Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 328 (2004) 135–149
Theoretical Computer Science
www.elsevier.com/locate/tcs

Bideterministic automata and minimal representations of regular languages☆

Hellis Tamm*, Esko Ukkonen

Department of Computer Science, University of Helsinki, P.O. Box 68, 00014, Helsinki, Finland

Abstract

Bideterministic automata are deterministic automata with the property of their reversal automata also being deterministic. It has been known that a bideterministic automaton is the minimal deterministic automaton accepting its language. This paper shows that any bideterministic automaton is the unique minimal automaton among all (including nondeterministic) automata accepting the same language. We also present a more general result that shows that under certain conditions a minimal deterministic automaton accepting some language or the reversal of the minimal deterministic automaton of the reversal language is a minimal automaton representation of the language. These conditions can be checked in polynomial time.

Keywords: Bideterministic automata; Minimal automata

1. Introduction

While the minimization of a deterministic finite automaton (DFA) can be done efficiently based on the Myhill–Nerode theorem [4, Theorem 3.9], finding a minimal nondeterministic finite automaton (NFA) for a given language seems to be a more difficult problem. The decision problem of finding a minimal NFA when given a DFA of a language is a PSPACE-complete problem [5]. Furthermore, for every regular language there is a unique (up to isomorphism) minimal DFA recognizing it, whereas there may exist more than one NFA

☆ Work supported by the Academy of Finland Grant 201560.

* Corresponding author.
E-mail addresses: hellis.tamm@cs.helsinki.fi (H. Tamm), esko.ukkonen@cs.helsinki.fi (E. Ukkonen).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.07.010

of minimal size accepting the given language. However, it is of interest to find sufficient conditions for an automaton to imply its minimality among all NFAs accepting its language.

This paper presents two minimality results. First, we show that any *bideterministic* automaton is the unique minimal automaton among all finite automata accepting the same language. Bideterministic automata or bideterministic languages have been considered, for example, in the context of machine learning [1], as a special case of reversible automata and languages [8], and in coding theory [9]. It has been observed that a bideterministic automaton is minimal among DFA [1,8]. In coding theory bideterministic trellises—which is a very restricted class of bideterministic automata—have been known to be minimal. This kind of trellises appear to correspond to certain codes (linear codes). It has been known that a minimal deterministic trellis is a minimal trellis for such codes [7]. But we are not aware of any such result for the general case of bideterministic automata. Second, we present some sufficient conditions under which a minimal DFA or the reversal of the minimal DFA of the reversal language is a minimal finite automaton. This is a more general result and actually, the minimality of a bideterministic automaton can be concluded from that result as well. We also show that these conditions can be tested in polynomial time. A preliminary version of this paper appeared in [10].

A finite automaton is a quintuple $A = (Q, \Sigma, \delta, I, F)$ where Q is a finite set of *states*, Σ is the *input alphabet*, $\delta : Q \times \Sigma \to 2^Q$ is the *transition function*, $I \subseteq Q$ is the set of *initial states* and $F \subseteq Q$ is the set of *final states*. An automaton A is *deterministic* if it has a unique initial state and if for every $q \in Q$ and every $a \in \Sigma$, $|\delta(q, a)| \leqslant 1$; otherwise it is *nondeterministic*. The *reversal* of an automaton A is the automaton $A^R = (Q, \Sigma, \delta^R, F, I)$ where $\delta^R(p, a) = \{q \mid p \in \delta(q, a)\}$ for all $p \in Q$ and $a \in \Sigma$. An automaton A is called *bideterministic* if both A and its reversal automaton A^R are deterministic.

The *empty string* is denoted by ε. For any string $x = x_1 \ldots x_k$, we denote by x^R the *reversal* of x which is the string $x_k \ldots x_1$.

We define the *extended transition function* $\hat{\delta} : Q \times \Sigma^* \to 2^Q$ so that $\hat{\delta}(q, \varepsilon) = \{q\}$ and $\hat{\delta}(q, xa) = \bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a)$ for all $q \in Q$, $x \in \Sigma^*$ and $a \in \Sigma$. A string $x \in \Sigma^*$ is *accepted* by A if there exists $q_0 \in I$ such that $\hat{\delta}(q_0, x) \cap F \neq \emptyset$. The set $L(A) = \{x \mid \bigcup_{q \in I} \hat{\delta}(q, x) \cap F \neq \emptyset\}$ is called the *language* accepted by A. The *reversal* of a language L, denoted by L^R, is the set of the reversals of all the strings belonging to L. A language accepted by a bideterministic automaton is a *bideterministic language*.

A *minimal* automaton is an automaton with the smallest number of states among all automata that accept the given language. A state q of A is *useful* if $L((Q, \Sigma, \delta, I, \{q\})) \neq \emptyset$ and $L((Q, \Sigma, \delta, \{q\}, F)) \neq \emptyset$. Two states q_i and q_j of A are *equivalent* if $L((Q, \Sigma, \delta, \{q_i\}, F)) = L((Q, \Sigma, \delta, \{q_j\}, F))$. Using the Myhill–Nerode theorem, it can be proved that a deterministic automaton is minimal among all DFAs accepting the same language if and only if all of its states are useful and no two states of it are equivalent. Two automata are equivalent if they accept the same language. Given an automaton A, using the well-known operation of the *subset construction*, we obtain an equivalent deterministic automaton $D(A) = (Q', \Sigma, \delta', q', F')$ [4, Theorem 2.1]. The automaton $D(A)$ is not necessarily the minimal DFA for $L(A)$.

Sometimes a stricter notion of determinism of an automaton than the definition given above is used by requiring that for all $q \in Q$ and $a \in \Sigma$, $|\delta(q, a)| = 1$ (instead of

$|\delta(q,a)| \leqslant 1$). This implies that some deterministic automata must have a so-called *dead state* $q_\emptyset$ such that $q_\emptyset \notin F$ and $\delta(q_\emptyset, a) = q_\emptyset$ for all $a \in \Sigma$. With this notion of determinism, the class of bideterministic automata is smaller when compared to the class of bideterministic automata obtained by using the definition of determinism as above. A deterministic automaton with a dead state $q_\emptyset$ cannot be bideterministic as there must be some $a \in \Sigma$ for which $\delta^R(q_\emptyset, a) > 1$. For example, while the language L_2^* of Example 12 in Section 3 is bideterministic by our definition of determinism, it is not bideterministic by this stricter notion of determinism. An example of a language that is bideterministic according to both definitions is the language $L((\mathbf{0}+\mathbf{1})((\mathbf{0}+\mathbf{1})(\mathbf{0}+\mathbf{1}))^*)$ of Example 13.

2. NFA minimization of Kameda and Weiner

Kameda and Weiner [6] have developed a theory for attacking the problem of minimization of nondeterministic automata. In the following, we present some definitions and results from this theory that we will need to prove our results.

Let $A = (Q, \Sigma, \delta, I, F)$ be an automaton, $B = D(A) = (Q', \Sigma, \delta', q', F')$ and $C = D(A^R) = (Q'', \Sigma, \delta'', q'', F'')$. As B and C are results of the subset construction applied on the set of states Q of A, both Q' and Q'' consist of subsets of Q.

Definition 1 (*Kameda and Weiner [6, Definition 7]*). The *states map (SM)* of A is a matrix which contains a row for each nonempty state of B, and a column for each nonempty state of C. The (i, j) entry contains $q_i' \cap q_j{}''$ (or is blank if $q_i' \cap q_j{}'' = \emptyset$), where q_i' is the ith element of Q' and $q_j{}''$ is the jth element of Q''. The *elementary automaton matrix (EAM)* of A is obtained from the SM of A by replacing each nonblank entry by a 1. Its (i, j) element is denoted by e_{ij}.

Theorem 2 (*Kameda and Weiner [6, Theorem 3]*).

$$L((Q', \Sigma, \delta', q_i', F')) = \bigcup_{j|e_{ij}=1} \{x^R \mid x \in L((Q'', \Sigma, \delta'', q'', q_j{}''))\}.$$

There are observations in [6] that according to Theorem 2, any states of B that have an identical pattern of 1's and blanks in the corresponding rows of the EAM of A, can be merged (by union, as the equivalent states). Also, because the definitions of B and C are symmetric, any states of C that have the same pattern of 1's and blanks in the corresponding columns, can be merged. These observations imply that two states of B (C) having the same pattern of blank entries in the corresponding rows (columns) of the SM of A can be merged. Rows (columns) of the SM with the same pattern of blank entries are called *equivalent*.

Definition 3 (*Kameda and Weiner [6, Definitions 8 and 10]*). The *reduced states map* (*RSM*) of A is obtained from the SM of A by merging all equivalent rows and columns. The merging of two rows (columns) means that they are replaced by a new row (column), the entries of which are the unions of the entries of the corresponding columns (rows). The *reduced automaton matrix* (RAM) of A is formed from the RSM of A by replacing each nonblank entry with a 1.

Let $\hat{B}$ be the minimal DFA for $L(A)$, obtained from B by merging by union the equivalent states, and let $\hat{C}$ be the minimal DFA, similarly obtained from C, for $L(C) = L(A)^R$.

Lemma 4 (*Kameda and Weiner [6, Lemma 3]*). *The RSM of A can be obtained from $\hat{B}$ and $\hat{C}$ in the same manner as the SM of A is obtained from B and C.*

Theorem 5 (*Kameda and Weiner [6, Theorem 4]*). *Equivalent automata have a RAM that is unique up to permutation of the rows and columns.*

Definition 6 (*Kameda and Weiner [6, Definitions 11–13]*). Given an EAM or RAM, if all the entries at the intersections of a set of rows $\{q'_{i_1}, \ldots, q'_{i_a}\}$ and a set of columns $\{q_{j_1}{}'', \ldots, q_{j_b}{}''\}$ are 1's then this set of 1's forms a *grid*. The grid is represented by $g = (q'_{i_1}, \ldots, q'_{i_a}; q_{j_1}{}'', \ldots, q_{j_b}{}'')$. The grid g *contains* the pair $(q'_i, q_j{}'')$ if $i \in \{i_1, \ldots, i_a\}$ and $j \in \{j_1, \ldots, j_b\}$. A set of grids forms a *cover* if every 1 in the EAM (or RAM) belongs to at least one grid in the set. A *minimum cover* is a cover that consists of the minimum number of grids. Given a cover of an EAM (or RAM), the corresponding *cover map* is obtained by replacing each 1 in the EAM (or RAM) by the names of all the grids (in the given cover) it belongs to.

Theorem 7 (*Kameda and Weiner [6, Theorem 5]*). *The SM* (*RSM*) *of an automaton A is a cover map, namely, the states of A appear as a cover of the EAM* (*RAM*) *of A.*

By a special rule, an NFA can be associated with any cover of the RAM of A. The rule is as follows. Let $\hat{B} = (\hat{Q}', \Sigma, \hat{\delta}', \hat{q}', \hat{F}')$, let Z be a cover of the RAM and let $f : \hat{Q}' \to 2^Z \setminus \emptyset$ be a function associated with Z which assigns to each state $\hat{p}$ of $\hat{B}$ the set of grids that intersect the row of the RAM that corresponds to $\hat{p}$. Then the NFA that is associated with the cover Z is given as $M = (Z, \Sigma, \gamma, Z_0, G)$ where for all $z \in Z$, $z' \in Z$, $\hat{p} \in \hat{Q}'$, and $a \in \Sigma$,

$$Z_0 = f(\hat{q}'),$$
$$z \in G \Leftrightarrow (z \in f(\hat{p}) \Rightarrow \hat{p} \in \hat{F}'),$$
$$z' \in \gamma(z, a) \Leftrightarrow (z \in f(\hat{p}) \Rightarrow z' \in f(\hat{\delta}'(\hat{p}, a))).$$

However, it may be the case that M is not equivalent to the original automaton A. To find a minimal NFA equivalent to A, [6] shows that an algorithm can be used that tests the covers of the RAM in increasing order of the size to find whether the NFA for the cover is equivalent to the original automaton. The first equivalent NFA found in this way is a minimal one. To check the equivalence of two automata, one may construct $D(M)$, find a minimal DFA equivalent to it and check if the resulting automaton is the same as $\hat{B}$, although [6] also proposes another procedure to accomplish the equivalence test.

3. Bideterministic automata are minimal

As we have stated in Section 1, bideterministic automata are deterministic automata with the property of their reversal automata also being deterministic. This means, among other things, that these automata have a unique initial state and a unique final state.

We also mentioned that it has been known that a bideterministic automaton is minimal among the DFAs. Indeed, to show that a bideterministic automaton is the minimal DFA for the language it accepts, one can, for example, use Brzozowski's minimization algorithm which involves reversing, determinizing, again reversing and determinizing the automaton [3,11]. That is, for an automaton A the minimal DFA can be obtained as $D((D(A^R))^R)$. As this algorithm, when applied to a bideterministic automaton, does not change it, it can be concluded that the automaton is minimal in the class of the DFAs. In the following we show, using the theory in Section 2, that a bideterministic automaton is also minimal in the class of the NFAs.

Let $A = (Q, \Sigma, \delta, q_0, q_f)$ be a bideterministic automaton. Its reversal automaton is $A^R = (Q, \Sigma, \delta^R, q_f, q_0)$ where $\delta^R(p, a) = \{q\}$ if and only if $\delta(q, a) = \{p\}$ for all $p \in Q$, $q \in Q$, and $a \in \Sigma$. Then the automata B and C from Section 2 are simply $B = D(A) = A$ and $C = D(A^R) = A^R$.

Let $Q = \{q_0, \ldots, q_{n-1}\}$. According to Definition 1, the SM of A consists of n rows and n columns, with exactly n non-blank entries $\{q_0\}, \ldots, \{q_{n-1}\}$ in the matrix which are positioned so that there is exactly one such entry in every row and every column. The corresponding EAM is basically the same as SM, only these non-blank entries are replaced with 1's. As there are no two equivalent rows nor columns in SM, it follows from Definition 3 that the RSM and RAM of A are the same as SM and EAM, respectively. Since there is exactly one 1 in every row and in every column of the RAM of A, we see by Definition 6 that every grid in the RAM contains just one row–column pair. Altogether there are n such grids in the RAM, and moreover, this set of grids is the only cover of the RAM. Because RAM is unique for all automata accepting $L(A)$ (Theorem 5) and any automaton accepting $L(A)$ has to have at least as many states as is the number of grids in the minimum cover of RAM (Theorem 7), we conclude that A is a minimal automaton. We have proved the following theorem.

Theorem 8. *A bideterministic automaton is minimal among all finite automata accepting the same language.*

Next, we show that a bideterministic automaton is uniquely minimal, that is, there do not exist any other automata with the same number of states that accept the same language. For this, we first note that as we discussed above, a bideterministic automaton is a minimal DFA which is known to be unique. Therefore, if any other automaton with the same number of states exists, it has to be nondeterministic. But this kind of automata do not exist as the following lemma shows.

Lemma 9. *Any nondeterministic automaton equivalent to a bideterministic automaton A has more states than A does.*

Proof. Let A be a bideterministic automaton with n states. Consider any nondeterministic automaton A' that accepts the same language as A does. Nondeterminism of A' means that either A' has multiple initial states or there is a state in A' from which there are transitions with the same label to more than one state. In any case, the determinized automaton $D(A')$

must have a state p—a subset of states of A'—of cardinality more than one. Let $p_1, \ldots, p_m$ be the states of A' comprising that subset, $m > 1$. Now, the row $\hat{p}$ of the states map SM of A' corresponding to the state p has to be such that every p_j, $j = 1, \ldots, m$, belongs to at least one entry in that row (because every state of A' belongs to some state of $D((A')^R)$). But, as we know that the RAM of A' is the same as the RAM of A (Theorem 5), then, according to the properties of the RAM of a bideterministic automaton as shown above, the RAM of A' has n rows and n columns with exactly one 1 in each row and each column. Hence the RSM of A' has exactly one non-blank entry in each row and each column. As an RSM is formed by merging the equivalent rows and columns of a SM (Definition 3), the RSM of A' must have a row—namely the row that contains the row $\hat{p}$ of the SM of A'—whose only non-blank entry contains all p_j, $j = 1, \ldots, m$ (and possibly some other states of A'). It has also to be the case that the intersection of any two entries of the RSM of A' is empty, or otherwise the RSM of A' could not have just one non-blank entry in each row and column. Because there are n rows and n columns and thus n non-blank entries in the RSM of A', it follows that A' has at least $n - 1 + m$ states. As we had $m > 1$, we get that A' must have more than n states. □

As a conclusion we may state the following theorem.

Theorem 10. *A bideterministic automaton is uniquely minimal.*

Proof. Follows from Lemma 9 and from the fact that a bideterministic automaton is the minimal DFA which is unique. □

Remark 11. The proof of Theorem 10 is independent from the result of Theorem 8.

Example 12. Let $L_k = \{ww^R \mid w \in \{0,1\}^k\}$ where $k \geqslant 0$, be a set of strings consisting of concatenations of any binary string of length k and its reversal string. Let L_k^* be the set that consists of strings obtained by concatenating zero or more times the elements of L_k. Then for every $k \geqslant 0$, L_k^* is accepted by a bideterministic automaton having $3 \times 2^k - 3$ states; the leftmost automaton in Fig. 1 is such an automaton with 9 states accepting L_2^*. By Theorem 8 we know that this is a minimal automaton recognizing this language and we cannot find a smaller automaton representation for it.

Example 13. A language $L((\mathbf{0}+\mathbf{1})((\mathbf{0}+\mathbf{1})(\mathbf{0}+\mathbf{1}))^*)$ that consists of all odd-length binary strings is accepted by a bideterministic automaton shown as the rightmost automaton in Fig. 1. By Theorem 8, this is a minimal automaton accepting this language.

4. More minimality results

From Theorem 8 we know that bideterminism is a sufficient condition for a language to have a property that the size of its minimal deterministic automaton is also the smallest

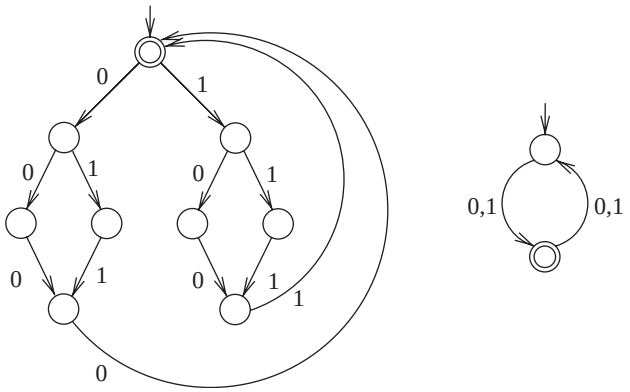

Fig. 1. Minimal automaton of the language L_2^* of Example 12 (at left) and minimal automaton of the language $L((\mathbf{0}+\mathbf{1})((\mathbf{0}+\mathbf{1})(\mathbf{0}+\mathbf{1}))^*)$ (at right).

possible size of any automaton—deterministic or nondeterministic—accepting the same language. It is of interest to find other conditions that imply similar minimalities. In this section, we present another, more general minimality result. Actually, Theorem 8 follows from this result as a special case.

The discussion below is based on two automata defined as follows. First, let $A = (Q, \Sigma, \delta, q, F)$ be a minimal deterministic automaton and second, let $A_1 = D(A^R) = (Q'', \Sigma, \delta'', q'', F'')$ be the automaton obtained from the reversal of A by the subset construction. Every state ${q_i}''$ of A_1 can also be seen as a subset of the state set Q of A.

Throughout the rest of the paper, we consider the partitions of the state sets of A and A_1, defined in the following way. Let $\{{Q_1}'', \ldots, {Q_k}''\}$ be the partition of the state set Q'' of A_1 into disjoint subsets (equivalence classes) such that any pair of states ${q_1}''$ and ${q_2}''$ of A_1 belongs to the same ${Q_i}''$, $i \in \{1, \ldots, k\}$, if and only if there exist states ${q_{i_1}}'', \ldots, {q_{i_l}}''$ of A_1 such that ${q_{i_1}}'' = {q_1}''$, ${q_{i_l}}'' = {q_2}''$ and ${q_{i_j}}'' \cap {q_{i_{j+1}}}'' \neq \emptyset$ for all $j = 1, \ldots, l-1$. And let $\{Q_1, \ldots, Q_k\}$ be the partition of the state set Q of A into disjoint subsets such that $Q_i = \bigcup_{q_j'' \in Q_i''} {q_j}''$ for $i = 1, \ldots, k$.

More intuitively, we divide the state set Q'' of A_1 into disjoint subsets in a way which ensures that the states of A_1 (as subsets of Q) belonging to different subsets of the partition do not intersect. Such partition of the states of A_1 accordingly induces a partition of the state set Q of A as well if we divide Q into subsets so that each subset is a union of the states belonging to the corresponding partition subset of Q''. These partitions divide the RAM of A into disjoint submatrixes, and in order to find a minimum cover of the RAM one can find a minimum cover for each such submatrix and take a union of those covers as it will be shown shortly below.

Consider the theory of Section 2 in the case where the automaton A of that section is the automaton A given above. Then the automata B and C of Section 2 are equal to the automata A and A_1, respectively, as $B = D(A) = A$ and $C = D(A^R) = A_1$. As B is equal to A, B is the minimal DFA accepting $L(A)$. According to the Brzozowski's minimization algorithm, C is the minimal DFA accepting $L(A^R)$ as we can write $C = D(A^R) = D(D(A)^R) =$

$D(D((A^{\mathrm{R}})^{\mathrm{R}})^{\mathrm{R}})$. Hence the SM of A is also the RSM of A (Definitions 1 and 3, Lemma 4). The number of rows and columns in the RSM (and the RAM) of A equals to the number of states of A and A_1, respectively. There is a one–one correspondence between the states of A and the rows of the RAM of A, as well as between the states of A_1 and the columns of the RAM of A. We denote by RAM_i, where $i \in \{1, \ldots, k\}$, a submatrix of the RAM of A, which is formed from the rows corresponding to the states belonging to Q_i and from the columns corresponding to the states in Q_i''.

Similarly to Definition 6, we say that a set of grids covers some RAM_i if every 1 in that RAM_i belongs to at least one grid in the set. Also, we say that a set of grids covers a set of rows and columns of the RAM if every 1 in these rows and columns belongs to at least some grid in that grid set.

Lemma 14. *Let G_i, where $i = 1, \ldots, k$, be any minimal set of grids covering* RAM_i. *Then $G_i \cap G_j = \emptyset$ if $i \neq j$, $i, j = 1, \ldots, k$, and $G_1 \cup \ldots \cup G_k$ is a minimum cover of the RAM of A.*

Proof. It is clear that any set of grids covers RAM_i if and only if it covers the set of rows and columns corresponding to Q_i and Q_i''. According to the definition of the partition of Q'', any two states taken from Q_i'' and Q_j'', where $i \neq j$, have an empty intersection. Therefore, any two 1's in the RAM of A, where the first 1 is in a column that corresponds to some state of Q_i'' and the second 1 is in a column corresponding to some state of Q_j'', cannot belong to the same grid. It follows that $G_i \cap G_j = \emptyset$. In order to find a minimum cover of the RAM, one can find a minimal set of grids covering the columns of Q_i'' (this set automatically covers also the rows of Q_i) for every $i \in \{1, \ldots, k\}$, and take a union of these sets. □

Definition 15. A grid is *elementary* if it consists of just one 1, that is, one row and one column. A grid with two or more 1's in it, that is, two or more rows or columns, is *non-elementary*. A non-elementary grid is *horizontal* (*vertical*) if all of its 1's are in the same row (column), that is, if it consists of one row (column).

Lemma 16. *Consider the following three conditions*:
(a) *Every state of A_1 consists of at most two states of A.*
(b) *Each state of A occurs in at most two states of A_1.*
(c) *Any two states of A_1 have at most one state of A in common.*
If any of the conditions (a), (b) *and* (c) *holds then every non-elementary grid in the RAM of A is either horizontal or vertical.*

Proof. Suppose that there is a non-elementary grid in the RAM of A that is not horizontal nor vertical. That is, there is a grid $g = (q_{i_1}, \ldots, q_{i_a}; q_{j_1}'', \ldots, q_{j_b}'')$ such that $a \geqslant 2$ and $b \geqslant 2$. This implies that there are at least two states q_{j_1}'' and q_{j_2}'' of A_1, both of which contain the states q_{i_1} and q_{i_2} of A. If (a) holds then the columns of the RAM of A corresponding to q_{j_1}''

and q_{j_2}'' must be equivalent. This is not possible, thus we have a contradiction. If (b) holds then the rows corresponding to q_{i_1} and q_{i_2} must be equivalent. This is not possible either, thus we have a contradiction in this case, too. If (c) holds then we have a direct contradiction. □

Lemma 17. *If a RAM_i, $i \in \{1, \ldots, k\}$, has more than one row or column and if every non-elementary grid in that RAM_i is either horizontal or vertical then there is a minimal set of grids covering the RAM_i, consisting of only horizontal and vertical grids, such that every horizontal grid in that set covers the corresponding row and every vertical grid in that set covers the corresponding column.*

Proof. Let the assumption of the lemma hold and let G_i be any minimal set of grids covering RAM_i. We modify G_i as described in the following. First, we observe that if G_i contains an elementary grid then there has to be two or more 1's in the row or the column involved by such grid, otherwise RAM_i would be a 1×1 matrix. Therefore we can and do replace any elementary grid in G_i either with a horizontal grid covering the corresponding row or with a vertical grid covering the corresponding column. Also, we replace any horizontal grid in G_i, which does not cover its corresponding row, with the horizontal grid covering that row entirely, and we replace any vertical grid in G_i, which does not cover its corresponding column, with the vertical grid covering that column entirely. After these replacements G_i still covers RAM_i and since the number of grids in G_i does not increase in the process (neither can it decrease because G_i was minimal in the beginning), G_i stays minimal. Thus, the modified G_i is as described in the statement of the lemma. □

In the proof of the following lemma we make use of one of the results given in Lemma 24 which will appear below in Section 5.

Lemma 18. *Consider Q_i and Q_i'', $i \in \{1, \ldots, k\}$, and assume that at least one of the following three conditions holds*:

(a) *Every state of A_1 consists of at most two states of A.*
(b) *Each state of A occurs in at most two states of A_1.*
(c+) *Any two states of A_1 have at most one state of A in common, and one of the next three conditions is true*: (i) $|Q_i| \leqslant 4$ *or* $|Q_i''| \leqslant 4$, (ii) $|Q_i| \geqslant 5$ *and* $|Q_i''| > |Q_i|(|Q_i| - 5)/2 + 5$, (iii) $|Q_i''| \geqslant 5$ *and* $|Q_i| > |Q_i''|(|Q_i''| - 5)/2 + 5$.

Then a minimum number of grids that cover RAM_i is the minimum of $|Q_i|$ and $|Q_i''|$.

Proof. Let at least one of the above conditions (a), (b), (c+) hold. If the RAM_i is a 1×1 matrix then it is clear that the statement of the lemma holds. Now consider the case where the RAM_i has more than one row or column. According to Lemma 16, any grid in the RAM_i is either horizontal, vertical or an elementary grid. Let G_i be a minimal set of grids covering RAM_i, which has the properties specified in Lemma 17. Then we may see G_i as a union of two non-intersecting sets of grids, denoted by G_i^{h} and G_i^{v}, where the set G_i^{h} consists of

horizontal grids where every grid covers its corresponding row and G_i^{v} consists of vertical grids with every grid covering the corresponding column. One of these two sets may be empty.

Obviously, the number of grids in G_i cannot be larger than the minimum of $|Q_i|$ and $|Q_i''|$. Neither can it be smaller, as we show in the following by contradiction.

Suppose that $|G_i| < |Q_i|$ and $|G_i| < |Q_i''|$. This implies that both G_i^{h} and G_i^{v} must be non-empty. As $|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| < |Q_i|$, then from the fact that G_i^{h} covers $|G_i^{\mathrm{h}}|$ rows it follows that G_i^{v} must cover the remaining of the $|Q_i|$ rows, i.e., at least $|Q_i| - |G_i^{\mathrm{h}}| > |G_i^{\mathrm{v}}|$ rows. Also, from $|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| < |Q_i''|$ and from the fact that G_i^{v} covers $|G_i^{\mathrm{v}}|$ columns it follows that G_i^{h} must cover at least $|Q_i''| - |G_i^{\mathrm{v}}| > |G_i^{\mathrm{h}}|$ columns. Due to the fact that the RAM_i cannot have equivalent columns nor rows, one horizontal grid can cover at most one column and one vertical grid can cover at most one row. Therefore, both G_i^{h} and G_i^{v} have to consist of at least two grids. Thus $|G_i^{\mathrm{h}}| \geqslant 2$ and $|G_i^{\mathrm{v}}| \geqslant 2$ which implies $|G_i| \geqslant 4$.

If (a) holds then there are exactly two 1's in every grid of G_i^{v}. This way, the grids in G_i^{v} can involve at most $|G_i^{\mathrm{v}}| + 1$ rows. We had above that G_i^{v} must cover more than $|G_i^{\mathrm{v}}|$ rows thus we conclude that the grids in G_i^{v} cover all the rows that they involve. But this implies $G_i^{\mathrm{v}} = G_i$, so we have obtained a contradiction.

If (b) holds then there are exactly two 1's in every grid of G_i^{h}. This way, the grids in G_i^{h} can involve at most $|G_i^{\mathrm{h}}| + 1$ columns. As we had above that G_i^{h} must cover more than $|G_i^{\mathrm{h}}|$ columns then the grids in G_i^{h} cover all the columns that they involve. But this implies $G_i^{\mathrm{h}} = G_i$, a contradiction.

If (c+) holds then in case (i) $|Q_i| \leqslant 4$ or $|Q_i''| \leqslant 4$ the hypothesis that $|G_i| < |Q_i|$ and $|G_i| < |Q_i''|$ leads to a contradiction since we showed above that $|G_i| \geqslant 4$. Now consider the case (ii) $|Q_i| \geqslant 5$ and $|Q_i''| > |Q_i|(|Q_i| - 5)/2 + 5$. Then we have $|Q_i''| > |Q_i|$ which means that $|G_i| \leqslant |Q_i| - 1$. Therefore $|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| \leqslant |Q_i| - 1$, which can also be written as $|Q_i| \geqslant |G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| + 1$. Using this last inequality with $|Q_i''| > |Q_i|(|Q_i| - 5)/2 + 5$, we get that $|Q_i''| > (|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| + 1)(|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| - 4)/2 + 5$. By Lemma 24 (c) we know that $|G_i^{\mathrm{h}}|$ horizontal grids can cover at most $|G_i^{\mathrm{h}}|(|G_i^{\mathrm{h}}| + 1)/2$ columns. Vertical grids in G_i^{v} cover $|G_i^{\mathrm{v}}|$ columns. Therefore $|Q_i''| \leqslant |G_i^{\mathrm{h}}|(|G_i^{\mathrm{h}}| + 1)/2 + |G_i^{\mathrm{v}}|$. Putting two last inequalities together, we get $|G_i^{\mathrm{h}}|(|G_i^{\mathrm{h}}| + 1)/2 + |G_i^{\mathrm{v}}| > (|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| + 1)(|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| - 4)/2 + 5$ from which it follows $(|G_i^{\mathrm{v}}| - 2)(2|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| - 3) < 0$. But this cannot hold, since neither $|G_i^{\mathrm{v}}| - 2$ nor $2|G_i^{\mathrm{h}}| + |G_i^{\mathrm{v}}| - 3$ can be negative. Thus we have obtained a contradiction. The proof for the case (iii) $|Q_i''| \geqslant 5$ and $|Q_i| > |Q_i''|(|Q_i''| - 5)/2 + 5$ is symmetric and similar to the case (ii). □

Theorem 19. *Consider a minimal DFA A and $A_1 = D(A^{\mathrm{R}})$ with the partitions of their state sets $\{Q_1, \ldots, Q_k\}$ and $\{Q_1'', \ldots, Q_k''\}$ as described above. Assume that at least one of the following three conditions holds*:

(a) *Every state of A_1 consists of at most two states of A.*

(b) *Each state of A occurs in at most two states of A_1.*

(c+) *Any two states of A_1 have at most one state of A in common, and for every $i = 1, \ldots, k$ one of the next three conditions is true*: (i) $|Q_i| \leqslant 4$ *or* $|Q_i''| \leqslant 4$, (ii) $|Q_i| \geqslant 5$ *and* $|Q_i''| > |Q_i|(|Q_i| - 5)/2 + 5$, (iii) $|Q_i''| \geqslant 5$ *and* $|Q_i| > |Q_i''|(|Q_i''| - 5)/2 + 5$.

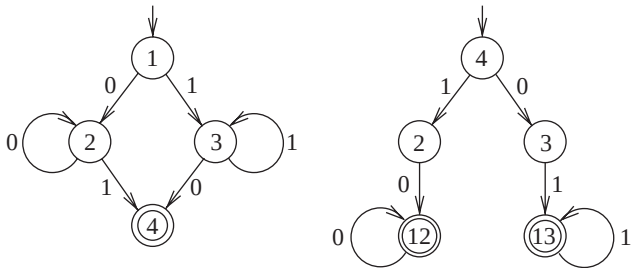

Fig. 2. Minimal DFA A of the language $L(\mathbf{00}^*\mathbf{1}+\mathbf{11}^*\mathbf{0})$ and $A_1 = D(A^{\mathrm{R}})$.

If $|Q_i| \leqslant |Q_i''|$ *for all* $i = 1, \ldots, k$, *then* A *is a minimal automaton accepting* $L(A)$. *If* $|Q_i''| \leqslant |Q_i|$ *for all* $i = 1, \ldots, k$, *then* A_1^{R} *is a minimal automaton accepting* $L(A)$.

Proof. According to the assumptions of this theorem along with Lemma 18, a minimum number of grids to cover RAM_i, for $i = 1, \ldots, k$, is the minimum of $|Q_i|$ and $|Q_i''|$. By Lemma 14, if $|Q_i| \leqslant |Q_i''|$ for all $i = 1, \ldots, k$, then a minimum cover of the RAM of A consists of $|Q_1| + \ldots + |Q_k| = |Q|$ grids. Similarly, if $|Q_i''| \leqslant |Q_i|$ for all $i = 1, \ldots, k$, then a minimum cover of the RAM of A consists of $|Q_1''| + \ldots + |Q_k''| = |Q''|$ grids. We know that any automaton equivalent to A cannot have less states than is the number of grids in a minimum cover of the RAM of A (Theorems 5 and 7). We know that A and A_1^{R} both accept $L(A)$ and their sizes are $|Q|$ and $|Q''|$, respectively. Therefore, if $|Q_i| \leqslant |Q_i''|$ for all $i = 1, \ldots, k$, then A is a minimal automaton accepting $L(A)$. Also, if $|Q_i''| \leqslant |Q_i|$ for all $i = 1, \ldots, k$, then A_1^{R} is a minimal automaton accepting $L(A)$. □

Example 20. In Fig. 2, the leftmost automaton is the minimal DFA A accepting the language $L(\mathbf{00}^*\mathbf{1}+\mathbf{11}^*\mathbf{0})$ and the rightmost one $A_1 = D(A^{\mathrm{R}})$ is obtained from the reversal of A by the subset construction. The partitions of the state sets of A and A_1 divide both sets into two subsets: $Q_1 = \{1, 2, 3\}$, $Q_2 = \{4\}$, $Q_1'' = \{\{1, 2\}, \{1, 3\}, \{2\}, \{3\}\}$, $Q_2'' = \{\{4\}\}$. It can be easily verified that all three conditions (a), (b) and (c+) of Theorem 19 hold. As $|Q_1| = 3 < 4 = |Q_1''|$ and $|Q_2| = 1 = |Q_2''|$ then by Theorem 19, A is a minimal automaton.

Corollary 21. *Theorem* 8 *follows from Theorem* 19.

Proof. Let A be a bideterministic automaton. Then A is a minimal DFA. The automaton A_1 of Theorem 19 is $A_1 = A^{\mathrm{R}}$. Since the state sets of A and A_1 coincide and $|Q_i| = |Q_i''| = 1$ for all $i = 1, \ldots, k$, it is clear that all three conditions (a), (b) and (c+) of Theorem 19 hold. As both $|Q_i| \leqslant |Q_i''|$ and $|Q_i''| \leqslant |Q_i|$ are true for all $i = 1, \ldots, k$, then we may conclude

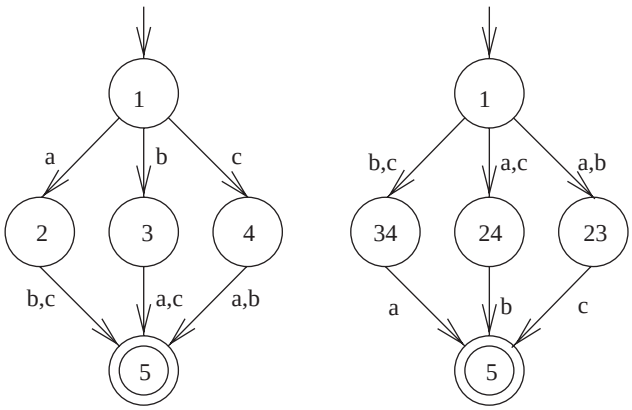

Fig. 3. Minimal DFA A of the language $\{ab, ac, ba, bc, ca, cb\}$ and $A_1^{\mathrm{R}} = (D(A^{\mathrm{R}}))^{\mathrm{R}}$.

that both A and $(A^{\mathrm{R}})^{\mathrm{R}}$ are minimal automata accepting $L(A)$. But these two automata are the same, so Theorem 8 follows. □

In certain cases Theorem 19 gives two different minimal automata:

Corollary 22. *Let A and A_1 be automata meeting the assumptions of Theorem 19. If $|Q_i| = |Q_i''|$ for all $i = 1, \ldots, k$, and A is not bideterministic then A and A_1^{R} are two different minimal automata accepting $L(A)$.*

Proof. As $|Q_i| \leqslant |Q_i''|$ and $|Q_i''| \leqslant |Q_i|$ for all $i = 1, \ldots, k$, then, according to Theorem 19, both A and A_1^{R} are minimal. If A is not bideterministic then A_1^{R} has to be nondeterministic. So, A and A_1^{R} must be different. □

Example 23. Fig. 3 presents two automata given by [2] as examples of two different minimal automata accepting the language $\{ab, ac, ba, bc, ca, cb\}$. In terms of our theory, the leftmost automaton is the minimal DFA A accepting that language and the rightmost one is $A_1^{\mathrm{R}} = (D(A^{\mathrm{R}}))^{\mathrm{R}}$. The partitions of the state sets of A and A_1 are the following: $Q_1 = \{1\}$, $Q_2 = \{2, 3, 4\}$, $Q_3 = \{5\}$, $Q_1'' = \{\{1\}\}$, $Q_2'' = \{\{2, 3\}, \{2, 4\}, \{3, 4\}\}$, $Q_3'' = \{\{5\}\}$. As the assumptions of Theorem 19 hold, $|Q_i| = |Q_i''|$ for all $i = 1, 2, 3$, and A is not bideterministic then by Corollary 22 both A and A_1^{R} are minimal, indeed.

5. Algorithms and time complexities

In this section we describe an algorithm for testing the sufficient conditions for minimality given in Theorem 19 and analyze the time complexity of the algorithm. For a given minimal DFA A, we want to test whether or not one of the conditions (a), (b) and (c+) of

Theorem 19 holds for $D(A^R)$, and if this is the case then we want also to test whether or not either $|Q_i| \leqslant |Q_i''|$ for all $i = 1, \ldots, k$, or $|Q_i''| \leqslant |Q_i|$ for all $i = 1, \ldots, k$. While the subset construction that is needed to produce $D(A^R)$ is exponential in the general case, it turns out that here we can obtain a polynomial time test. This is because we can stop the construction as soon as we know that conditions (a), (b) and (c+) cannot be true. They can be true only if $D(A^R)$ has a polynomial number of states. More precisely, the stopping criterion of the subset construction is based on the following lemma.

Lemma 24. *Let K be a collection of non-empty subsets of $\{1, 2, \ldots, n\}$. Then the following statements hold*:

(a) *If each member of K contains at most two elements, then $|K| \leqslant n(n+1)/2$.*
(b) *If any element of $\{1, 2, \ldots, n\}$ belongs to at most two different members of K, then $|K| \leqslant \lfloor 3n/2 \rfloor$.*
(c) *If any two different members of K have at most one element in common, then $|K| \leqslant n(n+1)/2$.*

Proof. In case (a) K is a subset of the set $\{\{i, j\} \mid 1 \leqslant i \leqslant j \leqslant n\}$ which has $n(n+1)/2$ elements. Thus $|K| \leqslant n(n+1)/2$. In case (b) we reason as follows. Let $i \in \{1, \ldots, n\}$. If i belongs to two members of K, let these members be A and B. If i belongs to only one member of K, let it be A and $B = \emptyset$. Otherwise let $A = B = \emptyset$. Then define $a_i = 0$ if $A = \emptyset$ and $a_i = 1/|A|$ otherwise, and $b_i = 0$ if $B = \emptyset$ and $b_i = 1/|B|$ otherwise. Then $a_i + b_i \leqslant 3/2$ because if either of a_i and b_i equals 1 and hence $A = \{i\}$ or $B = \{i\}$, the other of a_i and b_i must be $\leqslant 1/2$; otherwise we would have $A = B = \{i\}$. But then $|K| = \sum_{i=1}^{n} (a_i + b_i) \leqslant 3n/2$. As $|K|$ is an integer, the claim follows. In case (c), for each $S \in K$, let $P_S = \{\{i, j\} \mid i \neq j, i \in S, j \in S\}$ if S contains at least two elements and $P_S = \{\{i\}\}$ if the single element of S is i. As two different members of K can have at most one element in common, $P_S \cap P_{S'}$ is empty whenever $S \neq S'$, that is, the sets P_S are pairwise disjoint subsets of the set $\{\{i, j\} \mid 1 \leqslant i \leqslant j \leqslant n\}$ with $n(n+1)/2$ elements. Hence the number of different sets P_S, which is the same as $|K|$, is $\leqslant n(n+1)/2$. □

Theorem 25. *For a minimal DFA A with n states, one can test in time $O(n^4 \log n)$ whether or not A satisfies the sufficient conditions of Theorem 19 for minimality.*

Proof. We will describe a test algorithm and analyze its running time.

The algorithm first builds $A_1 = D(A^R)$, using the subset construction. Recall that (an incremental version of) the subset construction algorithm works by generating new candidate states for $D(A^R)$ from the already accepted states. Each new candidate is a subset of the states of A. The candidate is generated by putting together the states of A to which A^R has transitions with the same input symbol from states that constitute an already accepted state of $D(A^R)$; at the beginning the only accepted state consists of the initial states of A^R.

For each new candidate the algorithm tests whether or not the candidate has already been accepted. To make this test fast we keep the already accepted states in a tree structure,

called the *state trie*. In the state trie an accepted state $(a_1, \ldots, a_k)$ where $a_1 < \cdots < a_k$ are some states of A, in increasing order, is represented by a leaf that is reached along the path $a_1, a_2, \ldots, a_k$ from the root of the trie. As the branching factor of such a trie is $O(n)$, we can test in time $O(k \log n)$ whether or not a new candidate state whose k elements are given in the increasing order has already been accepted.

Now, to test the condition (a) of Theorem 19, we note first that by Lemma 24 (a) the condition can be true only if there are $O(n^2)$ states in $D(A^R)$. Each accepted state generates $\leqslant |\Sigma|$ candidate states where Σ is the alphabet of A whose size is assumed to be constant. Hence the total number of candidate states is $O(|\Sigma|n^2) = O(n^2)$. Each new candidate can be generated in time $O(n)$ by scanning through the transition function of A^R. In fact, to generate the next candidate state from a current accepted state X of $D(A^R)$, scan the transition function and find for each $q \in X$ the list of states that can be reached in one transition step from q with a fixed input symbol. As in our case $|X| \leqslant 2$, there can be at most two such lists, each of length $\leqslant n$. Then merge these lists into sorted order (to remove duplicates and to prepare for the state trie search). As A is deterministic, the size of the transition function of A and hence also of A^R is $O(|\Sigma|n) = O(n)$. Hence a new candidate can be generated from X as described above in time $O(n)$, and the total time for candidate generation becomes $O(n^3)$. If a new candidate has more than two states then we know that (a) is not true. Otherwise we add the candidate to the state trie if it is not already there. As the height of the trie is 2, this takes time $O(\log n)$ per candidate, hence $O(n^2 \log n)$ altogether.

Similarly, to test the condition (b) of Theorem 19, we note that by Lemma 24 (b) the condition can be true only if $D(A^R)$ has $O(n)$ states. The total number of the candidate states is therefore $O(n)$. An accepted state as well as a new candidate can now have $O(n)$ elements. Each candidate can be generated in time $O(n^2 \log n)$. This bound comes from merging the $O(n)$ lists, each of length $O(n)$, to put the elements of the candidate in the increasing order. The total time for candidate generation becomes $O(n^3 \log n)$. That no element of the candidate occurs in more than one already accepted state can be tested in $O(n)$ time using simple bookkeeping of the usage of each element. If some element already occurs in two accepted states, condition (b) is not true. Adding an accepted candidate to the state trie takes time $O(n \log n)$ as the height of the trie is $O(n)$. The total time for acceptance of the candidates is therefore $O(n^2 \log n)$.

Finally, consider the condition (c+) of Theorem 19. This condition consists of two parts: the first is the condition (c) of Lemma 16 and the second is a condition involving comparisons of the sizes of Q_i and Q_i'' which we will discuss shortly below. To test the first part of (c+), we see that by Lemma 24(c) the condition can be true only if $D(A^R)$ has $O(n^2)$ states. That makes the total number of the candidate states to be $O(n^2)$. A next candidate can be generated from a current accepted state in time $O(n^2 \log n)$ in the same way as in case (b). Thus the total time for candidate generation is $O(n^4 \log n)$. Checking that a candidate does not have more than one common state with any of the accepted states can again be done in $O(n)$ time by bookkeeping techniques. Adding an accepted candidate to the state trie takes time $O(n \log n)$. The total time for acceptance of the candidates is therefore $O(n^3 \log n)$.

To summarize, we have shown so far that testing (a), (b) and the first part of the condition (c+) can be achieved in time $O(n^4 \log n)$. We have still to evaluate the sizes of classes Q_i'' and Q_i. To this end, let us construct a graph (Q, E) with nodes Q and edges E where Q

is the set of the states of A and E contains (u, v) if and only if u and v belong to the same state of $D(A^R)$. Set E can be built from $D(A^R)$ at least in $O(n^4)$ time. As the set of the nodes of each connected component of this graph equals some Q_i, we get the sizes of the Q_i's just by counting the number of the nodes of each connected component. This can be done by forming the components using the standard algorithm in time $O(n^2)$. Finally, the number of states ${q_j}''$ of $D(A^R)$ that together form the class ${Q_i}''$ and hence cover the class Q_i which corresponds to our component can be evaluated combined with the construction of the component itself. We just need to associate with each node $q \in Q$ the names of all states ${q_j}''$ that contain q, and then count the total number of different names associated with a connected component. The total time requirement stays $O(n^4 \log n)$. □

The algorithm in the proof of Theorem 25 could possibly be made asymptotically faster. Our goal was just to show that the test can be done in polynomial time.

References

[1] D. Angluin, Inference of reversible languages, J. Assoc. Comput. Mach. 29 (3) (1982) 741–765.

[2] A. Arnold, A. Dicky, M. Nivat, A note about minimal non-deterministic automata, Bull. European Assoc. Theoret. Comput. Sci. 47 (1992) 166–169.

[3] J.A. Brzozowski, Canonical regular expressions and minimal state graphs for definite events, in: Proc. Symp. on Mathematical Theory of Automata, MRI Symposia Series, Vol. 12, Polytechnic Press, Polytechnic Institute of Brooklyn, New York, 1963, pp. 529–561.

[4] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, MA, 1979.

[5] T. Jiang, B. Ravikumar, Minimal NFA problems are hard, SIAM J. Comput. 22 (6) (1993) 1117–1141.

[6] T. Kameda, P. Weiner, On the state minimization of nondeterministic automata, IEEE Trans. Comput. C-19 (7) (1970) 617–627.

[7] D.J. Muder, Minimal trellises for block codes, IEEE Trans. Inform. Theory 34 (5) (1988) 1049–1053.

[8] J.-E. Pin, On reversible automata, in: Proc. 1st LATIN Conf., Lecture Notes in Computer Science, Vol. 583, Springer, Berlin, 1992, pp. 401–416.

[9] P. Shankar, A. Dasgupta, K. Deshmukh, B.S. Rajan, On viewing block codes as finite automata, Theoret. Comput. Sci. 290 (3) (2003) 1775–1797.

[10] H. Tamm, E. Ukkonen, Bideterministic automata and minimal representations of regular languages, in: Proc. CIAA 2003, Lecture Notes in Computer Science, Vol. 2759, Springer, Berlin, 2003, pp. 61–71.

[11] B.W. Watson, Taxonomies and toolkits of regular language algorithms, Ph.D. Dissertation, Faculty of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven, The Netherlands, 1995.