Automatic semigroup acts

Erzsébet R. Dombi[a], Miklós Hartmann[b,*]

[a]*Department of Mathematics and Statistics, University of Strathclyde, Glasgow, G1 1XH, U. K.*
[b]*Department of Mathematics, University of York, Heslington, York YO10 5DD, U. K.*

Abstract

To give a general framework for the theory of automatic groups and semigroups, we introduce the notion of automaticity for semigroup acts. We investigate their basic properties and discuss how the property of being automatic behaves under changing the generators of the acting semigroup and under changing the generators of the semigroup act. In particular, we prove that under some conditions on the acting semigroup, the automaticity of the act is invariant under changing the generators. Since automatic semigroups can be seen as a special case of automatic semigroup acts, our result generalizes and extends the corresponding result on automatic semigroups, where the semigroup S satisfies $S = SS$. We also give a geometric approach in terms of the fellow traveller property and discuss the solvability of the equality problem in automatic semigroup acts. Our notion gives rise to a variety of definitions of automaticity depending on the set chosen as a semigroup act and we discuss future research directions.

Keywords: semigroup acts, automaticity, change of generators
2000 MSC: 20M10, 20M30

1. Introduction

Building on the theory of automatic groups, researchers proposed the development of an analogous theory for semigroups in [5]. One of the main purposes of [5] was to investigate whether group theoretic results generalize. It was found that some properties do: for example the word problem in automatic semigroups is also solvable in quadratic time, but it became clear that automatic semigroups do not enjoy the many pleasant properties automatic groups enjoy. For example, it is not known whether the so-called uniform word problem is solvable (see [16]). Furthermore, the beautiful geometric theory that provides the backbone of the theory of automatic groups does not work in the semigroup theoretic case as the so called fellow traveller property does not characterize automatic structures for semigroups. This lead to an example showing that being automatic does depend on the choice of the generating set in the semigroup theoretic setting.

*Corresponding author
Email addresses: `e.dombi@strath.ac.uk` (Erzsébet R. Dombi), `hartm@math.u-szeged.hu` (Miklós Hartmann)

These negative results opened new doors in the theory of automatic semigroups. On one hand semigroup classes related to groups have been investigated ([1], [2], [6], [8], [11], [9]). For example, in [6] it is proved that automatic completely simple semigroups are characterized by the fellow traveller property, that they are finitely presented and that automaticity does not depend on the choice of the generating set. In [9], it is proved that for semigroups satisfying that $S = SS$, the property of being automatic does not depend on the choice of the semigroup generating set. On the other hand, the research community experimented with alternative definitions ([13], [20], [19]). For example, in [19], the notion of a prefix-automatic monoid is introduced and is proved that this property does not depend on the choice of the semigroup generating set.

Our aim is to give a general framework for the theory of automatic semigroups and groups by introducing the notion of automatic semigroup acts, or S-acts. Both groups and semigroups can be viewed as S-acts, where the set on which S acts is S itself. Our motivating examples for the initiation of this theory are free inverse semigroups. Free inverse semigroups have solvable word problem [18] but they are not automatic [7]. By choosing a suitable set closely related to a given semigroup such as an $\mathcal{R}$-class or $S/\mathcal{R}$ on which an action of S can be naturally defined, we envisage that this framework will be useful in investigating various properties of S given that the S-act is automatic.

The purpose of this paper is to lay the foundations of the theory of automatic S-acts. We verify that our notion is indeed a generalization of the semigroup and group theoretic notion and we investigate basic properties. The behaviour of automaticity with respect to changing the generators has been intensively investigated. It was shown that automaticity is invariant under changing the generators for groups ([12]), for monoids ([11]), for semigroups with local right identities ([10]) and for semigroups satisfying $S = S^2$ ([9]). In case of S-acts, there are two types of generators: one for the S-act and one for S. We show that automaticity is invariant under changing the generators of the S-act and we show that if S is a semigroup satisfying that $S^N = S^{N+1}$ for some $N \in \mathbb{N}$, then automaticity is invariant under changing the generators of S. Next, we discuss the equality problem for automatic S-acts. In particular we show that the equality problem is solvable in quadratic time. By introducing the fellow traveller property for S-acts, we show that automaticity of an S-act implies that the fellow traveller property holds in the graph associated to the S-act. The results of Sections 5, 7 and 8 are from the first author's PhD Thesis ([10]). We close the paper by discussing future research directions.

2. Preliminaries

In this section we introduce the notation, definitions and results we need. For background reading on automata and formal language theory we refer the reader to [14] and [17].

For any finite set X, we let X^+ denote the set of all non-empty words over X and let X^* denote the set of all words over X (including the empty word ϵ). Let $u = x_1 \dots x_n \in X^*$. We let $l(u)$ denote the length of u, namely n, and for $t \leq l(u)$ we let $u(t) = x_1 \dots x_t$ denote the prefix of u whose length is t. For any $t > l(u)$ we let $u(t) = u$. Let $N \in \mathbb{N}$. We let $X^N = \{w \in X^+ \mid l(w) = N\}$ and we let $X^{\leq N} = \{w \in X^* \mid l(w) \leq N\}$.

Consider the alphabet $X = \{x_1, x_2, \ldots, x_{n-1}, x_n\}$ and let $<$ be a total order on the alphabet X. We define the *shortlex order* on X^+ by

$$u < v \text{ if and only if } \begin{cases} l(u) < l(v), \text{ or} \\ l(u) = l(v) \text{ and } u \text{ precedes } v \text{ in the lexicographical order.} \end{cases}$$

Let X be a finite set, and let \$ be a symbol not contained in X. We define

$$X(2,\$) = ((X \cup \$) \times (X \cup \$)) \setminus \{(\$,\$)\}.$$

Throughout the paper we will often think of elements of $X(2,\$)^*$ as being pairs of words over $X \cup \{\$\}$ having the same length, that is, $(x_1, y_1)\ldots(x_n, y_n) \equiv (x_1 \ldots x_n, y_1, \ldots y_n)$ where $(x_1, y_1), \ldots, (x_n, y_n) \in X(2,\$)$. Let $u = x_1 \ldots x_n$ and $v = y_1 \ldots y_m$ be words over X. Define $\delta_X : X^* \times X^* \to X(2,\$)^*$ by

$$(u,v)\delta_X = \begin{cases} \epsilon & \text{if } m = n = 0, \\ (x_1, y_1)\ldots(x_n, y_n) & \text{if } n = m > 0, \\ (x_1, y_1)\ldots(x_n, y_n)(\$, y_{n+1})\ldots(\$, y_m) & \text{if } n < m, \\ (x_1, y_1)\ldots(x_m, y_m)(x_{m+1}, \$)\ldots(x_n, \$) & \text{if } n > m \end{cases}$$

Throughout the paper, if $\varphi : X \to Y$ is a map, then we will also use φ to denote the map

$$\varphi : X \times X \to Y \times Y, (x,y) \mapsto (x\varphi, y\varphi).$$

For any X we define $\iota_X \colon (X \cup \{\$\})^* \to X^*$ to be the homomorphism which fixes elements of X and sends \$ to ϵ.

The following useful notion was introduced in [4]:

Definition 2.1. Let $M, L \subseteq (X^* \times X^*)\delta_X$. The padded product of M and L is defined to be the language

$$M \odot L = \{(u_1 v_1, u_2 v_2) \mid (u_1, v_1)\delta_X \in M,\ (u_2, v_2)\delta_X \in L\}\delta_X.$$

A *regular language* L over X is a subset of X^* such that there is a finite state automaton accepting L. A *finite state automaton* $\mathcal{A}$ consists of a finite non-empty set Q called a *set of states*, a finite set X called an *alphabet*, an element $q_0 \in Q$ called an *initial state*, a partial function $\delta : Q \times X \to Q$ called a *transition function* and a subset T of Q called *final states* or *accept states*. We write $\mathcal{A} = (Q, X, \delta, q_0, T)$. We recall the following properties of regular languages:

Proposition 2.2. *Let X and Y be finite sets. Then the following hold:*

(1) X^+, X^* and finite subsets of X^* are regular languages.

(2) If $K \subseteq X^*$ and $L \subseteq X^*$ are regular languages, then $K \cup L,\ K \cap L,\ K - L, KL, K^*$ and $K^{\mathrm{rev}} = \{x_1 \ldots x_n \mid x_n \ldots x_1 \in K\}$ are regular languages.

(3) If $K \subseteq X^*$ is a regular language and $\varphi : X^+ \to Y^+$ is a semigroup homomorphism, then $K\varphi$ is a regular language.

(4) If $L \subseteq Y^*$ is a regular language and $\varphi : X^+ \to Y^+$ is a semigroup homomorphism, then $L\varphi^{-1}$ is a regular language.

The following Lemma will be useful in the paper ([4, Lemma 5.3]):

Lemma 2.3. *Let $M, L \subseteq (X^* \times X^*)\delta_X$ be regular languages. If there exists a constant c such that for all $(u,v)\delta_X \in M$ we have that $|l(u) - l(v)| \leq c$, then $M \odot L$ is a regular language. In particular, if $M = (u,v)\delta_X$ for some $u, v \in X^*$, then $M \odot L = ((u,v)L\iota_X)\delta_X$ is a regular language.*

Next, we define the notion of an automatic semigroup:

Definition 2.4. Let S be a semigroup generated by a finite set X. Let L be a regular language over X and $\varphi : X^+ \to S$ a homomorphism. We say that (X, L) is an automatic structure for S if

(1) $L\varphi = S$,

(2) $L_= = \{(u,v) \mid u, v \in L,\ u = v\}\delta_X$ is a regular language,

(3) $L_x = \{(u,v) \mid u, v \in L,\ ux = v\}\delta_X$ is a regular language for all $x \in X$.

If a semigroup S has an automatic structure, then we say that S is automatic. If L maps bijectively onto S, then we say that (X, L) is an automatic structure with uniqueness.

Let S be a semigroup. A (right) S-act is a set A together with a function $f\colon A \times S \to A, (a, s) \mapsto a.s$ such that $(a.s).t = a.(st)$ for all $a \in A$ and $s, t \in S$. If S is a monoid with identity 1 then we also assume that $a.1 = a$ for all $a \in A$. A typical S-act is S itself with action defined by $s.t = st$ for all $s, t \in S$.

3. Operations on regular languages

The results of the paper rely heavily on certain operations on regular languages. In this section we collect all these results. First we start with some basic results from [5].

Lemma 3.1. *Let X be a finite set.*

1. *If $w \in X^*$ and $L \subseteq X^*$ is regular, then so is $\{u \in X^* : uw \in L\}$.*
2. *If $L, K \subseteq X^*$ are regular then so is $(L \times K)\delta_X \subseteq X(2, \$)^*$.*
3. *If $U \subseteq (X^* \times X^*)\delta_X$ is regular, then so is*

$$\{u \in X^* : (u,v)\delta_X \in U \text{ for some } v \in X^*\}.$$

4. *If $L \subseteq X^*$ is regular then so is $\{(w,w) : w \in L\}\delta_X \subseteq X(2, \$)^*$.*

5. *If $U, V \subseteq X(2,\$)^*$ are regular then so is*

$$\{(u,w):\ \text{there exists } v \in X^* \text{ such that } (u,v)\delta_X \in U, (v,w)\delta_X \in V\}\delta_X \subseteq X(2,\$)^*.$$

6. *If $U \subseteq X(2,\$)^*$ is regular then so is*

$$\{u \in X^*:\ \text{if } (u,v)\delta_X \in U \text{ then } u \leq v\}.$$

Definition 3.2. A *deterministic finite state transducer (or just transducer)* is a tuple $\mathcal{A} = (Q, X, Y, q_0, \delta)$ where Q, X, Y are finite sets, Q is the set of states, X is the input alphabet, Y is the output alphabet, $q_0 \in Q$ is the initial state and $\delta\colon Q \times X \to Q \times Y^*$ is the transition function.

We use transducers to rewrite nonempty words over X to words over Y: we simply start at the initial state and traverse through the transducer as usual – the output word is obtained by simply concatenating the output words from each step. For every $w = x_1 \dots x_n \in X^+$ we define $q_1, \dots, q_n \in Q$ and $w_0, w_1, \dots w_n \in Y^*$ recursively by $w_0 = \epsilon$, $q_i = \delta(q_{i-1}, x_i)^{(1)}$ and $w_i = w_{i-1}\delta(q_{i-1}, x_i)^{(2)}$ for all $1 \leq i \leq n$ where ${}^{(k)}$ denotes the kth component. The *map defined by the transducer* $\mathcal{A}$ is $\tau_{\mathcal{A}}\colon X^+ \to Y^*, w \mapsto w_{l(w)}$.

Lemma 3.3. *[14] Let $\mathcal{A} = (Q, X, Y, q_0, \delta)$ be a transducer and let $L \subseteq X^+$ be a regular language. Then $L\tau_{\mathcal{A}}$ is also regular.*

Definition 3.4. A *deterministic finite state transducer with final output (or just transducer with final output)* is a tuple $\mathcal{A} = (Q, X, Y, q_0, \delta, \rho)$ where $\mathcal{B} = (Q, X, Y, q_0, \delta)$ is a transducer and $\rho\colon Q \to Y^*$. The map determined by $\mathcal{A}$ is $\tau_{\mathcal{A}}\colon X^* \to Y^*, w \mapsto w\tau_{\mathcal{B}} \cdot \rho(q)$ where q is the state in which $\mathcal{B}$ finishes reading w. For future reference we define $\tau'_{\mathcal{A}} = \tau_{\mathcal{B}}$.

Lemma 3.5. *Let $\mathcal{A} = (Q, X, Y, q_0, \delta, \rho)$ be a transducer with final output and let $L \subseteq X^*$ be a regular language. Then $L\tau_{\mathcal{A}}$ is also regular.*

Proof. Let $X' = X \cup \{\#\}$ where $\# \notin X$. Then $L' = L\{\#\}$ is a regular language. Define the transducer $\mathcal{A}' = (Q, X', Y, q_0, \delta')$ where δ' extends δ by $\delta'(q, \#) = (q, \rho(q))$. It is straightforward to see that $L\tau_{\mathcal{A}} = L'\tau_{\mathcal{A}'}$, finishing the proof. □

When we try to check how the automaticity of an S-act (or just a semigroup S) behaves with respect to changing the generators then it is very natural to use transducers to rewrite the regular languages associated with the automatic structure. The languages $L_1, \dots, L_n$ appearing in Definition 4.2 behave well with respect to transducing, however, one has to impose extra conditions for the other languages $L_{(i,j)=}$ and $L_{(i,j)x}$.

Definition 3.6. A transducer $\mathcal{A}$ is called *linear* if there exist constants $A, B \in \mathbb{N}$ such that for every $w \in X^*$ we have

$$|l(w\tau_A) - Al(w)| < B.$$

Definition 3.7. A transducer with final output $\mathcal{A}$ is called *linear* if there exist constants $A, B \in \mathbb{N}$ such that for every $w \in X^*$ we have

$$|l(w\tau_A) - Al(w)| < B, \ |l(w\tau'_A) - Al(w)| < B.$$

Definition 3.8. Let X be a finite set and let $(u_1, u_2) \in X^* \times X^*$. Then the *unilength factorisation* of (u_1, u_2) is the unique factorisation $(u_1, u_2) = (\tilde{u}_1, \tilde{u}_2)(\hat{u}_1, \hat{u}_2)$ satisfying $l(\tilde{u}_1) = l(\tilde{u}_2)$ and either $\hat{u}_1 = \epsilon$ or $\hat{u}_2 = \epsilon$ (depending on whether u_2 or u_1 is longer).

Lemma 3.9. *Let $\mathcal{A} = (Q, X, Y, q_0, \delta)$ be a linear transducer. Then there exists a transducer with final output $\mathcal{B} = (Q', X(2,\$), Y(2,\$), q'_0, \delta', \rho')$ such that for every $L \subseteq (X^* \times X^*)\delta_X$ we have that $L\iota_X\tau_{\mathcal{A}}\delta_Y = L\tau_{\mathcal{B}}$. As a consequence if L is regular then so is $L\iota_X\tau_{\mathcal{A}}\delta_Y$.*

Proof. The heuristic approach would be simply to double the transducer $\mathcal{A}$: the transducer $\mathcal{B}$ would have $Q \times Q$ as the set of states, and if it is at state (q_1, q_2) and reads (x_1, x_2) then it simply outputs $(\delta(q_1, x_1)^{(1)}, \delta(q_2, x_2)^{(1)})$, and moves to the state $(\delta(q_1, x_1)^{(2)}, \delta(q_2, x_2)^{(2)})$. The problem is that $\mathcal{B}$ has to output words in $Y(2,\$)^*$ instead of $Y^* \times Y^*$: it is only allowed to output pairs (y_1, y_2) where $l(y_1) = l(y_2)$. To circumvent this problem, if the output would be (y_1, y_2) at state (q_1, q_2), we only output the first $\min(l(y_1), l(y_2))$ letters of y_1 and y_2, and store the rest in the state itself – the linearity condition on $\tau_{\mathcal{A}}$ is there exactly to ensure that there are only finitely many 'remaining parts', so they can be stored in states. So in the state corresponding to (q_1, q_2), we need to incorporate two extra coordinates to store the remainder of the output (though in fact in those states which are visited while rewriting words in $(X^* \times X^*)\delta_X$, one of the coordinates will always be ϵ).

The tedious mathematical definition of $\mathcal{B}$ is the following: let A, B be the constants showing that $\mathcal{A}$ is linear and let $Q' = Q \times Q \times Y^{\leq 2B} \times Y^{\leq 2B}$, with $q'_0 = (q_0, q_0, \epsilon, \epsilon)$. We extend the map δ to $Q \times (X \cup \{\$\})$ by defining $\delta(q, \$) = (q_0, \epsilon)$ for all $q \in Q$. Let $(q'_1, q'_2, s'_1, s'_2) \in Q'$ and $(x_1, x_2) \in X(2,\$)$. For $l = 1, 2$, define $q_l = \delta(q'_l, x_l)^{(1)}$. If $x_l = \$$ for $l = 1$ or $l = 2$ and $l(s'_l) < l(s'_{3-l}\delta(q'_{3-l}, x_{3-l})^{(2)})$ then let

$$\delta'\big((q'_1, q'_2, s'_1, s'_2), (x_1, x_2)\big) = \big(\ (q_1, q_2, \epsilon, \epsilon), (s'_1\delta(q'_1, x_1)^{(2)}, s'_2\delta(q'_2, x_2)^{(2)})\delta_Y\ \big).$$

Otherwise let $(s'_1\delta(q'_1, x_1)^{(2)}, s'_2\delta(q'_2, x_2)^{(2)}) = (o_1, o_2)(\hat{s}_1, \hat{s}_2)$ be the unilength factorisation, and for $l = 1, 2$, let $s_l = \hat{s}_l(2B)$. We define

$$\delta'\big((q'_1, q'_2, s'_1, s'_2), (x_1, x_2)\big) = \big(\ (q_1, q_2, s_1, s_2), (o_1, o_2)\ \big).$$

The function ρ' is defined by

$$\rho'(q'_1, q'_2, s'_1, s'_2) = (s'_1, s'_2)\delta_Y.$$

Note that in the first case (when $x_l = \$$ and $l(s'_l) < l(s'_{3-l}\delta(q'_{3-l}, x_{3-l})^{(2)})$) we have that

$$\delta'\big(\ (q'_1, q'_2, s'_1, s'_2), (x_1, x_2)\ \big)^{(2)} \cdot (s_1, s_2)\delta_Y = \big(s'_1\delta(q'_1, x_1)^{(2)}, s'_2\delta(q'_2, x_2)^{(2)}\big)\delta_Y, \qquad (1)$$

and this equation also holds in the other case provided $s_1 = \hat{s}_1$ and $s_2 = \hat{s}_2$.

Here is an example of a run of the automaton $\mathcal{B}$ – in the example we assume that $\mathcal{A}$ is linear where $B > 4$. The actual states of $\mathcal{A}$ and $\mathcal{B}$ are not important for the demonstration how δ' works, so we ignore them - what matters is what the outputs of $\mathcal{A}$ are in both components and what is stored in the state of $\mathcal{B}$. The first line contains the output of the automata $\mathcal{A}$ after having read the corresponding input letter, the second line the output of $\mathcal{B}$ and the third line the stored remaining parts.

$(y_1y_2y_1y_3, y_1)$	$(y_2y_3y_1y_2, y_2)$	(ϵ, y_2y_1)	(ϵ, y_1y_2)	(ϵ, y_2)	$(\epsilon, y_3y_1y_2)$
(y_1, y_1)	(y_2, y_2)	(y_1y_3, y_2y_1)	(y_2y_3, y_1y_2)	(y_1, y_2)	$(y_1\$\$, y_1y_2)$
$(y_2y_1y_3, \epsilon)$	$(y_1y_3y_2y_3y_1y_2, \epsilon)$	$(y_2y_3y_1y_2, \epsilon)$	(y_1y_2, ϵ)	(y_1, ϵ)	(ϵ, ϵ)

An important property of $\mathcal{B}$ is that it outputs any \$s only in the first case: if $x_l = \$$ and $l(s'_l) < l(s'_{3-l}\delta(q'_{3-l}, x_{3-l})^{(2)})$ for $l = 1$ or $l = 2$. However, if the input is an element of $(X^* \times X^*)\delta_X$ then in this case the subsequent x_l's all will equal \$, and both words stored in the state will equal ϵ, so $\mathcal{B}$ will not output elements of Y in the lth component any more, only \$'s. Furthermore, in this case $\mathcal{B}$ will never output any \$'s in the $3 - l$th component, showing that $(u_1, u_2)\delta_X\tau_{\mathcal{B}} \in (Y^* \times Y^*)\delta_Y$ if $(u_1, u_2) \in X^* \times X^*$.

Now we prove by induction on $\max(l(u_1), l(u_2))$ that $(u_1, u_2)\tau_{\mathcal{A}}\delta_Y = (u_1, u_2)\delta_X\tau_{\mathcal{B}}$ for all $(u_1, u_2) \in X^* \times X^*$. If $\max(l(u_1), l(u_2)) = 0$ then both sides are equal to ϵ. Let us suppose that, for some $N \geq 0$, the equation holds for all $(u'_1, u'_2) \in X^* \times X^*$ satisfying $l(u'_1), l(u'_2) \leq N$, and let $(u_1, u_2) \in X^* \times X^*$ be such that $\max(l(u_1), l(u_2)) = N + 1$. Then $(u_1, u_2)\delta_X = (u'_1, u'_2)\delta_X(x_1, x_2)$ where $\max(l(u_1)', l(u'_2)) = N$ and $(x_1, x_2) \in X(2, \$)$. Let (q'_1, q'_2, s'_1, s'_2) be the state of $\mathcal{B}$ after having read $(u'_1, u'_2)\delta_X$. Then by the induction hypothesis we have

$$(u'_1, u'_2)\tau_{\mathcal{A}}\delta_Y = (u'_1, u'_2)\delta_X\tau_{\mathcal{B}} = (u'_1, u'_2)\delta_X\tau'_{\mathcal{B}} \cdot \rho(q'_1, q'_2, s'_1, s'_2) = (u'_1, u'_2)\delta_X\tau'_{\mathcal{B}} \cdot (s'_1, s'_2)\delta_Y \quad (2)$$

We use Equation (1) to finish the proof - it holds if $x_l = \$$ and $l(s'_l) < l(s'_{3-l}\delta(q'_{3-l}, x_{3-l})^{(2)}$ for $l = 1$ or $l = 2$. In the other cases let $(s'_1\delta(q'_1, x_1)^{(2)}, s'_2\delta(q'_2, x_2)^{(2)}) = (o_1, o_2)(\hat{s}_1, \hat{s}_2)$ be the unilength factorisation and let s_1, s_2 be as in the definition of δ'. If $x_1, x_2 \neq \$$ then $(u'_1, u'_2)\delta_X = (u'_1, u'_2)$ and so

$$(u'_1, u'_2)\tau'_{\mathcal{B}} \cdot (o_1, o_2) \cdot (\hat{s}_1, \hat{s}_2) = \big((u'_1, u'_2)\tau'_{\mathcal{B}} \cdot (s'_1, s'_2)\delta_Y \cdot (\delta(q'_1, x_1)^{(2)}, \delta(q'_2, x_2)^{(2)})\big)\iota_Y =$$

$$= [(u'_1, u'_2)\tau_{\mathcal{A}}\delta_Y \cdot (\delta(q'_1, x_1)^{(2)}, \delta(q'_2, x_2)^{(2)}))]\iota_Y = (u'_1\tau_{\mathcal{A}}, u'_2\tau_{\mathcal{A}}) \cdot (\delta(q'_1, x_1)^{(2)}, \delta(q'_2, x_2)^{(2)}) =$$
$$= (u_1\tau_A, u_2\tau_A),$$

where we used the induction hypothesis, the fact that $(u'_1, u'_2)\tau'_{\mathcal{B}}$ does not contain \$'s (because u'_1, u'_2 don't) and that δ_Y inserts extra \$'s, which are deleted by ι_Y. Altogether, we conclude that $(u_1\tau_{\mathcal{A}}, u_2\tau_{\mathcal{A}}) = (u'_1, u'_2)\tau'_{\mathcal{B}} \cdot (o_1, o_2) \cdot (\hat{s}_1, \hat{s}_2)$ is the unilength factorisation. As a consequence we have that

$$\max(l(\hat{s}_1), l(\hat{s}_2)) = |l(u_1\tau_{\mathcal{A}}) - l(u_2\tau_{\mathcal{A}})| \leq |l(u_1\tau_{\mathcal{A}}) - Al(u_1)| + |l(u_2\tau_{\mathcal{A}}) - Al(u_2)| < 2B$$

by the linearity of $\mathcal{A}$. Thus, $s_1 = \hat{s}_1$ and $s_2 = \hat{s}_2$, so Equation (1) holds.

The third case to check is when $x_l = \$$ and $l(s'_l) \geq l(s'_{3-l}\delta(q'_{3-l}, x_{3-l})^{(2)}$ for $l = 1$ or $l = 2$. In this case $(s'_1\delta(q'_1, x_1)^{(2)}, s'_2\delta(q'_2, x_2)^{(2)}) = (o_1, o_2)(\hat{s}_1, \hat{s}_2)$ implies that

$$2B > l(s'_l) \geq l(s'_{3-l}\delta(q'_{3-l}, x_{3-l})^{(2)}) \geq l(\hat{s}_{3-l}), \text{ and } 2B > l(s'_l) \geq l(\hat{s}_l),$$

so $s_l = \hat{s}_l$ for $l = 1, 2$, showing that Equation (1) indeed holds in all cases.

As a consequence

$$\begin{aligned}(u_1, u_2)\delta_X\tau_{\mathcal{B}} = (u'_1, u'_2)\delta_X\tau'_B \cdot \delta'\big((q'_1, q'_2, s'_1, s'_2), (x_1, x_2)\big)^{(2)} \cdot (s_1, s_2)\delta_Y = \\ = (u'_1, u'_2)\delta_X\tau'_{\mathcal{B}} \cdot (s'_1\delta(q'_1, x_1)^{(2)}, s'_2\delta(q'_2, x_2)^{(2)})\delta_Y =^{(1)} \\ = [(u'_1, u'_2)\delta_X\tau'_B \cdot (s'_1, s'_2)\delta_Y \cdot (\delta(q'_1, x_1)^{(2)}, \delta(q'_2, x_2)^{(2)})]\iota_Y\delta_Y = \\ = [(u'_1, u'_2)\tau_{\mathcal{A}}\delta_Y \cdot (\delta(q'_1, x_1)^{(2)}, \delta(q'_2, x_2)^{(2)})]\iota_Y\delta_Y =^{(2)} \\ = [(u'_1\tau_{\mathcal{A}}, u'_2\tau_{\mathcal{A}}) \cdot (\delta(q'_1, x_1)^{(2)}, \delta(q'_2, x_2)^{(2)})]\delta_Y =^{(3)} \\ = (u_1\tau_{\mathcal{A}}, u_2\tau_{\mathcal{A}})\delta_Y,\end{aligned}$$

where (1) holds because $(u_1, u_2)\delta_X\tau_B \in (Y^* \times Y^*)\delta_Y$, so no \$ is followed by elements of Y in the product preceding δ_Y, so if we apply ι_Y before the final δ_Y, we can include any \$'s in the previous words. Equation (2) holds because the element before δ_Y on the right hand side does not contain any \$'s, so ι_Y and the preceding δ_Y can be removed. And finally, equation (3) follows from the recursive definition of $\tau_{\mathcal{A}}$. The statement of the Lemma immediately follows from this equation. □

In some cases the transducer $\mathcal{A}$ in Lemma 3.9 needs to have final outputs. Incorporating final outputs in the proof of that lemma would complicate it too much, instead we follow the proof of Lemma 3.5.

Lemma 3.10. *Let $\mathcal{A} = (Q, X, Y, q_0, \delta, \rho)$ be a linear transducer with final output and let $L \subseteq (X^* \times X^*)\delta_X$ be regular. Then $L\iota_X\tau_{\mathcal{A}}\delta_Y \subseteq (Y^* \times Y^*)\delta_Y$ is also regular.*

Proof. Let $X' = X \cup \{\#\}$. Since $\mathcal{A}$ is linear, there exist $A, B \in \mathbb{N}$ such that

$$|l(w\tau_{\mathcal{A}}) - Al(w)|, |l(w\tau'_{\mathcal{A}}) - Al(w)| < B.$$

Let us fix a word $w \in Y^*$ such that $l(w) = A$. We define a linear transducer $\mathcal{A}' = (Q \cup \{F\}, X', Y, q_0, \delta')$ the following way: δ' extends δ by $\delta'(q, \#) = (F, \rho(q))$ and $\delta'(F, x) = (F, w)$ for all $x \in X'$ (this is necessary to ensure the linearity of $\mathcal{A}'$ on words which contain several #s). Then it is easy to see that $\mathcal{A}'$ is also linear and that for any $u \in X(2, \$)^*$ we have

$$u\tau_{\mathcal{A}} = u\#\tau_{\mathcal{A}'}. \tag{3}$$

We can think of L as being a regular language over X'. Now let $L' = (L\iota_{X'}(\#, \#))\delta_{X'}$, which is regular by the dual of Lemma 2.3. Note that Equation (3) implies $(L\iota_X\tau_{\mathcal{A}})\delta_Y = L'\iota_{X'}\tau_{\mathcal{A}'}\delta_Y$, which is regular by Lemma 3.9. □

4. Definition(s) of automatic semigroup acts

In this section we introduce two different notions of automaticity, namely $+$-automaticity and $*$-automaticity and investigate their relationship. The difference stems from the fact that generation of subacts of an act is more complicated when the underlying semigroup is not a monoid.

Definition 4.1. Let A be an S-act and let $A_g \subseteq A$. Then the subact *$+$-generated by A_g* is $\langle A_g \rangle_+ = A_g.S$. The subact *$*$-generated by A_g* is $\langle A_g \rangle_* = A_g.S \cup A_g$.

Note that for monoid acts, the two definitions coincide. For semigroup acts, the main problem is that $+$-generated subacts may not contain the generating sets – actually, some acts even cannot be $+$-generated at all. To avoid this problem we use $*$-generation throughout the paper, except for this section, which is mainly dedicated to comparing the two notions of automaticity arising from these two different definitions of generation.

For the remainder of the paper whenever S is an X-generated semigroup then we will denote by φ the surjective homomorphism $X^+ \to S$ which fixes elements of X. Note that if A is an S-act then A can be considered an X^+-act by defining $a.u = a.u\varphi$ for every $a \in A$ and $u \in X^+$. Furthermore, though the homomorphism φ may not extend to the monoid X^*, A can still be considered an X^*-act by defining furthermore $a.\epsilon = a$ for every $a \in A$. In the sequel we will make use of this action of X^* on A. Note that using this action we have that S is $+(*)$-generated by $A_g \subseteq A$ if and only if $A = A_g.X^+ (A = A_g.X^*)$.

Definition 4.2. Let S be a semigroup generated by X, let $A_g = \{a_1, \dots, a_n\}$ be a finite $+(*)$-generating set of A and let $L_1, \dots, L_n \subseteq X^+(X^*)$ be regular languages. We say that $(A_g, X, L_1, \dots, L_n)$ forms an $+(*)$-automatic structure for A if the following hold:

1. $A = \bigcup_{i=1}^n a_i.L_i$,
2. $L_{(i,j)=} = \{(u,v) \in L_i \times L_j : a_i.u = a_j.v\}\delta_X$ is a regular language for all $1 \leq i,j \leq n$,
3. $L_{(i,j)x} = \{(u,v) \in L_i \times L_j : a_i.ux = a_j.v\}\delta_X$ is a regular language for all $1 \leq i,j \leq n$ and $x \in X$.

We say that the automatic structure $(A_g, X, L_1, \dots, L_n)$ is *with uniqueness* if for every $1 \leq i,j \leq n, u \in L_i$ and $v \in L_j$ we have $a_i.u = a_j.v \Longrightarrow i = j, u = v$.

As the following Lemma shows, letters in the definition of the languages $L_{(i,j)x}$ may be replaced by words. We omit the proof, for it is essentially the same as that of Proposition 5.2 in [5].

Lemma 4.3. *Let $(A_g, X, L_1, \dots, L_n)$ be a $+(*)$-automatic structure for the S-act A. Then for every word $w \in X^+$ the language*

$$L_{(i,j)w} = \{(u,v) \in L_i \times L_j : a_i.uw = a_j.v\}\delta_X$$

is regular.

If an S-act has a $+(*)$-automatic structure $(A_g, X, L_1, \ldots, L_n)$ then we say that S is automatic with respect to the generating sets (A_g, X). We say that the S-act A is $+(*)$-automatic if it is $+(*)$-automatic with respect to some generating sets. In the remaining part of this section we show that $*$-automaticity is a more general notion than $+$-automaticity and that it generalizes the usual notion of automaticity for semigroups and monoids.

Lemma 4.4. *If $A = \langle A_g \rangle_+$ then A is $+$-automatic with respect to (A_g, X) if and only if A is $*$-automatic with respect to (A_g, X).*

Proof. Note that the direct part is obvious: if $(A_g, X, L_1, \ldots, L_n)$ is a $+$-automatic structure for A, then it is also a $*$-automatic structure. For the converse part let us suppose that $(A_g, X, L_1, \ldots, L_n)$ is a $*$-automatic structure for A. Since A_g is a $+$-generating set, for every $1 \leq i \leq n$ there exists a word $w_i \in X^+$ and $1 \leq k_i \leq n$ such that $a_i = a_{k_i}.w_i$. For any i, let $W_i = \{w_l : k_l = i\}$ and $L'_i = (L_i \setminus \{\epsilon\}) \cup W_i$ - note that these languages are all regular. Furthermore, the choice of the words w_i ensure that $A = \bigcup_{i=1}^n a_i.L'_i$.

To see that the languages $L_{(i,j)=}$ are regular, first we define the languages

$$\tilde{L}_{(l,j)} = \{v \in X^* : (\epsilon, v)\delta_X \in L_{(l,j)=}\},\ \tilde{L}^d_{(i,l)} = \{u \in X^* : (u, \epsilon)\delta_X \in L_{(i,l)=}\}.$$

By Lemma 3.1 we have that these languages are regular, and so the languages

$$O = \bigcup_{w_l \in W_i} (\{w_l\} \times \tilde{L}_{(l,j)})\delta_X,\ O^d = \bigcup_{w_l \in W_j} (\tilde{L}_{(i,l)} \times \{w_l\})\delta_X$$

are also regular by the same Lemma. For any $1 \leq i, j \leq n$, we have that

$$\begin{array}{rl} L'_{(i,j)=} & = \\ \{(u, v) \in L'_i \times L'_j : a_i.u = a_j.v\}\delta_X & = \\ \{(u, v) \in L_i \times L_j : u, v \neq \epsilon, a_i.u = a_j.v\}\delta_X & \cup \\ \{(w_l, v) : w_l \in W_i, (\epsilon, v)\delta_X \in L_{(l,j)=}\}\delta_X & \cup \\ \{(u, w_l) : w_l \in W_j, (u, \epsilon)\delta_X \in L_{(i,l)=}\}\delta_X & \cup \\ \{(w_l, w_m) : w_l \in W_i, w_m \in W_j, \epsilon \in L_{(l,m)=}\}\delta_X & = \\ O_1 \cup O \cup O^d \cup O_2. & \end{array}$$

Note that $O_1 = L_{(i,j)=} \cap (X \times X)X(2, \$)^*$, and O_2 is finite, so they are regular, showing that $L'_{(i,j)=}$ is regular for all $1 \leq i, j \leq n$.

The proof that the languages $L'_{(i,j)x}$ are regular parallels this one, so $(A_g, X, L'_1, \ldots, L'_n)$ is a $+$-automatic structure for A_g, showing that A is indeed $+$-automatic with respect to (A_g, X). $\square$

As Lemma 4.4 shows, whenever $+$-automaticity makes sense (that is, whenever the generating set $+$-generates the act), it is equivalent to $*$-automaticity, that is, $*$-automaticity extends the notion of $+$-automaticity. As a consequence we only use $*$-automaticity in the rest of the paper, and to simplify notation we simply call it automaticity.

Lemma 4.5. *A semigroup S is automatic with respect to a finite generating set X if and only if S as a right S-act is automatic with respect to (X, X).*

Proof. For the direct part, let (X, L) be an automatic structure for S where $X = \{x_1, \ldots, x_n\}$. Then the languages $L_=$ and L_x are regular for all $x \in X$. For every $1 \le j \le n$, let $L_j = \{u \in X^* : x_j u \in L\}$. By Lemma 3.1, the languages L_j are regular. Furthermore, since $L\varphi = S$ we have that $S = \bigcup_{i=1}^n x_j.L_j$. For every $1 \le i, j \le n$ we have (note that $(u, v) \in L_=$ can equal ϵ)

$$L_{(i,j)=} = \{(u, v) \in L_i \times L_j : x_i.u = x_j.v\}\delta_X = \{(u, v)\delta_X \in X(2, \$)^* : (x_i, x_j)(u, v) \in L_=\},$$

and similarly

$$L_{(i,j)x} = \{(u, v) \in L_i \times L_j : x_i.ux = x_j.v\}\delta_X = \{(u, v)\delta_X \in X(2, \$)^* : (x_i, x_j)(u, v) \in L_x\},$$

showing by Lemma 3.1, Part (1) that the languages $L_{(i,j)=}$ and $L_{(i,j)x}$ are regular.

For the converse part let $(X, X, L_1, \ldots, L_n)$ be an automatic structure for the right S-act S. Define the regular language $L = \bigcup_{i=1}^n x_i L_i \subseteq X^+$. Then $L\varphi = S$. Furthermore,

$$\begin{aligned} L_= = \{(u, v) \in L \times L : u\varphi = v\varphi\}\delta_X &= \\ \textstyle\bigcup_{1\le i,j\le n}\{(x_i u', x_j v') \in L \times L : (u', v') \in L_i \times L_j\}\delta_X &= \\ \textstyle\bigcup_{1\le i,j\le n}(x_i, x_j)L_{(i,j)=} & \end{aligned}$$

is a regular language. Similarly

$$L_x = \bigcup_{1\le i,j\le n} L_{(i,j)x}$$

is a regular language for all $x \in X$, completing the proof. □

5. An automatic structure with uniqueness

In this section we show that every automatic S-act has an automatic structure with uniqueness. Part of the proof follows the semigroup case.

Theorem 5.1. *Let A be an automatic S-act. Then there exists an automatic structure with uniqueness for A.*

Proof. Let $(A_g, X, L_1, \ldots, L_n)$ be an automatic structure for A. We will achieve uniqueness in two steps: first we will ensure that $a_i.L_i' \cap a_j.L_j' = \emptyset$ for all $1 \le i \ne j \le n$. This is quite easy to achieve, namely, for every $1 \le i < n$, let

$$L_i' = L_i \setminus \{u \in L_i : a_i.u = a_j.v \text{ for some } j > i \text{ and } v \in L_j\} =$$

$$= L_i \setminus \bigcup_{j=i+1}^{n} \{u \in L_i : (u, v)\delta_X \in L_{(i,j)=}\},$$

which is a regular language by Lemma 3.1. We also define $L'_n = L_n$. The definition of L'_i ensures that $\bigcup_{i=1}^n a_i.L'_i = \bigcup_{j=1}^n a_j.L_j$, and that $a_i.L'_i \cap a_j.L'_j = \emptyset$ for all $1 \leq i \neq j \leq n$.

The next step is to replace the languages L'_i by L''_i such that $a_i.u = a_i.v$ implies $u = v$ for all $u, v \in L''_i$. For every $1 \leq i \leq n$, let us define

$$L''_i = \{u \in L'_i : \text{ if } (u,v) \in L'_{(i,i)=} \text{ then } u < v \text{ in the shortlex order}\},$$

that is, for each $a \in a_i.L'_i$, we keep only the shortlex-minimal $u \in L'_i$ such that $a = a_i.u$ By Lemma 3.1 we have that L''_i is a regular language. Furthermore, clearly $a_i.L'_i = a_i.L''_i$, so $A = \bigcup_{i=1}^n a_i.L''_i$. The languages L''_i now satisfy that whenever $u \in L_i, v \in L_j$ such that $a_i.u = a_j.v$ then $i = j$ and $u = v$. As a consequence

$$L''_{(i,j)=} = \begin{cases} \emptyset & \text{if } i \neq j \\ \{(u,u) : u \in L_i\}\delta_X & \text{if } i = j. \end{cases}$$

Thus $L''_{(i,j)=}$ is regular for all $1 \leq i, j \leq n$ by Lemma 3.1. Moreover, since $L''_i \subseteq L_i$ for all $1 \leq i \leq n$, we also have that

$$L''_{(i,j)x} = L_{(i,j)x} \cap (L''_i \times L''_j)\delta_X,$$

and hence is a regular language. We deduce that $(A_g, X, L''_1, \ldots, L''_n)$ is indeed an automatic structure with uniqueness for the S-act A. □

6. Change of generators

In this section we show that under some conditions on S, automaticity of S-acts is independent of the choice of the generating sets. Note that for S-acts there are two types of generators: one is for the act itself and one is for S. First we show that automaticity is independent of the choice of the act generating set.

Lemma 6.1. *Let $S = \langle X \rangle$, let A be an S-act and let $A_g, A'_g \subset A$ be two finite generating sets for A. Then A is automatic with respect to the generating sets (A_g, X) if and only if it is automatic with respect to (A'_g, X).*

Proof. Let us suppose that A is automatic with respect to (A_g, X). Then there exists an automatic structure $(A_g, X, L_1, \ldots, L_n)$ for A where $A_g = \{a_1, \ldots, a_n\}$. Let $A'_g = \{a'_1, \ldots, a'_m\}$. Then for every $1 \leq i \leq n$ there exist $1 \leq p_i \leq m$ and $w_i \in X^*$ such that $a_i = a'_{p_i}.w_i$. For any $1 \leq p \leq m$ we define the languages

$$L'_p = \bigcup_{i:p_i=p} w_i L_i.$$

Clearly the languages L'_p are regular and by the definition of the indices p_i we also have that $A = \bigcup_{p=1}^m a'_p.L'_p$. It only remains to check that the languages $L'_{(p,q)=}$ and $L'_{(p,q)y}$ are regular. For this note that

$$\begin{aligned} L'_{(p,q)=} = \{(u',v') \in L'_p \times L'_q : a'_p.u' = a'_q.v'\}\delta_X &= \\ \bigcup_{i:p_i=p, j:p_j=q} \{(w_i u, w_j v) \in L'_p \times L'_q : a_i.u = a'_p.w_i u = a'_q.w_j v = a_j.v\}\delta_X &= \\ = \bigcup_{i:p_i=p, j:p_j=v} \big((w_i, w_j) L_{(i,j)=} \iota_X\big)\delta_X. \end{aligned}$$

Since the languages $((w_i, w_j)L_{(i,j)=}\iota_X)\delta_X$ are regular by Lemma 3.1, we conclude that $L_{(p,q)=}$ is regular for all $1 \le p, q \le m$. Similarly one can show that

$$L_{(p,q)x} = \bigcup_{i:p_i=p, j:p_j=v} \big((w_i, w_j)L_{(i,j)x}\iota_X\big)\delta_X,$$

showing that $(A'_g, X, L'_1, \ldots, L'_m)$ is indeed an automatic structure for A, and finishing the proof. □

Changing the generators of the semigroup S is more complicated, one has to expect some conditions, since in general, automaticity of semigroups does depend on the choice of the generators (see [5]). In the sequel we aim to strengthen the most general result currently existing, namely that of [9]. The main idea is to show that under a certain condition, almost all words over one generators can be rewritten to words over the other generators in both a short and a long way, which enables one to use Lemma 3.9. This idea is captured in the following lemma.

Lemma 6.2. *Let $A = \langle A_g \rangle$ be an S-act where S satisfies $S^N = S^{N+1}$ for some $N \in \mathbb{N}$. Let X and Y be finite generating sets of S. Then if A is automatic with respect to (A_g, X) then A is automatic with respect to (A_g, Y).*

Proof. Let $\varphi\colon X^+ \to S$ and $\psi\colon Y^+ \to S$ be the homomorphisms fixing X and Y, respectively. Since $S^N = S^{N+1}$ we have that if $u \in X^+$ satisfies $l_X(u) \ge N$ then for every $n \ge N$ there exists $v \in Y^+$ such that $l_Y(v) \ge n$ and $u\varphi = v\psi$. For every $u \in X^{\le N}$ let us fix $p_u^{(s)} \in Y^+$ (the superscript stands for 'short') such that $u\varphi = p_u^{(s)}\psi$. Let $L \ge \max(\{l_Y(p_u^{(s)}) : u \in X^{\le N}\})$ such that N divides L. For every $u \in X^N$ let us also fix $p_u^{(l)} \in Y^+$ (the superscript stands for 'long') such that $u\varphi = p_u^{(l)}\psi$ and $l_Y(p_u^{(l)}) > L$. Let $B = \max(\{l_Y(p_u^{(l)}) : u \in X^N\})$. Summing up, for every $u \in X^{\le N}$ and $v \in X^N$ we have that

$$u\varphi = p_u^{(s)}\psi,\ v\varphi = p_v^{(l)}\psi,\ 1 \le l_Y(p_u^{(s)}) \le L < l_Y(p_v^{(l)}) \le B. \tag{4}$$

Let $(A_g, X, L_1, \ldots, L_n)$ be an automatic structure for A where $A_g = \{a_1, \ldots, a_n\}$. We are going to define a linear transducer with final output $\mathcal{A} = (Q, X, Y, q_0, \delta, \rho)$ such that for every $u \in X^+$ we have that $u\varphi = u\tau_{\mathcal{A}}\psi$. Then by making use of Lemma 3.10 we will show that $(A_g, Y, L_1\tau_{\mathcal{A}}, \ldots, L_n\tau_{\mathcal{A}})$ is an automatic structure for A. The transducer will rewrite words in the following way: words u such that $l_X(u) \le N$ will be simply rewritten to $p_u^{(s)}$. Longer words will be rewritten by rewriting blocks of length N to their short or long counterparts over Y, depending on how far the length of the previous output is from $\frac{L}{N} l_X(u')$ where u' is the previous input. The tedious definition of $\mathcal{A}$ is the following: let $Q = \{-B, \ldots, B\} \times X^{\le N-1}, q_0 = (0, \epsilon)$. The transition function $\delta\colon Q \times X \to Q \times Y^*$ is defined by

$$\delta((i, w), x) = \begin{cases} ((i, wx), \epsilon) & \text{if } l_X(w) < N-1 \\ ((i + l_Y(p_{wx}^{(l)}) - L, \epsilon), p_{wx}^{(l)}) & \text{if } l_X(w) = N-1, i \le 0 \\ ((i + l_Y(p_{wx}^{(s)}) - L, \epsilon), p_{wx}^{(s)}) & \text{if } l_X(w) = N-1, i > 0, \end{cases}$$

and $\rho\colon Q \to Y^*$ is defined by $\rho((i,w)) = p_w^{(s)}$. First of all note that δ is well-defined, because if $i \leq 0$ then $i \leq i + l_Y(p_{wx}^{(l)}) - L \leq B - L \leq B$ and if $i > 0$ then $i \geq i + l_Y(p_{wx}^{(s)}) - L > -L > -B$. Furthermore, it is clear that $u\varphi = u\tau_{\mathcal{A}}\psi$ for every $u \in X^+$.

The next step is to show by induction on $l_X(w)$ that if $w = w_1 w_2 \dots w_k w'$ where $l_X(w_1) = \dots = l_X(w_k) = N$ and $l_X(w') < N$ then $\mathcal{A}$ finishes reading w in the state $(l_X(w\tau'_{\mathcal{A}}) - Lk, w')$. If $l_X(w) = 0$ then the statement is true. Let us suppose that there exists $O \in \mathbb{N}$ such that the statement is true whenever $l_X(w') < O$ and let $l_X(w) = O$. Write $w = w_1 \dots w_k w'$ where $l_X(w_1) = \dots = l_X(w_k) = N$ and $l_X(w') < N$. If $w' \neq \epsilon$ then let $w' = w''x$ where $x \in X$. In this case by the induction hypothesis we have that $\mathcal{A}$ is in the state $(l_X(w_1 \dots w_k\tau'_{\mathcal{A}}) - Lk, w'')$ after having read $w_1 \dots w_k w''$, and then after reading x, it will move to the state $(l_X(w_1 \dots w_k \tau\mathcal{A}') - Lk, w''x)$ by the definition of δ. On the other hand if $w' = \epsilon$ then let $w_k = w'_k x$ where $x \in X$. By the induction hypothesis $\mathcal{A}$ finishes reading $w_1 \dots w_{k-1} w'_k$ in the state $(l_X(w_1 \dots w_{k-1}\tau'_{\mathcal{A}}) - L(k-1), w'_k)$. In the next step, when reading x, $\mathcal{A}$ outputs u, which either equals $p_{w_k}^{(s)}$ or $p_{w_k}^{(l)}$, and moves to the state $(l_X(w_1 \dots w_{k-1}\tau'_{\mathcal{A}}) - L(k-1) + l_X(u) - L, \epsilon)$. Note that by the definition of $\tau'_{\mathcal{A}}$ and δ we have that $w_1 \dots w_k \tau'_{\mathcal{A}} = w_1 \dots w_{k-1}\tau'_{\mathcal{A}} \cdot u$, so certainly $l_X(w_1 \dots w_k\tau'_{\mathcal{A}}) = l_X(w_1 \dots w_{k-1}\tau'_{\mathcal{A}}) + l_X(u)$, showing that $\mathcal{A}$ indeed finishes reading w in the state $(l_X(w_1 \dots w_k\tau'_{\mathcal{A}}) - Lk, w')$. As a consequence, for every $w = w_1 \dots w_k w'$, $l_X(w_1) = \dots = l_X(w_k), l_X(w') < N$ we have

$$\left|l_X(w\tau_{\mathcal{A}}) - \tfrac{L}{N} l_X(w)\right| = \left|l_X(w\tau'_{\mathcal{A}}) + l_Y(p_{w'}^{(s)}) - Lk - l_X(w')\right| \leq \\ |l_X(w\tau'_{\mathcal{A}}) - Lk| + \left|l_Y(p_{w'}^{(s)}) - l_X(w')\right| \leq B + L$$

and

$$\left|l_X(w\tau'_{\mathcal{A}}) - \frac{L}{N} l_X(w)\right| = |l_X(w\tau'_{\mathcal{A}}) - Lk - l_X(w')| \leq |l_X(w\tau'_{\mathcal{A}}) - Lk| + l_X(w') \leq B + L,$$

so $\mathcal{A}$ is indeed linear with constants L/N and $B + L + 1$.

For every $1 \leq i \leq n$, let $L'_i = L_i\tau_{\mathcal{A}}$. These languages are regular by Lemma 3.5, and since $u\varphi = u\tau_{\mathcal{A}}\psi$ for every $u \in X^+$, they satisfy $A = \bigcup_{i=1}^n a_i.L'_i$. Furthermore, we have that for every $1 \leq i, j \leq n$,

$$L'_{(i,j)=} = \{(u', v') \in L'_i \times L'_j : a_i.u' = a_j.v'\}\delta_Y =$$

$$= \{(u, v)\iota_X\tau_{\mathcal{A}} : (u, v) \in L_{(i,j)=}\}\delta_Y = L_{(i,j)=}\iota_X\tau_{\mathcal{A}}\delta_Y,$$

and these languages are also regular by Lemma 3.10. Similarly we have that $L'_{(i,j)y} = L_{(i,j)w}\iota_X\tau_{\mathcal{A}}\delta_Y$, where $w \in X^+$ is a word such that $w\varphi = y\psi$. This shows by Lemma 4.3 that the languages $L'_{(i,j)y}$ are also regular, so $(A_g, Y, L'_1, \dots, L'_n)$ is an automatic structure for A.

□

By combining Lemmas 6.1 and 6.2, one obtains the result of the section.

Theorem 6.3. *Let S be a semigroup satisfying $S^N = S^{N+1}$ for some $N \in \mathbb{N}$ and let A be an S-act. Then the automaticity of A is independent of the choice of the generators of S and of A.*

Corollary 6.4. *Let S be a semigroup satisfying $S^N = S^{N+1}$ for some $N \in \mathbb{N}$. Then the automaticity of S is independent of the choice of the generators of S.*

7. Equality problem

Let S be a semigroup generated by a finite set X. The word problem is said to be solvable for S, if there exists an algorithm which decides whether given any two words $u, v \in X^+$ represent the same element in S or not. Automatic groups and semigroups have solvable word problem in quadratic time. In this section we will introduce the concept of the equality problem for S-acts, and show that the equality problem is solvable for the right S-act S if and only if the word problem is solvable for the semigroup S. Moreover we show that the equality problem for automatic semigroup acts is solvable in quadratic time.

Let A be an S-act, generated by a finite set $A_g \subseteq A$. If there exists an algorithm that decides whether for any $(a_i, a_j) \in A_g \times A_g$ and for any two given words $u, v \in X^*$, $a_i.u = a_j.v$ holds, then we say that the equality problem is solvable for the S-act A. Not surprisingly, the equality problem of the right S-act S is connected to the word problem of the semigroup S.

Proposition 7.1. *Let S be a semigroup generated by a finite set X. Then the word problem is solvable for S if and only if the equality problem is solvable for the right S-act S.*

Proof. Note that if S is generated by X, then the right S-act S is generated by X. Assume that the word problem is solvable for S. Let $x_i, x_j \in X,\ u, v \in X^+$ and $u' = x_i u,\ v' = x_j v$. By our assumptions, there exists an algorithm which decides whether or not $x_i.u = u'\varphi = v'\varphi = x_j.v$, proving that the equality problem is solvable for the right S-act S.

For the converse, assume that the equality problem is solvable for the right S-act S. Let $u, v \in X^+$, and assume that $u \equiv x_i u'$ and that $v \equiv x_j v'$. By our assumptions, there exists an algorithm that decides whether or not $u\varphi = x_i.u' = x_j.v' = v\varphi$ holds in S or not, proving that the word problem is indeed solvable for S. $\square$

To show that the equality problem is solvable for automatic S-acts, first we prove an analogue of the well-known Pumping Lemma for regular languages.

Proposition 7.2. *Let A be an automatic S-act. Let $(A_g, X, L_1, \ldots, L_n)$ be an automatic structure for A. Then, there exists a constant N such that for any $a_i \in A_g$ and $u \in L_i$ the following hold for the elements $a = a_i.u$ or $a = a_i.(u \cdot x)$ of A, where $x \in X$:*

(i) There exists $a_j \in A_g$ and $v \in L_j$ such that $|v| \leq |u| + N$ and $a = a_j.v$.

(ii) If there exists $a_j \in A_g$ and $v \in L_j$, such that $|v| > |u| + N$ and $a = a_j.v$, then there exist infinitely many $w \in L_j$ such that $a = a_j.w$.

Proof. (i) For all $(a_j, a_k) \in A_g \times A_g$ consider finite state automata $\mathcal{A}_{(j,k)=}$, $\mathcal{A}_{(j,k)x}$, $(x \in X)$ accepting the regular languages $L_{(j,k)=}$, $L_{(j,k)x}$ respectively. Let N be greater than the number of states in any of the automata defined.

Let $a_i \in A_g$ and $u \in L_i$. If $a = a_i.u$, then (i) is straightforward. Assume that $a = a_i.(u \cdot x)$, where $x \in X$. Since $(A_g, X, L_1, \ldots, L_n)$ is an automatic structure for A, there exists $a_j \in A_g$ and $v \in L_j$ such that $a = a_j.v$. If $|v| \le |u| + N$, then we are finished. Assume that $|v| > |u| + N$. Clearly $(u, v)\delta_X$ is accepted by $\mathcal{A} = \mathcal{A}_{(i,j)_x}$. After reading through all of u, we visit a state, say q, in $\mathcal{A}$ at least twice. Removing the subword of v between successive visits to q, we get a shorter word v_1, moreover $(u, v_1)\delta_X$ is still accepted by $\mathcal{A}_{(i,j)_x}$. Repeating this procedure as many times as necessary, we obtain a word w, which satisfies that $|w| \le |u| + N$ and $a = a_j.w$.

(ii) Assume that there exists $a_j \in A_g$ and $v \in L_j$, such that $|v| > |u| + N$ and $a = a_j.v$. In particular we have that $(u, v)\delta_X$ is accepted by one of the automata defined; say by $\mathcal{A}$. After reading through all of u, we visit a state, say q, in $\mathcal{A}$ more then once. Inserting the subword in between two successive visits to q in v in the appropriate place, we will get a longer word v_1 so that $(u, v_1)\delta_X$ is still accepted by $\mathcal{A}$. Repeating this process as many times as we want, we get the desired result. □

Theorem 7.3. *Let A be an S-act and let $(A_g, X, L_1, \ldots, L_n)$ be an automatic structure for A where $A_g = \{a_1, \ldots, a_n\}$. Then the equality problem for A is solvable in quadratic time.*

Proof. Let N be the number guaranteed by Proposition 7.2. That is, for every $a_i \in A_g$, $u \in X^*$ and $x \in X$ we have that there exist $a_j \in A_g$ and $v \in L_j$ such that $a_i.ux = a_j.v$ and $l(v) \le l(u) + N$.

For every $a_i \in A_g$, let us fix $a_{k_i} \in A_g$ and $w_i \in L_{k_i}$ such that $a_i = a_{k_i}.w_i$. First we show that for any $a_i \in A_g$ and $u \in X^*$ we can find $a_j \in A_g$ and $w \in L_j$ such that $a_i.u = a_j.w$ in quadratic time. Let $u = x_1 \ldots x_l$. First of all note that $a_i.u = a_{k_i}.w_i u$. The algorithm finding a_j and w is as follows: first we search for $a_{o_1} \in A_g$ and $v_1 \in L_{o_1}$ such that $a_{k_i}.w_i x_1 = a_{o_1}.v_1$.

This is done by scanning the regular languages $L_{(k_i,1)x_1}, \ldots, L_{(k_i,n)x_1}$ for an element $(w_i, v_1)\delta_X$ satisfying $l(v_1) \le l(w_i u) + N$. This scanning is done by running first the automaton corresponding to $L_{(k_i,1)x_1}$ - we start at the initial vertex, and check where the $n+1$ inputs $(w_i^{(1)}, x')$ lead us (where $w_i^{(p)}$ is the pth letter of w_i if $p \le l(w_i)$ and $w_i^{(p)} = \$$ for $p > l(w_i)$) and x' runs through $X \cup \{\$\}$. We then check from the reached states where the (at most) $n+1$ inputs $(w_i^{(2)}, x')$ lead us where x' again runs through $X \cup \{\$\}$ (note that if we chose \$ in the first step then we have to choose \$ now). Since in every step we have at most $n+1$ choices, we can decide in at most $(n+1)(l(w_i u) + N) = O(l)$ time whether such a $v_1 \in L_1$ exists - if we reach a final state after having read all of w_i, then we have found v_i, otherwise we stop after $l(w_i u) + N$ steps. So it takes $O(l)$ time to decide whether there exists $v_1 \in L_1$ such that $a_{k_i}.w_i x_1 = a_1.v_1$ satisfying $l(v_1) \le l(w_1 u) + N$, and if there exists, we can also find it. If such a $v_1 \in L_1$ does not exist, then we check for $v_1 \in L_2, \ldots, L_n$ - by Proposition 7.2, we will find v_1 in at most $n \cdot O(l) = O(l)$ time.

So we can find $v_1 \in L_{o_1}$ in $O(l)$ time. Similarly it takes $O(l)$ time to find $v_2 \in L_{o_2}$ such that $a_{k_1}.w_i x_1 x_2 = a_{o_1}.v_1 x_2 = a_{o_2}.v_2, l(v_2) \leq l(w_1 u) + N$, and so on. Altogether we need l steps, each step taking at most $O(l)$ time, so to find $v_l \in L_{o_l}$ such that $a_i.u = a_{k_i}.w_i x_1 \dots x_l = a_{o_l}.v_l$ takes at most $O(l^2)$ time.

Now let $a, a' \in A_g, u, u' \in X^*$ and let $l = \max(l(u), l(u'))$. Then we can find $a_i, a_j \in A_g, v \in L_i$ and $v' \in L_j$ such that $a.u = a_i.v$ and $a.u' = a_j.v'$ in $O(l^2)$ time. After this, it takes $O(l)$ time to check whether $(v, v')\delta_X \in L_{(i,j)=}$, showing that the equality problem is solvable in quadratic time. □

Remark 7.4. Note that Theorem 7.3 does not imply that the so-called uniform word problem is solvable. It shows that for every automatic act A, there exists an algorithm solving the equality problem, however, the algorithm uses more information about A than is given by the automatic structure. In other words, we do not know currently if there exists an algorithm which inputs the automatic structure $(A_g, X, L_1, \dots, L_n)$, the corresponding automata for the regular languages $L_{(i,j)=}, L_{(i,j)x}$ and elements $a_i, a_j \in A_g$ and $u, v \in X^*$, and decides whether $a_i.u = a_j.v$.

8. Fellow traveller property

In this section we first associate a directed labelled graph $\Gamma_X(A, S)$ to each S-act A and introduce the notion of distance in $\Gamma_X(A, S)$. With these tools, we give the definition of the fellow traveller property and claim that the introduced notion is a generalization of the fellow traveller property given for semigroups and groups. Finally we prove in this section that if A is an automatic S-act, then $\Gamma_X(A, S)$ possesses the fellow traveller property.

As before, S will denote a semigroup, X a finite generating set for S. We denote by $\varphi : X^+ \to S$ the homomorphism extending the identity map $\mathrm{id}_X : X \to S$. If S is a group, then we will assume that X is closed under taking inverses.

Intuitively we can think of an S-act A as a directed labelled graph $\Gamma_X(A, S)$, in which the vertices are elements of A, the labels are elements of X and there is an arrow from a to b with label x precisely when $a.x = b$. We write the arrows of $\Gamma_X(A, S)$ as ordered triples (a, x, b) indicating that $a.x = b$. We let $\mathcal{V}(\Gamma_X(A, S))$ denote the set of vertices and $\mathcal{A}(\Gamma_X(A, S))$ denote the set of arrows of $\Gamma_X(A, S)$. Clearly $\Gamma_X(A, S)$ is not necessarily a connected graph.

We define a *path* between two vertices a and b of $\Delta = \Gamma_X(A, S)$ to be a sequence of edges:

$$a = a_0 \overset{x_1}{\text{———}} a_1 \overset{x_2}{\text{———}} a_2 \dots a_{n-1} \overset{x_n}{\text{———}} a_n = b$$

such that either $(a_i, x_i, a_{i+1}) \in \mathcal{A}(\Delta)$ or $(a_{i+1}, x_i, a_i) \in \mathcal{A}(\Delta)$, $(0 \leq i \leq n-1)$ and say that the *length of the path* is n. For $a, b \in \mathcal{V}(\Delta)$, we define the *distance* $d_\Delta(a, b)$ between a and b to be the length of the shortest path connecting a and b and say that the *distance is infinite* if a and b belong to different components of Δ.

Example 8.1. If A is the right S-act S, then $\Delta = \Gamma_X(A, S)$ is the right Cayley graph $\Gamma = \Gamma_X(S)$ of S. We have for all $a, b \in S$ that $d_\Delta(a, b) = d_\Gamma(a, b)$. If S is a group then Δ is a connected graph and $(g, x, h) \in \mathcal{A}(\Delta)$ if and only if $(h, x^{-1}, g) \in \mathcal{A}(\Delta)$. In other words, any two vertices are connected via a directed path.

Definition 8.2. Let $L_1, \ldots, L_n \subseteq X^*$ satisfy $A = \bigcup_{i=1}^{n} a_i.L_i$. The graph $\Delta = \Gamma_X(A, S)$ is said to have the *fellow traveller property with respect to L_i, L_j and a_i, a_j*, if there exists a constant $k \in \mathbb{N}$ such that whenever $(u, v) \in L_i \times L_j$ with $d_\Delta(a_i.u, a_j.v) \leq 1$, then $d_\Delta(a_i.u(t), a_j.v(t)) \leq k$ for all $t \geq 1$. We say that $\Gamma_X(A, S)$ possesses the *fellow traveller property* with respect to $L_1, \ldots, L_n$ and A_g, if it possesses the fellow traveller property with respect to any two languages L_i, L_j and corresponding generators a_i, a_j.

We have seen that if S is a semigroup then $\Gamma_X(S, S)$ is the Cayley graph of S. Now we show that the fellow traveller property for S-acts is a generalization of the fellow traveller property given for semigroups and groups.

Proposition 8.3. *Let S be a semigroup generated by a finite set $X = \{x_1, \ldots, x_n\}$. Then the Cayley graph of S possesses the fellow traveller property with respect to some regular language L if and only if $\Gamma_X(S, S)$ possesses the fellow traveller property with respect to some regular languages $L_1, \ldots, L_n$ and X.*

Proof. ($\Rightarrow$) Assume that the Cayley graph $\Gamma = \Gamma_X(S)$ of S has the fellow traveller property with respect to a language L. Then $L\varphi = S$ and there exists a constant $k \in \mathbb{N}$ such whenever $d_\Gamma(u, v) \leq 1$ with $u, v \in L$ then $d_\Gamma(u(t), v(t)) \leq k$ for all $t \geq 1$. Clearly, $A = \bigcup_{i=1}^{n} a_i.L_i$ for the languages $L_i = \{u \in X^+ \mid x_i u \in L\}$ $(1 \leq i \leq n)$. Let $\Delta = \Gamma_X(S, S)$. Choose languages L_i, L_j and let $(u, v) \in L_i \times L_j$ such that $d_\Delta(x_i \cdot u, x_j \cdot v) \leq 1$. Bearing in mind that $d_\Delta(a, b) = d_\Gamma(a, b)$ for all $a, b \in S$, (see Example 8.1) we have that $d_\Gamma(x_i u, x_j v) \leq 1$, and hence $d_\Gamma((x_i u)(t), (x_j v)(t)) \leq k$ holds for all $t \geq 1$. In particular we have that $d_\Delta(x_i \cdot (u(t)), x_j \cdot (v(t))) \leq k$ for all $t \geq 1$, proving that the fellow traveller property holds in Δ with respect to L_i, L_j and x_i, x_j. Since L_i, L_j were arbitrarily chosen, we may deduce that $\Gamma_X(S, S)$ possesses the fellow traveller property.

($\Leftarrow$) Assume that $\Delta = \Gamma_X(S, S)$ possesses the fellow traveller property with respect to some regular languages $L_1, \ldots, L_n$ and X. Then $\bigcup_{j=1}^{n} x_j L_j \varphi = S$ and for any two chosen languages L_i, L_j and corresponding generators x_i, x_j, there exists a constant $k \in \mathbb{N}$ such that whenever $d_\Delta(x_i \cdot u, x_j \cdot v) \leq 1$ with $(u, v) \in L_i \times L_j$ then $d_\Delta(x_i \cdot (u(t)), x_j \cdot (v(t))) \leq k$ for all $t \geq 1$. We let $L = \bigcup_{j=1}^{n} x_j L_j$. Then $L\varphi = S$. We claim that the Cayley graph $\Gamma = \Gamma_X(S)$ has the fellow traveller property with respect to L. Let $N \in \mathbb{N}$ be a constant such that for any two generators x_i and x_j that are connected via a path in Δ the distance $d_\Delta(x_i, x_j) \leq N$. Let $M = \max(k, N)$. Assume that $d_\Gamma(u, v) \leq 1$, $(u, v \in L)$. Then $u = x_i \cdot \tilde{u}$ and $v = x_j \cdot \tilde{v}$, where $x_i, x_j \in X$ and $(\tilde{u}, \tilde{v}) \in L_i \times L_j$. It follows that $d_\Delta(x_i \cdot \tilde{u}, x_j \cdot \tilde{v}) \leq 1$ holds and we obtain that for all $t \geq 1$, $d_\Delta(x_i \cdot (\tilde{u}(t)), x_j \cdot (\tilde{v}(t))) \leq k \leq M$ holds. Since $d_\Delta(a, b) = d_\Gamma(a, b)$ for all $a, b \in S$, we have that for all $t \geq 1$, $d_\Gamma(x_i \cdot (\tilde{u}(t)), x_j \cdot (\tilde{v}(t))) \leq k \leq M$. To finish the proof we need to verify that $d_\Gamma(x_i, x_j) \leq M$ or equivalently that x_i and x_j are connected in $\Gamma = \Delta$. But the latter fact follows by our assumptions, since $x_i \cdot (\tilde{u} \cdot x) = x_j \cdot \tilde{v}$ holds for some $x \in X \cup \{\epsilon\}$. That is, we have the following path in Δ

$$x_i \xrightarrow{\quad \tilde{u} \quad} x_i \cdot \tilde{u} \xrightarrow{x} x_i \cdot \tilde{u} \cdot x = x_j \cdot \tilde{v} \xleftarrow{\quad \tilde{v} \quad} x_j$$

We may deduce that the Cayley graph of S possesses the fellow traveller property with respect to L. $\square$

Next we verify that if A is an automatic S-act, then $\Gamma_X(A,S)$ possesses fellow traveller property. We follow the group and semigroup theoretical proofs.

Proposition 8.4. *Let S be a semigroup generated by a finite set X. Let A be an automatic S-act with an automatic structure $(A_g, X, L_1, \ldots, L_n)$. Then $\Gamma_X(A,S)$ has the fellow traveller property with respect to $L_1, \ldots, L_n$ and A_g.*

Proof. Let $A_g = \{a_1, \ldots, a_n\}$. For each regular language $L_{(i,j)x}$, $x \in X \cup \{=\}$, consider a finite state automaton $\mathcal{A}_{(i,j)x}$ accepting it and choose a constant $N \in \mathbb{N}$, such that N is greater then the number of states of any of the automata defined. Let $\Delta = \Gamma_X(A,S)$.

Choose regular languages L_i, L_j and assume that $d_\Delta(a_i.u, a_j.v) \leq 1$ holds, where $(u,v) \in L_i \times L_j$. Without loss of generality we can assume that $a_i.(u \cdot y) = a_j.v$ for some $y \in X \cup \{\epsilon\}$. Then $(u,v)\delta_X$ is accepted by the automaton $\mathcal{A}_{(i,j)x}$, where $x = y$ if $y \in X$ and x is the symbol $=$ if $y = \epsilon$. Start reading the word $(u,v)\delta_X$ in $\mathcal{A}_{(i,j)x}$ and assume that after reading the first t letters $(u(t), v(t))\delta_X$, we are at state q. Let $(\tilde{u}, \tilde{v})\delta_X$ be the shortest word over $X(2,\$)$ such that reading $(\tilde{u}, \tilde{v})\delta_X$ from state q, we arrive at a final state of $\mathcal{A}_{(i,j)x}$. Clearly, the length of $(\tilde{u}, \tilde{v})\delta_X$ is less then N, since the number of states of the considered automaton is less then N. Furthermore, since $(u(t), v(t))\delta_X(\tilde{u}, \tilde{v})\delta_X$ is accepted by $\mathcal{A}_{(i,j)x}$, we have the following diagram in Δ: That is,

$$d_\Delta(a_i.u(t), a_j.v(t)) \leq |\tilde{u}| + |\tilde{v}| + |x| \leq 2N + 1,$$

which proves that the fellow traveller property holds in Δ with respect to L_i, L_j and a_i, a_j. Since L_i, L_j were arbitrarily chosen, we may deduce that Δ possesses the fellow traveller property with respect to $L_1, \ldots, L_n$ and A_g. $\square$

Corollary 8.5. *Let S be an automatic semigroup. Then the Cayley graph of S possesses the fellow traveller property.*

9. Conclusions and future research directions

In this paper, we gave a general framework for the theory of automaticity by introducing the notion of an automatic S-act. We proved that this notion is indeed a generalization of the semigroup theoretic notion and that basic properties also generalize. If the S-act is S itself, then we arrive at the usual notion of automaticity. If we choose the S-act to be a set closely related to S, then we arrive at a wide variety of notions of automaticity. Indeed, we may choose the S-act A to be an $\mathcal{R}$-class of S or $S/\mathcal{R}$ or $S/\mathcal{L}$ to enable us to define Schützenberger-automaticity, $\mathcal{R}$-class automaticity or $\mathcal{L}$-class automaticity and we may combine these definitions. This leads to a plethora of interesting questions. For example how different notions of automaticity relate to each other? Under what conditions (if any) does automaticity of a semigroup imply Schützenberger automaticity of an $\mathcal{R}$-class and vica versa. If S is a regular semigroup, then how does Schützenberger automaticity of an

$\mathcal{R}$-class relate to automaticity of a maximal subgroup within that $\mathcal{R}$-class. Is it possible to give a geometric characterization of Schützenberger automatic regular semigroups in terms of the fellow traveller property? Under what conditions does Schützenberger automaticity imply finite presentability? What can one say about the word problem of a Schützenberger automatic semigroup? We propose these questions for further investigation to gain more insight into the nature and properties of semigroups S given that the chosen S-act is automatic.

10. Acknowledgement

The authors acknowledge the support of EPSRC grant no. EP/I032312/1. Research also partially supported by the Hungarian Scientific Research Fund (OTKA) grant no. K83219.

[1] A. J. Cain, Automatic semigroups and Bruck-Reilly extensions, *Acta Math. Hungar.* **47** (2010) 1–15.

[2] A. J. Cain, Automatic Clifford semigroups, preprint.

[3] A. J. Cain, Automatic structures for subsemigroups of Baumslag—Solitar semigroups, *Semigroup Forum* **87** (2013) 537–552.

[4] I. Andrade, L.Desçalco, M. A. Martins, Automatic structures for semigroup constructions, *Semigroup Forum* **76** (2008), 239–255.

[5] C. M. Campbell, E. F. Robertson, N. Ruskuc, R. M. Thomas, Automatic semigroups, *Theoret. Comput. Sci.* **365** (2001), 365 – 391.

[6] C. M. Campbell, E. F. Robertson, N. Ruškuc and R. M. Thomas, Automatic completely-simple semigroups, *Acta Math. Hungar.* **95** (2002) 201–215.

[7] A.Cutting, A.Solomon, Remarks concerning finitely generated semigroups having regular sets of unique normal forms, *J.Aust.Math.Soc.* **70** (2001) 293–309.

[8] L. Descalço and N. Ruškuc, On automatic Rees matrix semigroups, *Comm. Algebra* **30** (2002) 1207–1226.

[9] E. R. Dombi, A note on automatic semigroups *Semigroup Forum* **86** (2013) 555–570.

[10] E. R. Dombi, Automatic S-acts and inverse semigroup presentations, PhD Thesis.

[11] A. J. Duncan, E. F. Robertson and N. Ruškuc, Automatic monoids and change of generators, *Math. Proc. Cambridge Philos. Soc.* **127** (1999) 403–409.

[12] D. B. A. Epstein, J. W. Cannon, D. F. Holt, S. V. F. Levy, M. S. Paterson, W. P. Thurston: Word Processing in Groups, Boston, MA: Jones and Bartlett Publishers, 1992.

[13] M. Hoffmann and R. M. Thomas, Notions of automaticity in semigroups, *Semigroup Forum* **66** (2003) 337–367.

[14] J. E. Hopcroft, J. D. Ullman, Formal languages and their relation to automata, Addison-Wesley, 1969.

[15] J.M. Howie, Fundamentals of semigroup theory, Clarendon Press, Oxford, 1995.

[16] M. Kambites, F. Otto, Uniform decision problems for automatic semigroups, *Journal of Algebra* **303** (2006) 789 – 809.

[17] M. V. Lawson, Finite automata, Chapman & Hall/CRC, 2004.

[18] W. D. Munn, Free inverse semigroups, *Proc. London Math. Soc.* (3) **29** (1974), 385 – 404.

[19] P. V. Silva and B. Steinberg, A geometric characterization of automatic monoids, *Q. J. Math.* **55** (2004) 333–356.

[20] P. V. Silva, B. Steinberg, Extensions and submonoids of automatic monoids, *Theoret. Comput. Sci.* **289** (2002) 727–754.