atom of a rational expression over  $B$  by its image under  $\varphi$ . Substitutions are consistent with rational operations in the sense that for every expressions  $E$  and  $F$  in  $\text{RatE}(B)$ , the following equalities hold:

$$\varphi(E + F) = \varphi(E) + \varphi(F), \quad \varphi(E \cdot F) = \varphi(E) \cdot \varphi(F) \quad \text{and} \quad \varphi((E)^*) = (\varphi(E))^*.$$

Naturally,  $\varphi$  also induces a substitution from  $B^*$  into  $A^*$  and the equality  $|\varphi(E)| = \varphi(|E|)$  holds. A substitution  $\varphi$  is *continuous* if  $c(\varphi(b))$  is null for every  $b$  in  $B$ . If  $\varphi$  is *continuous*, then, for every expression  $E$  in  $\text{RatE}(B)$ , we have:

$$c(\varphi(E)) = c(E).$$

The following lemma states that the derivation behaves with respect to continuous rational substitutions exactly as it is expected from an operator called derivation. (Note also this lemma deals with the *derivation* and not with the breaking derivation. It turns out that we shall make use of the two notions in parallel.)

**Lemma 3.1.** *Let  $\varphi: \text{RatE}(B) \rightarrow \text{RatE}(A)$  be a continuous rational substitution. Then*

$$\forall a \in A \quad \frac{\partial}{\partial a} \varphi(E) = \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} E \right). \quad (9)$$

*Proof.* By induction on the depth of  $E$ . If  $E = 0$  or  $1$ , both sides of (9) are zero. If  $E = b$ , the right-hand side of (9) reduces to  $\frac{\partial}{\partial a} \varphi(b)$  which is the left-hand side, and the base of the induction is established. The following three sequences of equalities give the three possible induction steps:

$$\begin{aligned}
\frac{\partial}{\partial a} \varphi(F + G) &= \frac{\partial}{\partial a} [\varphi(F) + \varphi(G)] = \frac{\partial}{\partial a} \varphi(F) \cup \frac{\partial}{\partial a} \varphi(G) \\
&= \left[ \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} F \right) \right] \cup \left[ \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} G \right) \right] \\
&= \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \left[ \varphi \left( \frac{\partial}{\partial b} F \right) \cup \varphi \left( \frac{\partial}{\partial b} G \right) \right] \\
&= \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} (F + G) \right) \\
\\
\frac{\partial}{\partial a} \varphi(F \cdot G) &= \frac{\partial}{\partial a} [\varphi(F) \cdot \varphi(G)] = \left[ \frac{\partial}{\partial a} \varphi(F) \right] \cdot \varphi(G) \cup c(\varphi(F)) \frac{\partial}{\partial a} \varphi(G) \\
&= \left[ \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} F \right) \right] \cdot \varphi(G) \cup c(F) \left[ \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} G \right) \right] \\
&= \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} (F \cdot G) \right) \\
\\
\frac{\partial}{\partial a} \varphi(F^*) &= \frac{\partial}{\partial a} (\varphi(F))^* = \left[ \frac{\partial}{\partial a} \varphi(F) \right] \cdot (\varphi(F))^* \\
&= \left[ \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} F \right) \right] \cdot (\varphi(F))^* \\
&= \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \left[ \varphi \left( \frac{\partial}{\partial b} F \right) \cdot (\varphi(F))^* \right] = \bigcup_{b \in B} \left[ \frac{\partial}{\partial a} \varphi(b) \right] \varphi \left( \frac{\partial}{\partial b} F^* \right).
\end{aligned}$$

□

### 3.2. NORMALISED AUTOMATA

We define standard, co-standard, and normalised automata. Indeed, with every automaton we associate a standard, a co-standard, and finally a normalised automaton. In contrast to what is usually done with such constructions, these new automata are not exactly equivalent to the one from which they are derived, but in each case, the added transitions bear a new letter that keeps track of the transformation.

**Definition 3.2.** An automaton is *standard* if it has exactly one initial state with no incoming transition.

With any automaton  $\mathcal{A} = \langle Q, A, E, I, T \rangle$ , we associate the standard automaton  $\mathcal{A}_\triangleleft = \langle Q \cup \{i\}, A \cup \{\$, \}, F, \{i\}, T \rangle$ , where  $i$  does not belong to  $Q$ , nor  $\$$  to  $A$ , and where

$$F = E \cup \{(i, \$, p) \mid p \in I\}.$$

Note that  $\mathcal{A}$  is co-deterministic if, and only if,  $\mathcal{A}_\triangleleft$  is. Clearly  $|\mathcal{A}_\triangleleft| = \$|\mathcal{A}|$ . This equality indeed generalises to expressions and their broken derived terms in the following way (when the state elimination method is applied to  $\mathcal{A}_\triangleleft$ , it is understood that  $i$  will be the last state to be eliminated).

**Lemma 3.3.** *Let  $\mathcal{A}$  be an automaton over  $A$ ,  $\mathcal{A}_\triangleleft$  the standard automaton defined as above,  $E = \Phi_\omega(\mathcal{A})$  and  $E_\triangleleft = \Phi_\omega(\mathcal{A}_\triangleleft)$  the expressions obtained by the state elimination method (with respect to the same order  $\omega$  on the states of  $\mathcal{A}$ ). Then we have:*

$$B(E) = \frac{\partial_b}{\partial \$} E_\triangleleft \quad \text{and} \quad BD(E) = TBD(E_\triangleleft).$$

**Definition 3.4.** An automaton is *co-standard* if it has exactly one final state with no outgoing transition, that is, if its transpose is a standard automaton.

With any automaton  $\mathcal{A} = \langle Q, A, E, I, T \rangle$ , we associate the co-standard automaton  $\mathcal{A}_\triangleright = \langle Q \cup \{t\}, A \cup \{\mathcal{L}\}, F, I, \{t\} \rangle$ , where  $t$  does not belong to  $Q$ , nor  $\mathcal{L}$  to  $A$ , and where

$$F = E \cup \{(p, \mathcal{L}, t) \mid p \in T\}.$$

Note that  $\mathcal{A}$  is co-deterministic if, and only if,  $\mathcal{A}_\triangleright$  is. Clearly  $|\mathcal{A}_\triangleright| = |\mathcal{A}| \mathcal{L}$ . We first consider more closely expressions for which such an equality holds. Let  $\mathcal{F}_\mathcal{L}$  be the family of expressions in  $\text{RatE}(A \cup \mathcal{L})$  that denote rational subsets of  $A^*$  followed by  $\mathcal{L}$ :

$$\mathcal{F}_\mathcal{L} = \{E \in \text{RatE}(A \cup \mathcal{L}) \mid |E| = L \mathcal{L} \quad L \in \text{Rat } A^*\}.$$

Since any word in the language denoted by an element of  $\mathcal{F}_\mathcal{L}$  does not contain the symbol  $\mathcal{L}$  but at the end, the following property holds.

**Property 3.5.** Any element of  $\mathcal{F}_\mathcal{L}$  is of one of the following forms:  $\mathcal{L}$ ,  $E_\mathcal{L} + F_\mathcal{L}$ , or  $G \cdot F_\mathcal{L}$ , where  $E_\mathcal{L}$  and  $F_\mathcal{L}$  are in  $\mathcal{F}_\mathcal{L}$  and  $G$  is in  $\text{RatE}(A)$ .

The notions of standard and co-standard automata are dual, but as the derivation is also a directed notion, the statement corresponding to Lemma 3.3 is not exactly its dual.

**Lemma 3.6.** *Let  $\mathcal{A}$  be an automaton over  $A$ ,  $\mathcal{A}_\triangleright$  the co-standard automaton defined as above,  $E = \Phi_\omega(\mathcal{A})$  and  $E_\triangleright = \Phi_\omega(\mathcal{A}_\triangleright)$ .*

*The projection  $\pi: (A \cup \{\mathcal{L}\})^* \rightarrow A^*$  induces a bijection between  $BD(E_\triangleright) \setminus \{1\}$  and  $BD(E)$  and in particular between  $B(E_\triangleright)$  and  $B(E)$ .*

*Proof.* As the algorithm  $\Phi$  acts symbolically on the labels of the transitions,  $\pi(\mathbf{E}_{\triangleright}) = \mathbf{E}$ . The expression  $\mathbf{E}_{\triangleright}$  belongs to  $\mathcal{F}_{\mathcal{L}}$ . By induction on the depth of the expression, we first show that  $\pi(\mathbf{B}(\mathbf{E}_{\triangleright})) = \mathbf{B}(\pi(\mathbf{E}_{\triangleright})) = \mathbf{B}(\mathbf{E})$ . The interesting case is when  $\mathbf{E}_{\triangleright} = \mathbf{G} \cdot \mathbf{F}_{\mathcal{L}}$ :

$$\pi(\mathbf{B}(\mathbf{G} \cdot \mathbf{F}_{\mathcal{L}})) = \pi(\mathbf{B}(\mathbf{G}) \cdot \mathbf{F}_{\mathcal{L}}) \cup \delta_{\mathbf{B}(\mathbf{G})} \cdot \pi(\mathbf{B}(\mathbf{F}_{\mathcal{L}})) = \mathbf{B}(\mathbf{G}) \cdot \mathbf{F} \cup \delta_{\mathbf{B}(\mathbf{G})} \mathbf{B}(\mathbf{F}) = \mathbf{B}(\mathbf{G} \cdot \mathbf{F}).$$

In the same way, for any  $\mathbf{H}_{\mathcal{L}}$  in  $\mathcal{F}_{\mathcal{L}}$ , and for any letter  $a$  in  $A$ , we have:

$$\pi\left(\frac{\partial_{\mathbf{b}}}{\partial a} \mathbf{H}_{\mathcal{L}}\right) = \frac{\partial_{\mathbf{b}}}{\partial a} \pi\left(\mathbf{H}_{\mathcal{L}}\right). \quad (10)$$

The statement is then shown by induction on the ‘length’ of the derivation. Every (broken) derived term of  $\mathbf{E}_{\triangleright}$  different from 1 is in  $\mathcal{F}_{\mathcal{L}}$ . Let  $\mathbf{K}_{\mathcal{L}}$  be a derived term in  $\text{BD}(\mathbf{E}_{\triangleright})$  and  $\mathbf{K}$  its projection, in  $\text{BD}(\mathbf{E})$  by induction hypothesis. The derivation of  $\mathbf{K}_{\mathcal{L}}$  with respect to  $\mathcal{L}$  gives either 1, or the empty set. The derivation of  $\mathbf{K}_{\mathcal{L}}$  with respect to any letter  $a$  in  $A$  is, by (10), in bijection with the derivation of  $\mathbf{K}$  with respect to  $a$ .  $\square$

Putting the notions of standard and co-standard automata together, we get the following definition and statement that are even more classical than the previous ones (*cf.* [5] for instance).

**Definition 3.7.** An automaton is *normalised* if, and only if, it has exactly one initial state and one final state such that there is no incoming transition on the initial state and no outgoing transition from the final state<sup>7</sup>.

With any automaton  $\mathcal{A} = \langle Q, A, E, I, T \rangle$ , we associate the normalised automaton  $\mathcal{A}_{\diamond} = \langle Q \cup \{i, t\}, A \cup \{\$, \mathcal{L}\}, F, \{i\}, \{t\} \rangle$ , where neither  $i$  nor  $t$  belongs to  $Q$ , neither  $\$$  nor  $\mathcal{L}$  belongs to  $A$ , and where

$$F = E \cup \{(i, \$, p) \mid p \in I\} \cup \{(p, \mathcal{L}, t) \mid p \in T\}.$$

Note that  $\mathcal{A}$  is co-deterministic if, and only if,  $\mathcal{A}_{\diamond}$  is. Clearly  $|\mathcal{A}_{\diamond}| = \$|\mathcal{A}|\mathcal{L}$ . And both Lemmas 3.3 and 3.6 are put together in the following statement.

**Lemma 3.8.** *Let  $\mathcal{A}$  be an automaton over  $A$ ,  $\mathcal{A}_{\diamond}$  the normalised automaton defined as above,  $\mathbf{E} = \Phi_{\omega}(\mathcal{A})$  and  $\mathbf{E}_{\diamond} = \Phi_{\omega}(\mathcal{A}_{\diamond})$ .*

*The projection  $\pi: (A \cup \{\mathcal{L}\})^* \longrightarrow A^*$  induces a bijection between  $\frac{\partial_{\mathbf{b}}}{\partial \$} \mathbf{E}_{\diamond}$  and  $\mathbf{B}(\mathbf{E})$  on one hand and between  $\text{TBD}(\mathbf{E}_{\diamond}) \setminus \{1\}$  and  $\text{BD}(\mathbf{E})$  on the other hand.  $\square$*

---

<sup>7</sup>In spite of its name, a normalised automaton is *not necessarily* the result of the normalisation of another automaton.

## 3.3. FUTURE OF STATES AND DERIVED TERMS

We end these preliminaries by putting in relation the derived terms and the broken derived terms of an expression obtained from an automaton by the state elimination method and the future of the states of that automaton. In the next three statements,  $\mathcal{A}$  is an automaton over  $A$  with set of states  $Q$  and  $E = \Phi(\mathcal{A})$  is an expression computed from  $\mathcal{A}$  by the state elimination method. For clear typography, we write  $a_q$  instead of  $(a, q)$  for the elements of  $A \times Q$ .

**Proposition 3.9.** *If  $\mathcal{A}$  is standard, then for every derived term  $K$  of  $E$  there exists a state  $q$  in  $Q$  such that  $|K|$  is included in the future of  $q$ .*

*Proof.* Let  $\widehat{\mathcal{A}} = \langle Q, A \times Q, \widehat{E}, I, T \rangle$  be obtained from  $\mathcal{A} = \langle Q, A, E, I, T \rangle$  by relabelling every transition  $(p, a, q)$  in  $E$  as  $(p, a_q, q)$ . The projection  $\pi$  maps  $A \times Q$  onto  $A$  and is extended to a map from  $\text{RatE}(A \times Q)$  onto  $\text{RatE}(A)$ .

As  $\pi$  is a continuous (rational) substitution, with the property that for any  $a, b$  in  $A$ , (and any  $q$  in  $Q$ ),  $\frac{\partial}{\partial a} \pi(b_q)$  is equal to 1 or 0 according to whether  $a$  is equal to  $b$  or not, Lemma 3.1 implies that for any  $a$  in  $A$  and for any expression  $F$  in  $\text{RatE}(A \times Q)$  we have:

$$\frac{\partial}{\partial a} \pi(F) = \bigcup_{q \in Q} \left[ \pi \left( \frac{\partial}{\partial a_q} F \right) \right]. \quad (11)$$

Let  $\widehat{E} = \Phi(\widehat{\mathcal{A}})$ . As the algorithm  $\Phi$  acts symbolically on the labels of the transitions,  $\pi(\widehat{E}) = E$  holds. Then, by iteration of (11), for any  $K$  in  $D(E)$  there exists  $K'$  in  $D(\widehat{E})$  such that  $K = \pi(K')$ .

Either  $K' = \widehat{E}$ , then  $K = E$  and  $|K'|$  is (in) the future of the (unique) initial state. Or there exists a non empty word  $w$  in  $(A \times Q)^+$  such that  $K'$  is in  $\frac{\partial}{\partial w} \widehat{E}$  and then

$$|K'| \subseteq \left| \frac{\partial}{\partial w} \widehat{E} \right| = w^{-1} |\widehat{\mathcal{A}}|.$$

Let  $a_q$  be the last letter of  $w$ , then  $w^{-1} |\widehat{\mathcal{A}}|$  is contained in  $\text{Fut}_{\widehat{\mathcal{A}}}(q)$ . The projection by  $\pi$  gives

$$\text{Fut}_{\mathcal{A}}(q) = \pi(\text{Fut}_{\widehat{\mathcal{A}}}(q)) \quad \text{and thus} \quad |K| \subseteq \text{Fut}_{\mathcal{A}}(q). \quad \square$$

The same property holds then for broken derived terms.

**Corollary 3.10.** *Without further hypothesis on  $\mathcal{A}$ , for every broken derived term  $K$  of  $E$  there exists a state  $q$  in  $Q$  such that  $|K|$  is included in the future of  $q$ .*

*Proof.* Let  $\mathcal{A}_{\triangleleft}$  be the standard automaton associated with  $\mathcal{A}$  as in Section 3.2 and let  $E_{\triangleleft} = \Phi_{\omega}(\mathcal{A}_{\triangleleft})$ . For every  $K_{\triangleleft}$  in  $\text{BD}(E_{\triangleleft})$  there exists  $H_{\triangleleft}$  in  $D(E_{\triangleleft})$  such that  $K_{\triangleleft}$  is in  $B(H_{\triangleleft})$  (cf. (8)) and thus  $|K| \subseteq |H|$ . Moreover, there exists, by Proposition 3.9, a state  $q$  in  $Q$  such that  $|H| \subseteq \text{Fut}_{\mathcal{A}_{\triangleleft}}(q)$  hence  $|K_{\triangleleft}| \subseteq \text{Fut}_{\mathcal{A}_{\triangleleft}}(q)$ . By Lemma 3.3,  $\text{TBD}(E_{\triangleleft}) = \text{BD}(E)$ , therefore  $|K| \subseteq \text{Fut}_{\mathcal{A}}(q)$ .  $\square$

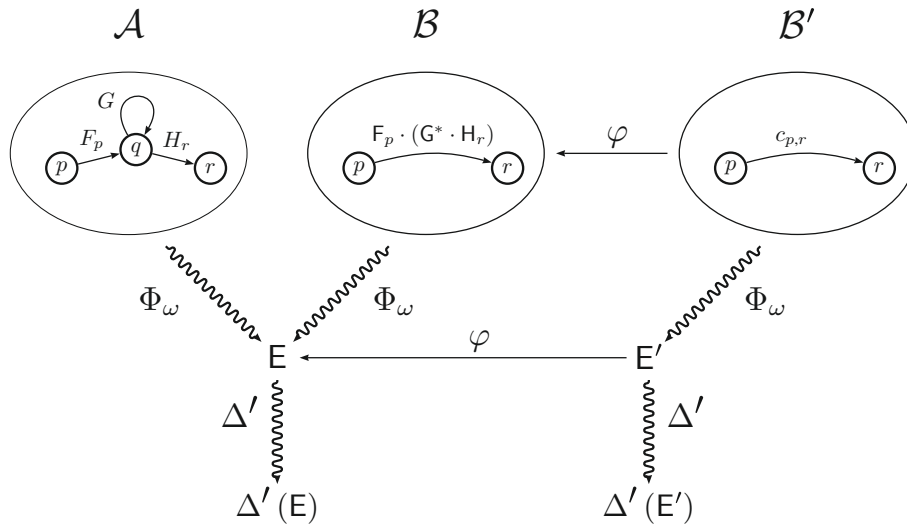


FIGURE 4. Diagram for an induction step in the proof of Theorem 1.1.

As the futures of states are pairwise disjoint in a co-deterministic automaton (Prop. 2.3), we then can state the following.

**Corollary 3.11.** *If  $\mathcal{A}$  is a co-deterministic normalised automaton, then for every derived term (resp. every broken derived term)  $K$  of  $D(E)$ , there exists a unique  $q$  in  $Q$  such that  $|K| \subseteq \text{Fut}_{\mathcal{A}}(q)$ .*

#### 4. PROOF OF THEOREM 1.1

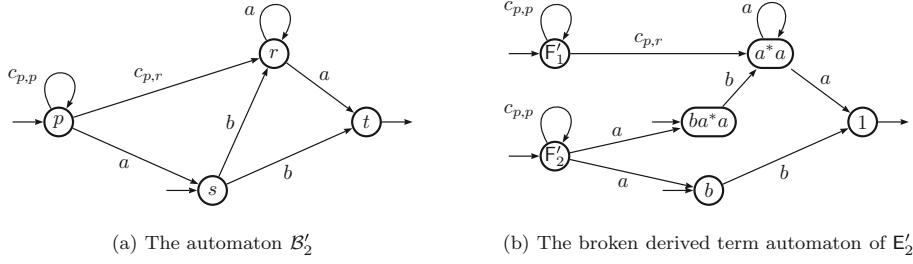
The proof goes by induction on the number of states of the automaton  $\mathcal{A}$  as sketched in Figure 4.

Let  $E = \Phi_{\omega}(\mathcal{A})$  the rational expression computed by the state elimination method following the order  $\omega$  on the states of  $\mathcal{A}$  and let  $q$  be the smallest state with respect to  $\omega$ . For every predecessor  $p$  of  $q$ , let  $F_p$  be the set of letters  $a$  such that  $(p, a, q)$  is a transition of  $\mathcal{A}$  and  $F_p$  the expression ‘sum of letters in  $F_p$ ’. Let  $R$  be the set of successors of  $q$  in  $\mathcal{A}$ ; for every  $r$  in  $R$ , let  $H_r$  be the set of letters  $a$  such that  $(q, a, r)$  is a transition of  $\mathcal{A}$  and  $H_r$  the expression ‘sum of letters in  $H_r$ ’. Let  $G$  be the set of letters  $a$  such that  $(q, a, q)$  is a transition of  $\mathcal{A}$  and  $G$  the expression ‘sum of letters in  $G$ ’. The set  $G$  may be empty and then  $G = 0$ ; the trivial identities  $(\mathbf{T})$  allows to treat this special case in the same way as the general one in almost every instance.

Let  $\mathcal{B}$  be the *generalised* automaton obtained from  $\mathcal{A}$  by elimination of  $q$  – by the state elimination method. For every  $p$  and every  $r$  as above, a transition from  $p$  to  $r$  labelled by  $F_p \cdot (G^* \cdot H_r)$  is thus created<sup>8</sup>. Let  $\mathcal{B}'$  be the (normalized)

<sup>8</sup>The automaton  $\mathcal{B}$  is called ‘generalised’ because some of its labels are rational expressions.



FIGURE 5. The first step first step of the state elimination method on  $\mathcal{A}_2$ .

automaton obtained from  $\mathcal{B}$  by replacing every label of the form  $F_p \cdot (G^* \cdot H_r)$  by a fresh letter  $c_{p,r}$ . Let  $C$  be the alphabet of these fresh letters and  $A' = A \cup C$ . Let  $\varphi$  be the (continuous) substitution from  $\text{RatE}(A')$  to  $\text{RatE}(A)$  which maps every letter of  $A$  onto itself and every letter  $c_{p,r}$  onto  $F_p \cdot (G^* \cdot H_r)$ .

Since the construction of  $\mathcal{B}$  is the first step of  $\Phi_\omega(\cdot)$ , then  $E = \Phi_\omega(\mathcal{B})$  and let  $E' = \Phi_\omega(\mathcal{B}')$ . As the algorithm  $\Phi_\omega$  acts symbolically on the labels of the transitions,  $\varphi(E') = E$ .

If  $\mathcal{A}$  is co-deterministic, so is  $\mathcal{B}'$  and  $\Delta'(E')$  is co-deterministic by induction hypothesis. The proof amounts to transferring the properties from  $\Delta'(E')$  to  $\Delta'(E)$  via the substitution  $\varphi$ .

**Example 4.1.** [Ex. 2.16 cont.] The first step of the state elimination method, followed by relabelling of automaton  $\mathcal{A}_2$  is the automaton  $\mathcal{B}'_2$  of Figure 5a, with:

$$\varphi(c_{p,p}) = ad^*b \quad \text{and} \quad \varphi(c_{p,r}) = ad^*d.$$

The expression  $E'_2$  computed by the state elimination method with the ordering  $r \leq p \leq s \leq t$  is:

$$E'_2 = c_{p,p}^* c_{p,r} a^* a + (1 + c_{p,p}^* a)(b + ba^* a),$$

and  $\varphi(E'_2) = E_2$  holds. Then,  $B(E'_2) = \{c_{p,p}^* c_{p,r} a^* a, c_{p,p}^* a(b + ba^* a), b, ba^* a\}$  and the broken derived term automaton of  $E'_2$  is shown at Figure 5b, with the notation  $F'_1 = c_{p,p}^* c_{p,r} a^* a$  and  $F'_2 = c_{p,p}^* a(b + ba^* a)$ .

#### 4.1. PREPARATION FOR THE INDUCTION

From now on – and until after Corollary 4.9 –  $\mathcal{A}$ , and thus  $\mathcal{B}$  and  $\mathcal{B}'$ , are *normalised* automata and the state  $q$  is neither the initial nor the final state. With the above notation, we have:

**Property 4.2.** For every state  $s$ ,  $s \neq q$ ,  $\text{Fut}_{\mathcal{A}}(s) = \varphi(\text{Fut}_{\mathcal{B}'}(s))$ .

The remainder of this subsection is devoted to the description of the derived terms and broken derived terms of  $\mathbf{E}$ , that is, the elements of  $\mathbf{D}(\mathbf{E})$  and  $\mathbf{BD}(\mathbf{E})$ , in terms of the elements of  $\mathbf{D}(\mathbf{E}')$ .

From Lemma 3.1 follows, for every  $\mathbf{K}'$  in  $\mathbf{RatE}(A')$ , and every letter  $a$  in  $A$ :

$$\frac{\partial}{\partial a} \varphi(\mathbf{K}') = \varphi \left( \frac{\partial}{\partial a} \mathbf{K}' \right) \cup \bigcup_{c \in C} \left[ \frac{\partial}{\partial a} \varphi(c) \right] \varphi \left( \frac{\partial}{\partial c} \mathbf{K}' \right).$$

For every  $c$  in  $C$ , there exists  $p$  and  $r$  such that  $\varphi(c) = F_p \cdot (\mathbf{G}^* \cdot H_r)$ . Hence, either  $a$  does not belong to any  $F_p$  and  $\frac{\partial}{\partial a} \varphi(\mathbf{K}') = \varphi \left( \frac{\partial}{\partial a} \mathbf{K}' \right)$  or it belongs to  $F_p$  for a unique  $p$ , since  $\mathcal{A}$  is co-deterministic, and (recall that  $R$  is the set of successors of  $q$  in  $\mathcal{A}$ ):

$$\frac{\partial}{\partial a} \varphi(\mathbf{K}') = \varphi \left( \frac{\partial}{\partial a} \mathbf{K}' \right) \cup \left[ \bigcup_{r \in R} \mathbf{G}^* \cdot H_r \varphi \left( \frac{\partial}{\partial c_{p,r}} \mathbf{K}' \right) \right]. \quad (12)$$

For every  $c$  in  $C$ , and as  $\mathbf{G}$  and  $H_r$  are sums of letters, the derivation of  $\mathbf{G}^* \cdot H_r \varphi \left( \frac{\partial}{\partial c} \mathbf{K}' \right)$  with respect to any letter  $x$  of  $A'$  gives  $\mathbf{G}^* \cdot H_r \varphi \left( \frac{\partial}{\partial c} \mathbf{K}' \right)$  if  $x$  is in  $G$ ,  $\varphi \left( \frac{\partial}{\partial c} \mathbf{K}' \right)$  if  $x$  is in  $H_r$ , both sets of expressions if  $x$  is in  $G \cap H_r$ , and  $\emptyset$  in any other case. Hence, by a straightforward induction, we have:

$$\mathbf{D}(\mathbf{E}) = \varphi(\mathbf{D}(\mathbf{E}')) \cup \bigcup_{c_{p,r} \in C} \left[ \bigcup_{a \in F_p} \frac{\partial}{\partial a} \varphi(c_{p,r}) \right] \varphi \left( \frac{\partial}{\partial c_{p,r}} \mathbf{D}(\mathbf{E}') \right). \quad (13)$$

The broken derived terms are obtained by applying  $\mathbf{B}()$  to (12). Either the letter  $a$  does not belong to any  $F_p$  and  $\frac{\partial_b}{\partial a} \varphi(\mathbf{K}') = \mathbf{B} \left( \varphi \left( \frac{\partial}{\partial a} \mathbf{K}' \right) \right)$  holds or it belongs to  $F_p$  for a unique  $p$ , and we have:

$$\frac{\partial_b}{\partial a} \varphi(\mathbf{K}') = \begin{cases} \mathbf{B} \left( \varphi \left( \frac{\partial}{\partial a} \mathbf{K}' \right) \right) \cup \bigcup_{r \in R} \mathbf{G}^* H_r \varphi \left( \frac{\partial}{\partial c_{p,r}} \mathbf{K}' \right) & \text{if } \mathbf{G} \neq 0 \\ \mathbf{B} \left( \varphi \left( \frac{\partial}{\partial a} \mathbf{K}' \right) \right) \cup \bigcup_{r \in R} \bigcup_{b \in H_r} b \varphi \left( \frac{\partial}{\partial c_{p,r}} \mathbf{K}' \right) & \text{otherwise.} \end{cases} \quad (14)$$

As above, and for every  $c$  in  $C$ , the *breaking* derivation of  $\mathbf{G}^* H_r \varphi \left( \frac{\partial}{\partial c} \mathbf{K}' \right)$  (*resp.* of  $b \varphi \left( \frac{\partial}{\partial c} \mathbf{K}' \right)$  with  $b$  in  $H_r$ ) with respect to any letter gives a *subset* of  $\left\{ \mathbf{G}^* H_r \varphi \left( \frac{\partial}{\partial c} \mathbf{K}' \right), \mathbf{B} \left( \varphi \left( \frac{\partial}{\partial c} \mathbf{K}' \right) \right) \right\}$ . Hence, by a straightforward induction,

we get the following equality:

$$\text{BD}(\mathbf{E}) = \text{B}(\varphi(\text{D}(\mathbf{E}')))) \cup \bigcup_{c_{p,r} \in C} \left[ \bigcup_{a \in F_p} \frac{\partial_{\mathbf{b}}}{\partial a} \varphi(c_{p,r}) \right] \varphi\left(\frac{\partial}{\partial c_{p,r}} \text{D}(\mathbf{E}')\right). \quad (15)$$

**Example 4.3.** [Ex. 4.1 cont.] Equation (14) applied to  $F_1 = c_{p,p}^* c_{p,r} a^*$  yields:

$$\begin{aligned} \frac{\partial_{\mathbf{b}}}{\partial a} \varphi(c_{p,p}^* c_{p,r} a^*) &= \text{B}\left(\varphi\left(\frac{\partial}{\partial a}(c_{p,p}^* c_{p,r} a^*)\right)\right) \cup d^* b \varphi\left(\frac{\partial}{\partial c_{p,p}}(c_{p,p}^* c_{p,r} a^*)\right) \\ &\quad \cup d^* d \varphi\left(\frac{\partial}{\partial c_{p,r}}(c_{p,p}^* c_{p,r} a^*)\right) \\ &= \{(ad^*b)^* ad^* da^* a, d^* b(ad^*b)^* ad^* da^* a, d^* da^* a\}. \end{aligned}$$

#### 4.2. PROOF OF THEOREM 1.1

The comparison between (13) and (15) is quite instructive. The presence in the right-hand side of both equations of a *non breaking derivation* makes it understandable that an induction proof will run more smoothly on the derived term automata. This is done with Proposition 4.7. On the other hand, when it comes to compare the derived terms of an expression obtained from an automaton  $\mathcal{A}$  and those obtained from its normalisation  $\mathcal{A}_\zeta$ , the *broken derived terms* are to be used (Lem. 3.8). Corollary 4.9 will fill the gap and Theorem 1.1 will follow.

We keep the notation of the preceding subsection and we suppose from now on that  $\mathcal{A}$  is (normalised) *co-deterministic*. This implies that  $\mathcal{B}'$  is co-deterministic and moreover that for every  $p$ ,  $G$  and  $F_p$  are disjoint; likewise, if  $p$  and  $p'$  are two distinct predecessors of  $q$ ,  $F_p$  and  $F_{p'}$  are disjoint. For every successor  $r$  of  $q$ , let  $I_r$  be the set of labels of incoming transitions of  $r$  that does not come from  $q$ ; then  $I_r$  and  $H_r$  are disjoint. For every state  $s$  of  $\mathcal{B}'$  (therefore different from  $q$ ), let  $J_s$  be the set of labels of incoming transitions of  $s$ . The following property then holds:

**Property 4.4.** For every state  $s$  in  $\mathcal{B}'$ ,  $\varphi(J_s)$  is a suffix code in  $A^*$  and the images of the elements of  $J_s$  by  $\varphi$  are pairwise disjoint.

**Lemma 4.5.** For every state  $s$  in  $\mathcal{A}$ ,  $s \neq q$ , and for every  $u$  in  $\text{Fut}_{\mathcal{A}}(s)$  there exists a unique  $v$  in  $\text{Fut}_{\mathcal{B}'}(s)$  such that  $u \in \varphi(v)$ .

*Proof.* Suppose by way of contradiction that there exist  $v$  and  $v'$  in  $\text{Fut}_{\mathcal{B}'}(s)$  such that  $u \in \varphi(v) \cap \varphi(v')$ . We write  $v = x_n \dots x_2 x_1$  and  $v' = x'_m \dots x'_2 x'_1$ . There exist  $w_i \in \varphi(x_i)$  and  $w'_i \in \varphi(x'_i)$  such that  $u = w_n \dots w_2 w_1 = w'_m \dots w'_2 w'_1$ .

Let  $j$  be the smallest index such that either  $w_j \neq w'_j$  or  $x_j \neq x'_j$ . Let  $r$  be the unique state of  $\mathcal{B}'$  such that  $x_{j-1} \dots x_1 \in \text{Fut}_{\mathcal{B}'}(r)$ . By Property 4.4,  $\varphi(J_r)$  is a suffix code and then  $w_j = w'_j$ , and  $\varphi(x_j) \cap \varphi(x'_j) \neq \emptyset$  implies  $x_j = x'_j$ .  $\square$

Let  $K$  be a derived term in  $\text{D}(\mathbf{E})$ . As  $\mathcal{A}$  is co-deterministic, and by Proposition 3.9, the analysis will be split into two cases:

either there exists a *unique* state  $s \neq q$  in  $\mathcal{A}$  such that  $|K| \subseteq \text{Fut}_{\mathcal{A}}(s)$  – **Case 1**,  
or  $|K| \subseteq \text{Fut}_{\mathcal{A}}(q)$  – **Case 2**.

**Lemma 4.6.** *Let  $K$  in  $D(E)$ . There exists  $K'$  in  $D(E')$  such that either:*  
**Case 1** *there exists  $s \neq q$  such that  $|K'| \subseteq \text{Fut}_{\mathcal{B}'}(s)$  and  $K = \varphi(K')$ , or*  
**Case 2** *there exists a successor  $r$  of  $q$  such that  $|K'| \subseteq \text{Fut}_{\mathcal{B}'}(r)$  and  $K = G^*H_r\varphi(K')$  or  $K = H_r\varphi(K')$  (if  $G$  is null).*

*Proof.* From (13),  $K$  is either (**Case 1**) in  $\varphi(D(E'))$  or (**Case 2**) in

$$\bigcup_{c_{p,r} \in C} \left[ \bigcup_{a \in F_p} \frac{\partial}{\partial a} \varphi(c_{p,r}) \right] \varphi(D_{c_{p,r}}(E')).$$

In **Case 1**, there exists  $K'$  such that  $K = \varphi(K')$ , by Proposition 3.9 there exists  $s$  in  $\mathcal{B}'$  such that  $|K'| \subseteq \text{Fut}_{\mathcal{B}'}(s)$  and by Property 4.2  $|K| \subseteq \text{Fut}_{\mathcal{A}}(s)$ .

In **Case 2**, there exist  $c_{p,r} \in C$ ,  $a \in A$  and  $K' \in D_{c_{p,r}}(E')$  such that

$$K = \left[ \frac{\partial}{\partial a} \varphi(c_{p,r}) \right] \varphi(K') = G^*H_r\varphi(K').$$

As  $K' \in D_{c_{p,r}}(E')$ ,  $|K'| \subseteq \text{Fut}_{\mathcal{B}'}(r)$ . By Property 4.2  $\varphi(|K'|) \subseteq \text{Fut}_{\mathcal{A}}(r)$  and thus, by definition of  $G$  and  $H_r$ ,  $|K| \subseteq \text{Fut}_{\mathcal{A}}(q)$ .  $\square$

**Proposition 4.7.** *Let  $\mathcal{A}$  be a normalised co-deterministic automaton, and let  $E = \Phi(\mathcal{A})$ . Then, the derived term automaton  $A(E)$  of  $E$  is co-deterministic.*

*Proof.* A normalised automaton has at least two states. If  $\mathcal{A}$  has only two states,  $\Phi(\mathcal{A})$  is reduced to a sum of letters and  $A(\Phi(\mathcal{A}))$  is clearly co-deterministic.

By induction,  $A(E')$  is co-deterministic, which means, by Property 2.15, that the interpretations of the derived terms of  $E'$  are pairwise disjoint. We prove now that the interpretations of elements of  $D(E)$  are disjoint, which implies that  $A(E)$  is co-deterministic.

Let  $K_1$  and  $K_2$  be two distinct derived terms of  $D(E)$  and assume that there exists  $u$  in  $|K_1| \cap |K_2|$ . By (13) and Lemma 4.6, there exist  $K'_1$  and  $K'_2$  in  $D(E')$  such that one of the following three cases holds:

**Case 1**  $K_1 = \varphi(K'_1)$  and  $K_2 = \varphi(K'_2)$ .

**Case 2.1**  $K_1 = \varphi(K'_1)$  and  $K_2 = G^*H_r\varphi(K'_2)$ .

**Case 2.2**  $K_1 = G^*H_{r_1}\varphi(K'_1)$  and  $K_2 = G^*H_{r_2}\varphi(K'_2)$ .

**Case 1** From Lemma 4.6 there exist states  $s_1$  and  $s_2$  such that  $|K'_1| \subseteq \text{Fut}_{\mathcal{B}'}(s_1)$  and  $|K'_2| \subseteq \text{Fut}_{\mathcal{B}'}(s_2)$ , and  $|K_1| \subseteq \text{Fut}_{\mathcal{A}}(s_1)$  and  $|K_2| \subseteq \text{Fut}_{\mathcal{A}}(s_2)$ . As  $\mathcal{A}$  is co-deterministic,  $s_1 = s_2 = s \neq q$  and by Lemma 4.5, there exists a unique  $v$  in  $\text{Fut}_{\mathcal{B}'}(s)$  such that  $u \in \varphi(v)$ . Hence  $v \in |K'_1| \cap |K'_2|$ . The induction hypothesis implies  $K'_1 = K'_2$  and thus  $K_1 = K_2$ .

**Case 2.1** From Lemma 4.6 there exist states  $s_1 \neq q$  such that  $|K_1| \subseteq \text{Fut}_{\mathcal{A}}(s_1)$  and  $|K_2| \subseteq \text{Fut}_{\mathcal{A}}(q)$ . As  $\mathcal{A}$  is co-deterministic,  $|K_1| \cap |K_2| = \emptyset$ .

**Case 2.2** From Lemma 4.6, there exist successors of  $q$ ,  $r_1$  and  $r_2$  such that  $|K'_1| \subseteq \text{Fut}_{\mathcal{B}'}(r_1)$  and  $|K'_2| \subseteq \text{Fut}_{\mathcal{B}'}(r_2)$ , and  $|K_1| \subseteq \text{Fut}_{\mathcal{A}}(q)$  and  $|K_2| \subseteq \text{Fut}_{\mathcal{A}}(q)$ . Hence  $u = g_1 h_1 w_1 = g_2 h_2 w_2$ , with  $g_1, g_2 \in G^*$ ,  $h_1 \in H_{r_1}$ ,  $h_2 \in H_{r_2}$ ,  $w_1 \in \varphi(|K'_1|)$  and  $w_2 \in \varphi(|K'_2|)$ . As  $G \cap H_{r_1} = G \cap H_{r_2} = \emptyset$ ,  $g_1 = g_2$ ,  $h_1 = h_2$ , and thus  $w_1 = w_2$ . Since  $\mathcal{A}$  is co-deterministic, there is a unique state  $s$ , different from  $q$ , such that  $w$  is in  $\text{Fut}_{\mathcal{A}}(s)$ . Hence, by Case 1,  $\varphi(K'_1) = \varphi(K'_2)$  and thus  $K_1 = K_2$ .  $\square$

A last lemma to deal with the breaking of the images by  $\varphi$ .

**Lemma 4.8.** *For every broken derived term  $K'$  of  $E'$ ,  $B(\varphi(K'))$  is a set of expressions whose interpretations are pairwise disjoint.*

*Proof.* As  $K'$  is a broken derived term, it is either a star, a letter or a product whose first factor is a star or a letter. If it begins with a star or a letter which is not in  $C$ ,  $B(\varphi(K')) = \{\varphi(K')\}$ . Otherwise,  $K'$  begins with a letter  $c_{p,r}$  and  $B(\varphi(K')) = \bigcup_{a \in F_p} aG^*H_rL$ .  $\square$

**Corollary 4.9.** *Let  $\mathcal{A}$  be a normalised co-deterministic automaton, and  $E = \Phi(\mathcal{A})$ . Then, the broken derived term automaton  $\Delta'(E)$  of  $E$  is co-deterministic.*

*Proof.* As above, the corollary trivially holds if  $\mathcal{A}$  has only two states.

By induction,  $\Delta'(E')$  is co-deterministic, which means, by Property 2.15, that the interpretations of the broken derived terms of  $E'$  are pairwise disjoint. We prove now that the interpretations of elements of  $\text{BD}(E)$  are disjoint, which implies that  $\Delta'(E)$  is co-deterministic.

By Equation (8),  $\text{BD}(E) = \text{B}(D(E))$  and, by Proposition 4.7, the interpretations of the derived terms of  $E$  are disjoint. If there exists a word  $u$  in the intersection of the interpretation of two broken derived terms  $K_1$  and  $K_2$ , then, there exists a derived term  $L$  of  $E$ , such that both  $K_1$  and  $K_2$  are in  $\text{B}(L)$ .

**Case 1** There exists  $L'$  in  $D(E')$  such that  $L = \varphi(L')$ . Then, there exists  $K'_1$  and  $K'_2$  in  $\text{B}(L')$  such that  $K_1 \in \text{B}(\varphi(K'_1))$  and  $K_2 \in \text{B}(\varphi(K'_2))$ . By Lemmas 4.6 and 4.5, there exists a unique word  $v$  in  $|L'|$  such that  $\varphi(v) = u$ , hence,  $v$  is in  $|K'_1| \cap |K'_2|$ . By induction hypothesis, the broken derived terms of  $E'$  have disjoint interpretations, thus  $K'_1 = K'_2$ . By Lemma 4.8,  $K_1 = K_2$ .

**Case 2** There exist  $L'$  in  $D(E')$ ,  $G$  and  $H_r$ , such that  $L = G^*H_r\varphi(L')$  or  $L = H_r\varphi(L')$  (if  $G$  is null).

In the first subcase,  $\text{B}(L) = \{L\}$ , hence  $K_1 = K_2 = L$ ; in the second subcase,  $K_1 = a_1\varphi(L')$  and  $K_2 = a_2\varphi(L')$ ,  $a_1$  and  $a_2$  are both the first letter of  $u$  and are therefore equal, hence  $K_1 = K_2$ .  $\square$

Theorem 1.1 is now established by the following simple argument. Indeed, either  $\mathcal{A}$  is a normalised automaton, and nothing is to be added, or we apply Corollary 4.9 to  $\mathcal{A}_{\diamond}$  and as the interpretations of any two broken derived terms of  $E_{\diamond}$  are disjoint, so are the interpretations of the corresponding broken derived terms of  $E$ : the broken derived term automaton is co-deterministic.  $\square$

## CONCLUSION

As we wanted to be complete, and not to leave another flaw behind us, the proof of the theorem is long, much longer than we expected, and somewhat laboured. It goes repeatedly forth and back between the properties of the terms and of their interpretation. It is also based on an interplay between the derived terms and the broken ones, which let us think that the truth may be hidden somewhere in between...

As we explained in the introduction, the statement whose proof is corrected in this corrigendum was considered in the original paper as the core of an algorithm to find a converse to state elimination method. It applies to minimal co-deterministic automata but its scope is more far reaching. First, the new like the old construction of the broken derived term automaton happens to give the solution in many cases where the automaton is *not co-deterministic*. Second, the main part of the paper [7] was devoted to the shortcomings of this construction and to the description of different tracks to obtain a converse algorithm when this construction alone fails. All these tracks remain valid and the problem as appealing as before.

On the other hand, the proof presented here brings to light these broken derived terms that we defined in [7] in an exploratory way. They deserve to be further studied, a task that we have already begun [1].

*Acknowledgements.* We first want to thank our student Florent Terrones who was in charge of implementing in the VAUCANSON platform the construction described in [8] and who drew our attention to the problem caused by the erroneous definition of  $\Delta(E)$ <sup>9</sup>. We are thankful to Pierre-Yves Angrand who contributed in his Master thesis to the final definition of the breaking operator. We are also grateful to the devoted referee who carefully read a first version of this paper and made a series of remarks and comments which helped us in writing the final version.

## REFERENCES

- [1] P.-Y. Angrand, S. Lombardy and J. Sakarovitch, On the number of broken derived terms of a rational expression. *J. Automata, Languages and Combinatorics*, to appear.
- [2] V. Antimirov, Partial derivatives of regular expressions and finite automaton constructions. *Theoret. Computer Sci.* **155** (1996) 291–319.
- [3] J.A. Brzozowski, Derivatives of regular expressions. *J. Assoc. Comput. Mach.* **11** (1964) 481–494.
- [4] P. Caron and M. Flouret, Glushkov construction for series: the non commutative case. *Int. J. Comput. Math.* **80** (2003) 457–472.
- [5] S. Eilenberg, *Automata, Languages, and Machines*. **A**, Academic Press (1974).

---

<sup>9</sup>The version 1.3.3 of VAUCANSON provides the construction of the derived term automaton of an expression, not the one of the broken derived term automaton; cf. <http://www.lrde.epita.fr/cgi-bin/twiki/view/Vaucanson/>.

- [6] V. Glushkov, The abstract theory of automata. *Russ. Math. Surv.* **16** (1961) 1–53.
- [7] S. Lombardy and J. Sakarovitch, Derivatives of rational expressions with multiplicity. *Theoret. Computer Sci.* **332** (2005) 141–177. (Journal version of *Proc. MFCS 02*, LNCS 2420 (2002) 471–482.)
- [8] S. Lombardy and J. Sakarovitch, How expressions can code for automata. *RAIRO – Inform. theor. appl.* **39** (2005) 217–237 (Journal version *Proc. LATIN*, LNCS **2976** (2004) 242–251.)
- [9] J. Sakarovitch, *Éléments de théorie des automates*. Vuibert (2003), Corrected English edition: *Elements of Automata Theory*. Cambridge University Press (2009).

Communicated by C. Choffrut.

Received January 15, 2008. Accepted June 23, 2010.