Scattered deletion and commutativity

Alexandru Mateescu*

Faculty of Mathematics, University of Bucharest, Academiei 14, sector 1, Bucharest, Romania

Communicated by A. Salomaa
Received November 1992
Revised March 1993

Abstract


This paper deals with the scattered deletion of a language by another language, i.e. with scattered residuals. We introduce the notion of the scattered syntactical monoid. The main result is a Myhill–Nerode-like theorem for languages $L$ with the property that the commutative closure of $L, \text{com}(L)$, is a regular language. We investigate some properties of these families of languages related to the cardinality of the scattered syntactical monoid.

1. Preliminaries

Operations of inserting and/or deleting strings between two languages are very important in theoretical computer science and, also in practice. We mention here only some few examples: top-down parsing (insertion), bottom-up parsing (deletion). Some recent papers in this area are [3, 4]. For the basic notions in formal language theory, the reader is referred to the monographs [1, 8, 12].

The residual or the scattered sequential deletion, of two languages $L_1, L_2$, introduced in [3] is defined by the following definition.
Definition 1.1. Let $L_1$ and $L_2$ be languages over $\Sigma$. The scattered residual or the scattered sequential deletion of $L_1$ by $L_2$ is defined as

$$L_1 \rightarrow_s L_2 = \bigcup_{u \in L_1, v \in L_2} (u \rightarrow_s v),$$

where

$$u \rightarrow_s v = \{u_1 u_2 \ldots u_k+1 \in \Sigma^* | k \geq 1, u = u_1 v_1 u_2 v_2 \ldots u_k v_k u_{k+1}, v = v_1 v_2 \ldots v_k, u_i \in \Sigma^*, 1 \leq i \leq k+1, v_i \in \Sigma^*, 1 \leq i \leq k\}.$$

Definition 1.2. The shuffle operation between words, denoted $[,]$, is defined recursively by

$$(au[,]bv) = a(u[,]bv) \cup b(au[,]v),$$

and

$$(u[,]\lambda) = (\lambda[,]u) = \{u\},$$

where $u, v \in \Sigma^*$ and $a, b \in \Sigma$.

The shuffle operation is extended in a natural way to languages.

Definition 1.3. The shuffle of two languages $L_1$ and $L_2$ is

$$L_1[,]L_2 = \bigcup_{u \in L_1, v \in L_2} (u[,]v).$$

The proof of the following lemma is obvious and thus omitted.

Lemma 1.4. For any languages $L_1$ and $L_2$

$$L_1 \rightarrow_s L_2 = \{w \in \Sigma^* | \exists v \in L_2, (w[,]v) \cap L_1 \neq \emptyset\}.$$

2. Some algebraic properties

Remark 2.1. The shuffle operation is a commutative and associative operation on $\mathcal{P}(\Sigma^*)$ and, moreover $\{\lambda\}$ is the unit element. Therefore, $\mathcal{S}=(\mathcal{P}(\Sigma^*), [,], \{\lambda\})$ is a commutative monoid, see also [8].

Notation. If $x \in \Sigma^*$, then $|x|$ is the length of $x$. If $a \in \Sigma$, then $|x|_a$ is the number of occurrences of the letter $a$ in $x$.

Definition 2.2. If $x \in \Sigma^*$, then the commutative closure of $x$ is

$$com(x) = \{w \in \Sigma^* | \text{for any } a \in \Sigma, |w|_a = |x|_a\}.$$
If $L$ is a language, then the commutative closure of $L$ is
\[ \text{com}(L) = \bigcup_{x \in L} \text{com}(x). \]

The following proposition enumerates some elementary properties of shuffle and commutative closure operations. We will omit the proof of Proposition 2.3, but most of these properties will be used in sequel. For many other important results concerning commutativity and shuffle operation see [5, 6, 10, 13].

**Proposition 2.3.** Let $L, L_1$ and $L_2$ be languages.

1. $L \subseteq \text{com}(L)$.
2. If $L_1 \subseteq L_2$, then $\text{com}(L_1) \subseteq \text{com}(L_2)$.
3. $\text{com}(\text{com}(L)) = \text{com}(L)$.
4. $\text{com}(L_1 \cup L_2) = \text{com}(L_1) \cup \text{com}(L_2)$.
5. $\text{com}(L_1 \cup L_2) = \text{com}([L_1 \cup L_2])$.
6. $\text{com}(L_1 \cup L_2) = \text{com}(L_1) \cup \text{com}(L_2)$.
7. $\text{com}(L_1 \cup L_2) = \text{com}(L_1) \cup \text{com}(L_2)$.
8. $\text{com}(L) = \text{com}(\text{mi}(L))$, where $\text{mi}(L)$ is the mirror image of $L$.

**Definition 2.4.** Let $L$ be a language, $L \subseteq \Sigma^*$. The relation induced by $L$ in $\mathcal{S}$, denoted $\sim_L$, is defined as follows: for any $L_1, L_2 \subseteq \Sigma^*$,
\[ L_1 \sim_L L_2 \text{ if and only if } \text{com}(L) \rightarrow_{L} L_1 \Rightarrow \text{com}(L) \rightarrow_{L} L_2. \]

**Lemma 2.5.** For any language $L, L \subseteq \Sigma^*$, the relation $\sim_L$ is a congruence on the monoid $\mathcal{S}$.

**Proof.** Obviously, $\sim_L$ is an equivalence relation on $\mathcal{S}$. Assume that $I_1 \sim_L I_2$ and let $A$ be an arbitrary language over $\Sigma$. We have to prove that $(A \mid \mid L_1) \sim_L (A \mid \mid L_2)$, or equivalently, that $\text{com}(L) \rightarrow_{I} (A \mid \mid L_1) = \text{com}(L) \rightarrow_{I} (A \mid \mid L_2)$. Let $u$ be in $\text{com}(L) \rightarrow_{I} (A \mid \mid L_1)$. From Lemma 1.4 it follows that there exists a $t \in A \mid \mid L_1$, such that $(u \mid \mid t) \cap \text{com}(L) \neq \emptyset$. Hence, there are $x \in A$ and $y \in L_1$ such that $t \in x \mid \mid y$ and, moreover, $(u \mid \mid t) \cap \text{com}(L) \neq \emptyset$. Therefore, $(u \mid \mid x \mid \mid y) \cap \text{com}(L) \neq \emptyset$. Consequently, $(u \mid \mid x \mid \mid y) \cap \text{com}(L) \neq \emptyset$.

Thus, $(u \mid \mid x) \subseteq (\text{com}(L) \rightarrow_{I} I_1)$. But, by our assumption $\text{com}(L) \rightarrow_{I} I_1 = \text{com}(L) \rightarrow_{I} I_2$. Hence, we can deduce that $(u \mid \mid x) \subseteq (\text{com}(L) \rightarrow_{I} I_2)$. Therefore, there exists a $y' \in L_2$ such that $(u \mid \mid y') \cap \text{com}(L) \neq \emptyset$. Note that $(x \mid \mid y') \subseteq (A \mid \mid I_2)$. Thus, there is a $t' \in A \mid \mid L_2$, such that $(u \mid \mid t') \cap \text{com}(L) \neq \emptyset$. But, this means that $u \in \text{com}(L) \rightarrow_{I} A \mid \mid L_2$, and therefore $\text{com}(L) \rightarrow_{I} A \mid \mid L_2 \subseteq \text{com}(L) \rightarrow_{I} A \mid \mid L_2$. The converse inclusion is similar. Hence, the relation $\sim_L$ is invariant on the right with respect to the operation of $\mathcal{S}$. But $\mathcal{S}$ is commutative and, thus, $\sim_L$ is a congruence on the monoid $\mathcal{S}$. □

**Corollary 2.6.** For any language $L, L \subseteq \Sigma^*$, $\mathcal{S}_L = \mathcal{S} / \sim_L$ is a commutative submonoid of $\mathcal{S}$.
Definition 2.7. If \( L \) is a language, then the monoid \( \mathcal{S}_L \) is the scattered syntactic monoid. (It can be called also the shuffle or the commutative syntactic monoid defined by \( L \).)

Comment. The remainder of this paper is dedicated to the study of different properties of these monoids. The main result is that the family of languages that have finite scattered syntactic monoids is exactly the family of languages with the property that their commutative closure is a regular language. This result resembles the famous theorem of Myhill–Nerode for the classical syntactic monoids.

For some other extensions of the Myhill–Nerode theorem, see [7, 11, 14].

Definition 2.8. A language \( L \) over \( \Sigma \) is commutatively saturated by an equivalence relation \( \sim \) on \( \mathcal{P}(\Sigma^*) \) if and only if, for any languages \( L_1, L_2 \), over \( \Sigma \), if \( L_1 \cap \text{com}(L) \neq \emptyset \) and \( L_1 \sim L_2 \), then \( L_2 \cap \text{com}(L) \neq \emptyset \).

Lemma 2.9.

(i) \( \text{com}(L) \) is commutatively saturated by \( \sim_L \).

(ii) \( \sim_L \) is the greatest congruence on \( \mathcal{S} \) with the above property.

Proof. (i) Assume that \( L_1 \sim_L L_2 \) and that \( L_1 \cap \text{com}(L) \neq \emptyset \). There exists \( w \in \text{com}(L) \cap L_1 \) and thus \( \lambda \in \text{com}(L) \to_L L_1 \). Consequently, \( \lambda \in \text{com}(L) \to_L L_2 \). Therefore, there is, \( u \in \text{com}(L) \) and, \( u \in L_2 \) such that \( u \to_L u \neq \lambda \), i.e. \( \text{com}(L) \cap L_2 \neq \emptyset \).

(ii) Let \( \approx \) be a congruence on \( \mathcal{S} \) such that \( L \) is commutatively saturated by \( \approx \). Now, assume that \( L_1 \approx L_2 \) and consider the language \( \text{com}(L) \to_L L_1 \). If \( \text{com}(L) \to_L L_1 = \emptyset \), then also \( \text{com}(L) \to_L L_2 = \emptyset \). (Otherwise, there is \( y \in L_2 \) such that \( x \to_L y \in \text{com}(L) \). Hence, \( \text{com}(L) \cap (L_2 \cap [y]) \neq \emptyset \) and moreover, \( (L_1 \cap [y]) \approx (L_2 \cap [y]) \). It follows that \( \text{com}(L) \cap (L_1 \cap [y]) \neq \emptyset \) and thus there is a word \( z \in \text{com}(L) \) and \( z \in t \cap [y] \), for some \( t \in L_1 \). Hence, \( y \in (z \to_L t) \) and thus \( y \in \text{com}(L) \to_L L_1 \), a contradiction with the assumption that \( \text{com}(L) \to_L L_1 = \emptyset \). Therefore, we can assume that \( \text{com}(L) \to_L L_1 \neq \emptyset \) and let \( x \in \text{com}(L) \to_L L_1 \). Note that, \( \text{com}(L) \cap (L_1 \cap [x]) \neq \emptyset \), \( (L_1 \cap [x]) \approx (L_2 \cap [x]) \) and thus \( \text{com}(L) \cap (L_2 \cap [x]) \neq \emptyset \). Hence, there exists a \( w \in L_2 \) such that \( (w \cap [x]) \cap \text{com}(L) \neq \emptyset \). We can conclude that \( x \in \text{com}(L) \to_L L_2 \). Therefore, \( \text{com}(L) \to_L L_1 \subseteq \text{com}(L) \to_L L_2 \). The converse inclusion is analogous. Thus, \( L_1 \sim_L L_2 \) and it follows that \( \approx \subseteq \sim_L \). □

Comment. The congruence \( \sim_L \) is the syntactic or principal congruence on \( \mathcal{S} \) with respect to the family: \( \mathcal{S}_L = \{ L' \mid L' \cap L \neq \emptyset \} \).

3. The main result

Theorem 3.1. Let \( L \) be a language over \( \Sigma \). The following conditions are equivalent:

(a) \( \text{com}(L) \) is a regular language.

(b) there exists a congruence \( \approx \) on \( \mathcal{S} \), of finite index, such that \( L \) is commutatively saturated with respect to \( \approx \).

(c) the scattered syntactic monoid of \( L, \mathcal{S}_L \), is finite.
Proof. (a)⇒(b). Let \( \mathcal{A} = (Q, \Sigma, \delta, q_0, F) \) be the minimal automaton such that \( L(\mathcal{A}) = \text{com}(L) \). Now consider the finite monoid, \( M = (X^X, \cdot, 1_X) \), where \( X = \mathcal{P}(Q) \) and the operation \( \cdot \) is defined as \( f \cdot g = g \circ f \) (here \( \circ \) is the usual composition of functions). Obviously, \( M \) is a monoid. Let \( \phi \) be the function, \( \phi : \mathcal{A} \to M \), such that, for any \( A \in \mathcal{A}, \phi(K) \in X^X \), that is, \( \phi(K) : X \to X \), i.e. \( \phi(K) : \mathcal{P}(Q) \to \mathcal{P}(Q) \), such that

\[
\phi(K)(Y) = \delta(Y, K).
\]

In order to prove that \( \phi \) is a morphism of monoids, we will show a property of the automaton \( \mathcal{A} \), i.e.

\[(*) \text{ for any } Y \subseteq Q \text{ and for any } K \subseteq \Sigma^*: \delta(Y, K) = \delta(Y, \text{com}(K)).\]

Obviously, \( \delta(Y, K) \subseteq \delta(Y, \text{com}(K)) \). Now, let \( w \) be in \( \text{com}(K) \). There exists a word \( w' \in K \) such that \( w \in \text{com}(w') \). It is easy to see that for any \( x, y \in \Sigma^* \) it is true that

\[xwy \in \text{com}(L) \text{ if and only if } xw' \in \text{com}(L).\]

Hence, \( w \) and \( w' \) are equivalent with respect to the classical syntactic congruence defined by \( \text{com}(L) \). Let \( p \) be in \( Y \) such that \( \delta(p, w) = q \). It follows that \( \delta(p, w') = q \), too, and thus \( q \in \delta(Y, K) \). It results that \( \delta(Y, \text{com}(K)) \subseteq \delta(Y, K) \). Therefore, the above property \( (*) \) does hold. Now, let us prove, using the properties from Proposition 2.3 and the property \( (*) \), that \( \phi \) is a morphism of monoids. For any \( J, K \subseteq \Sigma^* \) and for any \( Y \subseteq Q \):

\[
\phi(J \cdot K)(Y) = \delta(Y, J \cdot K) = \delta(Y, \text{com}(J \cdot K)) = \delta(Y, \text{com}(JK))
\]

\[
= \delta(Y, JK) = \delta(\delta(Y, J), K) = \delta(\phi(J)(Y), K) = \phi(K)(\phi(J)(Y))
\]

\[
= (\phi(K) \circ \phi(J))(Y) = (\phi(J), \phi(K))(Y).
\]

But \( Y \) is arbitrary and thus \( \phi(J \cdot K) = \phi(J) \cdot \phi(K) \), for any \( J, K \subseteq \Sigma^* \). Therefore, \( \phi \) is a morphism of monoids.

Consequently, the kernel of \( \phi \), i.e.

\[
\text{Ker} \phi = \{(A, B) \mid \phi(A) = \phi(B)\}
\]

is a congruence on the monoid \( \mathcal{A} \). Moreover, because \( M \) is finite, \( \text{Ker}(\phi) \) is a congruence of finite index, i.e. has only a finite number of equivalence classes.

Observe that, moreover, \( L \) is commutatively saturated by \( \text{Ker}(\phi) \). Assume that \( L_1, L_2 \) are such that \( \phi(L_1) = \phi(L_2) \) and that \( L_1 \cap \text{com}(L) \neq \emptyset \). Therefore, \( \delta(q_0, L_1) \cap F \neq \emptyset \). But, \( \phi(L_1)(q_0) = \phi(L_2)(q_0) \) and thus \( \delta(q_0, L_1) = \delta(q_0, L_2) \). It follows that \( \delta(q_0, L_2) \cap F \neq \emptyset \) and hence, \( L_2 \cap \text{com}(L) \neq \emptyset \).

(b)⇒(c) By Lemma 2.9(ii), it results that \( \text{Ker}(\phi) \subseteq \sim_L \) and therefore, \( \text{card}(\mathcal{A}/\text{Ker} \phi) \geq \text{card}(\mathcal{A}/\sim_L) \). But, \( \text{card}(\mathcal{A}/\text{Ker} \phi) \) is finite and hence, \( \mathcal{A}_L \), is a finite monoid.
(c)⇒(a) Define the finite deterministic automaton \( \mathcal{A} = (Q, \Sigma, \delta, q_0, F) \), where \( Q = \mathcal{P}(\Sigma^*)/\sim \). Note that \( Q \) is a finite set. The equivalence class of a language \( K \) will be denoted by \([K]\). Moreover, \( q_0 = [\{\lambda\}] \), \( F = \{[K] \mid K \cap \text{com}(L) \neq \emptyset \} \) and the transition function, \( \delta : Q \times \Sigma \rightarrow Q \) is

\[
\delta([B], \sigma) = [B \sigma].
\]

Note that \( \delta \) is well defined, i.e. if \( B \approx B' \), then \( B \sigma \approx B' \sigma \). In order to prove that \( L(\mathcal{A}) = \text{com}(L) \), assume first that \( w \in L(\mathcal{A}) \). Therefore, \( \delta([\lambda], w) \in F \). If \( w = w_1 w_2 \ldots w_n \), where \( w_i \in \Sigma, i = 1, \ldots, n \), then it is easy to see that \( \delta([\lambda], w) = [w_1 \sigma_1 w_2 \sigma_2 \ldots \sigma_n w_n] \).

Hence, there is a language \( K \), such that \( K \cap \text{com}(L) \neq \emptyset \) and \( K \approx (w_1 \sigma_1 \ldots \sigma_n w_n) \). But \( L \) is commutatively saturated by \( \approx \) and hence \( (w_1 \sigma_1 \ldots \sigma_n w_n) \cap \text{com}(L) \neq \emptyset \). Therefore, there is a permutation of \( w \) in \( L \). It follows that \( w \in \text{com}(L) \). Conversely, assume that \( w \in \text{com}(L) \). Hence, there exists a \( u \in L \) such that \( w \\in \text{com}(u) \). Now, if \( w = w_1 w_2 \ldots w_n \), \( w_i \in \Sigma, i = 1, \ldots, n \), then \( \delta([\lambda], w) = [w_1 \sigma_1 w_2 \sigma_2 \ldots \sigma_n w_n] \). Moreover, \( u \\in (w_1 \sigma_1 \ldots \sigma_n w_n) \) and thus \( (w_1 \sigma_1 \ldots \sigma_n w_n) \cap \text{com}(L) \neq \emptyset \). Consequently, \( [w_1 \sigma_1 w_2 \sigma_2 \ldots \sigma_n w_n] \in F \) and therefore \( w \in L(\mathcal{A}) \). It follows that \( \text{com}(L) = L(\mathcal{A}) \).

**Corollary 3.2.** A language \( \text{com}(L) \) has a finite number of scattered residuals if and only if \( \text{com}(L) \) is a regular language.

4. Related problems

**Remark 4.1.** Because \( \text{card}(\mathcal{S}) = \aleph_1 \), a scattered syntactic monoid \( \mathcal{S}_L \) may have a finite cardinal, a \( \aleph_0 \) cardinal or even an \( \aleph_1 \) cardinal. Theorem 3.1 provides a characterization of languages with a finite scattered monoid.

**Definition 4.2.** The family of languages with a finite scattered syntactic monoid will be denoted by \( \mathcal{F} \), i.e.

\[
\mathcal{F} = \{L \mid \text{\mathcal{S}_L is a finite monoid}\}.
\]

Analogously,

\[
\mathcal{F}_L = \{L \mid \text{\text{card} } (\mathcal{S}_L) = \aleph_0\},
\]

\[
\mathcal{F}_L = \{L \mid \text{\text{card} } (\mathcal{S}_L) = \aleph_1\}.
\]

**Corollary 4.3.** Any finite language \( L \) is in \( \mathcal{F} \).

**Proof.** Obviously, \( \text{com}(L) \) is finite and hence regular. □
Definition 4.4. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a minimal deterministic automaton. The monoid $M(A) = (X, \circ, l_X)$, where $X = \mathcal{P}(Q)$ is the monoid of hyperactions.

Definition 4.5. Let $A$ be the above automaton. Any subset $K$ of $\Sigma^*$ defines in a natural way a function from $X$ to $X$, $K(Y) = \delta(Y, K)$, for any $Y$ subset of $Q$. The submonoid of $M(A)$, generated by these functions, is the monoid of hypertransitions, denoted $\mathcal{F}(A)$.

Theorem 4.6. Let $A$ be a minimal deterministic automaton. The monoid $\mathcal{F}_L$ is isomorphic with a submonoid of $\mathcal{F}(A)$, where $L = L(A) = \text{com}(L)$.

Proof. Define the function $\phi$: 

$$\phi: \mathcal{F}(A) \rightarrow \mathcal{F}_L, \quad \phi(K) = [K].$$

It is easy to prove that $\phi$ is a surjective homomorphism. \qed

Remark 4.7. Theorem 4.6 provides a method to compute the scattered syntactic monoid. See the following example.

Example 4.8. Let us compute the scattered syntactic monoid for the language $L = \{a^n b a^n | n \geq 0\} \cup \{a^n b a^n | n \geq 0\}$. It is easy to observe that $\text{com}(L) = a^* b a^*$, and hence $\text{com}(L)$ is a regular language. Now, consider the minimal automaton $A = (Q, \{a, b\}, \delta, q_0, \{q_1\})$ that accepts the language $\text{com}(L)$. Thus, $Q = \{q_0, q_1, q_2\}$ and $\delta(p, a) = p$, for any $p \in Q$, $\delta(q_0, b) = q_1$ and in any other case, $\delta(r, b) = q_2$. By a well-known algorithm, we can compute the classical syntactic monoid of $\text{com}(L)$. We obtain the relations: $a \equiv \lambda$ and $b^2 \equiv b^3$ and the elements are: $\lambda, b, b^2$. Moreover, $b^2$ is a zero of the syntactic monoid and will be identified with $0$ in the scattered syntactic monoid. Now, using the above theorem we have to consider only the following subsets of $\mathcal{P}(\Sigma^*)$: $\emptyset, \lambda, u = \{b\}, v = \{\lambda, b\}$. The table of hypertransitions (Table 1) for $\emptyset, \lambda, u, v$ is

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>$\emptyset$</th>
<th>$\lambda$</th>
<th>$u$</th>
<th>$v$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$\emptyset$</td>
<td>$q_0$</td>
<td>$q_1$</td>
<td>${q_0, q_1}$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$\emptyset$</td>
<td>$q_1$</td>
<td>$q_2$</td>
<td>${q_1, q_2}$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$\emptyset$</td>
<td>$q_2$</td>
<td>$q_2$</td>
<td>$q_2$</td>
</tr>
</tbody>
</table>

Using the above method and the table of hypertransitions, we obtain that the scattered syntactic monoid consists of $\emptyset$ that is a zero of the monoid, $\{\lambda\}$ which is the unit element of the monoid and the following two elements: $u, v$. Moreover, the multiplication table (Table 2) of the above elements is

Table 1

Table 2
Therefore, \( \mathcal{F}_L \) has four elements, \( \mathcal{F}_L = \{ \emptyset, \lambda, u, v \} \). Note that the above monoid is indeed commutative and moreover, \( \text{com}(L) \) has only four scattered residuals: \( \emptyset, \text{com}(L), a^* \) and \( \text{com}(L) \cup a^* \), corresponding to: \( \emptyset, \lambda, u = \{ b \} \) and, respectively, to \( v = \{ \lambda, b \} \).

**Remark 4.9.** The language from the above example is a context-free language that is not regular. Hence, the classical syntactic monoid for this language is infinite and cannot be "computed".

**Comment.** The language \( L \), from the above example, provides us an example of a language with an infinite number of scattered residuals whereas \( \text{com}(L) \) has only a finite number of scattered residuals. Indeed, if \( i < j \), then \( L \rightarrow_s \{ ba^i \} \) contains the word \( a^i \) but, \( L \rightarrow_s \{ ba^j \} \) does not contain this word \( a^i \). Therefore, the infinite sequence \( \{ ba^i \}_{i \geq 0} \) leads to an infinite number of scattered residuals. It follows that, in this respect, our approach is the best possible.

**Theorem 4.10.** The family \( \mathcal{F}_\text{in} \) is not comparable with any family in the Chomsky hierarchy.

**Proof.** Note that the language \( L = (ab)^* \) is in every Chomsky family, but \( \text{com}(L) = \{ w \mid w_a = |w_b| \} \), is not regular. Thus, \( L \) is not in \( \mathcal{F}_\text{in} \).

Assume that, \( K, K \subseteq a^*b^* \) is any, (arbitrary !) language and consider \( K' = K \cup b^*a^* \). Then \( \text{com}(K') = \{ a, b \}^* \), which is a regular language. Therefore, \( K' \) is in \( \mathcal{F}_\text{in} \). Observe that \( K \), and thus \( K' \), could be very complex languages, even not recursively enumerable, and still \( K' \) is in \( \mathcal{F}_\text{in} \). \( \Box \)

**Comment.** Note the fact that there are languages \( L \) that are not recursively enumerable, but the scattered syntactic monoid of \( L \) is finite.

Next theorem is useful for proving a decidability result.

**Theorem 4.11** (Ginsburg and Spanier [2]). It is decidable to determine for an arbitrary regular language \( R \) whether \( \text{com}(R) \) is regular.

**Theorem 4.12.** For any context-free language \( L \), it is decidable whether or not \( L \) is in \( \mathcal{F}_\text{in} \).
Proof. Let $L$ be a context-free language. There exists a regular language $R$ letter equivalent with $L$, and $R$ can be effectively determined (see [1] or [12]). Obviously, $\text{com}(L) = \text{com}(R)$ and therefore $\mathcal{S}_L = \mathcal{S}_R$. Now, from Theorem 4.11, it is decidable whether or not $R$ has the property that $\text{com}(R)$ is regular. If $\text{com}(R)$ is regular, then $L$ is in $\text{Fin}$, otherwise $L$ is not in $\text{Fin}$. \hfill $\square$

**Theorem 4.13.** The family $\text{Fin}$ is closed under the following operations: shuffle, catenation, union, mirror, com, cycle, where, cycle($L$) = $\{uv | u \in L, v \in L\}$.

**Proof.** The closure of $\text{Fin}$ under union, mirror, com and cycle is trivial.

To prove the closure under shuffle, consider $L_1, L_2 \in \text{Fin}$. Then, from Proposition 2.3 (the identity 6), it follows that $L_1 \cup L_2$ is in $\text{Fin}$. The closure under catenation follows from the closure under shuffle easily by Proposition 2.3 (the identity 4). \hfill $\square$

**Theorem 4.14.** The family $\text{Fin}$ is not closed under complement, intersection, intersection with regular languages, Kleene closure, homomorphisms and inverse homomorphisms.

**Proof.** For complement, let $L$ be the language $L = \{a, b\}^* - (ab)^*$. Obviously, $L$ is in $\text{Fin}$, but the complement of $L$ is not in $\text{Fin}$. For intersection, assume that $L_1 = \{a^n b^n | n \geq 0\} \cup b a^* a^*$ and $L_2 = a^* b^*$. Note that both $L_1$ and $L_2$ are in $\text{Fin}$ but their intersection is $\{a^* b^* | n \geq 0\}$ which is not in $\text{Fin}$. To prove the nonclosure of intersection with regular languages, consider $L_3 = \{a, b\}^*$ and $L_4 = (ab)^*$ and note that the intersection of $L_3$ with $L_4$ is $L_4$, which is not in $\text{Fin}$. Let us consider Kleene closure. Assume that $L_5 = \{ab\}$ and observe that $L_5$ is in $\text{Fin}$, but $L_5^* = L_4$ and hence is not in $\text{Fin}$. Now, define the homomorphism $h$, $h(a) = ab$, $h(b) = b$, and the language $L_5 = a^* b^*$. Note that $h(L_5) = (ab)^* b^*$ which is not in $\text{Fin}$. For the inverse homomorphism consider the same homomorphism as above, and the same languages. Observe that $h^{-1}(L_6)$ is not in $\text{Fin}$. \hfill $\square$

**Comment.** The nonclosure of $\text{Fin}$ under complement is surprising. For the classical syntactic monoids it is likely that for any language, of any complexity, $L$ and complement of $L$ have exactly the same syntactic monoid.

**Theorem 4.15.** $\mathcal{S} \neq \emptyset$.

**Proof.** Let $\Sigma$ be $\{a, b\}$ and $L = \{w | w|_a = |w|_b\}$. We will prove that $\text{card}(\mathcal{S}_L) = \aleph_1$. Define $f: \mathcal{P}(a^*) \rightarrow \mathcal{S}_L$ as follows: for any $A \subseteq a^*$, $f(A) = [A]$. Now, note that $f$ is injective. Let $A_1, A_2$ be subsets of $a^*$ such that $A_1 \neq A_2$. There exists a word $a^k$ in $A_1$ such that $a^k$ is not in $A_2$. Thus, $\text{com}(L) \rightarrow A_1$ contains the word $b^k$, but $\text{com}(L) \rightarrow A_2$ does not contain $b^k$. Thus, $[A_1] \neq [A_2]$ and therefore $f$ is injective. But, $\text{card}(\mathcal{P}(a^*))$ is $\aleph_1$ and hence $\text{card}(\mathcal{S}_L)$ is $\aleph_1$. \hfill $\square$

**Theorem 4.16.** $\mathcal{F}_0 \neq \emptyset$.
Proof. We will consider the language:

\[ L = \{ w | w|_a \neq |w|_b, w \in \Sigma^* \}, \]

where \( \Sigma = \{ a, b \} \). Now, let \( A \) be a language over \( \Sigma \). There are only some possibilities to produce a scattered residual. If any word \( w \) in \( A \) has the property that \(|w|_a = |w|_b\), then the residual is the language \( L \). If \( A \) contains only one word \( w \) with \(||w|_a - |w|_b| = j\), where \( j \neq 0 \), then the residual is \( R_j = \Sigma^* \setminus \{ u | |u|_a - |u|_b| = kj, k \geq 0 \} \). In any other case the residual is \( \Sigma^* \) or \( \emptyset \). Therefore, there is only a denumerable number of scattered residuals. □

5. Conclusions and open problems

We have proved a Myhill–Nerode-like theorem for all languages that have a finite scattered syntactic monoid. Moreover, we obtained a theorem of representation of all regular languages that are in the class \( \mathcal{F}in \). We noticed that the class of languages that have the scattered syntactic monoid finite is not comparable with any class from the Chomsky hierarchy and, hence, there are languages in \( \mathcal{F}in \) that are not even recursive enumerable languages. On the other hand, there remains a large number of problems for further research in this topic. Can we characterize in a similar way the languages from \( \mathcal{F}_0 \) and/or \( \mathcal{F}_f \)? We do not know yet results concerning decidable properties of languages in these classes. Is it possible to obtain theorems like the Myhill–Nerode theorem for other deletion operations considered in [3] (sequential, parallel, controlled, iterated, etc.).

Acknowledgments

The author would like to express his gratitude to Juhani Karhumaki and to Tero Harju for many fruitful discussions related to this paper.

Special thanks are due to Michel Latteux for his kind help in finding some useful references.

References