Math. Systems Theory 24, 117-146 (1991)

Mathematical
Systems Theory

A Geometrical View of the Determinization and Minimization of Finite-State Automata *

Bruno Courcelle,[1] Damian Niwinski,[2] and Andreas Podelski[3]

[1] Laboratoire d'Informatique, Université Bordeaux-1,
351 cours de la Libération, 33405 Talence cedex, France
E-mail: courcell@geocub.greco-prog.fr

[2] Institute of Mathematics, Warsaw University,
PKiN IX, 00-901 Warszawa, Poland

[3] LITP, Université Paris-7, 2 place Jussieu,
75251 Paris Cedex 05, France
E-mail: andreas@litp.ibp.fr

Abstract. With every finite-state word or tree automaton, we associate a binary relation on words or trees. We then consider the "rectangular decompositions" of this relation, i.e., the various ways to express it as a finite union of Cartesian products of sets of words or trees, respectively. We show that the determinization and the minimization of these automata correspond to simple geometrical reorganizations of the rectangular decompositions of the associated relations.

Introduction

Many results of Automata Theory can be conveniently formulated in the setting of Universal Algebra, which facilitates an immediate extension from words to trees. Regular languages and regular sets of finite trees can thus be considered as two instances of the notion of a recognizable set, a notion which can be defined with respect to arbitrary algebras as shown by Mezei and Wright [MW]. Courcelle [C1] applies this formal framework to sets of finite trees of various kinds, ordered or not, with bounded or unbounded degree. Appropriate notions of finite-state automata follow. Recognizable sets of graphs are considered in [C2]. However, due

* This work was supported by the "Programme de Recherches Coordonnées: Mathématiques et Informatique." It was initiated during a stay in Bordeaux by D. Niwinski in 1988.

to the intrinsically unstructured nature of graphs, no natural notion of a graph automaton arises.

In this setting, a deterministic finite-state automaton can be viewed as a finite algebra over an appropriate signature. The minimizations of deterministic word and tree automata can be thus formulated as constructions of quotient algebras.

Nondeterministic finite-state word and tree automata can also be defined. Classical algorithms transform them into deterministic ones which define the same sets of words or trees. The aforementioned algebraic approach to automata is not convenient for expressing the corresponding *determinization* algorithms, because we cannot represent the behaviors of nondeterministic automata by algebras with single-valued functions. (Algebras with multivalued functions might help, but this direction remains to be explored.)

In this paper we present in a unified way the determinization and the minimization of finite-state word and tree automata. The idea is to associate with a language $L \subseteq X^*$ the set of pairs of words (u, v) such that uv belongs to L. This set is a binary relation s_L on the set X^*. A language L is regular iff the associated relation can be expressed as a finite union of "rectangles," i.e., of relations of the form $A \times B$, where A and B are languages over the alphabet X. We call such an expression a rectangular decomposition of s_L. In particular, a rectangular decomposition of s_L is canonically associated with every automaton defining L. The determinization and the minimization of finite-state automata can then be expressed as reorganizations of the rectangular decompositions associated with the given automata (and visualized geometrically, see Figures 1–4). This method also works for sets of trees: in this case, the appropriate relation is the set of pairs (t, c) such that t is a tree, c is a context (i.e., a tree with a "hole"), and $c[t]$, the tree obtained by filling up the hole of c with t, belongs to the considered set of trees.

As an application, we get a transparent proof of the fact that the minimization of a finite-state (word) automaton can be realized by a determinization of the reversed automaton. (See [B] and Proposition 3.8(2) below for a precise statement.)

We also show that the study of recognizable sets in arbitrary algebras can be done in the setting of rectangular decompositions of relations. Hence, our approach subsumes the algebraic one.

Finally, we can treat the root-to-frontier tree automata introduced by Podelski [P], that are in a certain sense deterministic, while being able to define all of the recognizable sets of trees.

This paper is organized as follows. The geometry of rectangular decompositions of relations is introduced in Section 1. Recognizable sets in arbitrary algebras are dealt with in Section 2. Applications to the recognizability of sets of finite and infinite words are given in Section 3. Our major applications concern finite-state tree automata. They are given in Section 4.

1. Rectangular Decompositions of Relations

By a *relation* we mean in this section a subset of $A \times B$, where A and B are nonempty sets.

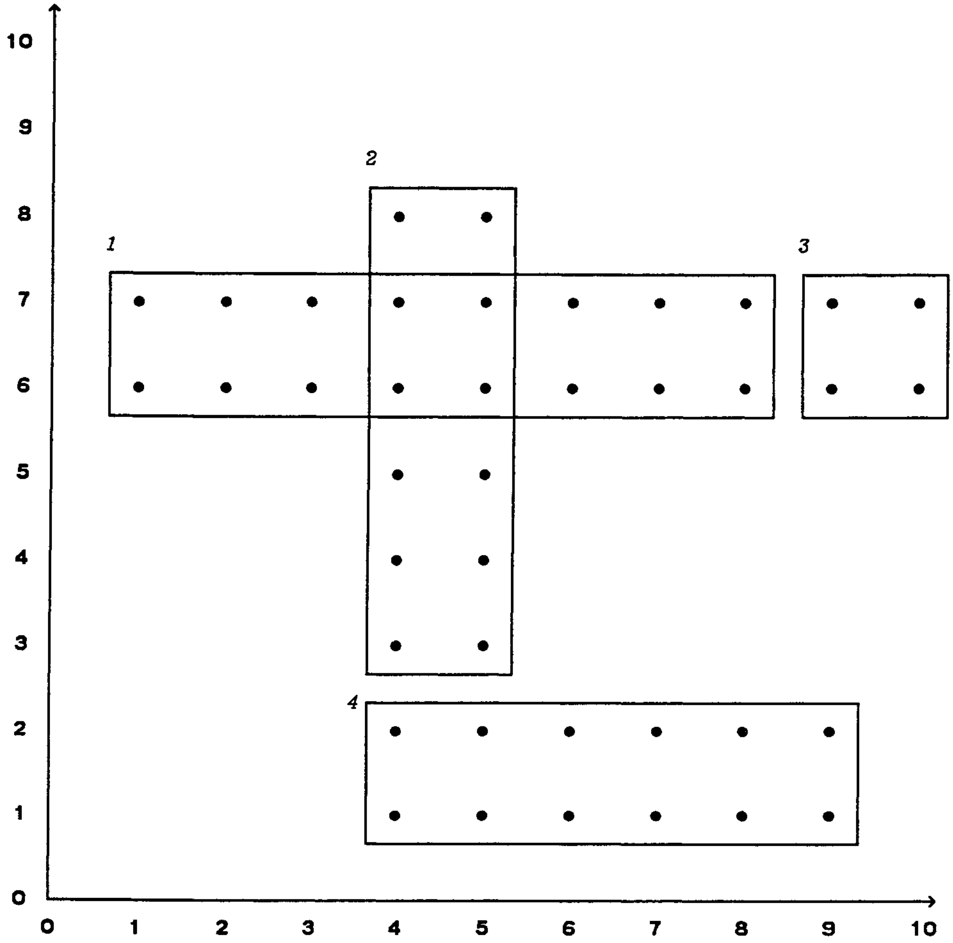

Fig. 1

Definition 1.1 (Rectangular Decompositions of Relations). A *rectangular decomposition* of a relation s is an indexed set of the form $\mathscr{D} = \{(A_i, B_i) | i \in I\}$, such that $s = \bigcup\{A_i \times B_i | \in I\}$ and the sets A_i, B_i are all nonempty. Each set of the form $A_i \times B_i$ is called a *rectangle* of the decomposition.

Figure 1 shows a decomposition of a relation $r \subseteq A \times B$ consisting of four rectangles. To be precise, $A = [0, 10]$, $B = [0, 10]$; the rectangles are $A_i \times B_i$, for $i = 1, \ldots, 4$, where $A_1 = [1, 8]$, $B_1 = [6, 7]$, $A_2 = [4, 5]$, $B_2 = [3, 8]$, $A_3 = [9, 10]$, $B_3 = [6, 7]$, $A_4 = [4, 9]$, and $B_4 = [1, 2]$. (We denote by $[n, m]$ the set of integers between n and m, inclusive of n and m.) An element (a, b) of r is represented as the point of the plane with coordinates a and b. The A-axis is horizontal and the B-axis is vertical. Other decompositions of the relation r are shown in Figures 2-5.

In a decomposition, two indices may refer to the same rectangle. Two decompositions are equal if they consist of the same rectangles, irrespective of the sets of indices. The cardinality of a decomposition is the number of distinct rectangles forming it, and not the cardinality of the index set. A relation is *finitely decomposable* if it has a finite decomposition.

The diagonal relation $\{(a, a) | a \in A\}$ on a set A has a unique decomposition. The rectangles of this decomposition are singletons.

Canonical decompositions of arbitrary relations will be obtained from the syntactical equivalences that we now define.

Definition 1.2 (Syntactical Equivalences). Letting s be a relation, $s \subseteq A \times B$, we also denote by s the total mapping: $A \to \mathscr{P}(B)$ defined by $s(a) := \{b \in B | (a, b) \in s\}$. We define an equivalence relation on A as follows:

$$a \sim_s a' \qquad \text{iff} \quad s(a) = s(a').$$

We denote by s^{-1} the relation $\subseteq B \times A$ equal to $\{(b, a) | (a, b) \in s\}$. Finally, we let $\tilde{s}(A) := \{s(a) | a \in A\}$ and $\tilde{s}(B) := \{s^{-1}(b) | b \in B\}$. These two sets should not be confused with the sets $s(A) = \bigcup\{s(a) | a \in A\}$ (called the *codomain* of s) and $s^{-1}(B) = \bigcup\{s^{-1}(b) | b \in B\}$ (called the *domain* of s).

Since we have

$$s(a) = \{b | a \in s^{-1}(b)\}$$

we have $a \sim_s a'$ iff a and a' belong to exactly the same sets of the form $s^{-1}(b)$, where b ranges over B. It follows that the equivalence class $[a]_s$ of an element a of A with respect to $\sim_s$ can be expressed as follows:

$$[a]_s = (\bigcap\{s^{-1}(b) | b \in s(a)\}) - (\bigcup\{s^{-1}(b) | b \notin s(a)\}). \tag{1}$$

The index m of $\sim_s$ (i.e., the cardinality of its set of equivalence classes) is equal to the cardinality of $\tilde{s}(A)$; similarly, the index n of $\sim_{s^{-1}}$ is equal to the cardinality of the set $\tilde{s}(B)$, and we have, by (1), $m \leq 2^n$. By symmetry, we also have $n \leq 2^m$. In particular, m and n are both finite or both infinite.

Proposition 1.3. *For every relation $s \subseteq A \times B$, the following conditions are equivalent:*

(i) *s is finitely decomposable,*
(ii) *$\tilde{s}(A)$ is finite,*
(iii) *$\tilde{s}(B)$ is finite,*
(iv) *the index of $\sim_s$ is finite,*
(v) *the index of $\sim_{s^{-1}}$ is finite.*

Proof. (i) ⇒ (ii) If $\mathscr{D}$ is a decomposition of s as in Definition 1.1, then $s(a)$ is a union of sets B_i. Hence there are finitely many such sets $s(a)$ if the decomposition $\mathscr{D}$ is finite.

The equivalence of (ii), (iii), (iv), and (v) follows from (1) and the related observations.

The implication (iv) ⇒ (i) follows from the definition of the decomposition **min**(s) that we give in the next subsection. □

Definition 1.4 (Reduced and Deterministic Decompositions). Let $\mathscr{D}$ be a decomposition of a relation s, of the general form of Definition 1.1. We denote by $\mathscr{D}^{-1}$ the decomposition $\{(B_i, A_i) | i \in I\}$ of s^{-1}.

We say that $\mathscr{D}$ is *reduced* if, for all indices i and j in I, $B_i = B_j$ implies $A_i = A_j$. It is *coreduced* if the decomposition $\mathscr{D}^{-1}$ is reduced.

We let **red**($\mathscr{D}$) be the decomposition $\{(A'_i, B_i) | i \in I\}$ of s, such that

$$A'_i = \bigcup \{A_j | j \in I,\ B_j = B_i\}.$$

This decomposition is reduced, and is equal to $\mathscr{D}$ if $\mathscr{D}$ already reduced. Hence, we say that it is obtained from $\mathscr{D}$ by *reduction*. We obtain dually a coreduced decomposition **cored**($\mathscr{D}$), by exchanging the roles of A and B, or, formally,

$$\mathbf{cored}(\mathscr{D}) = (\mathbf{red}(\mathscr{D}^{-1}))^{-1}.$$

We say that a decomposition $\mathscr{D}$ is *deterministic* (resp. *codeterministic*) if $A_i \cap A_j = \varnothing$ (resp. $B_i \cap B_j = \varnothing$) whenever the rectangles $A_i \times B_i$ and $A_j \times B_j$ are distinct. Hence, $\mathscr{D}$ is codeterministic iff $\mathscr{D}^{-1}$ is deterministic.

If $\mathscr{D}$ is deterministic, and if a belongs to A_i, then $B_i = s(a)$. It follows that the second component of any pair (A_i, B_i) in a deterministic decomposition is uniquely determined by the first one. If, in addition, $\mathscr{D}$ is reduced, then each first component of such a pair is uniquely determined by the second as follows: A_i is the set of elements a of A such that $B_i = s(a)$. Hence, any two reduced deterministic decompositions of a relation are equal.

If $\mathscr{D}$ is a deterministic decomposition, then **red**($\mathscr{D}$) is deterministic.

That every relation has a reduced deterministic decomposition is easy to see from the above remarks. Hence, every relation has one and only one such decomposition. It is *canonical* in the sense that it depends only on the relation, and is called its *minimal* decomposition. Dually, every relation has a unique coreduced codeterministic decomposition, that is also canonical, and is called its *cominimal* decomposition. These two canonical decompositions are investigated in detail below. We first look at some examples.

Figures 2 and 3 show respectively a deterministic and a reduced deterministic decomposition of the relation r of Figure 1. The decomposition of Figure 2 is not reduced because we have $B_7 = B_8$ and $B_5 = B_9$, while rectangles 7 and 8, on one hand, and rectangles 5 and 9, on the other, are distinct. The reduced decomposition of Figure 3 is obtained from that of Figure 2 by reduction, that is, by merging rectangles 7 and 8 into a single one, numbered 11. Rectangles 5 and 9 are also merged into a single one, numbered 10. In Figure 3, rectangle 10 is split into two parts, by the necessity of the graphic representation. Figure 4 shows the unique coreduced codeterministic decomposition of r.

The two aforementioned canonical decompositions of a relation s can be obtained from the equivalence relations defined in Definition 1.2. We have

$$\begin{aligned} s &= \bigcup \{[a]_s \times s(a) | a \in A\} \\ &= \bigcup \{s^{-1}(b) \times [b]_{s^{-1}} | b \in B\}. \end{aligned}$$

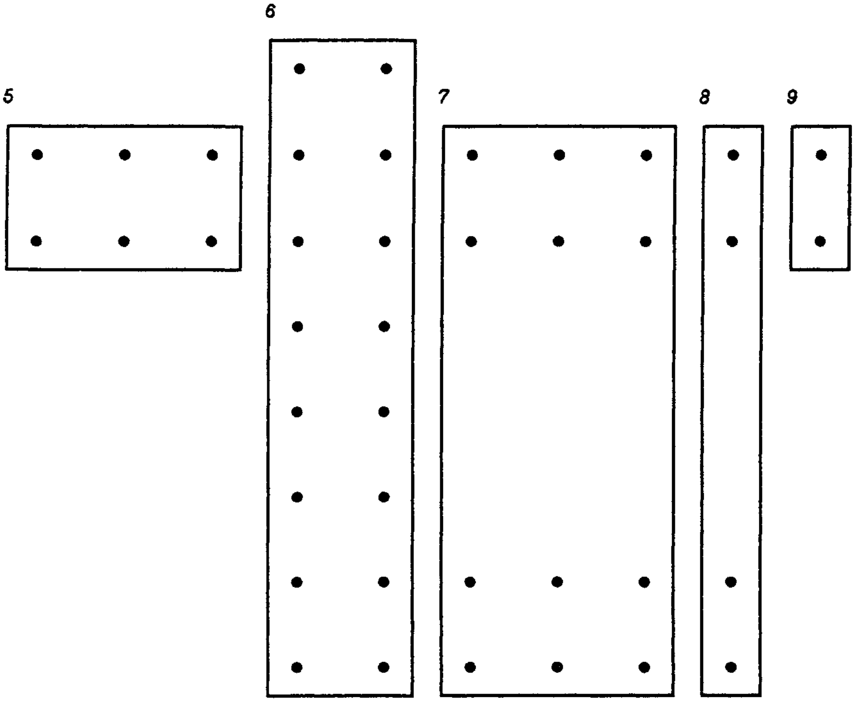

Fig. 2

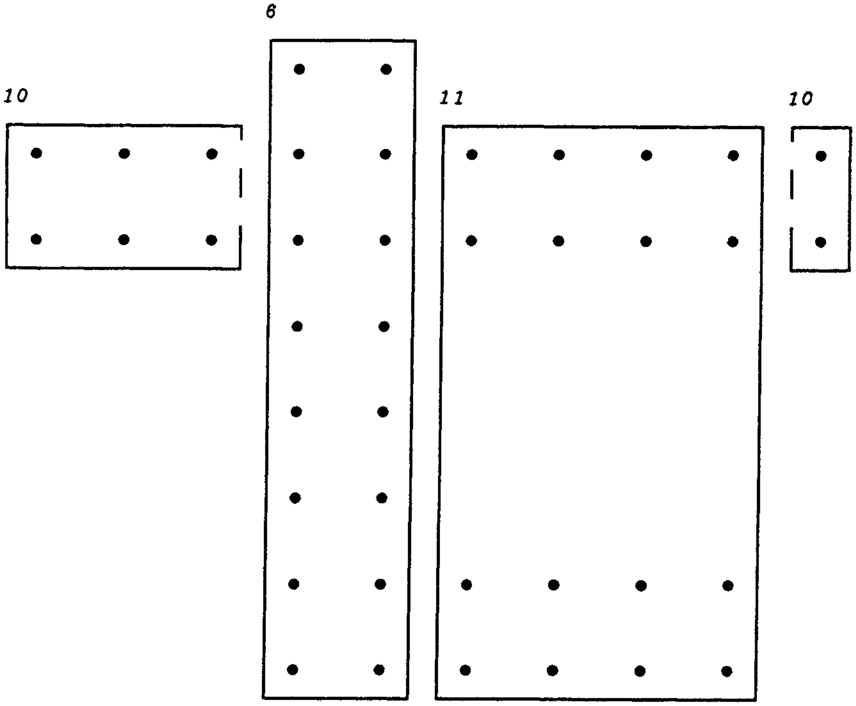

Fig. 3

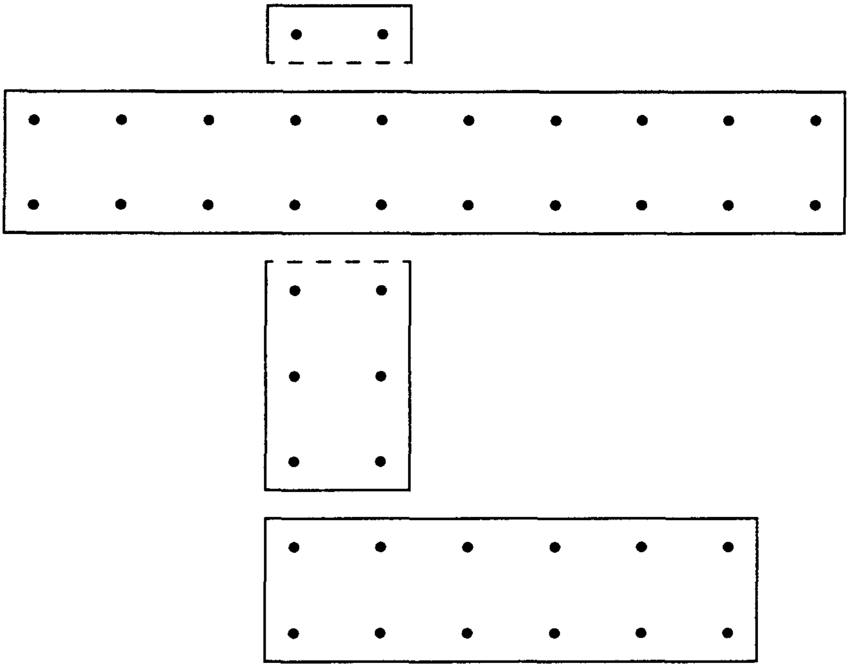

Fig. 4

Hence, the decompositions of s,

$$\mathbf{min}(s) := \{([a]_s, s(a)) \mid a \in A, s(a) \text{ is not empty}\}$$

and

$$\mathbf{comin}(s) := \{(s^{-1}(b), [b]_{s^{-1}}) \mid b \in B, s^{-1}(b) \text{ is not empty}\},$$

are respectively reduced deterministic and coreduced codeterministic. Hence, they are *the unique* such decompositions. As mentioned above, we call them respectively the *minimal* and the *cominimal* decomposition of s.

These decompositions are minimal with respect to the following partial order. If $\mathcal{D}$ and $\mathcal{D}'$ are two decompositions of s, we write $\mathcal{D}' \preccurlyeq \mathcal{D}$ if every rectangle of $\mathcal{D}$ is contained in some rectangle of $\mathcal{D}'$. Let $\mathcal{E}$ be any deterministic decomposition of s. Let $E \times F$ be a rectangle of $\mathcal{E}$ and let $a \in E$. It follows from the determinism of $\mathcal{E}$ that $F = s(a)$ and that $E \subseteq [a]_s$. Hence, every rectangle of $\mathcal{E}$ is contained in some rectangle of $\mathbf{min}(s)$, and $\mathbf{min}(s) \preccurlyeq \mathcal{E}$. We also have $\mathbf{Card}(\mathbf{min}(s)) \leq \mathbf{Card}(\mathcal{E})$. These remarks establish the following minimality properties of $\mathbf{min}(s)$:

Proposition 1.5.

(1) *The decomposition* $\mathbf{min}(s)$ *is the unique reduced deterministic decomposition of s. It is finite iff s is finitely decomposable.*

(2) *It is the least deterministic decomposition of s with respect to the partial order* $\preccurlyeq$.

(3) *If we assume in addition that s is finitely decomposable, then, for every deterministic decomposition $\mathscr{E}$ of s, we have*

$$\mathbf{Card}(\mathbf{min}(s)) < \mathbf{Card}(\mathscr{E}),$$

unless $\mathscr{E} = \mathbf{min}(s)$. Hence, $\mathbf{min}(s)$ *is the unique deterministic decomposition of s of minimal cardinality.*

A similar characterization of $\mathbf{comin}(s)$ holds with respect to *the same* partial order on decompositions.

We now wish to show that the decompositions $\mathbf{min}(s)$ and $\mathbf{comin}(s)$ can be constructed in a uniform way from any decomposition of s. A special case of this construction is that of the minimal deterministic automaton of a regular language, taking a nondeterministic automaton as input. See Section 3.

Definition 1.6 (The Determinization of a Decomposition). Let

$$\mathscr{D} = \{(A_i, B_i) \mid i \in I\}$$

be a decomposition of $s \subseteq A \times B$. For $a \in A$, we let

$$\alpha(a) := \bigcap\{A_i \mid i \in I,\ a \in A_i\} - \bigcup\{A_i \mid i \in I,\ a \notin A_i\}.$$

(This set is the equivalence class of a relative to the equivalence relation $\equiv$ on A such that $x \equiv x'$ iff, for all i in I, $x' \in A_i$ iff $x \in A_i$. It is nonempty.)

Since $\mathscr{D}$ is a decomposition of s, we have

$$s(a) = \bigcup\{B_i \mid i \in I,\ a \in A_i\} \tag{2}$$

and this set is also nonempty whenever a belongs to $s^{-1}(B)$, because in such a case, $s(a)$ is a nonempty union of nonempty sets. We have

$$s = \bigcup\{\alpha(a) \times s(a) \mid a \in s^{-1}(B)\}.$$

Hence, the indexed set

$$\mathbf{det}(\mathscr{D}) := \{(\alpha(a), s(a)) \mid a \in s^{-1}(B)\}$$

is a decomposition of s. We now establish that this decomposition is deterministic. Any two sets of the form $\alpha(a)$ are equal or disjoint since they are equivalence classes of $\equiv$. If $a \equiv a'$, i.e., if $\alpha(a)$ and $\alpha(a')$ are not disjoint, then $s(a) = s(a')$ by equality (2) and the definition of $\equiv$. This proves that $\mathbf{det}(\mathscr{D})$ is deterministic. We call it the *determinization* of $\mathscr{D}$. It is worth noting that the equivalence $\equiv$ is finer than $\sim_s$, hence that $\mathbf{det}(\mathscr{D}) \geqslant \mathbf{min}(s)$.

The *codeterminization* of $\mathscr{D}$, denoted by $\mathbf{codet}(\mathscr{D})$, is defined symmetrically, by exchanging the roles of A and B, i.e., by letting:

$$\mathbf{codet}(\mathscr{D}) := (\mathbf{det}(\mathscr{D}^{-1}))^{-1}.$$

Note that $\mathbf{det}(\mathscr{D})$ is finite if $\mathscr{D}$ is finite, and, more precisely, that $\mathbf{card}(\mathbf{det}(\mathscr{D})) \leq 2^m$ if $m = \mathbf{card}(\mathscr{D})$.

Figure 2 shows $\mathbf{det}(\mathscr{D})$ where $\mathscr{D}$ is the decomposition of the relation r shown in

Figure 1. Figure 3 shows **red(det($\mathscr{D}$))**, that is, by Proposition 1.5(1), the canonical decomposition **min**(r). Figure 4 shows the codeterminization of the decomposition **det**($\mathscr{D}$), shown in Figure 2.

Proposition 1.7.

(1) *Let $\mathscr{D}$ be a decomposition of a relation s. We have*

min(s) = **red(det($\mathscr{D}$))**.

(2) *If, furthermore, $\mathscr{D}$ is codeterministic, then* **det**($\mathscr{D}$) *is reduced and* **det**($\mathscr{D}$) = **min**(s).

Proof. (1) follows from the first part of Proposition 1.5.

(2) Let a, a' be such that $s(a) = s(a')$, which means

$$\bigcup\{B_i \mid a \in A_i, i \in I\} = \bigcup\{B_i \mid a' \in A_i, i \in I\}. \qquad (3)$$

Since $\mathscr{D}$ is codeterministic, if $B_i \cap B_j$ is not empty, then $B_i = B_j$, and $A_i = A_j$. It follows from (3) that $a \in A_i$ iff $a' \in A_i$, for every i in I, Hence, $\alpha(a) = \alpha(a')$. This proves that **det**($\mathscr{D}$) is reduced. □

The following corollary is immediate:

Corollary 1.8. *For every decomposition $\mathscr{D}$ of a relation s, we have*

min(s) = **det(codet($\mathscr{D}$))**.

The following fact is an easy consequence of the definitions. We state it for further reference.

Fact 1.9.

(1) *Let $\mathscr{D}$ be a decomposition of a relation s that is both deterministic and codeterministic. It is reduced, coreduced, minimal, and cominimal. It is unique with these properties.*

(2) *The relation s has such a decomposition iff whenever three pairs (a, b), (a, b'), and (a', b) belong to s, then (a', b') also belongs to s.*

Proof. (1) is clear from the definitions and the previous remarks.

(2) Let s have a decomposition $\mathscr{D}$ that is both deterministic and codeterministic. Let (a, b), (a, b'), and (a', b) belong to s. There is a unique rectangle $C \times D$ of $\mathscr{D}$ such that a belongs to C and b belongs to D. We also have a' in C and b' in D. Hence, (a', b) belongs to $C \times D$, hence to s.

Conversely, let s satisfy the closure condition of Fact 1.9(2). For every (a, b) in s, let $s(a, b) = \{(a', b') \mid (a', b) \text{ and } (a, b') \text{ belong to } s\}$. This set is a rectangle included in s, and the set of all such rectangles forms a deterministic and codeterministic decomposition of s. □

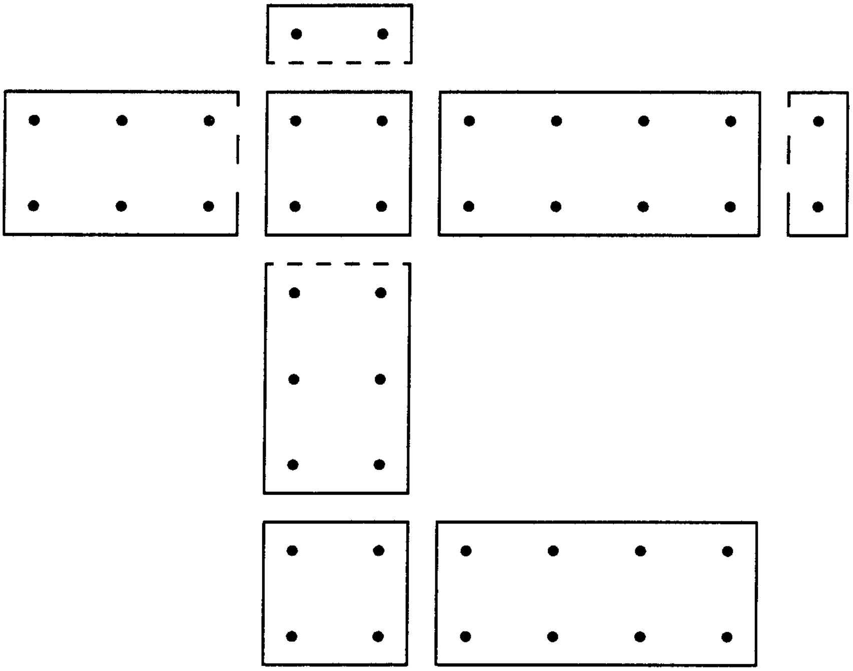

Fig. 5

We finally mention a third type of canonical decomposition that works for arbitrary relations. Any relation s can be written as follows:

$$s = \bigcup\{[a]_s \times [b]_{s^{-1}} \mid (a, b) \in s\}.$$

The corresponding decomposition is larger with respect to $\leqslant$ than both **min**(s) and **comin**(s). It is in general neither deterministic nor codeterministic. It is finite iff s is finitely decomposable.

Figure 5 shows it in the case of the relation r of Figure 1. Its rectangles are the nonempty intersections of the rectangles of Figures 3 and 4.

The reader wishing to see some applications immediately can look at Section 3, where the determinization and the minimization of finite-state automata are considered in this framework. In the next section we apply these definitions and results to recognizable sets in arbitrary algebraic structures.

2. Recognizability in Algebraic Structures

We apply the results of the preceding section and obtain a new characterization of recognizability in arbitrary algebraic structures.

Notation 2.1. Let A be a set and let B be a set of mappings from A into itself. With every subset L of A we associate a relation $\mathbf{s}_L \subseteq A \times B$ defined by

$$(a, b) \in \mathbf{s}_L \quad \text{iff} \quad b(a) \in L.$$

We apply the constructions of the preceding section to the relation $\mathbf{s}_L$. We use the following notations:

$\sim_{\mathbf{s}_L}$ is denoted by $\sim_L$,
$\sim_{\mathbf{s}_L^{-1}}$ is denoted by $\approx_L$,

so that

$$a \sim_L a' \text{ iff, } \forall b \in B,\ b(a) \in L \Leftrightarrow b(a') \in L,$$
$$b \approx_L b' \text{ iff, } \forall a \in A,\ b(a) \in L \Leftrightarrow b'\ (a) \in L.$$

Definition 2.2 (Many-Sorted Algebras). Let S be a finite set called the set of *sorts*. An S-signature is a finite set F of function symbols, such that each symbol is given with a profile. The *profile* of a symbol f in F is a sequence of the form $s_1 \times \cdots \times s_k \longrightarrow s$, where $s_1, \ldots, s_k$, and s belong to S. (The sequence $s_1, \ldots, s_k$ is called the arity of f, and s is called its sort.)

We use terms written with symbols from F and variables. Each variable will have a fixed sort in S. Clearly, terms will have to be well-formed with respect to sorts in a classical way.

An *F-algebra* is an object $\mathbf{A} = \langle (A_s)_{s \in S}, (f_A)_{f \in F} \rangle$. Each A_s is the *domain* of sort s of $\mathbf{A}$, and each f_A is a mapping: $A_{s_1} \times \cdots \times A_{s_k} \longrightarrow A_s$ where $s_1 \times \cdots \times s_k \longrightarrow s$ is the profile of f.

For sake of convenience, we assume that the domains of an algebra are pairwise disjoint. The union of the domains of $\mathbf{A}$ is denoted by A.

A *congruence* on $\mathbf{A}$ as above in an equivalence relation $\approx$ on A such that any two equivalent objects are of the same sort (i.e., belong to the same domain A_s), and that is stable under the operations of $\mathbf{A}$ in a well-known way. We denote by $\approx_s$ the restriction of the relation $\approx$ to the domain A_s. We say that $\approx$ is *finite* if it has finitely many classes. It *saturates* a subset L of A_s if this set is a union of equivalence classes.

Let t be a finite term, constructed with the symbols of F, with finitely many elements of the domains of $\mathbf{A}$, and one variable x, of sort s, having a unique occurrence in t. Let r be the sort of t, i.e., by definition, the sort of its first (topmost) symbol. Then t defines in a classical way a mapping $A_s \longrightarrow A_r$. Such a mapping is called a *linear unary derived operation* of $\mathbf{A}$, or more simply a *linear operation*. We denote by $\mathbf{Lin}(\mathbf{A})$ the set of linear operations of $\mathbf{A}$. The identity mapping on each domain A_s is in $\mathbf{Lin}(\mathbf{A})$ (it is defined by the term reduced to the variable x of sort s). Sort compatible compositions of mappings in $\mathbf{Lin}(\mathbf{A})$ are also in $\mathbf{Lin}(\mathbf{A})$.

Let us recall from [MW], and from [C1] for the many-sorted case, that a set $L \subseteq A_s$ is said to be $\mathbf{A}$-*recognizable* if there exist a finite F-algebra $\mathbf{B}$, a homomorphism $h: \mathbf{A} \to \mathbf{B}$, and a subset C of B_s, such that $L = h^{-1}(C)$, or equivalently that L is a union of classes of a finite congruence.

Our aim is to characterize recognizability in terms of finite decomposability. We do that for one-sorted algebras. We have introduced many-sorted algebras because we embed one-sorted algebras into two-sorted ones. Our construction actually extends in a straightforward manner to many-sorted algebras.

We let **A** be an F-algebra, where F is a signature with a single sort. Its domain is denoted by A. For every subset L of A, we let $\mathbf{s}_L$ be the relation $\subseteq A \times \mathbf{Lin}(\mathbf{A})$ associated with L by Notation 2.1, namely,

$$(a, b) \in \mathbf{s}_L \quad \text{iff} \quad b(a) \in L.$$

The following lemma is well known in the case where **A** is a free monoid [E]. Its proof in the general case is essentially the same (see [C2] or Theorem 7.1, p. 94, of [GS]) and is anyway easy to establish.

Lemma 2.3. *Let $L \subseteq A$. Then $\sim_L$ is a congruence relation on* **A** *that saturates L. This congruence is finite iff L is* **A***-recognizable.*

Definition 2.4. We extend **A** into an algebra **LIN(A)** with two sorts, the sort *ob* of objects, that is nothing but the unique sort of **A**, and the sort *ln* of linear operations.

The signature LIN(F) of **LIN(A)** consists of F augmented with the symbols **o**, **app**, **id**, and $[f, i]$ for all f in F and i in $\mathbb{N}$ such that $1 \le i \le \rho(f)$. (We denote by $\rho(f)$ the number of arguments of f; all of them are of sort *ob*.) We now define the profiles of these symbols, and the operations they denote in **LIN(A)**. We let

$\mathbf{LIN(A)} := \langle A, \mathbf{Lin(A)}, (f_A)_{f\in F}, \mathbf{app}, \mathbf{o}, \mathbf{id}, ([f, i]_{LIN(A)})_{f \in F, 1 \le i \le \rho(f)} \rangle$, where

A is the domain of sort *ob*,

Lin(A) is the domain of sort *ln*,

app denotes the application of an argument to a linear operation; its profile is $ln \times ob \longrightarrow ob$, and it is defined by $\mathbf{app}(b, a) = b(a)$,

o denotes the composition of linear operations, its profile is $ln \times ln \longrightarrow ln$,

id is a constant of type *ln*, denoting the identity operation,

$[f, i]_{LIN(A)}$ is defined for $f \in F$, $1 \le i \le \rho(f)$, its profile is $ob^{\rho(f)-1} \longrightarrow ln$, and

$[f, i]_{LIN(A)}(a_1, a_2, \ldots, a_{\rho(f)-1})$ is the linear operation $\lambda x \cdot f_A(a_1, \ldots, x, \ldots, a_{\rho(f)-1})$, with one occurrence of x at the ith position.

Proposition 2.5. *A subset L of A is* **A***-recognizable iff it is* **LIN(A)***-recognizable.*

Proof. The "if" direction is obvious because **A** is a "part" of **LIN(A)**.

"Only if." Let L be **A**-recognizable. Let $\simeq_L$ be the equivalence relation on **Lin(A)** defined as follows:

$$b \simeq_L b' \quad \text{iff,} \quad \text{for all } c \text{ in } \mathbf{Lin(A)}, \text{ for all } a \text{ in } A, \; c(b(a)) \in L \quad \text{iff} \quad c(b'(a)) \in L.$$

It is easy to verify that the pair $(\sim_L, \simeq_L)$ forms a congruence on **LIN(A)**. Note that $\simeq_L \subseteq \approx_L$. (This inclusion is strict in general.)

We know that $\sim_L$ has finitely many classes (because L is $\mathbf{A}$-recognizable and by Lemma 2.3). It follows from Proposition 1.3 that $\approx_L$ has finitely many classes. We need only prove that the same holds for $\simeq_L$. Let us consider b in $\mathbf{Lin}(\mathbf{A})$ and a, a' in A, such that $a \sim_L a'$. Then $b(a) \sim_L b(a')$. Hence b defines a unique mapping $\bar{b}$ of $A/\sim_L$ into itself. It is easy to see that $b \simeq_L b'$ iff $\bar{b} = \bar{b}'$. Hence $\simeq_L$ is of finite index since $A/\sim_L$ is a finite set.

It follows that L is $\mathbf{LIN}(A)$-recognizable. □

Theorem 2.6. *Let* $\mathbf{A}$ *be a one-sort algebra and let* $L \subseteq A$. *The following conditions are equivalent*:

(i) *L is $\mathbf{A}$-recognizable,*
(ii) *L is $\mathbf{LIN}(\mathbf{A})$-recognizable,*
(iii) *the equivalence $\sim_L$ is finite,*
(iv) *the equivalence $\simeq_L$ is finite,*
(v) *the equivalence $\approx_L$ is finite,*
(vi) *the relation $\mathrm{s}_L \subseteq A \times \mathbf{Lin}(\mathbf{A})$ is finitely decomposable.*

Proof. (i)$\Leftrightarrow$(ii) is Proposition 2.5.
(i)$\Leftrightarrow$(iii) is Lemma 2.3.
(iii)$\Leftrightarrow$(v)$\Leftrightarrow$(vi) is Proposition 1.3.
(i)$\Rightarrow$(iv) by the proof of Proposition 2.5.
(iv)$\Rightarrow$(v) because $\simeq_L \subseteq \approx_L$ (whence the index of $\approx_L$ is at most that of $\simeq_L$). □

The equivalence of (i) and (iv) is proved in [NP] and that of (i) and (v) is proved in [P] in the case where $\mathbf{A}$ is an initial algebra, namely, the algebra of finite binary trees.

3. Applications to Automata on Finite and Infinite Words

We now apply the definitions and results of the previous sections to finite-state automata on finite words, and to finite-state languages of infinite words.

Definition 3.1. Let X be a finite alphabet. For every language $L \subseteq X^*$, we let $\mathrm{s}_L := \{(u, v) | u, v \in X^*, uv \in L\} \subseteq X^* \times X^*$.

An *automaton* is a 5-tuple $\mathscr{A} = \langle X, Q, \delta, Q_\mathrm{I}, Q_\mathrm{F}\rangle$ consisting of an input alphabet X, a possibly infinite set of states Q, a transition relation δ, a set of initial states Q_I, and a set of final states Q_F. The transition relation is any subset of $Q \times X \times Q$. Such an automaton may be nondeterministic, but has no ε-transition.

For every state q in Q, we let $\mathbf{L}(\mathscr{A}, \rightarrow q)$ be the set of words u in X^* for which there is a computation from an initial state to q, and we let $\mathbf{L}(\mathscr{A}, q \rightarrow)$ be the set of words for which there is a computation from q to a final state. All automata are assumed to be *trim*, i.e., to be such that the languages $\mathbf{L}(\mathscr{A}, \rightarrow q)$ and $\mathbf{L}(\mathscr{A}, q \rightarrow)$ are all nonempty. The language $\mathbf{L}(\mathscr{A})$ defined by $\mathscr{A}$ is $\bigcup\{\mathbf{L}(\mathscr{A}, \rightarrow q) | q \in Q_F\}$ and is also equal to $\bigcup\{\mathbf{L}(\mathscr{A}, q \rightarrow) | q \in Q_\mathrm{I}\}$.

The indexed set $\mathbf{D}(\mathscr{A}) := \{(\mathbf{L}(\mathscr{A}, \rightarrow q), \mathbf{L}(\mathscr{A}, q \rightarrow)) | q \in Q\}$ is thus a decomposition of the relation $\mathbf{s}_L$.

Not every decomposition of the relation $\mathbf{s}_L$ is of the form $\mathbf{D}(\mathscr{A})$ for some automaton $\mathscr{A}$ defining L, even if it is finite. We can state the following characterization.

Proposition 3.2. *Let $L \subseteq X^*$ and $\mathscr{D}$ be a decomposition $\{(A_i, B_i) | i \in J\}$ of the relation $\mathbf{s}_L$.*

(1) *There exists an automaton $\mathscr{A}$ such that $\mathbf{D}(\mathscr{A}) = \mathscr{D}$ iff the following conditions hold for all $u \in X^*$, $x \in X$, and $i \in J$:*

(i) *If $ux \in A_i$, then there exists $j \in J$ such that $u \in A_j$, $A_jx \subseteq A_i$, and $xB_i \subseteq B_j$.*

(ii) *If $xu \in B_i$, then there exists $j \in J$ such that $u \in B_j$, $xB_j \subseteq B_i$, and $A_ix \subseteq A_j$.*

(2) *If $\mathscr{D}$ is deterministic, then conditions* (i) *and* (ii) *can be replaced by:*

(iii) *If $ux \in A_i$, then $u \in A_j$ and $A_jx \subseteq A_i$ for some $j \in J$.*

(3) *The minimal decomposition* $\mathbf{min}(\mathbf{s}_L)$ *satisfies these conditions.*

Proof. We only indicate the main steps.

(1) The decomposition associated with an automaton satisfies conditions (i) and (ii). Let us conversely consider $\mathscr{D}$ satisfying them. We define $\mathscr{A} = \langle X, Q, \delta, Q_I, Q_F \rangle$ by letting $Q := J$, $Q_I := \{i | \varepsilon \in A_i\}$, $Q_F := \{i | \varepsilon \in B_i\}$, and δ be the set of triples (i, x, j) such that $A_ix \subseteq A_j$ and $xB_j \subseteq B_i$. We then prove by induction on the length of a word w that

$$w \in A_i \quad \text{iff} \quad w \in \mathbf{L}(\mathscr{A}, \rightarrow i)$$

and

$$w \in B_i \quad \text{iff} \quad w \in \mathbf{L}(\mathscr{A}, i \rightarrow).$$

(2) If a decompostion is deterministic and fulfills (iii), then it also fulfills (i) and (ii).

(3) Easy to verify. □

If two automata are isomorphic, then their associated decompositions are equal, but the converse is not true. Let $\mathscr{A}$ be the automaton of Figure 6. Let $\mathscr{A}'$ be the one obtained from $\mathscr{A}$ by the deletion of one transition labeled by b. Its states are the same as those of $\mathscr{A}$. For every state q, we have

$$\mathbf{L}(\mathscr{A}, \rightarrow q) = \mathbf{L}(\mathscr{A}', \rightarrow q) \quad \text{and} \quad \mathbf{L}(\mathscr{A}, q \rightarrow) = \mathbf{L}(\mathscr{A}', q \rightarrow). \tag{4}$$

It follows that the decompositions associated with $\mathscr{A}$ and $\mathscr{A}'$ are the same although these automata are not isomorphic. Note that these decompositions are the same in a stronger sense than that of Definition 1.1: by (4), not only are the rectangles the same, but their indices also are. Hence, choosing the "strong equality" of decompositions (where two decompositions are equal iff they are equal as multisets of rectangles) would not yield a one-to-one mapping from automata to decompositions.

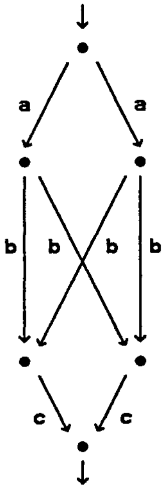

Fig. 6

An automaton $\mathscr{A}$ as in Definition 3.1 is *deterministic* if it has only one initial state, and if, for every $q \in Q$ and $a \in X$, there is at most one transition (q, a, q') in δ. It is *reduced* if, for every q, q' in Q, $\mathbf{L}(\mathscr{A}, q \rightarrow) \neq \mathbf{L}(\mathscr{A}, q' \rightarrow)$ if $q \neq q'$. An automaton is *minimal* if it is both deterministic and reduced. (This terminology is borrowed from [E].)

Fact 3.3. *An automaton* $\mathscr{A}$ *is deterministic iff* $\mathbf{L}(\mathscr{A}, \rightarrow q) \cap \mathbf{L}(\mathscr{A}, \rightarrow q')$ *is empty whenever* $q \neq q'$. *In this case, the decomposition* $\mathbf{D}(\mathscr{A})$ *is deterministic, and it is reduced iff* $\mathscr{A}$ *is minimal.*

The proof of this fact is a straightforward verification. The converse of the second implication does not hold. If $\mathscr{A}$ is a deterministic automaton, and $\mathscr{B}$ consists of two disjoint copies of $\mathscr{A}$, then $\mathbf{D}(\mathscr{B}) = \mathbf{D}(\mathscr{A})$, hence is a deterministic decomposition, although $\mathscr{B}$ is not deterministic (it has two initial states). However, we have:

Fact 3.4.

(1) *Two deterministic automata* $\mathscr{A}$ *and* $\mathscr{A}'$ *are isomorphic iff* $\mathbf{D}(\mathscr{A}) = \mathbf{D}(\mathscr{A}')$.

(2) *Any two minimal automata defining the same language are isomorphic.*

Proof. (1) Let Q and Q' be the sets of states of the two automata. If the associated decompositions are the same, the relation $\mathbf{L}(\mathscr{A}, \rightarrow q) = \mathbf{L}(\mathscr{A}', \rightarrow q')$ defines a bijection of Q onto Q' and, moreover, an isomorphism of the automata. (Easy to verify.)

(2) If two minimal automata define the same language L, the associated decompositions are two minimal decompositions of the relation $\mathbf{s}_L$. Hence, they are equal. Since the automata are deterministic, they are isomorphic by (1). □

For every automaton $\mathscr{A}$, we denote by $\mathbf{rev}(\mathscr{A})$ the automaton obtained by reversing the arrows (in the oriented graph representing $\mathscr{A}$) and exchanging the sets Q_{I} and Q_{F}. It is clear that $\mathbf{L}(\mathbf{rev}(\mathscr{A})) = \mathbf{rev}(\mathbf{L}(\mathscr{A}))$ where $\mathbf{rev}(u)$ denotes the mirror image of a word u, and $\mathbf{rev}(L)$ denotes the set of mirror images of the words of a language L. It is clear that

$$\mathbf{D}(\mathbf{rev}(\mathscr{A})) = \{(\mathbf{rev}(B), \mathbf{rev}(A)) \mid (A, B) \in \mathbf{D}(\mathscr{A})\}.$$

This immediately yields the following fact:

Fact 3.5. *For every automaton $\mathscr{A}$, the decomposition $\mathbf{D}(\mathscr{A})$ is codeterministic if $\mathbf{rev}(\mathscr{A})$ is deterministic.*

We now reformulate in the present framework some classical transformations of automata. Given an automaton $\mathscr{A} = \langle X, Q, \delta, Q_{\mathrm{I}}, Q_{\mathrm{F}}\rangle$, the associated deterministic (trim) automaton $\mathbf{det}(\mathscr{A})$ is constructed as follows:

$$\mathbf{det}(\mathscr{A}) := \langle X, Q', \delta', \{Q_{\mathrm{I}}\}, Q'_{\mathrm{F}}\rangle,$$

where

$$Q' := \{\Delta(u) \mid u \in X^*, \Delta(u) \neq \varnothing\},$$

where, for every word u,

$$\Delta(u) := \{q \mid u \in \mathbf{L}(\mathscr{A}, \rightarrow q)\}.$$

(Since $\mathscr{A}$ is trim, $\Delta(u)$ is not empty iff u is a prefix of some word in $\mathbf{L}(\mathscr{A})$.)

$$Q'_{\mathrm{F}} := \{p \in Q' \mid p \cap Q_{\mathrm{F}} \neq \varnothing\},$$
$$(p, x, p') \in \delta' \quad \text{iff} \quad p, p' \in Q' \text{ and } p' = \{q' \in Q \mid (q, x, q') \in \delta, q \in p\}.$$

Fact 3.6. *For every automaton $\mathscr{A}$,*

$$\mathbf{D}(\mathbf{det}(\mathscr{A})) = \mathbf{det}(\mathbf{D}(\mathscr{A})).$$

Proof. Let $\mathscr{A}$ and $\mathbf{det}(\mathscr{A})$ be as above. Since $\mathbf{det}(\mathscr{A})$ is deterministic and, by Fact 3.3, we have, for every $p \in Q'$,

$$\mathbf{L}(\mathbf{det}(\mathscr{A}), \rightarrow p) = \{u \in X^* \mid p = \Delta(u)\}. \tag{5}$$

Let p be given as $\Delta(w)$ for some w. We easily verify that the right-hand side of equality (5) can be written as follows:

$$\left(\bigcap\{\mathbf{L}(\mathscr{A}, \rightarrow q) \mid q \in p\}\right) - \left(\bigcup\{\mathbf{L}(\mathscr{A}, \rightarrow q) \mid q \notin p\}\right),$$

hence, is equal to $\alpha(w)$, where α is defined as in Definition 1.6 relatively to the decomposition $\mathbf{D}(\mathscr{A})$.

We also obtain, if $p = \Delta(w)$,

$$\begin{aligned}\mathbf{L}(\mathbf{det}(\mathscr{A}), p \rightarrow) &= \bigcup\{\mathbf{L}(\mathscr{A}, q \rightarrow) \mid q \in p\} \\ &= \mathbf{s}_{L(\mathscr{A})}(w) \end{aligned} \tag{6}$$

(see Definition 1.6). It follows from (5), (6), and Definition 1.6 that $\mathbf{D}(\mathbf{det}(\mathscr{A})) = \mathbf{det}(\mathbf{D}(\mathscr{A}))$, since $\mathbf{D}(\mathscr{A})$ is defined as the following decomposition of $\mathbf{s}_{L(\mathscr{A})}$:

$$\mathbf{D}(\mathscr{A}) = \{(\mathbf{L}(\mathscr{A}, \rightarrow q), \mathbf{L}(\mathscr{A}, q \rightarrow)) \mid q \in Q\}. \qquad \square$$

Given a deterministic automaton $\mathscr{A}$ of the general form of Definition 3.1, the associated *reduced* automaton, also called its *minimal* automaton, is

$$\mathbf{min}(\mathscr{A}) := \langle X, Q'', \delta'', \{\bar{q}_{\mathrm{I}}\}, Q_{\mathrm{F}}'' \rangle,$$

where

$$Q'' := \{\bar{q} \mid q \in Q\},$$

and, for every q in Q,

$$\bar{q} := \{q' \in Q \mid \mathbf{L}(\mathscr{A}, q' \rightarrow) = \mathbf{L}(\mathscr{A}, q \rightarrow)\}.$$

The initial state of $\mathbf{min}(\mathscr{A})$ is $\bar{q}_{\mathrm{I}}$ where q_{I} is the (unique) initial state of $\mathscr{A}$. Its set of final states is

$$Q_{\mathrm{F}}'' := \{\bar{q} \mid q \in Q_{\mathrm{F}}\}$$

and

$$\delta'' := \{(\bar{q}, x, \bar{q}') \mid (q, x, q') \in \delta\}.$$

It is clear that, for every $p \in Q''$, $\mathbf{L}(\mathbf{min}(\mathscr{A}), p \rightarrow) = \mathbf{L}(\mathscr{A}, q \rightarrow)$ for all $q \in p$, and that $\mathbf{L}(\mathbf{min}(\mathscr{A}), \rightarrow p) = \bigcup\{\mathbf{L}(\mathscr{A}, \rightarrow q) \mid q \in p\}$. From this observation, and Definition 1.4, we immediately have the following fact:

Fact 3.7. *For every deterministic automaton* $\mathscr{A}$, *we have*

$$\mathbf{D}(\mathbf{min}(\mathscr{A})) = \mathbf{red}(\mathbf{D}(\mathscr{A})).$$

Proposition 3.8. *Let L be the language defined by an automaton* $\mathscr{A}$.

(1) *The automaton* $\mathbf{min}(\mathbf{det}(\mathscr{A}))$ *is the unique deterministic automaton* $\mathscr{B}$ *such that* $\mathbf{D}(\mathscr{B}) = \mathbf{min}(\mathbf{s}_L)$.

(2) *We have* $\mathbf{min}(\mathbf{det}(\mathscr{A})) = \mathbf{det}(\mathbf{rev}(\mathbf{det}(\mathbf{rev}(\mathscr{A}))))$.

Proof. (1) That $\mathbf{D}(\mathbf{min}(\mathbf{det}(\mathscr{A}))) = \mathbf{min}(\mathbf{s}_L)$ follows from Facts 3.6 and 3.7. The unicity follows from Fact 3.4.

(2) The automaton $\mathbf{rev}(\mathscr{A})$ defines $\mathbf{rev}(L)$. So does the deterministic automaton $\mathbf{det}(\mathbf{rev}(\mathscr{A}))$. Hence, $\mathbf{rev}(\mathbf{det}(\mathbf{rev}(\mathscr{A})))$ defines L, and its associated decomposition is codeterministic by Fact 3.5. It follows from Proposition 1.7(2) that $\mathbf{det}(\mathbf{D}(\mathbf{rev}(\mathbf{det}(\mathbf{rev}(\mathscr{A})))))$ is reduced. This decomposition of $\mathbf{s}_L$ is equal to $\mathbf{D}(\mathbf{det}(\mathbf{rev}(\mathbf{det}(\mathbf{rev}(\mathscr{A})))))$ by Fact 3.6. Since it is reduced, the deterministic automaton $\mathbf{det}(\mathbf{rev}(\mathbf{det}(\mathbf{rev}(\mathscr{A}))))$ is reduced, by Fact 3.3. This automaton defines L, it is deterministic and reduced, hence it is equal to $\mathbf{min}(\mathbf{det}(\mathscr{A}))$, by the first part. $\square$

The second assertion of this proposition is known from [B]. We think that its best explanation is the "geometrical fact" stated as Corollary 1.8.

A few words now on regular languages, defined by automata with finitely many states.

Proposition 3.9. *A language $L \subseteq X^*$ is regular iff the relation* $\mathbf{s}_L$ *is finitely decomposable.*

Proof. If L is regular, then the decomposition associated with a finite automaton defining it is finite. Conversely, if $\mathbf{s}_L$ is finitely decomposable, then a finite automaton defining L can be constructed from the finite decomposition $\mathbf{min}(\mathbf{s}_L)$ by Proposition 3.2. □

We now consider what can be obtained from the algebraic framework of Section 2. We let **A** be the algebra $\langle X^*, (\hat{x})_{x\in X}, \varepsilon\rangle$ where ε is a constant denoting the empty word, and $\hat{x}$ is the mapping $u \mapsto ux$ (for $u \in X^*$, $x \in X$). From Theorem 2.6 we get that $L \subseteq X^*$ is **A**-recognizable (i.e., regular) iff any of the three equivalence relations $\sim_L$, $\approx_L$, and $\simeq_L$ has a finite index. These equivalences are the following classical ones:

$$\begin{aligned} u \sim_L u' &\Leftrightarrow \forall w \in X^* \cdot [uw \in L \Leftrightarrow u'w \in L],\\ u \approx_L u' &\Leftrightarrow \forall v \in X^* \cdot [vu \in L \Leftrightarrow vu' \in L],\\ u \simeq_L u' &\Leftrightarrow \forall v, w \in X^* \cdot [vuw \in L \Leftrightarrow vu'w \in L]. \end{aligned}$$

The first one is the canonical right-invariant equivalence (its finiteness characterizes regularity by Nerode's well-known result [N]), the second one is the "dual" of the first (with respect to **rev** as a duality mapping), and the last one is the syntactic congruence of L. We refer the reader to [E] or to [RS] for more details.

We conclude this section by giving some applications to ω-languages.

Definition 3.10. For every set of ω-words $L \subseteq X^\omega$, we let

$$\mathbf{s}_L := \{(u, v) \mid u \in X^*, v \in X^\omega, uv \in L\}.$$

An ω-language (i.e., a set of ω-words) L is *finite-state*, as defined by Trakhtenbrot [T], if the set $\{\mathbf{s}_L(u) \mid u \in X^*\}$ is finite. The *regular* ω-languages, defined by finite-state automata in the sense of McNaughton, Büchi, and Muller, are finite-state. There are uncountably many finite-state ω-languages, hence, some of them are not regular. (See [S] for the proofs of these facts.)

Let **B** be the algebra $\langle X^\omega, (\hat{x})_{x\in X}\rangle$, where $\hat{x}$ is the mapping of X^ω into itself such that $\hat{x}(u) = xu$ for every ω-word u. (This algebra is not finitely generated if X has at least two symbols). We say that an ω-language is *recognizable* iff it is recognizable with respect to **B**.

Proposition 3.11. *For every ω-language $L \subseteq X^\omega$, the following conditions are equivalent*:

(i) *L is finite-state,*

(ii) *L is recognizable,*

(iii) *the equivalence relation* $\approx$ *on* X^* *such that*
$u \approx u'$ *iff* $uw \in L \quad \Leftrightarrow \quad u'w \in L$ *for every* ω*-word* w.
has a finite index,

(iv) *the congruence* $\sim$ *on* $\mathbf{B}$ *such that*
$w \sim w'$ *iff* $uw \in L \quad \Leftrightarrow \quad uw' \in L$ *for every word* u,
has a finite index,

(v) *the congruence* $\simeq$ *on* X^* *such that*
$v \simeq v'$ *iff* $uvw \in L \quad \Leftrightarrow \quad uv'w \in L$ *for every word* u *and every* ω*-word* w, *has a finite index.*

Proof. (i)$\Leftrightarrow$(iii)$\Leftrightarrow$(iv) by Proposition 1.3.
(ii)$\Leftrightarrow$(iv) by Lemma 2.3.
(iv)$\Leftrightarrow$(v) by (iv)$\Leftrightarrow$(v) of Theorem 2.6. □

Note that $\simeq$ is a congruence on words that extends, in a natural way, the syntactical congruence classically associated with a language. See [A] for a refinement of $\simeq$, the finiteness of which characterizes the regularity of an ω-language.

Note also that, in a pair (u, v) as in Definition 3.10, the component v is an element of the domain of the relevant algebra, namely $\mathbf{B}$, while u corresponds to an element of $\mathbf{Lin(B)}$.

4. Applications to Tree Automata

We first review the determinization and the minimization of finite-state frontier-to-root (bottom-up) tree automata. These constructions are well known (see [GS]) and are actually very close to those of Section 3. We then consider certain root-to-frontier tree automata and we reformulate several determinization and minimization results originally presented in [P].

Definition 4.1. We let F be a finite one-sorted signature consisting of a set F_2 of binary symbols and a set F_0 of nullary symbols. We denote by $\mathbf{M}(F)$ the initial F-algebra. A *tree* is an element of the domain of $\mathbf{M}(F)$, that we also denote by $\mathbf{M}(F)$. A *forest* (or a *tree language*) is a set of trees. It follows from this definition that a tree is either a symbol of F_0 or an expression of the form $f(t_1, t_2)$, where f is a symbol in F_2 and t_1 and t_2 are trees.

Trees can also be represented by labeled directed graphs in a well-known way. We thus refer to the set of *nodes* $\mathbf{N}(t)$ of a tree t. One of the nodes of a tree is its *root*. The nodes labeled by symbols from F_0 are the *leaves*. Any other node has a label belonging to F_2 and an ordered pair of successors belonging to $\mathbf{N}(t) \times \mathbf{N}(t)$. The set of leaves of a tree is called its *frontier*.

A *tree automaton* is a tuple $\mathscr{A} = \langle F, Q, \delta, Q_R \rangle$ consisting of a finite signature F as above, a set of *states* Q, a set of *root states* $Q_R \subseteq Q$, and a transition relation

$$\delta \subseteq (F_0 \times Q) \cup (Q \times Q \times F_2 \times Q).$$

If t is a tree in $\mathbf{M}(F)$, a *run* of $\mathscr{A}$ on t is a mapping r: $\mathrm{N}(t) \to Q$ satisfying the following conditions, for every node v of t:

(i) if v is labeled by some symbol a from F_0, then $(a, r(v)) \in \delta$ (in this case v is a leaf of t);

(ii) if v is labeled by some symbol f from F_2, then it has a pair of successors (v_1, v_2) and we require that the 4-tuple $(r(v_1), r(v_2), f, r(v))$ belongs to δ.

We denote by $\mathbf{T}(\mathscr{A}, q)$ (or by $\mathbf{T}(q)$ if the context makes $\mathscr{A}$ clear) the set of trees t on which $\mathscr{A}$ has a run r such that $r(\mathbf{root}(t)) = q$. We denote by $\mathbf{T}(\mathscr{A})$ the forest *recognized* by $\mathscr{A}$, and defined as $\bigcup\{\mathbf{T}(q) \mid q \in Q_{\mathrm{R}}\}$.

It is well known that a forest T is $\mathbf{M}(F)$-recognizable (in the sense of Section 2) iff it is recognized by a *finite* tree automaton, i.e., by a tree automaton with finitely many states. (See [GS], [C1], [NP], and [P].)

The set $\mathbf{Lin}(\mathbf{M}(F))$ is in one-to-one correspondence with the set of trees in $\mathbf{M}(F \cup \{x\})$ having one and only one occurrence of the variable x. These special trees are called *contexts*. (They are called *pointed* trees in [NP].) Theorem 2.6 yields a characterization of recognizable sets of trees, some parts of which can be found in [NP] and in [P].

The set of contexts relative to F is denoted by $\mathbf{Ctxt}(F)$. We denote by $c[t]$ the tree obtained by substituting the tree t for the variable x in the context c. Note that x is a context, called the *identity* context (since $x[t] = t$). Since contexts are trees over a larger set of symbols, we can refer to them with the terminology that we already use for trees.

By a *q-run* r of $\mathscr{A}$ on a context c, we mean a mapping r: $\mathrm{N}(c) \to Q$ satisfying conditions (i) and (ii) above, together with the following two conditions for every v in $\mathrm{N}(c)$:

(iii) if v is labeled by x, then $r(v) = q$,

(iv) $r(\mathbf{root}(c)) \in Q_{\mathrm{R}}$.

Note that $\mathscr{A}$ has a q-run on the context x iff $q \in Q_{\mathrm{R}}$.

We let $\mathbf{C}(\mathscr{A}, q)$ denote the set of contexts on which $\mathscr{A}$ has a q-run. (This set is denoted by $\mathbf{C}(q)$ if $\mathscr{A}$ is clear from the context.) Let us observe that

$$\mathbf{C}(q) \subseteq \{c \in \mathbf{Ctxt}(F) \mid c[t] \in \mathbf{T}(\mathscr{A})\}, \tag{7}$$

for every tree t in $\mathbf{T}(q)$, and that if $(p, p', f, q) \in \delta$, $c \in \mathbf{C}(q)$, then

$$\mathbf{T}(p') \subseteq \{t \in \mathbf{M}(F) \mid c[f(x, t)] \in \mathbf{C}(p)\}. \tag{8}$$

All tree automata are assumed to be *trim*, i.e., to be such that the sets $\mathbf{T}(q)$ and $\mathbf{C}(q)$ are nonempty for all states q.

With every forest $T \subseteq \mathbf{M}(F)$, we associate a relation $\mathrm{s}_{\mathrm{T}} := \{(t, c) \in \mathbf{M}(F) \times \mathbf{Ctxt}(F) \mid c[t] \in T\}$. With every automaton $\mathscr{A}$, we associate the indexed set of pairs of sets:

$$\mathbf{D}(\mathscr{A}) := \{(\mathbf{T}(q), \mathbf{C}(q)) \mid q \in Q\}.$$

Fact 4.2. $\mathbf{D}(\mathscr{A})$ *is a decomposition of the relation* $\mathbf{s}_{T(\mathscr{A})}$.

A characterization of the decompositions of a relation of the form $\mathbf{s}_T$ that correspond to tree automata, fully analogous to the one of Proposition 3.2, could be given. It is quite complicated to write, and we omit it.

An automaton $\mathscr{A}$ as above is *frontier-to-root deterministic* (*fr-deterministic*) if there are no two tuples in δ of the form (a, q) and (a, q'), or of the form (q_1, q_2, f, q) and (q_1, q_2, f, q') for any a in F_0, f in F_2, and q_1, q_2 in Q, with $q \neq q'$. If $\mathscr{A}$ is fr-deterministic, then it has at most one run on every tree t in $\mathbf{M}(F)$, and on every context c in $\mathbf{Ctxt}(F)$, it has at most one q-run for each state q. If, furthermore, $\mathscr{A}$ is finite (and given), then its run on t or its q-run on c can be computed *deterministically from the leaves to the root.* (This justifies the terminology.)

A tree automaton is *fr-reduced*, if, for every two distinct states q and q', the sets $\mathbf{C}(q)$ and $\mathbf{C}(q')$ are not equal. It is *fr-minimal* if it is fr-deterministic and fr-reduced.

Fact 4.3. *A tree automaton* $\mathscr{A}$ *is fr-deterministic iff, for any two distinct states* q *and* q', *the sets* $\mathbf{T}(q)$ *and* $\mathbf{T}(q')$ *are disjoint. If this is the case, then the decomposition* $\mathbf{D}(\mathscr{A})$ *is deterministic and, furthermore,* $\mathscr{A}$ *is fr-minimal iff* $\mathbf{D}(\mathscr{A})$ *is reduced.*

Fact 4.4. *Two fr-deterministic tree automata* $\mathscr{A}$ *and* $\mathscr{A}'$ *are isomorphic iff* $\mathbf{D}(\mathscr{A}) = \mathbf{D}(\mathscr{A}')$. *For every forest* T, *there exists a unique fr-minimal tree automaton* $\mathscr{B}$, *such that* $\mathbf{D}(\mathscr{B}) = \mathbf{min}(\mathbf{s}_T)$.

Note the similarity with the case of languages in Facts 3.3 and 3.4: here, the sets of trees $\mathbf{T}(q)$ play the role of the languages $\mathbf{L}(\rightarrow q)$, while the sets of contexts $\mathbf{C}(q)$ play that of the languages $\mathbf{L}(q \rightarrow)$.

It is well known that fr-deterministic tree automata can be determinized and minimized (see [GS]). The formal constructions are so similar to the ones for (word) automata recalled in Section 3, that we do not give them. In particular, the subset construction can be performed on tree automata and makes it possible to transform a tree automaton $\mathscr{A}$ into an equivalent fr-deterministic one, denoted by $\mathbf{fr\text{-}det}(\mathscr{A})$. Similarly, an fr-deterministic automaton $\mathscr{A}$ can be transformed into an fr-minimal one, denoted by $\mathbf{fr\text{-}min}(\mathscr{A})$. As in the case of words, we have:

Fact 4.5. *For every tree automaton* $\mathscr{A}$:

$$\mathbf{D}(\mathbf{fr\text{-}det}(\mathscr{A})) = \mathbf{det}(\mathbf{D}(\mathscr{A})).$$

If $\mathscr{A}$ *is fr-deterministic, then*

$$\mathbf{D}(\mathbf{fr\text{-}min}(\mathscr{A})) = \mathbf{red}(\mathbf{D}(\mathscr{A})).$$

Note that, for every tree automaton $\mathscr{A}$, the decomposition $\mathbf{codet}(\mathbf{D}(\mathscr{A}))$ exists. It does not correspond to any tree automaton in general. Since we do not have on trees any operation analogous to the mirror image on words (denoted by **rev**), the results of Section 3 involving **rev** have no counterpart concerning forests and tree automata.

Definition 4.6 (Root-to-Frontier Determinism). An automaton $\mathscr{A}$ as in Definition 4.1 is *root-to-frontier deterministic* (*rf-deterministic*) if it has one and only one root state and if there are no two distinct tuples of the forms

$$(q_1, q_2, f, q) \quad \text{and} \quad (q'_1, q'_2, f, q)$$

in its transition relation δ.

It is clear that if $\mathscr{A}$ is rf-deterministic, then $\mathbf{C}(\mathscr{A}, q) \cap \mathbf{C}(\mathscr{A}, q')$ is empty whenever $q \neq q'$, hence that $\mathbf{D}(\mathscr{A})$ is codeterministic. It is well known that some sets of trees, like the finite one $\{f(a, a), f(b, b)\}$, cannot be recognized by any rf-deterministic automaton.

We consider alternative less restrictive determinism conditions, ensuring in particular, for every automaton, the existence and unicity (up to isomorphism) of the minimal automaton recognizing the same forest.

We say that an automaton $\mathscr{A}$ as above is *l-deterministic* if it has a unique root state, and if, for any two distinct tuples (q_1, q_2, f, q) and (q'_1, q'_2, f, q) in δ, we have

$$\text{(L)} \quad \begin{cases} \mathbf{T}(\mathscr{A}, q_1) \neq \mathbf{T}(\mathscr{A}, q'_1), \\ \mathbf{T}(\mathscr{A}, q_2) \cap \mathbf{T}(\mathscr{A}, q'_2) = \varnothing. \end{cases}$$

By exchanging the roles of (q_1, q'_1) and (q_2, q'_2), we get the analogous notion of *r-determinism*. Finally, an automaton is *lr-deterministic* if it is both l- and r-deterministic, that is if it has a unique root state, and if it satisfies the following condition, for every two distinct tuples as above:

$$\text{(LR)} \quad \begin{cases} \mathbf{T}(\mathscr{A}, q_1) \cap \mathbf{T}(\mathscr{A}, q'_1) = \varnothing, \\ \mathbf{T}(\mathscr{A}, q_2) \cap \mathbf{T}(\mathscr{A}, q'_2) = \varnothing. \end{cases}$$

(This condition implies that the decompositions

$$\{(\mathbf{T}(\mathscr{A}, p), \mathbf{T}(\mathscr{A}, p')) \mid (p, p', f, q) \in \delta\}$$

are deterministic and codeterministic for all f in F_2 and q in Q. See the proof of Proposition 4.9 below.)

Lemma 4.7.

(1) *An automaton $\mathscr{A}$ is lr-deterministic iff, for every two distinct states q and q', the sets $\mathbf{C}(q)$ and $\mathbf{C}(q')$ are disjoint. If this is the case, then the associated decomposition $\mathbf{D}(\mathscr{A})$ is codeterministic.*

(2) *Two lr-deterministic tree automata are isomorphic iff the associated decompositions are equal.*

Proof. (1) Let $\mathscr{A}$ be such that $\mathbf{C}(q) \cap \mathbf{C}(q') = \varnothing$ for every two distinct states q and q'. Let us consider two distinct tuples (q_1, q_2, f, q) and (q'_1, q'_2, f, q) in the transition relation δ of $\mathscr{A}$. We prove that $\mathbf{T}(q_1) \cap \mathbf{T}(q'_1) = \varnothing$ and $\mathbf{T}(q_2) \cap \mathbf{T}(q'_2) = \varnothing$.

Let us assume without loss of generality that $q_1 \neq q'_1$. Let also c be some context in $\mathbf{C}(q)$. Then, by formula (8) of Definition 4.1,

$$\mathbf{T}(q_2) \subseteq \{t \mid c[f(x, t)] \in \mathbf{C}(q_1)\},$$

and similarily with q'_2 and q'_1. Since $\mathrm{C}(q_1) \cap \mathrm{C}(q'_1)$ is empty, we get that $\mathrm{T}(q_2) \cap \mathrm{T}(q'_2)$ is also empty. We obtain in particular that $q_2 \neq q'_2$, hence, by a symmetric argument, we obtain that $\mathrm{T}(q_1)$ and $\mathrm{T}(q'_1)$ have an empty intersection. Hence, we have established conditions (LR).

If $\mathscr{A}$ had two distinct root states q and q', then we would have $x \in \mathrm{C}(q) \cap \mathrm{C}(q')$, contradicting the initial assumption on $\mathscr{A}$. Hence, $\mathscr{A}$ is lr-deterministic.

Let us now assume that $\mathscr{A}$ is lr-deterministic. We prove that for every context c there is at most one q such that $c \in \mathrm{C}(q)$.

If $c = x$, then $c \in \mathrm{C}(q)$ iff $q \in Q_{\mathrm{R}}$, and we know that Q_{R} is singleton. Otherwise, $c = c'[f(x, t)]$ or, symmetrically, $c = c'[f(t, x)]$. Let us assume the first. By induction, there is at most one state q' such that $c' \in \mathrm{C}(q')$. If $c \in \mathrm{C}(q_1) \cap \mathrm{C}(q'_1)$, then this means that there are two tuples (q_1, q_2, f, q') and $(q'_1, q'_2, f\ q')$ in δ such that $t \in \mathrm{T}(q_2) \cap \mathrm{T}(q'_2)$. By the definition of lr-determinism, we get $q_1 = q'_1$. The proof is similar in the symmetric case.

(2) Let $\mathscr{A}$ and $\mathscr{A}'$ be two lr-deterministic automata with respective sets of states Q and Q' and equal decompositions. The relation $\mathrm{C}(\mathscr{A}, q) = \mathrm{C}(\mathscr{A}', q')$ (or equivalently $\mathrm{C}(\mathscr{A}, q) \cap \mathrm{C}(\mathscr{A}', q') \neq \varnothing$) defines an isomorphism of the two automata. We omit the details. □

Definition 4.8 (Minimal lr-Deterministic Automata). We say that an lr-deterministic automaton $\mathscr{A}$ is *minimal* iff, for every two distinct states q and q', we have $\mathrm{T}(q) \neq \mathrm{T}(q')$. By Lemma 4.7, this is equivalent to requiring that the decomposition $\mathrm{D}(\mathscr{A})$ is codeterministic and coreduced, hence, equivalently, that it is the (unique) cominimal decomposition of $s_{T(\mathscr{A})}$. This shows, by the second part of Lemma 4.7, that there exists at most one minimal lr-deterministic automaton recognizing a forest. (The unicity should be understood up to isomorphism.)

Proposition 4.9. *If $\mathscr{A}$ is an lr-deterministic automaton, then there exists a minimal lr-deterministic automaton $\mathscr{B}$ recognizing* $\mathrm{T}(\mathscr{A})$. *It is unique with these properties, and* $\mathbf{D}(\mathscr{B}) = \mathbf{cored}(\mathbf{D}(\mathscr{A}))$.

Proof. Let $\mathscr{A} = \langle F, Q, \delta, \{q_{\mathrm{R}}\}\rangle$ be lr-deterministic. Let $\sim$ be the equivalence relation on Q such that $q \sim q'$ iff $\mathrm{T}(q) = \mathrm{T}(q')$. We let $\mathscr{B} = \langle F, Q/\sim, \eta, \{[q_{\mathrm{R}}]\}\rangle$ where $Q/\sim$ is the quotient set of Q by $\sim$, where $[q_{\mathrm{R}}]$ is the equivalence class of q_{R}, and where η is defined as the set of tuples of the form $([q_1], [q_2], f, [q])$ or $(a, [q])$ where (q_1, q_2, f, q) or, respectively (a, q), belongs to δ.

For every $q \in Q$ and $f \in F_2$, we let

$$\delta^{-1}(f, q) = \{(q_1, q_2) | (q_1, q_2, f\ q) \in \delta\}.$$

For every subset T of $\mathbf{M}(F)$, we let

$$f^{-1}(T) = \{(t_1, t_2) | f(t_1, t_2) \in T\}.$$

We denote by $\Delta^{-1}(f, q)$ the decomposition

$$\Delta^{-1}(f, q) = \{(\mathrm{T}(q_1), \mathrm{T}(q_2)) | (q_1, q_2) \in \delta^{-1}(f, q)\}$$

of the relation $f^{-1}(\mathrm{T}(q))$.

Claim 1. *If $q \sim q'$, if (r, s, f, q) belongs to δ, then there is some (r', s', f, q') in δ such that $r \sim r'$ and $s \sim s'$.*

Proof. Since $q \sim q'$, we have $f^{-1}(\mathbf{T}(q)) = f^{-1}(\mathbf{T}(q'))$. The decompositions $\Delta^{-1}(f, q)$ and $\Delta^{-1}(f, q')$ are both deterministic and codeterministic by condition (LR). Hence they are equal. (See Fact 1.9.) The rectangle $\mathbf{T}(r) \times \mathbf{T}(s)$ of the former is equal to some rectangle $\mathbf{T}(r') \times \mathbf{T}(s')$ of the latter, and (r', s', f, q) is the desired tuple. □

Claim 2. $\mathbf{T}(\mathscr{B}, [q]) = \mathbf{T}(\mathscr{A}, q)$ *for every $q \in Q$.*

Proof. Let $\varphi\colon Q \to Q/\sim$ be the canonical surjection that maps any q to its equivalence class $[q]$. Let $t \in \mathbf{T}(\mathscr{A}, q)$ and let $r\colon \mathrm{N}(t) \to Q$ be a run of $\mathscr{A}$ on t. Then $\varphi \circ r$ is a run of $\mathscr{B}$ on t, and we have $t \in \mathbf{T}(\mathscr{B}, \varphi(q))$ ($= \mathbf{T}(\mathscr{B}, [q])$). This proves the inclusion $\supseteq$.

For the other, consider t in $\mathbf{T}(\mathscr{B}, [q])$ and a run r of $\mathscr{B}$ on t. By a repeated use of Claim 1 and by traversing t from the root to the leaves, we can construct from r a run of $\mathscr{A}$ on t with root state q. This proves that $t \in \mathbf{T}(\mathscr{A}, q)$. □

It follows in particular that $\mathbf{T}(\mathscr{B}) = \mathbf{T}(\mathscr{A})$, that $\mathscr{B}$ is lr-deterministic, and, in addition, that $\mathscr{B}$ is minimal lr-deterministic.

Claim 3. $\mathbf{C}(\mathscr{B}, [q]) = \bigcup\{\mathbf{C}(\mathscr{A}, q') \mid q' \sim q\}$ *for every $q \in Q$.*

Proof. If $q \sim q'$, we obtain $\mathbf{C}(\mathscr{A}, q) \subseteq \mathbf{C}(\mathscr{B}, [q])$ as in the first part of the proof of Claim 2.

For proving the other inclusion, namely $\subseteq$, let us consider c in $\mathbf{C}(\mathscr{B}, [q])$. By the construction of the second part of the proof of Claim 2, we can find a run of $\mathscr{A}$ on c. Hence, $c \in \mathbf{C}(\mathscr{A}, q')$ for some q'. Hence, by the first part, c is in $\mathbf{C}(\mathscr{B}, [q'])$. Since $\mathscr{B}$ is lr-deterministic, we have $[q'] = [q]$ by Lemma 4.7, hence, $q' \sim q$, as was to be proved. □

It follows then from Claims 2 and 3 that $\mathbf{D}(\mathscr{B}) = \mathbf{cored}(\mathbf{D}[\mathscr{A}])$. □

Next we characterize the forests defined by lr-deterministic automata.

Definition 4.10 (Homogeneous forests). For every $T \subseteq \mathbf{M}(F)$ and every c in $\mathbf{Ctxt}(F)$, we let

$$c^{-1}(T) := \{t \mid c[t] \in T\}.$$

Let us recall that, for $f \in F_2$, we let $f^{-1}(T)$ denote

$$\{(t_1, t_2) \in \mathbf{M}(F) \times \mathbf{M}(F) \mid f(t_1, t_2) \in T\}.$$

We say that T is *homogeneous* if, for every c in $\mathbf{Ctxt}(F)$ and every f in F_2, the relation $f^{-1}(c^{-1}(T))$ on $\mathbf{M}(F)$ has a (possibly infinite) decomposition that is both deterministic and codeterministic. As noticed in Fact 1.9, this is equivalent to requiring that, for every t_1, t_2, t'_1, t'_2 in $\mathbf{M}(F)$, if (t_1, t_2), (t_1, t'_2), and (t'_1, t_2) belong to $f^{-1}(c^{-1}(T))$, then (t'_1, t'_2) also belongs to $f^{-1}(c^{-1}(T))$. In this way we obtain the original definition of [P].

Proposition 4.11. *A forest is homogeneous if it is defined by an lr-deterministic automaton.*

Proof. Let $L = \mathrm{T}(\mathscr{A})$ for some lr-deterministic automaton $\mathscr{A}$. Let $w = f^{-1}(c^{-1}(L))$ for some context c and some $f \in F_2$, with $w \neq \varnothing$. By Lemma 4.7, there is a unique state q of $\mathscr{A}$ such that $c \in \mathrm{C}(\mathscr{A}, q)$. Hence $\mathrm{T}(\mathscr{A}, q) = c^{-1}(L)$, and

$$w = \bigcup\{\mathrm{T}(\mathscr{A}, q_1) \times \mathrm{T}(\mathscr{A}, q_2) \mid (q_1, q_2) \in \delta^{-1}(f, q)\}.$$

Since $\mathscr{A}$ is lr-deterministic $\mathrm{T}(\mathscr{A}, q_i) \cap \mathrm{T}(\mathscr{A}, q'_i) = \varnothing$, for $i = 1, 2$, where (q_1, q_2) and (q'_1, q'_2) are distinct pairs in $\delta^{-1}(f, q)$. It follows that $\{(\mathrm{T}(\mathscr{A}, q_1), \mathrm{T}(\mathscr{A}, q_2)) \mid (q_1, q_2) \in \delta^{-1}(f, q)\}$ is a deterministic and codeterministic decomposition of w. Hence L is homogeneous. □

Our next proposition yields the converse of the previous one.

Proposition 4.12. *Let $\mathscr{A}$ be an automaton defining a homogeneous forest L. There exists a minimal lr-deterministic automaton $\mathscr{B}$ that defines L and is such that* $\mathbf{D}(\mathscr{B}) = \mathbf{codet}(\mathbf{det}(\mathbf{D}(\mathscr{A})))$.

Proof. Let $\mathscr{A} = \langle F, Q, \delta, Q_{\mathrm{R}}\rangle$ be an automaton defining a homogeneous forest L. For every context c in $\mathbf{Ctxt}(F)$, the forest $c^{-1}(L)$ is homogeneous. We let

$$\mathrm{Q}(c) := \{q \in Q \mid \mathrm{T}(\mathscr{A}, q) \subseteq c^{-1}(L)\}.$$

It is not hard to verify that $c^{-1}(L) = \mathrm{T}(\mathscr{A}, \mathrm{Q}(c))$, where, for every $\alpha \subseteq Q$, we let $\mathrm{T}(\mathscr{A}, \alpha)$ denote $\bigcup\{\mathrm{T}(\mathscr{A}, q) \mid q \in \alpha\}$.

We now let $\mathscr{B} = \langle F, Q', \delta', \{\mathrm{Q}(x)\}\rangle$ where $Q' = \{\mathrm{Q}(c) \mid c^{-1}(L) \neq \varnothing\}$. The unique root state of $\mathscr{B}$ is $\mathrm{Q}(x)$, associated with the identity context x. We have $Q_{\mathrm{R}} \subseteq \mathrm{Q}(x)$ but the inclusion may be strict.

We define δ' as the set of tuples of the following two possible forms:

(i) (a, γ) where a $\in \mathrm{T}(\mathscr{A}, \gamma)$, $\gamma \in Q'$.

(ii) $(\alpha, \beta, f, \gamma)$ where we have

$$\alpha = \{q \in Q \mid f(\mathrm{T}(\mathscr{A}, q), s_2) \subseteq \mathrm{T}(\mathscr{A}, \gamma)\},$$
$$\beta = \{q \in Q \mid f(s_1, \mathrm{T}(\mathscr{A}, q)) \subseteq \mathrm{T}(\mathscr{A}, \gamma)\},$$
$$s_1, s_2 \in \mathbf{M}(F),\ f(s_1, s_2) \in \mathrm{T}(\mathscr{A}, \gamma), \text{ and } \gamma \in Q'.$$

(We say that this 4-tuple is *associated* with the pair (s_1, s_2).)

In case (ii) we must prove that α and β belong to Q'. It is actually easy to verify that $\alpha = \mathrm{Q}(c')$, where c' is the context $c[f(x, s_2)]$. Since $f(s_1, s_2) \in \mathrm{T}(\mathscr{A}, \gamma)$, $\alpha \neq \varnothing$, hence $\alpha \in Q'$. Similarly, $\beta \in Q'$. It follows that $s_1 \in \mathrm{T}(\mathscr{A}, \alpha)$ and that $s_2 \in \mathrm{T}(\mathscr{A}, \beta)$.

Claim. *For every $\alpha \in Q'$, we have* $\mathrm{T}(\mathscr{B}, \alpha) = \mathrm{T}(\mathscr{A}, \alpha)$.

Proof. We prove by induction on the structure of t that, for every t in $\mathbf{M}(F)$,

$$t \in \mathbf{T}(\mathscr{B}, \alpha) \quad \Leftrightarrow \quad t \in \mathbf{T}(\mathscr{A}, \alpha).$$

The case where $t = a \in F_0$ follows immediately from the definition of δ'. Otherwise, let $t = f(t_1, t_2)$.

Let us assume that $t \in \mathbf{T}(\mathscr{A}, \alpha)$. A tuple in δ' of the form $(\beta, \gamma, f, \alpha)$ with $t_1 \in \mathbf{T}(\mathscr{A}, \beta)$, $t_2 \in \mathbf{T}(\mathscr{A}, \gamma)$, can be associated with (t_1, t_2) by case (ii) of the definition of δ'. By induction we have $t_1 \in \mathbf{T}(\mathscr{B}, \beta)$ and $t_2 \in \mathbf{T}(\mathscr{B}, \gamma)$, hence, $t = f(t_1, t_2) \in \mathbf{T}(\mathscr{B}, \alpha)$.

Conversely, let us assume that $t = f(t_1, t_2) \in \mathbf{T}(\mathscr{B}, \alpha)$. We have a tuple $(\beta, \gamma, f, \alpha)$ such that $t_1 \in \mathbf{T}(\mathscr{B}, \beta)$, $t_2 \in \mathbf{T}(\mathscr{B}, \gamma)$, associated by (ii) with a pair (s_1, s_2). Hence, we have $t_1 \in \mathbf{T}(\mathscr{A}, \beta)$ and $t_2 \in \mathbf{T}(\mathscr{A}, \gamma)$ by the induction hypothesis. We also have

$$\begin{array}{ll} f(s_1, s_2) \in \mathbf{T}(\mathscr{A}, \alpha), & \\ f(\mathbf{T}(\mathscr{A}, \beta), s_2) \subseteq \mathbf{T}(\mathscr{A}, \alpha), & \text{whence } \; f(t_1, s_2) \in \mathbf{T}(\mathscr{A}, \alpha), \\ f(s_1, \mathbf{T}(\mathscr{A}, \gamma)) \subseteq \mathbf{T}(\mathscr{A}, \alpha), & \text{whence } \; f(s_1, t_2) \in \mathbf{T}(\mathscr{A}, \alpha). \end{array}$$

Since $\mathbf{T}(\mathscr{A}, \alpha)$ is homogeneous, we have $f(t_1, t_2) \in \mathbf{T}(\mathscr{A}, \alpha)$ as was to be proved. □

It follows from the claim that $\mathbf{T}(\mathscr{B}) = \mathbf{T}(\mathscr{A}, \mathbf{Q}(x)) = L$. We now prove that $\mathscr{B}$ is lr-deterministic. Note first that $\mathscr{B}$ has a single root state. Let $(\beta, \gamma, f, \alpha)$ and $(\beta', \gamma', f, \alpha)$ be two tuples in δ' associated with pairs (s_1, s_2) and (s_1', s_2'). Let t belong to $\mathbf{T}(\mathscr{B}, \beta) \cap \mathbf{T}(\mathscr{B}, \beta')$, assumed to be nonempty.

We have

$$\gamma = \{q \in Q \mid f(s_1, \mathbf{T}(\mathscr{A}, q)) \subseteq \mathbf{T}(\mathscr{A}, \alpha)\}.$$

Since $f(s_1, s_2)$ and $f(t, s_2)$ belong to $\mathbf{T}(\mathscr{A}, \alpha)$, we also have, since $\mathbf{T}(\mathscr{A}, \alpha)$ is homogeneous,

$$f(s_1, \mathbf{T}(\mathscr{A}, q)) \subseteq \mathbf{T}(\mathscr{A}, \alpha) \quad \text{iff} \quad f(t, \mathbf{T}(\mathscr{A}, q)) \subseteq \mathbf{T}(\mathscr{A}, \alpha),$$

hence,

$$\gamma = \{q \in Q \mid f(t, \mathbf{T}(\mathscr{A}, q)) \subseteq \mathbf{T}(\mathscr{A}, \alpha)\}.$$

The same argument applies to γ', hence, $\gamma' = \gamma$.

By using some t' in $\mathbf{T}(\mathscr{B}, \gamma)$, which is equal to $\mathbf{T}(\mathscr{B}, \gamma')$, we get by a similar argument that $\beta = \beta'$. Hence, $\mathscr{B}$ is lr-deterministic.

Finally, we show that $\mathscr{B}$ is minimal. Let us consider $\mathbf{T}(\mathscr{B}, \alpha)$ with $\alpha = \mathbf{Q}(c)$ for some c. Every state q, such that $\mathbf{T}(\mathscr{A}, q) \subseteq \mathbf{T}(\mathscr{B}, \alpha)$, belongs to α (because, then, $c[t]$ belongs to L for every t in $\mathbf{T}(\mathscr{A}, q)$, whence $q \in \mathbf{Q}(c) = \alpha$). Hence, if $\mathbf{T}(\mathscr{B}, \alpha) = \mathbf{T}(\mathscr{B}, \alpha')$ we obtain by the claim and this remark that $\alpha = \mathbf{Q}(c) = \alpha'$.

It follows that $\mathbf{D}(\mathscr{B})$ is the unique cominimal decomposition of $\mathbf{s}_L$. Hence, it must be equal to $\mathbf{codet}(\mathbf{det}(\mathbf{D}(\mathscr{A})))$ by Corollary 1.8. □

Proposition 4.12 is also a consequence of Proposition 4.15 given below, because it can be proved that if an l-deterministic automaton recognizes a homogeneous forest, then this automaton is lr-deterministic. See [P] for more details.

Corollary 4.13. *A forest is homogeneous iff it is recognized by an lr-deterministic automaton. It is homogeneous and recognizable iff it is recognized by a finite lr-deterministic automaton. If a forest is given by a finite automaton, then we can decide whether it is homogeneous. If it is, we can construct its finite minimal lr-deterministic automaton.*

Proof. If in Proposition 4.12 the automaton $\mathscr{A}$ is finite, then the automaton $\mathscr{B}$ is finite and can be effectively constructed. If it is lr-deterministic, which can be tested, then we can test whether $\mathbf{T}(\mathscr{A}) = \mathbf{T}(\mathscr{B})$. If the equality does not hold or if $\mathscr{B}$ is not lr-deterministic, then the given forest is not homogeneous. Otherwise it is, and $\mathscr{B}$ is the desired automaton. □

Remark 4.14. Let $\mathscr{A}$ and $\mathscr{B}$ be as in Proposition 4.12. By this proposition and Proposition 1.7(1), we have

$$\begin{aligned} \mathbf{D}(\mathscr{B}) &= \mathbf{cored}(\mathbf{codet}(\mathbf{D}(\mathscr{A}))) && (9)\\ &= \mathbf{codet}(\mathbf{det}(\mathbf{D}(\mathscr{A}))). && (10)\end{aligned}$$

If $\mathscr{A}$ is fr-deterministic, then $\mathbf{D}(\mathscr{A})$ is deterministic and equality (10) reduces to

$$\mathbf{D}(\mathscr{B}) = \mathbf{codet}(\mathbf{D}(\mathscr{A})).$$

We ask the following question: does there exist a general construction by which we can obtain from $\mathscr{A}$ an automaton $\mathscr{C}$ such that $\mathbf{D}(\mathscr{C}) = \mathbf{codet}(\mathbf{D}(\mathscr{A}))$, where $\mathscr{A}$ is not necessarily fr-deterministic? This would give, by (9),

$$\mathbf{D}(\mathscr{B}) = \mathbf{cored}(\mathbf{D}(\mathscr{C})),$$

and a two-step construction of the minimal lr-deterministic automaton recognizing the forest $\mathbf{T}(\mathscr{A})$, similar to the construction of the minimal automaton recognizing a language that consists of a determinization followed by a reduction.

Equation (10) actually corresponds to the construction of the minimal automaton of a language consisting of two determinizations (see Proposition 3.8(2).)

Finally, the "one-step construction" that we give in Proposition 4.12 has the following counterpart in the case of languages. Let $\mathscr{A} = \langle X, Q, \delta, Q_\mathrm{I}, Q_\mathrm{F}\rangle$ be a (nondeterministic) word automaton recognizing $L \subseteq X^*$. For every word $u \in X^*$, we let $\mathbf{Q}(u) := \{q \in Q \mid u\mathbf{L}(\mathscr{A}, q \rightarrow) \subseteq L\}$. We then let $\mathscr{B} = \langle X, Q', \delta', \{Q(\varepsilon)\}, Q'_\mathrm{F}\rangle$ where

$$\begin{aligned} &Q' = \{\mathbf{Q}(u) \mid u \in X^*, \mathbf{Q}(u) \neq \varnothing\},\\ &Q'_\mathrm{F} = \{\mathbf{Q}(u) \mid u \in X^*, Q_\mathrm{F} \cap \mathbf{Q}(u) \neq \varnothing\},\\ &(\alpha, a, \beta) \in \delta' \quad \text{iff} \quad \alpha, \beta \in Q', \quad a \in X,\end{aligned}$$

and

$$\beta = \{q \in Q \mid a\mathbf{L}(\mathscr{A}, q \rightarrow) \subseteq \bigcup\{\mathbf{L}(\mathscr{A}, q' \rightarrow) \mid q' \in \alpha\}\}.$$

The verification that $\mathscr{B}$ is the minimal automaton defining L is nothing but an easy adaptation of the proof of Proposition 4.12.

We now consider the recognition power and the minimization of l-deterministic automata.

Definition 4.6 can be reformulated as follows: an automaton $\mathscr{A} = \langle F, Q, \delta, Q_R\rangle$ is l-deterministic iff Q_R is a singleton and, for every $q \in Q$ and $f \in \mathrm{F}_2$, the decomposition

$$\Delta^{-1}(\mathscr{A}, f, q) := \{(\mathrm{T}(\mathscr{A}, q_1), \mathrm{T}(\mathscr{A}, q_2)) \mid (q_1, q_2) \in \delta^{-1}(f, q)\}$$

of the relation $f^{-1}(\mathrm{T}(\mathscr{A}, q))$ is its unique cominimal decomposition.

An l-deterministic automaton $\mathscr{A}$ is *minimal* iff $\mathrm{T}(\mathscr{A}, q) \neq \mathrm{T}(\mathscr{A}, q')$ for any two distinct states q and q'. This implies that $\mathbf{D}(\mathscr{A})$ is coreduced. The following proposition is fully similar to Proposition 4.9.

Proposition 4.15. *If $\mathscr{A}$ is an l-deterministic automaton, then* $\mathrm{T}(\mathscr{A})$ *is recognized by a unique minimal l-deterministic automaton $\mathscr{B}$, and then* $\mathbf{D}(\mathscr{B}) = \mathbf{cored}(\mathbf{D}(\mathscr{A}))$.

Proof. We first establish that any two minimal l-deterministic automata, $\mathscr{A}$ and $\mathscr{A}'$, defining the same forest L are isomorphic. We let $\mathscr{A} = \langle F, Q, \delta, \{q_R\}\rangle$ and $\mathscr{A}' = \langle F, Q', \delta', \{q'_R\}\rangle$. We let ρ be the relation $\subseteq Q \times Q'$ such that $(q, q') \in \rho$ iff $\mathrm{T}(\mathscr{A}, q) = \mathrm{T}(\mathscr{A}', q')$. This relation is a one-to-one partial function since the two automata are minimal. We prove that it defines a bijection, $Q \to Q'$, and, moreover, an isomorphism of the two automata.

We have $(q_R, q'_R) \in \rho$ since $\mathrm{T}(\mathscr{A}) = \mathrm{T}(\mathscr{A}')$. Let q, q' be two states such that $q \in Q$, $q' \in Q'$, and $(q, q') \in \rho$. We have

$$\mathrm{T}(\mathscr{A}, q) = \mathrm{T}(\mathscr{A}', q'),$$

hence, for every $f \in F_2$,

$$f^{-1}(\mathrm{T}(\mathscr{A}, q)) = f^{-1}(\mathrm{T}(\mathscr{A}', q'))$$

from which we get

$$\Delta^{-1}(\mathscr{A}, f, q) = \Delta^{-1}(\mathscr{A}', f, q'),$$

since, as noted above, they are two cominimal decompositions of a same relation.

It follows that, for each tuple (q_1, q_2, f, q) in δ, we have a unique tuple (q'_1, q'_2, f, q') in δ' with $(q_1, q'_1) \in \rho$, $(q_2, q'_2) \in \rho$.

For every a in F_0, we also have $(a, q) \in \delta$ iff $(a, q') \in \delta'$, since we assume that q and q' are related by ρ.

It follows that, for every q in Q, if there is a context c, such that $c \in \mathrm{C}(\mathscr{A}, q)$, then there is a state q' in Q' such that $(q, q') \in \rho$ and $c \in \mathrm{C}(\mathscr{A}', q')$. A similar property holds by exchanging the roles of the two automata. Since they are trim, the relation ρ is a bijection of Q onto Q'. We have seen above that it defines a bijection of δ onto δ'.

Hence, there is at most one minimal l-deterministic automaton recognizing a given forest.

Let us now assume that a forest L is $\mathrm{T}(\mathscr{A})$ for some l-deterministic automaton $\mathscr{A}$. The quotient construction already used in the proof of Proposition 4.9 yields the desired automaton $\mathscr{B}$. The remainder of the proof is the same as that of Proposition 4.9, and we omit it. □

Theorem 4.16. *Every forest L is defined by some l-deterministic automaton $\mathscr{A}$. If L is recognizable, then $\mathscr{A}$ can be constructed finite.*

Proof. Let $L \subseteq \mathrm{M}(F)$ be defined by a (possibly infinite) fr-deterministic (trim) automaton $\mathscr{B} = \langle F, Q, \delta, Q_{\mathrm{R}}\rangle$. For $f \in F_2$ and $\alpha \subseteq Q$, we let

$$\mathrm{r}(f, \alpha) := \{(q_1, q_2) | f(\mathrm{T}(\mathscr{B}, q_1), \mathrm{T}(\mathscr{B}, q_2)) \subseteq \mathrm{T}(\mathscr{B}, \alpha)\},$$

where $\mathrm{T}(\mathscr{B}, \alpha) := \bigcup\{\mathrm{T}(\mathscr{B}, q) | q \in \alpha\}$.

We now let $\mathscr{A}'$ be the possibly not trim automaton $\langle F, Q', \delta', Q'_{\mathrm{R}}\rangle$ where

$$Q' := \mathscr{P}(Q),$$
$$Q'_{\mathrm{R}} := \{Q_{\mathrm{R}}\},$$
$$\delta' := \{(a, \alpha) | p \in \alpha, \text{ and } (a, p) \in \delta\}$$
$$\cup \{(\alpha_1, \alpha_2, f, \alpha) | (\alpha_1, \alpha_2) \text{ is a pair belonging to } \mathbf{comin}(\mathrm{r}(f, \alpha))\}.$$

Let us recall that **comin**(s) denotes the unique cominimal decomposition of a relation s. (See Definition 1.4.)

Claim 1. $\mathrm{T}(\mathscr{A}', \alpha) = \mathrm{T}(\mathscr{B}, \alpha)$ *for all* $\alpha \in Q'$.

From this claim it follows that $\mathrm{T}(\mathscr{A}) = \mathrm{T}(\mathscr{A}') = L$ where $\mathscr{A}$ is the trim automaton obtained from $\mathscr{A}'$ by deleting all states α such that $\mathrm{T}(\mathscr{A}', \alpha)$ or $\mathrm{C}(\mathscr{A}', \alpha)$ is empty.

We prove later that $\mathscr{A}$ is l-deterministic.

Proof of Claim 1. We prove that, for every t in $\mathrm{M}(F)$,

$$t \in \mathrm{T}(\mathscr{A}', \alpha) \quad \Leftrightarrow \quad t \in \mathrm{T}(\mathscr{B}, \alpha)$$

by using an induction on the structure of t.

Basis: $t = a \in F_0$. The equivalence follows immediately from the definition of $\mathscr{A}'$.

Inductive step: $t = f(t_1, t_2)$. Let $t \in \mathrm{T}(\mathscr{A}', \alpha)$. Then we have $t_1 \in \mathrm{T}(\mathscr{A}', \alpha_1)$, $t_2 \in \mathrm{T}(\mathscr{A}', \alpha_2)$ for some $(\alpha_1, \alpha_2, f, \alpha)$ in δ'. By the induction hypothesis, we have $t_i \in \mathrm{T}(\mathscr{B}, q_i)$, $q_i \in \alpha_i$, for $i = 1$, 2. Since $f(\mathrm{T}(\mathscr{B}, q_1), \mathrm{T}(\mathscr{B}, q_2)) \subseteq \mathrm{T}(\mathscr{B}, \alpha)$ by the definitions of δ' and of $\mathrm{r}(f, \alpha)$, we have $t = f(t_1, t_2) \in \mathrm{T}(\mathscr{B}, \alpha)$, as was to be proved. The proof is similar in the other direction. □

Claim 2. $\mathscr{A}$ *is l-deterministic.*

Proof. By construction, $\mathscr{A}$ has a unique root state. We now verify condition (L). Let $(\alpha_1, \alpha_2, f, \alpha)$ and $(\alpha'_1, \alpha'_2, f, \alpha)$ be distinct tuples in δ', and let us establish that

$\mathrm{T}(\mathscr{A}, \alpha_2) \cap \mathrm{T}(\mathscr{A}, \alpha'_2) = \varnothing$. Let us assume by contradiction that $t \in \mathrm{T}(\mathscr{A}, \alpha_2) \cap \mathrm{T}(\mathscr{A}, \alpha'_2)$. Since $\mathscr{B}$ is fr-deterministic, there is a unique $q \in Q$ such that $t \in \mathrm{T}(\mathscr{B}, q)$ and $q \in \alpha_2$ by Claim 1. Similarly, $q \in \alpha'_2$, but we get a contradiction since $\alpha_2 \cap \alpha'_2 = \varnothing$ by the definition of δ'.

In order to complete the proof, we need only prove that $\mathrm{T}(\mathscr{A}, \alpha_1) \neq \mathrm{T}(\mathscr{A}, \alpha'_1)$. By the definition of δ', we have $\alpha_1 \neq \alpha'_1$. Let q distinguish these two sets, say q belongs to α_1 and not to α'_1. Let t belong to $\mathrm{T}(\mathscr{B}, q)$. Since $\mathscr{B}$ is fr-deterministic, and by Claim 1, t cannot belong to $\mathrm{T}(\mathscr{A}, \alpha'_1)$. This finishes the proof. □

This concludes the proof for the case where L is an arbitrary forest. If L is recognizable and $\mathscr{B}$ is a given finite automaton, then the l-automaton $\mathscr{A}$ is finite and can be effectively constructed. □

Acknowledgments

We thank G. Sénizergues for helpful comments.

References

[A] Arnold, A., A syntactic congruence for rational ω-languages, *Theoret. Comput. Sci.*, **39** (1985), 333-335.

[B] Brzozowski, J., Canonical regular expressions and minimal state graphs for definite events, in *Mathematical Theory of Automata*, Vol. 12, MRI Symposium Series, Polytechnic Press of the Polytechnic Institute of Brooklyn, 1963, pp. 529-561.

[C1] Courcelle, B., On recognizable sets and tree automata, in *Resolution of Equations in Algebraic Structures*, Vol. 1 (H. Aït-Kaci and M. Nivat, eds.), Academic Press, New York, 1989, pp. 93-126.

[C2] Courcelle, B., The monadic second-order logic of graphs, I: Recognizable sets of finite graphs, *Inform. and Comput.*, **85** (1990), 12-75.

[E] Eilenberg, S., *Automata, Languages, and Machines*, Vol. A, Academic Press, New York, 1974.

[GS] Gecseg, F., and Steinby, M., *Tree Automata*, Akademiai Kiado, Budapest, 1984.

[MW] Mezei, J., and Wright, J., Algebraic automata and context-free sets, *Informat. and Control*, **11** (1967), 3-29.

[N] Nerode, A., Linear automata transformations, *Proc. Amer. Math. Soc.*, **9** (1958), 541-544.

[NP] Nivat, M., and Podelski, A., Tree monoids and recognizable sets of finite trees, in *Resolution of Equations in Algebraic Structures*, Vol. 1 (H. Aït-Kaci and M. Nivat, eds.), Academic Press, New York, 1989, pp. 351-367.

[P] Podelski, A., Monoïdes d'arbres et automates d'arbres, Thèse, Université Paris-7, 1989.

[RS] Rabin, M., and Scott, D., Finite automata and their decision problems, *IBM J. Res. Develop.*, **3** (1959), 114-125. Reprinted in *Sequential Machines* (E. Moore, ed.), Addison-Wesley, Reading, MA, 1964.

[S] Staiger, L., Finite-state ω-languages, *J. Comput. System Sci.*, **27** (1983), 434-448.

[T] Trakhtenbrot, B., Finite automata and the logic of 1-place predicates, *Siberian Math. J.*, **3** (1962), 103-131 (in Russian).

Received July 20, 1990, and in revised form December 18, 1990.