The Commutative Closure of Shuffle Languages over Group Languages is Regular

Stefan Hoffmann, Universität Trier, Germany

July, 2021

International Conference on Implementation and Application of Automata (CIAA) 2021,
July 19 – July 22, 2021 (virtuell)
Organizers: Sebastian Maneth, Peter Leupold, Kathryn Lorenz, Martin Vu

Let me Introduce, The Three Main Protagonists:

The Class of Group Languages $\mathcal{G}$.

Languages recognized by permutation automata, i.e, where every letter permutes the states.

The Shuffle Product and the Iterated Shuffle.

All ways to interleave the words of two languages, and to interleave the words of a single language n-times for all $n \geq 0$.

The Commutative Closure.

Closure of a language L under permuting the letters of the words in L.

Let me Introduce, The Three Main Protagonists:

The Class of Group Languages $\mathcal{G}$.

Languages recognized by permutation automata, i.e, where every letter permutes the states.

The Shuffle Product and the Iterated Shuffle.

All ways to interleave the words of two languages, and to interleave the words of a single language n-times for all $n \geq 0$.

The Commutative Closure.

Closure of a language L under permuting the letters of the words in L.

Let me Introduce, The Three Main Protagonists:

The Class of Group Languages $\mathcal{G}$.

Languages recognized by permutation automata, i.e, where every letter permutes the states.

The Shuffle Product and the Iterated Shuffle.

All ways to interleave the words of two languages, and to interleave the words of a single language n-times for all $n \geq 0$.

The Commutative Closure.

Closure of a language L under permuting the letters of the words in L.

Let me Introduce, The Three Main Protagonists:

The Class of Group Languages $\mathcal{G}$.

Languages recognized by permutation automata, i.e, where every letter permutes the states.

The Shuffle Product and the Iterated Shuffle.

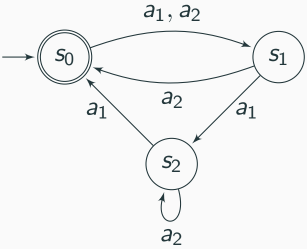
All ways to interleave the words of two languages, and to interleave the words of a single language n-times for all $n \geq 0$.

The Commutative Closure.

Closure of a language L under permuting the letters of the words in L.

An Example of a Permutation Automaton

Both letters partition the state set into cycles.



$$Q = \{s_0, s_1, s_2\}$$

Shuffle Operation

Definition (Shuffle operation)
The shuffle operation, denoted by $\sqcup\!\sqcup$, is defined by

$$u \sqcup\!\sqcup v := \left\{ x_1 y_1 x_2 y_2 \cdots x_n y_n \;\middle|\; \begin{array}{l} u = x_1 x_2 \cdots x_n, v = y_1 y_2 \cdots y_n, \\ x_i, y_i \in \Sigma^*, 1 \leq i \leq n, n \geq 1 \end{array} \right\},$$

for $u, v \in \Sigma^*$ and $L_1 \sqcup\!\sqcup L_2 := \bigcup_{x \in L_1, y \in L_2} (x \sqcup\!\sqcup y)$ for $L_1, L_2 \subseteq \Sigma^*$.

Example
$\{ab\} \sqcup\!\sqcup \{cd\} = \{abcd, acbd, acdb, cadb, cdab, cabd\}$

The Commutative Closure

The commutative closure, mapping languages L into commutative languages, is denoted by $\mathrm{perm}(L)$.

Example: $\mathrm{perm}(\{abc\}) = \{abc, bac, acb, cba, bca, cab\}$.

The Parikh map $\psi : \Sigma^* \to \mathbb{N}_0^{|\Sigma|}$ is

$$\psi(w) = (|w|_{a_1}, \ldots, |w|_{a_k}),$$

where $|w|_{a_i}$ denotes the number of occurrences of the letter a_i in w.

A Prequel, Featuring Two Protagonists:

A. Gomez, G. Guaiana, J.-E. Pin (2013) have shown that the commutative closure on group languages is regularity-preserving. Actually, this is also implied by more general results about language equation due to M. Kunc (2005), see also the chapter Language Equations in the upcoming Handbook of Automata Theory by Kunc & Okhotin (available online).

The proofs were based on well-quasi-order arguments. How to construct an automaton was not clear.

Last year (Hoffmann, DCFS 2020), an automaton was constructed using the state label method.

Theorem
Let $\Sigma = \{a_1, \ldots, a_k\}$ *and* $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ *be a permutation automaton. Then* $\mathrm{perm}(L(\mathcal{A}))$ *is recognizable by an automaton with at most* $\left(|Q|^k \prod_{i=1}^{k} L_i\right)$ *states, where* L_i *for* $i \in \{1, \ldots, k\}$ *denotes the order of* a_i*. Furthermore, the recognizing automaton is computable.*

The Shuffle Languages, or The First Two Protagonists Bond

Definition
Let $\mathcal{L}$ be a class of languages.

1. $\mathcal{SE}(\mathcal{L})$ is the closure of $\mathcal{L}$ under shuffle, iterated shuffle, union, concatenation and Kleene star.

Here, we look at $\mathcal{SE}(\mathcal{G})$ for the class of group languages $\mathcal{G}$.

The Shuffle Languages, The Third Protagonist Enters

We will show that the commutative closure is (effectively) regular on shuffle languages over $\mathcal{G}$, i.e., we can compute a recognizing automaton from the input automata. We need:

Theorem

Let $\Sigma = \{a_1, \ldots, a_k\}$ and $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ be a permutation automaton. Then

$$\mathrm{perm}(L(\mathcal{A})^{\sqcup\!\sqcup,*})$$

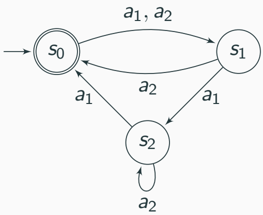
is recognizable by an automaton with at most $\left(|Q|^k \prod_{j=1}^k L_j\right) + 1$ many states, where L_j for $j \in \{1, \ldots, k\}$ denotes the order of a_j, and this automaton is effectively computable.

The Method of Proof

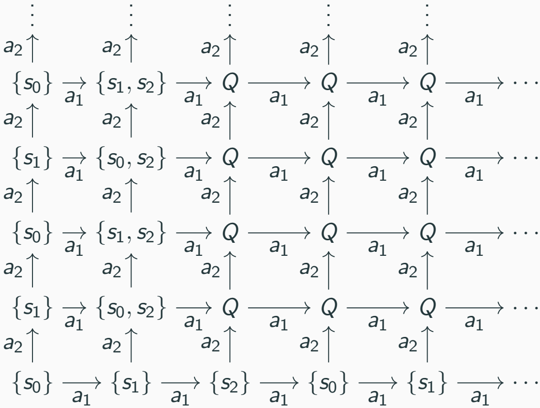
The proof uses an adaption of the state label method introduced in (Hoffmann, DCFS 2020). Let $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$.

The method in a nutshell: We label points $\mathbb{N}_0^{|\Sigma|}$ with subsets of states from $\mathcal{A}$. When $\mathcal{A}$ is a permutation automaton, these labelings induce (uniform) periodicities that allow us to construct automata for the commutative closure.

Method of Proof – The Basic Case



$Q = \{s_0, s_1, s_2\}$



State Label at point p: $S_p = \{\delta(q_0, u) \mid \psi(u) = p\}$.

Method of Proof

How to adapt this method to the iterated shuffle of group languages?

Let $\Sigma = \{a_1, \ldots, a_k\}$ and $e_i = \psi(a_i) = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{N}_0^k$ be the vector with 1 precisely at the i-th position and zero everywhere else. If $\mathcal{A} = (\Sigma, Q, \delta, q_0, F)$ is an automaton, set

$$T_{(0,\ldots,0)} = \{q_0\} \quad \text{and} \quad T_p = \bigcup_{\exists i \in \{1,\ldots k\} : p = q + e_i} \delta(S_q^+, a_i) \text{ for } p \neq (0, \ldots, 0),$$

where

$$S_p^+ = \begin{cases} T_p \cup \{q_0\} & \text{if } T_p \cap F \neq \emptyset; \\ T_p & \text{if } T_p \cap F = \emptyset. \end{cases}$$

Then, $v \in \text{perm}(L(\mathcal{A})^*) \Leftrightarrow S_p^+ \cap F \neq \emptyset$ or $v = \varepsilon$.

Method of Proof

$$
\begin{array}{ccccc}
\vdots & & \vdots & & \\
b\uparrow & & b\uparrow & & \\
S^+_{(p_a-1,p_b+1)} & \xrightarrow[a]{} & S^+_{(p_a,p_b+1)} & \xrightarrow[a]{\qquad} & \cdots \\
b\uparrow & & b\uparrow & & \\
S^+_{(p_a-1,p_b)} & \xrightarrow[a]{} & \underline{S^+_{(p_a,p_b)}} & \xrightarrow[a]{\qquad} & \cdots
\end{array}
$$

$$T_{(p_a,p_b+1)} = \delta(S^+_{(p_a-1,p_b+1)}, a) \cup \delta(S^+_{(p_a,p_b)}, b) \tag{1}$$

$$S^+_{(p_a,p_b+1)} = \begin{cases} T_{(p_a,p_b+1)} \cup \{s_0\} & \text{if } T_{(p_a,p_b+1)} \cap F \neq \emptyset; \\ T_{(p_a,p_b+1)} & \text{otherwise,} \end{cases} \tag{2}$$

Method of Proof

As we have permutation automata, the state labels grow in size and become ultimately period for some uniform constant.

We have a universal bound $N \geq 0$ and a period $P > 0$ such that, for $p = (p_1, \ldots, p_k) \in \mathbb{N}_0^K \setminus ([N] \times \ldots \times [N])$ and $j \in \{1, \ldots, k\}$, the labels at p and $(p_1, \ldots, p_{j-1}, p_j - P, p_{j+1}, \ldots, p_k)$ are equal.

The commutative closure of the language described by the original automaton is regular and could be accepted by an automaton of size at most N^k.

The Shuffle Languages, The Third Protagonist Enters

Finally:

1. With respect to the commutative closure we can replace concatenation and Kleene star by shuffle and iterated shuffle.
2. Every language in $\mathcal{S}huf(\mathcal{G})$ could be written as a finite union of languages of the form

$$L_1 \sqcup\!\sqcup \ldots \sqcup\!\sqcup L_k \sqcup\!\sqcup L_{k+1}^{\sqcup\!\sqcup,*} \sqcup\!\sqcup \ldots \sqcup\!\sqcup L_n^{\sqcup\!\sqcup,*}$$

   with $1 \leq k \leq n$ and $L_i \in \mathcal{G}$.

Hence, as the commutative closure respects the shuffle operation and union, the result follows, and all operations are effective.

(the size of an accepting automaton has size $\exp(\sqrt{n \log n})$, where n is the sum of the number of states of all the involved automata in the atomic languages of a shuffle expression for $L \in \mathcal{S}huf(\mathcal{G})$).

A Natural Question

We have shown that the commutative closure of a language in $\mathcal{S}huf(\mathcal{G})$ is regular. But, are all languages in $\mathcal{S}huf(\mathcal{G})$ regular?

Regularity of the closure is not sufficient for regularity of the original language: The non-regular context-free language given by the grammar G over $\{a, b\}$ with rules

$$S \to aTaS \mid \varepsilon, \quad T \to bSbT \mid \varepsilon.$$

for example as a regular language as the commutative closure.

Open Problem
This was posed as an open problem in the CIAA paper.

A Natural Question

Good News!

The Question was solved, and indeed they are!

More generally, this follows as polynomials (unions of marked products) of 0-group languages[1] languages are closed under iterated shuffle.

Still an open question
Size of accepting automata?

[1] Using 0-groups instead of groups is a technicality if we allow operations of group languages over different alphabets, for example $(aa)^* \cup (bbb)^*$ – I was vague in the initial definition on this. If we view $\mathcal{G}$ as a variety, such things are excluded.

A Natural Question

An insertion system is a special type of rewriting system whose rules are of the form $\varepsilon \to r$ for all r in a given language $R \subseteq \Sigma^*$. We write $u \to_R v$ if $u = u'u''$ and $v = u'ru''$ for some $r \in R$. We denote by $\to_R^*$ the reflexive transitive closure of the relation $\to_R$. Then, for $L \subseteq \Sigma^*$, we set

$$[L]_{\to_R^*} = \{v \in \Sigma^* \mid \text{there exists } u \in L \text{ such that } u \to_R^* v\}.$$

Clearly, we have $L \subseteq [L]_{\to_R^*}$.

Let $L = L_0 a_1 L_1 \cdots a_n L_n$ be a marked product with group languages $L_i \subseteq \Sigma^*$. Choose recognizing morphisms $\varphi_i : \Sigma^* \to G_i$ with finite groups G_i and set $R = \bigcap_{i=1}^n \varphi_i^{-1}(1)$.

1. For any language $U \subseteq \Sigma^*$, $[U]_{\to_R^*}$ is a polynomial of group languages.
2. For $L = L_0 a_1 L_1 \cdots a_n L_n$ and $R \subseteq \Sigma^*$ as above, we have $[L]_{\to_R^*} \subseteq L$.
3. For any $U, K \subseteq \Sigma^*$, we have $[\mathrm{perm}(U)]_{\to_K^*} \subseteq \mathrm{perm}([U]_{\to_K^*})$.

A Natural Question

As $(U \cup V)^{⧢,+} = (U^{⧢,+} ⧢ V^{⧢,+}) \cup U^{⧢,+} \cup V^{⧢,+}$ for any $U, V \subseteq \Sigma^*$ and $\mathrm{Pol}(\mathcal{G})$ is closed under shuffle and union by (Gomez,Guiana,Pin), we only need to show the claim for marked products.

Let $L = L_0 a_1 L_1 \cdots a_n L_n$ be a marked product over $\mathcal{G}$ and $R \subseteq \Sigma^*$ defined as in the beginning of this section with respect to L. We show that $L^{⧢,*}$ is closed under the relation $\to_R$, which inductively yields closure under $\to_R^*$. Let $uv \in u_1 ⧢ \ldots ⧢ u_m$ with $u_i \in L$ for some $m > 0$. Hence,

$$uv = u_{1,1} u_{2,1} \cdots u_{m,1} u_{1,2} u_{2,2} \cdots u_{m,2} \cdots u_{1,r} u_{2,r} \cdots u_{m,r}$$

with $u_i = u_{i,1} \cdots u_{i,r}$ for some $r \geq 0$. Choose $x \in R$.

Suppose $uxv = u_{1,1} u_{2,1} \cdots u_{m,1} \cdots u'_{i,j} x u''_{i,j} u_{i+1,j} \cdots u_{1,r} u_{2,r} \cdots u_{m,r}$ with $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, r\}$ with $u'_{i,j} u''_{i,j} = u_{i,j}$ (note that the argument to come entails $uxv = u_{1,1} u_{2,1} \cdots u_{m,1} \cdots u'_{m,r} x u''_{m,r}$ as the case $i = m$ and $j = r$ and, as $i = m$ and $j < r$, the case $uxv = u_{1,1} u_{2,1} \cdots u_{m,1} \cdots u'_{m,j} x u''_{m,j} u_{1,j+1} \cdots u_{m,r}$, even if the notation suggests there might be an index $i + 1$ or $j + 1$ the argument is the same without them).

Set $w = u_{i,1} \cdots u'_{i,j} x u''_{i,j} u_{i,j+1} \cdots u_{i,r}$. Then, $uxv \in u_1 ⧢ \ldots ⧢ u_{i-1} ⧢ w ⧢ u_{i+1} ⧢ \ldots u_m$. So, $w \in L$ (L is closed for insertion from identity language) and so $uxv \in L^{⧢,m} \subseteq L^{⧢,+}$.

Hence, $L^{⧢,+}$ is upward-closed for $\to_R^*$, which gives $[L^{⧢,+}]_{\to_R^*} = L^{⧢,+}$.