Conjugacy and equivalence of weighted automata and functional transducers

Marie-Pierre Béal[1], Sylvain Lombardy[2], and Jacques Sakarovitch[3]

[1] Institut Gaspard-Monge, Université Marne-la-Vallée.
[2] LIAFA, Université Paris 7.
[3] LTCI, CNRS / Ecole Nationale Supérieure des Télécommunications. (UMR 5141)

beal@univ-mlv.fr lombardy@liafa.jussieu.fr sakarovitch@enst.fr

Abstract. We show that two equivalent $\mathbb{K}$-automata are conjugate to a third one, when $\mathbb{K}$ is equal to $\mathbb{B}$, $\mathbb{N}$, $\mathbb{Z}$, or any (skew) field and that the same holds true for functional tranducers as well.

EXTENDED ABSTRACT

1 Presentation of the results

In a recent paper ([1]), we have studied the equivalence of $\mathbb{Z}$-automata. This equivalence is known to be decidable (with polynomial complexity) for more than forty years but we showed there two results that give more *structural information* on two equivalent $\mathbb{Z}$-automata. We first proved that two equivalent $\mathbb{Z}$-automata are related by a series of *three* conjugacies — we shall define conjugacy later in the paper — and then that every conjugacy relation can be decomposed into a sequence of three operations: state (out-)splitting (also known as *covering*), circulation of coefficients, and state (in-)merging (also known as *co-covering*). Altogether, we reached a decomposition of any equivalence between $\mathbb{Z}$-automata as the one described at Figure 1 [Conjugacy is represented by double-line arrows, coverings by simple solid arrows, co-coverings by simple dashed arrows, and circulation by simple dotted arrows].

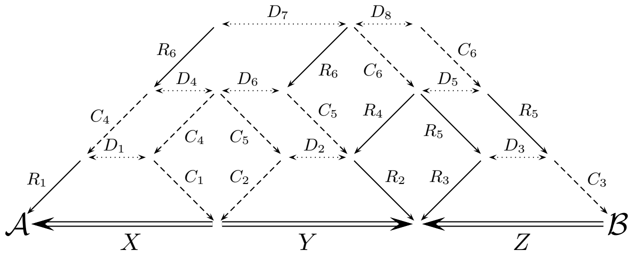

Fig. 1. Structural decomposition of the equivalence of two $\mathbb{Z}$-automata.

At the end of our ICALP paper we mentioned two problems open by the gap between these results and those that were formerly known. First, whether *three* conjugacies are necessary (in general), and, if yes, whether it is decidable when *two* conjugacies suffice. Second, whether, in the case of $\mathbb{N}$-automata, the whole chain of conjugacies could be always realized with transfer matrices in $\mathbb{N}$ and, if not, whether it is decidable when this property holds.

We answer these two questions here. By means of techniques different from the ones that where developed in [1], we show that *two conjugacies* always suffice and that this property holds not only for $\mathbb{Z}$-automata but also for $\mathbb{N}$-automata and other families of automata as stated by the following.

Theorem 1. *Let $\mathbb{K}$ be $\mathbb{B}$, $\mathbb{N}$, $\mathbb{Z}$, or any (skew) field. Two $\mathbb{K}$-automata are equivalent if and only if there exists a third $\mathbb{K}$-automaton that is conjugate to both of them.*

Moreover, an analoguous result holds for functional transducers as well.

Theorem 2. *Two functional transducers are equivalent if and only if there exists a third functional transducer that is conjugate to both of them*

Together with these results on conjugacy, we extend the decomposition of conjugacy by means of covering, co-covering and "circulation" as follow (we shall define covering and co-covering more precisely at Section 3). We state the first one for sake of completeness.

Theorem 3 ([1]). *Let $\mathbb{K}$ be a field $\mathbb{F}$ or the ring $\mathbb{Z}$ and let $\mathcal{A}$ and $\mathcal{B}$ be two $\mathbb{K}$-automata. We have $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{B}$ if and only if there exists two $\mathbb{K}$-automata $\mathcal{C}$ and $\mathcal{D}$ and a circulation matrix D such that $\mathcal{C}$ is a co-$\mathbb{K}$-covering of $\mathcal{A}$, $\mathcal{D}$ a $\mathbb{K}$-covering of $\mathcal{B}$ and $\mathcal{C} \overset{D}{\Longrightarrow} \mathcal{D}$.*

Theorem 4. *Let $\mathbb{K}$ be the semiring $\mathbb{N}$ or the Boolean semiring $\mathbb{B}$ and let $\mathcal{A}$ and $\mathcal{B}$ be two trim $\mathbb{K}$-automata. We have $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{B}$ if and only if there exists a $\mathbb{K}$-automaton $\mathcal{C}$ that is a co-$\mathbb{K}$-covering of $\mathcal{A}$ and a $\mathbb{K}$-covering of $\mathcal{B}$.*

Theorem 5. *Let $\mathcal{A}$ and $\mathcal{B}$ be two trim functional transducers. We have $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{B}$ if and only if there exists two (functional) transducers $\mathcal{C}$ and $\mathcal{D}$ and a diagonal matrix of words D such that $\mathcal{C}$ is a co-covering of $\mathcal{A}$, $\mathcal{D}$ a covering of $\mathcal{B}$ and $\mathcal{C} \overset{D}{\Longrightarrow} \mathcal{D}$.*

In other words, Figure 1 can be replaced by Figure 2 where $\mathcal{A}$ and $\mathcal{B}$ are taken in any family considered in Theorems 1 and 2.

The present result on conjugacy is both *stronger* and *broader* than the preceeding ones. Stronger as the number of conjugacies is reduced from *three* to *two*, broader as the result apply not only to $\mathbb{Z}$-automata (indeed to automata with multiplicity in an Euclidean domain) but to a much larger family of automata. It answers in particular to what was a long standing problem for the authors: is it possible to transform an $\mathbb{N}$-automaton into any other equivalent one using only state splitting and state merging? The answer is thus positive, and the chain of operations is rather short. The benefit brought by the change from $\mathbb{Z}$ into $\mathbb{N}$ is well illustrated by the following consequence.

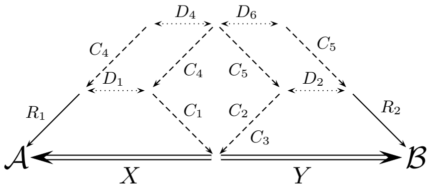

Fig. 2. Structural decomposition of the equivalence of two $\mathbb{K}$-automata.

Theorem 6. *If two regular languages have the same generating function (i.e. the numbers of words of every length is the same in both languages) then there exists a letter-to-letter rational function that realizes a bijection between the two languages.*

2 Conjugacy and covering of automata

A *finite* automaton $\mathcal{A}$ over an alphabet A with multiplicity in a semiring $\mathbb{K}$, or $\mathbb{K}$-automaton for short, can be written in a compact way as $\mathcal{A} = \langle I, E, T \rangle$ where E is a square matrix of finite dimension Q whose entries are linear combinations (with coefficients in $\mathbb{K}$) of letters in A and where I and T are two vectors — respectively row vector and column vector — with entries in $\mathbb{K}$ as well. We can view each entry $E_{p,q}$ as the label of a unique arc which goes from state p to state q in the graph whose set of vertices is Q (if $E_{p,q} = 0_{\mathbb{K}}$, we consider that there is *no* arc from p and q).

The *behaviour* of $\mathcal{A}$, denoted $|\mathcal{A}|$, is the series such that the coefficient of a word w is the coefficient of w in $I\,E^{|w|}\,T$. It is part of Kleene-Schützenberger Theorem that every $\mathbb{K}$-rational series is the behaviour of a $\mathbb{K}$-automaton of the form we have just defined. For missing definitions, we refer to [4, 2, 10].

2.1 Conjugacy

Definition 1. *A $\mathbb{K}$-automaton $\mathcal{A} = \langle I, E, T \rangle$* is conjugate to *a $\mathbb{K}$-automaton $\mathcal{B} = \langle J, F, U \rangle$ if there exists a matrix X with entries in $\mathbb{K}$ such that*

$$I\,X = J, \qquad E\,X = X\,F, \quad \text{and} \quad T = X\,U.$$

The matrix X is the transfer matrix *of the conjugacy and we write $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{B}$.*

Remark that in spite of the idea conveyed by the terminology, the conjugacy relation *is not an equivalence* but a *preorder* relation. Suppose that $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{C}$ holds; if $\mathcal{C} \overset{Y}{\Longrightarrow} \mathcal{B}$ then $\mathcal{A} \overset{XY}{\Longrightarrow} \mathcal{B}$, but if $\mathcal{B} \overset{Y}{\Longrightarrow} \mathcal{C}$ then $\mathcal{A}$ is not necessarily conjugate to $\mathcal{B}$, and we write $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{C} \overset{Y}{\Longleftarrow} \mathcal{B}$ or even $\mathcal{A} \overset{X}{\Longrightarrow}\overset{Y}{\Longleftarrow} \mathcal{B}$.

This being well understood, we shall speak of "conjugate automata" when the orientation does not matter. For instance, we state that, obviously, two conjugate automata are equivalent (*i.e.* have the same behaviour).

2.2 Covering

The standard notion of morphisms of automata — which consists in merging states and does not tell enough on transitions — is not well-suited to $\mathbb{K}$-automata. Hence the definitions of $\mathbb{K}$-coverings and co-$\mathbb{K}$-coverings. These have probably stated independently a number of times. We describe them here in terms of conjugacy. A definition closer to the classical morphisms could be given and then the definitions below become propositions (*cf.* [1, 10]).

Let $\varphi\colon Q \to R$ be a surjective map and H_φ the $Q \times R$-matrix where the (q, r) entry is 1 if $\varphi(q) = r$, 0 otherwise. Since φ is a map, each row of H_φ contains exactly one 1 and since φ is surjective, each column of H_φ contains at least one 1. Such a matrix is called an amalgamation matrix ([6, Def. 8.2.4]).

Let $\mathcal{A}$ and $\mathcal{B}$ be two $\mathbb{K}$-automata of dimension Q and R respectively. We say that $\mathcal{B}$ is *a* $\mathbb{K}$-*quotient* of $\mathcal{A}$ and conversely that $\mathcal{A}$ is *a* $\mathbb{K}$-*covering* of $\mathcal{B}$ if there exists a surjective map $\varphi\colon Q \to R$ such that $\mathcal{A}$ is conjugate to $\mathcal{B}$ by H_φ

The notion of $\mathbb{K}$-quotient is *lateralized* since the conjugacy relation is not symmetric. Somehow, it is the price we pay for extending the notion of morphism to $\mathbb{K}$-automata. Therefore the *dual* notions *co*-$\mathbb{K}$-*quotient* and *co*-$\mathbb{K}$-*covering* are defined in a natural way. We say that $\mathcal{B}$ is *a co*-$\mathbb{K}$-*quotient* of $\mathcal{A}$ and conversely that $\mathcal{A}$ is *a co*-$\mathbb{K}$-*covering* of $\mathcal{B}$ if there exists a surjective map $\varphi\colon Q \to R$ such that $\mathcal{B}$ is conjugate to $\mathcal{A}$ by ${}^{t}H_\varphi$.

We also write $\varphi\colon \mathcal{A} \to \mathcal{B}$ and call φ, by way of metonymy, *a* $\mathbb{K}$-*covering, or a co*-$\mathbb{K}$-*covering from* $\mathcal{A}$ *onto* $\mathcal{B}$.

3 The joint reduction

The proof of Theorems 1 and 2 relies on the idea of *joint reduction* which is defined by means of the notion of *representation.*

An automaton $\mathcal{A} = \langle I, E, T \rangle$ of dimension Q can equivalently be described as the representation $\mathcal{A} = (I, \mu, T)$ where $\mu\colon A^* \to \mathbb{K}^{Q\times Q}$ is the morphism defined by the equality

$$E = \sum_{a\in A} \mu(a)\, a \quad .$$

This equality makes sense since the entries of E are assumed to be linear combinations of letters of A with coefficients in $\mathbb{K}$. And the coefficient of any word w in the series $|\mathcal{A}|$ is $I\, \mu(w)\, T$.

The set of vectors $\{I\, \mu(w) \mid w \in A^*\}$ (row vectors of dimension Q), that is, the *phase space* of $\mathcal{A}$, plays a key role in the study of $\mathcal{A}$, as exemplifyed by the following two contrasting cases.

If $\mathcal{A}$ is a Boolean automaton, this set of vectors (each vector represents a subset of the dimension Q) is finite and makes up the states of the *determinized automaton* $\mathcal{D}$ of $\mathcal{A}$ (by the subset construction). Moreover, if we form the matrix X whose rows are the states of $\mathcal{D}$, then $\mathcal{D}$ is conjugate to $\mathcal{A}$ by X.

If $\mathcal{A}$ is a $\mathbb{K}$-automaton with $\mathbb{K}$ a field, the *left reduction* of $\mathcal{A}$ — recalled with more detail below — consists in choosing a prefix-closed set P of words such that the vectors $\{I\,\mu(p) \mid p \in P\}$ is a basis of the vector space generated by $\{I\,\mu(w) \mid w \in A^*\}$ (*cf.* [2]). Moreover the (left-)reduced automaton is conjugate to $\mathcal{A}$ by the matrix X whose rows are the vectors $\{I\,\mu(p) \mid p \in P\}$.

Let now $\mathcal{A} = (I, \mu, T)$ and $\mathcal{B} = (I', \kappa, T')$ be two $\mathbb{K}$-automata of dimension Q and R respectively. We consider the *union* of $\mathcal{A}$ and $\mathcal{B}$ and thus the vectors $[I\,\mu(w)|I'\,\kappa(w)]$ of dimension $Q \cup R$. These vectors, for w in A^*, generate a $\mathbb{K}$-module W. The *(left) joint reduction* of $\mathcal{A}$ and $\mathcal{B}$ consists in computing — when it is possible — a finite set G of vectors $[x|y]$ which generate the same $\mathbb{K}$-module W. Then the matrix M whose *rows* are these vectors $[x|y]$ provides in some sense a $\mathbb{K}$-automaton $\mathcal{C}$ which is conjugate to both $\mathcal{A}$ and $\mathcal{B}$ with the transfer matrices X and Y respectively, where X and Y are the 'left' and 'right' parts of the matrix M respectively.

In every case listed in the above Theorems 1 and 2, and which we consider now, the finite set G is effectively computable.

3.1 Joint reduction in fields

Let $\mathbb{K}$ be a field and let $\mathcal{A} = (I, \mu, T)$ be a $\mathbb{K}$-automaton of dimension n.

The reduction algorithm for $\mathbb{K}$-automata is split into two dual parts. The first part consists in computing a prefix-closed subset P of A^* such that the set $G = \{I\,\mu(w) \mid w \in P\}$ is free and, for every letter a, and every word in P, $I\,\mu(wa)$ is lineary dependant from G. The set G has at most n elements and an automaton $\mathcal{C} = (J, \kappa, U)$, whose states are the elements of G, is defined by:

$$J_x = \begin{cases} 1 & \text{if } x = I \ , \\ 0 & \text{otherwise} \ , \end{cases} \qquad \forall x \in G, \quad U_x = xT \ ,$$

$$\forall a, \quad \exists! \kappa(a), \qquad \forall x \in G,\ x\mu(a) = \sum_{y \in G} \kappa(a)_{x,y} y \ .$$

This can be viewed as a change of basis: the set G generates the smallest subspace of $\mathbb{K}^n$ that contains every $I\,\mu(w)$ and if G is completed into a basis B, after changing the canonical basis by B and projection, one gets the automaton $\mathcal{C}$. Finally, if M is the matrix whose *rows* are the elements of G, it holds $\mathcal{C} \overset{M}{\Longrightarrow} \mathcal{A}$.

The second part is similar and consists in computing a basis of the subspace of $\mathbb{K}^{|G|}$ generated by the vectors $\kappa(w)\,U$. It is a nice result (by Schützenberger) that after these two semi-reductions, the outcome is a $\mathbb{K}$-automaton of smallest dimension that is equivalent to $\mathcal{A}$.

We focus here on the first part which we call *left reduction*. Let $\mathcal{A} = (I, \mu, T)$ and $\mathcal{B} = (I', \mu', T')$ be two equivalent $\mathbb{K}$-automata and let $\mathcal{C}_0 = (J, \kappa, U)$ be the automaton obtained by left reduction of $\mathcal{A} + \mathcal{B}$. The automaton $\mathcal{A} + \mathcal{B}$ has a representation equal to $([I|I'], \mathsf{diag}(\mu, \mu'), [T|T'])$, where $[I|I']$ is obtained by horizontally joining the row vectors I and I', $[T|T']$ by vertically stacking the column vectors T and T', and for every letter a, $[\mathsf{diag}(\mu|\mu')](a)$ is the matrix whose diagonal blocks are $\mu(a)$ and $\mu'(a)$.

The automaton $\mathcal{C}_0$ is conjugate to $\mathcal{A}+\mathcal{B}$ by the matrix $[X|Y]$, in which every row has the form $[I\,\mu(w)|I'\,\mu'(w)]$ where w is a word. It holds:

$$J\,[X|Y] = [I|I'], \quad \forall a,\ \kappa(a)\,[X|Y] = [X|Y]\,(\mathsf{diag}(\mu|\mu'))(a), \quad U = [X|Y]\,[T|T']$$

As $\mathcal{A}$ and $\mathcal{B}$ are equivalent, $X\,T = Y\,T'$ and thus $U = 2X\,T = 2Y\,T'$. Let $\mathcal{C} = \langle J, \kappa, U/2\rangle$; it immediatly comes $\mathcal{C} \xrightarrow{X} \mathcal{A}$ and $\mathcal{C} \xrightarrow{Y} \mathcal{B}$.

3.2 Joint reduction in $\mathbb{Z}$

The result and the algorithm are basically the same as the previous ones if the multiplicity semiring is $\mathbb{Z}$. As in vector spaces, there is a dimension theory in the free $\mathbb{Z}$-modules and it is still possible to compute a basis G of the submodule of $\mathbb{Z}^n$ generated by the vectors $I\,\mu(w)$. However, this basis does not correspond any more to a prefix-closed set of words. and the algorithm to compute it is explained in [1].

3.3 Joint reduction in $\mathbb{N}$

There is no dimension theory in the $\mathbb{N}$-modules and thus no reduction algorithm for $\mathbb{N}$-automata similar to the previous ones.

However, given $\mathcal{A}+\mathcal{B}$ our aim is not the reduction itself but the computation of a set G of vectors with the 3 properties: for every $z = [x|y]$ in G, $xT = yT'$ holds, the $\mathbb{N}$-module $\langle G\rangle$ generated by G is closed under multiplication by $(\mathsf{diag}(\mu|\mu'))(a)$, for every letter a (which is important to effectively build the automaton $\mathcal{C}$), and finally G is finite. It can be noted that in the preceeding algorithms, the freeness of the generating set G is used only to garantee finiteness. An algorithm that compute such a G for $\mathbb{N}$-automata can be roughly sketched as follows.

Start from $G = \{[I|I']\}$. While $\langle G\rangle$ is not closed under $(\mathsf{diag}(\mu|\mu'))(a)$, take $z = [x|y]$ in $G\,(\mathsf{diag}(\mu|\mu'))(a) \setminus \langle G\rangle$ add z to G, and *reduce* G. The reduction goes as follow: while G contains z and z' such that $z < z'$ (in the product order of $\mathbb{N}^{Q\cup R}$) replace z' by $z' - z$. This algorithm ends since at every step, either the size of vectors of G decreases or the size of G increases. The size of vectors cannot decrease infinitely and as vectors of G are pairwise incomparable (after the reduction step), G has only a finite number of elements.

The outcome of this algorithm is not canonically associated to $\mathcal{A}$ and $\mathcal{B}$ and even its size (in contrast to what happens with fields) may depend on the order in which comparable pairs are considered during the reduction step. Yet, an automaton $\mathcal{C}$ whose states are the elements of G is built as the previous cases.

3.4 Joint reduction in $\mathbb{B}$

In $\mathbb{B}$, as in many semirings that cannot be embedded in rings, there is no subtraction. Therefore it is quite difficult to *reduce* vectors $[I\,\mu(w)|I'\,\mu'(w)]$ to find a “minimal” set of generators. As $\mathbb{B}$ is finite, the simplest way is to keep all the vectors $[I\,\mu(w)|I'\,\mu'(w)]$. The automaton $\mathcal{C}$ obtained from this set is nothing else that the *determinised* automaton of $\mathcal{A} \cup \mathcal{B}$. For the same reason as above, this automaton is conjugate both to $\mathcal{A}$ and $\mathcal{B}$.

3.5 Joint reduction of functional transducers

With transducers, difficulties of automata with multiplicities and Boolean automata meet. On the one hand, if $\mathcal{T} = (I, \mu, T)$, the set $\{I\mu(w) \mid w \in A^*\}$ may be infinite and, in the other hand, as in the Boolean case, the substraction is not allowed in the semiring of multiplicities that can be associated to them.

If the transducers $\mathcal{A}$ and $\mathcal{B}$ were sequentialisable, it would be sufficient to consider the sequentialised transducer of their union that would be conjugate to each of them. The idea of the sequentialisation (*cf.* [3, 7]) is to compute a (finite) set of vectors of words, each vector being the information that can not be output and that is necessary for further computation.

On general functional transducers, this algorithm does not always end. We present now a *pseudo-sequentialisation*, that stops on any functional transducer. This algorithm allows to split vectors of words when their components are different enough, which induces non deterministic transitions.

We describe first this algorithm on one functional transducer and then explain how to use it for the joint reduction.

Definition 2. *Let k be a positive integer and X be a se of words. Two words u and v of A^* are* k-related *in X, if there exists a finite sequence $w_0, ..., w_n$ of words such that $u = w_0$, $v = w_n$, for every i in $[1; n]$, $d(w_{i-1}, w_i) \leqslant k$ and there exists i in $[1; n]$ such that w_i is a prefix of u and v. The set X is k-related if every pair of its elements is k-related in X.*

The k-relation is an equivalence on X.

Definition 3. *Let α be a vector of words. The k-decomposition of α is the smallest set of vectors $D_k(\alpha)$ such that, for every $\beta \in D_k(\alpha)$ the set of components of β is k-related and $\alpha = \sum_{\beta \in D_k(\alpha)} \beta$.*

Obviously, the vectors of $D_k(\alpha)$ have disjoint supports. We shall applies this decomposition to vectors of words and then reduce them with respect to their greatest common prefix; this second step is exactly the same as in the classical sequentialisation algorithm.

Definition 4. *Let α be a vector of words. We denote $\overset{\circ}{\alpha}$ the greatest common prefix of non zero components of α and $\alpha^\sharp = \overset{\circ}{\alpha}^{-1}\alpha$.*

Definition 5. *Let $\mathcal{T} = (I, \mu, T)$ be a functional transducer and let k be non negative integer The k-pseudo-sequentialised transducer $\mathcal{S}$ of $\mathcal{T}$ is defined by:*

– *for every β in $D_k(I)$, $\beta^\sharp$ is an initial state with initial weight $\overset{\circ}{\beta}$;*
– *for every state α, for every letter a, for every β in $D_k(\alpha\mu(a))$, there is a transition labeled by a with output $\overset{\circ}{\beta}$ from α to $\beta^\sharp$.*
– *for every state α, α is final with output $w = \alpha T$ if w is non zero.*

Proposition 1. *For k, the k-pseudo-sequentialised transducer $\mathcal{S}$ of a functional transducer $\mathcal{T}$ is a finite transducer that is conjugate to $\mathcal{T}$.*

The transducer $\mathcal{S}$ is finite since the components of its states (that are vectors) are bounded by k.

If the k-pseudo-sequentialisation is applied to the union of two equivalent functional transducers $\mathcal{A} = (I, \mu, T)$ and $\mathcal{B} = (I', \mu', T')$, it gives a transducer $\mathcal{C}$ which is conjugate to $\mathcal{A} \cup \mathcal{B}$ with a matrix $M = [X|Y]$, but, in general, this transducer is not conjugate to $\mathcal{A}$ with X and to $\mathcal{B}$ with Y. Actually, if k is too small, there may be rows $[x|y]$ of M such that $xT \neq yT'$.

Let k be equal to n^2L, where n is the maximum of dimensions of $\mathcal{A}$ and $\mathcal{B}$ and L is the longest output of transitions or terminal functions of $\mathcal{A}$ and $\mathcal{B}$. In this case, the k-pseudo-sequentialised transducer is unambiguous, which implies that $xT = yT'$ for every state $[x|y]$ of $\mathcal{C}$. Therefore, the transducer $\mathcal{C}$ is conjugate both to $\mathcal{A}$ and $\mathcal{B}$.

Example 1. Figure 3 shows the transducer $\mathcal{T}_1$ and its k-pseudo-sequentialised $\mathcal{S}_1$ (the result is the same with any positive k), where $\mathcal{T}_1$ is the (left) transducer that replaces systematically factors *abb* by *baa* when reading words *from right to left*; $\mathcal{T}_1$ is thus co-sequential, that is, input co-deterministic (*cf.* [10]). The transducer $\mathcal{S}_1$ is conjugate to $\mathcal{T}_1$ with the transfer matrix M:

$$M = \begin{bmatrix} bb & b & 1 \\ b & 1 & 0 \\ 1 & 0 & 0 \\ 0 & b & 1 \\ 0 & 0 & 1 \end{bmatrix} .$$

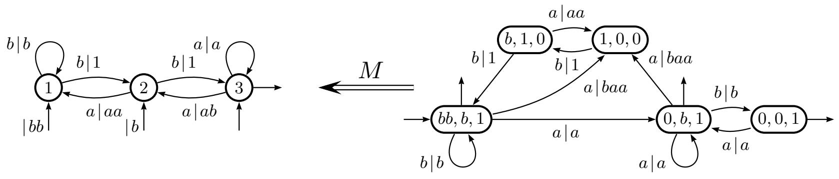

Fig. 3. The transducers $\mathcal{T}_1$ and $\mathcal{S}_1$

The above list may lead to think that a joint reduction procedure may be found for any semiring. This is certainly not the case and the tropical semirings for instance, or the non functional transducers, are not likely to admit a joint reduction procedure.

4 From conjugacy to coverings

It remains to show Theorems 3, 4 and 5.

4.1 The case of fields and integers

We have proved Theorem 3 in [1]. Actually, every matrix M can be decomposed in a product HDK, where ${}^tH$ and K are amalgamation matrices and D is a diagonal matrix whose entries are invertible. If $\mathbb{K}$ is a field, the dimension of D is the number of non zero entries of M , and if $\mathbb{K} = \mathbb{Z}$, as the only invertible elements are 1 and -1, every non zero element has to be decomposed in a sum of ± 1 and the dimension of D is the sum of the absolute values of the entries of M.

The proof consists then in proving that there exist automata $\mathcal{C}$ and $\mathcal{D}$ such that $\mathcal{A} \overset{H}{\Longrightarrow} \mathcal{C} \overset{D}{\Longrightarrow} \mathcal{D} \overset{K}{\Longrightarrow} \mathcal{B}$. The construction of $\mathcal{C}$ and $\mathcal{D}$ amounts to fill in blocks of their transition matrix knowing the sum of the rows and the columns.

For natural integers, the proof is exactly the same. The unique invertible element of $\mathbb{N}$ is 1, thus D is the identity matrix. However, to get the expected form, the matrix M must have no zero row or column.[4] This is ensured by the assumption that $\mathcal{A}$ and $\mathcal{B}$ are trim.

4.2 The Boolean case

Let $\mathcal{A} = (I, \mu, T)$ and $\mathcal{B} = (J, \kappa, U)$ be two trim automata such that there exists a $n \times m$ Boolean matrix X that verifies $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{B}$.

Let k be the number of non zero entries of matrix X. We define $\varphi \colon [1;k] \to [1;n]$ and $\psi \colon [1;k] \to [1;m]$, such that $x_{\varphi(i),\psi(i)}$ is the i-th non zero entry of X. Let H_φ and H_ψ be the matrices associated to these applications. It holds $X = {}^tH_\varphi \; H_\psi$. We define $\mathcal{C} = (K, \zeta, V)$ with dimension k by:

$$K = I \, {}^tH_\varphi \quad , \quad V = H_\psi U \quad ,$$
$$\forall (p,q) \in [1;k]^2, \qquad \zeta(a)_{p,q} = \mu(a)_{\varphi(p),\varphi(q)} \wedge \kappa(a)_{\psi(p),\psi(q)} \quad .$$

It is then easy to check that $\mathcal{C} \overset{{}^tH_\varphi}{\Longrightarrow} \mathcal{A}$ and $\mathcal{C} \overset{H_\psi}{\Longrightarrow} \mathcal{B}$, which means that $\mathcal{C}$ is a co-$\mathbb{B}$-covering of $\mathcal{A}$ and a $\mathbb{B}$-covering of $\mathcal{B}$.

In the case were $\mathcal{A}$ is the determinised automaton of $\mathcal{B}$ (which arises if one applies the algorithm given in the previous section), the automaton built in this way is the Schützenberger covering of $\mathcal{B}$, a construction that appears naturally in a number of problems for automata with multiplicity (*cf.* [5, 9, 10]).

4.3 The functional transducer case

Let $\mathcal{A} = (I, \mu, T)$ and $\mathcal{B} = (J, \kappa, U)$ be two trim functional transducers and let X be a $n \times m$ matrix of words such that $\mathcal{A} \overset{X}{\Longrightarrow} \mathcal{B}$. Let k be the number of non zero entries of X. The matrix X can be decomposed into HDK, where H and K are Boolean matrices and D is a diagonal matrix of words of dimension k.

[4] In the previous case, this technical item is handled by considering that $0 = 1 + (-1)$.

This diagonal matrix corresponds to a circulation of words. Actually, in the framework of transducers, the circulation of words is a well-known operation that is needed for instance in the minimisation of sequential transducers. This operation can be related to the circulation of invertible elements for fields if we consider words as elements of the free group.

We want to prove that there exists $\mathcal{A}' = (I', \mu', T')$ and $\mathcal{B}' = (J', \kappa', U')$ such that $\mathcal{A} \overset{H}{\Longrightarrow} \mathcal{A}' \overset{D}{\Longrightarrow} \mathcal{B}' \overset{K}{\Longrightarrow} \mathcal{B}$. We set $I' = I\,H$, $J' = I'\,D$, $U' = K\,U$ and $T' = D\,U'$.

For every letter a, there exists a matrix $\zeta(a)$ such that $H\,\zeta(a) = \mu(a)\,H\,D$ and $\zeta(a)\,K = D\,K\,\kappa(a)$. As H and K are Boolean matrices, $\zeta(a)$ can be factorised in $\mu'(a)\,D$ and $D\,\kappa'(a)$, which gives the solutions.

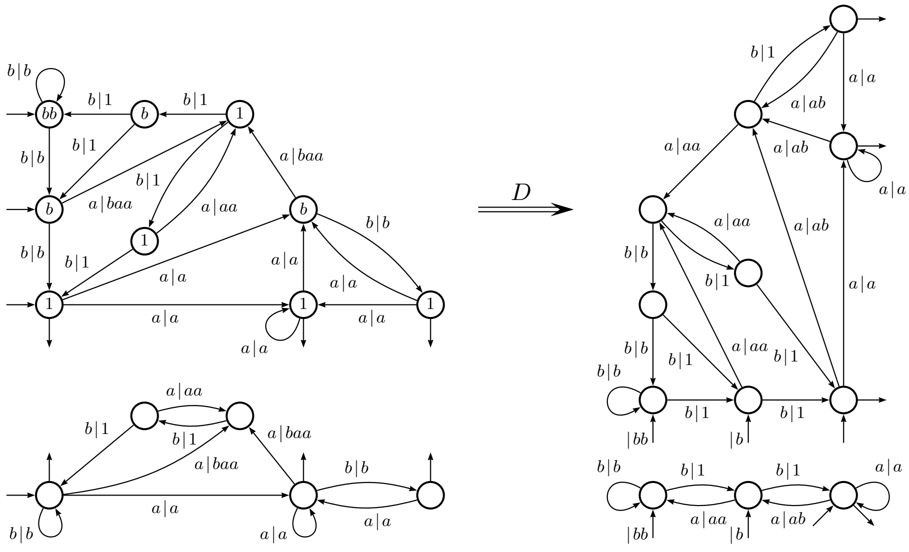

Fig. 4. The instance of Theorem 5 for $\mathcal{S}_1$ and $\mathcal{T}_1$

5 An application

Theorem 6 is a striking consequence of the strengthening of our conjugacy result of [1] and answers a question on automatic structures.

Let $\mathcal{A}$ and $\mathcal{B}$ be two (Boolean) *unambiguous* automata the languages L and K respectively and suppose that L and K have the same generating functions. It amounts to say that if we forget the labels in $\mathcal{A}$ and $\mathcal{B}$ (and replace them all by the same letter x) we have two equivalent $\mathbb{N}$-automata $\mathcal{A}'$ and $\mathcal{B}'$: the coefficient of x^n in $|\mathcal{A}'|$ and thus in $|\mathcal{B}'|$ is the number of words of length n in L and thus in K.

By Theorem 1, $\mathcal{A}'$ and $\mathcal{B}'$ are both conjugate to a same $\mathbb{N}$-automaton $\mathcal{C}'$ (on x^*). By Theorem 4 there exist $\mathcal{D}'$ and $\mathcal{E}'$ such that $\mathcal{D}'$ is a co-$\mathbb{N}$-covering of $\mathcal{C}'$ and a $\mathbb{N}$-covering of $\mathcal{A}'$ and $\mathcal{E}'$ is a co-$\mathbb{N}$-covering of $\mathcal{C}'$ and a $\mathbb{N}$-covering of $\mathcal{B}'$. By a diamond lemma ([1, Proposition 6]) there exists a $\mathbb{N}$-automaton $\mathcal{T}'$ (on x^*) which is a co-$\mathbb{N}$-covering of $\mathcal{D}'$ and of $\mathcal{E}'$.

Every transition of $\mathcal{T}'$ is mapped, via the co-$\mathbb{N}$-coverings and the $\mathbb{N}$-coverings onto a transition of $\mathcal{A}'$ and onto a transition of $\mathcal{B}'$. But these are transitions of $\mathcal{A}$ and $\mathcal{B}$ and every transition of $\mathcal{T}'$ may thus be labelled by a pair of letters (one coming from $\mathcal{A}$ and one coming from $\mathcal{B}$) and hence turned into a letter-to-letter transducer $\mathcal{T}$. As the projection on each component gives an unambiguous automaton, $\mathcal{T}$ realises a bijective function.

Remark 1. Theorem 6 bears some similarity with an old result by Maurer and Nivat (*cf.* [8]) on rational bijections. It is indeed completely different: it is more restricted in the sense it applies only to languages with the same generating functions whereas Maurer and Nivat considered bijections between languages with 'comparable' growth functions, and it is much more precise in the sense that the transducer which realizes the bijection is letter-to-letter. It is this last property that makes the result interesting for the study of automatic structures.

Figure 5 shows the construction for the two languages $L = a(a+b)^*$ and $K = (c+dc+dd)^* \setminus cc(c+d)^*$ recognized by their minimal deterministic (and thus unambiguous) automata $\mathcal{A}$ and $\mathcal{B}$.

References

1. Béal, M.-P., Lombardy, S. and Sakarovitch, J. On the equivalence of $\mathbb{Z}$-automata. *Proc. ICALP'05*, LNCS 3580, Springer (2005) 397–409.
2. Berstel, J., and Reutenauer, Ch. *Rational Series and their Languages.* Springer, 1988.
3. Choffrut, Ch. Une caractrisation des fonctions squentielles et des fonctions sous-squentielles en tant que relations rationnelles. *Theoret. Comput. Sci. 5* (1977), 325–337.
4. Eilenberg, S. *Automata, Languages, and Machines. Vol. A.* Academic Press, 1974.
5. Klimann, I., Lombardy, S., Mairesse J., and Prieur, Ch. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoret. Comput. Sci. 327* (2004), 349–373.
6. Lind, D., and Marcus, B. *An Introduction to Symbolic Dynamics and Coding.* Cambridge University Press, 1995.
7. Lombardy, S. and Sakarovitch, J. Sequential ?. *Theoret. Comput. Sci.*, to appear.
8. Maurer, H., and Nivat, M. Rational Bijection of Rational Sets. *Acta Informatica 13* (1980) 365–378.
9. Sakarovitch, J., A construction on automata that has remained hidden. *Theoret. Computer Sci. 204* (1998), 205–231.
10. Sakarovitch, J., *Eléments de théorie des automates*, Vuibert, 2003. English translation, Cambridge Universit Press, to appear.

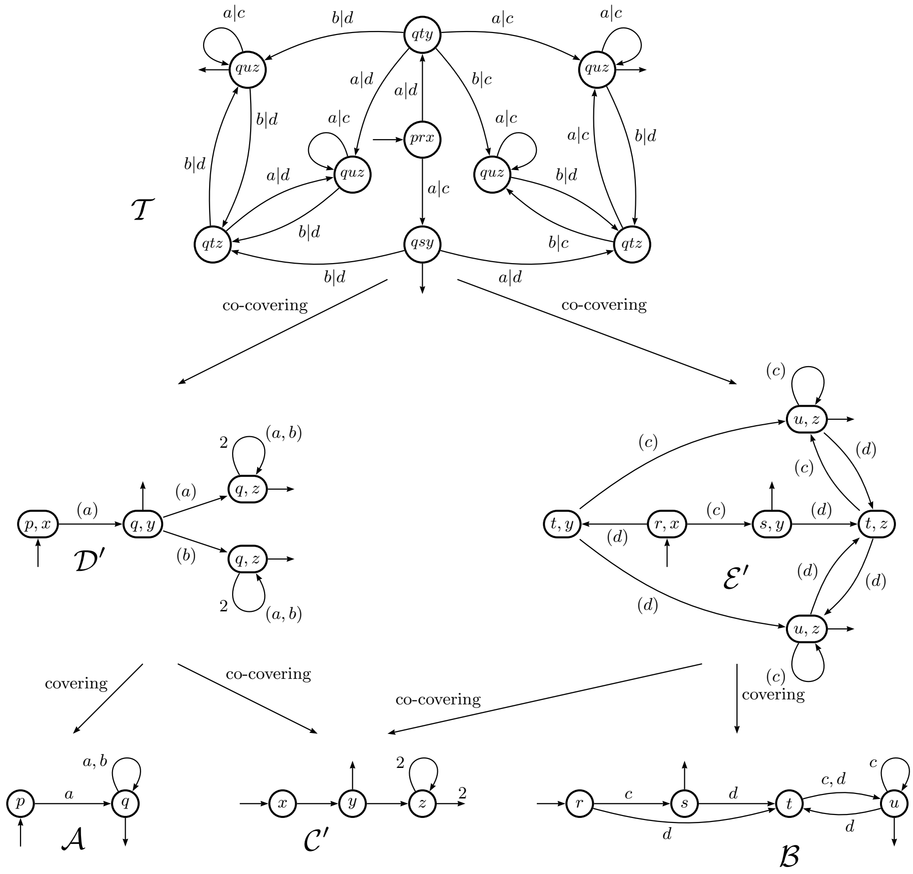

Fig. 5. Construction of a letter-to-letter bijective rational function