The n-ary Initial Literal and Literal Shuffle

Stefan Hoffmann, Universität Trier, Germany

August, 2021

The Prague Stringology Conference (PSC) 2021
Prague, Czech Republic, August 30–31, 2021

Outline

Notation & the Shuffle Operation

Notation

Σ: A finite alphabet

Σ^*: Set of a all words (finite sequences), ε denotes empty word of length zero

Language: Subset of Σ^*

Projection for $\Gamma \subseteq \Sigma$; Mapping $\pi_{\Sigma,\Gamma} : \Sigma^* \to \Gamma^*$ given by

$$\pi_{\Sigma,\Gamma}(x) = \begin{cases} x & \text{if } x \in \Gamma \\ \varepsilon & \text{otherwise} \end{cases}$$

and $\pi_{\Sigma,\Gamma}(ux)$ for $u \in \Sigma^*$, $x \in \Sigma$.

Language Families: We will consider regular, context-free, context-sensitive, recursive and recursively enumerable languages.

Automata: We will mention deterministic finite automata (DFA) and non-deterministic finite automata (NFA)

$\mathcal{P}(M)$: Power set of set M

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

Initial and Literal Shuffle & Motivation

Initial and Literal Shuffle

$I : (\Sigma^k)^n \to \Sigma^{nk}$ denotes *interleaving*.

Example: $I(aab, bbb, aaa) = abaababba$.

Berard (1987) Let $U, V \subseteq \Sigma^*$. The *initial literal shuffle* of U and V is

$$U \sqcup\!\sqcup_1 V = \{I(u,v)w \mid u,v,w \in \Sigma^*, |u| = |v|, \\ (uw \in U, v \in V) \text{ or } (u \in U, vw \in V)\}.$$

and the *literal shuffle* is

$$U \sqcup\!\sqcup_2 V = \{w_1 I(u,v) w_2 \mid w_1, u, v, w_2 \in \Sigma^*, |u| = |v|, \\ (w_1 u w_2 \in U, v \in V) \text{ or } (u \in U, w_1 v w_2 \in V) \text{ or } \\ (w_1 u \in U, v w_2 \in V) \text{ or } (u w_2 \in U, w_1 v \in V)\}.$$

Example: $\{abc\} \sqcup\!\sqcup_1 \{de\} = \{adbec\}$,
$\{abc\} \sqcup\!\sqcup_2 \{de\} = \{abcde, abdce, adbec, daebc, deabc\}$.

Motivation

1. Introduced to model the synchronization of two processes (the general shuffle models full concurrency).
2. Initial and Literal Shuffle are not associative.
3. So, how to model more than two synchronous process? How to define iterated versions naturally?
4. Here, we present *n*-ary variants of this operation to remedy these defects.

Introducing the n-ary Variants

The n-ary Initial Literal and Literal Shuffle

Instead of two words that are interleaved letter-wise, we interleave n input words that way.

Write $w_1, \ldots, w_n$ as $w_i = x_1^{(i)} \cdots x_m^{(i)}$ where $x_j^{(i)} \in \{\varepsilon\} \cup \Sigma$.

- $⧢_1^n : (\Sigma^*)^n \to \Sigma^*$ denotes the n-ary initial literal shuffle:

$$⧢_1^n(w_1, \ldots, w_n) = x_1^{(1)} \cdots x_1^{(n)} x_2^{(1)} \cdots x_2^{(n)} \cdot \ldots \cdot x_m^{(1)} \cdots x_m^{(n)}$$

  where $\forall j \in \{1, \ldots, m-1\} : x_j^{(i)} = \varepsilon \Rightarrow x_{j+1}^{(i)} = \varepsilon$. ($i \in \{1, \ldots, n\}$)
- $⧢_2^n : (\Sigma^*)^n \to \mathcal{P}(\Sigma^*)$ denotes the n-ary literal shuffle:

$$\begin{aligned} ⧢_2^n (w_1, \ldots, w_n) = \{ & x_1^{(1)} \cdots x_1^{(n)} x_2^{(1)} \cdots x_2^{(n)} \cdot \ldots \cdot x_m^{(1)} \cdots x_m^{(n)} \mid \\ \forall i \in \{1, \ldots, n\}\ \exists j \leq j' \quad & (\forall k \in \{j, \ldots, j'\} \cap [1, m] : x_k^{(i)} \neq \varepsilon) \wedge \\ & (\forall k \in \{1, \ldots, j-1\} \cap [1, m] : x_k^{(i)} = \varepsilon) \wedge \\ & (\forall k \in \{j'+1, \ldots, m\} \cap [1, m] : x_k^{(i)} = \varepsilon).\ \} \end{aligned}$$

  where $[1, m] = \{1, \ldots, m\}$ (case $\{j, \ldots, j'\} \cap [1, m] = \emptyset$ possible)

The n-ary Literal Shuffle, Example

Graphical depiction of the word

$$a_1^{(1)} a_2^{(1)} a_3^{(1)} a_4^{(1)} a_1^{(2)} a_5^{(1)} a_2^{(2)} a_6^{(1)} a_3^{(2)} a_1^{(3)} a_7^{(1)} a_4^{(2)} a_2^{(3)} a_5^{(2)} a_3^{(3)} a_6^{(2)} a_7^{(2)} a_8^{(2)} a_9^{(2)}$$

from $\shuffle_2^3(u, v, w)$ with

$$\begin{aligned} u &= a_1^{(1)} a_2^{(1)} a_3^{(1)} a_4^{(1)} a_5^{(1)} a_6^{(1)} a_7^{(1)}, \\ v &= a_1^{(2)} a_2^{(2)} a_3^{(2)} a_4^{(2)} a_5^{(2)} a_6^{(2)} a_7^{(2)} a_8^{(2)} a_9^{(2)}, \\ w &= a_1^{(3)} a_2^{(3)} a_3^{(3)}. \end{aligned}$$

$a_1^{(1)}$	$a_2^{(1)}$	$a_3^{(1)}$	$a_4^{(1)}$	$a_5^{(1)}$	$a_6^{(1)}$	$a_7^{(1)}$						
			$a_1^{(2)}$	$a_2^{(2)}$	$a_3^{(2)}$	$a_4^{(2)}$	$a_5^{(2)}$	$a_6^{(2)}$	$a_7^{(2)}$	$a_8^{(2)}$	$a_9^{(2)}$	
					$a_1^{(3)}$	$a_2^{(3)}$	$a_3^{(3)}$					

Iterated Variants

Let $L \subseteq \Sigma^*$ be a language. Then, for $i \in \{1, 2\}$, define

$$L^{\sqcup\!\sqcup_i, \circledast} = \{\varepsilon\} \cup \bigcup_{n \geq 1} \sqcup\!\sqcup_i^n (L, \ldots, L).$$

Formal Properties

Formal Properties: Let $L_1, \ldots, L_n \subseteq \Sigma^*$ and $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ a permutation. Then

1. $L_{\pi(1)} \cdot \ldots \cdot L_{\pi(n)} \subseteq ⧢_2^n(L_1, \ldots, L_n)$

   $L_1^* \subseteq L_1^{⧢_2, \circledast}$; (concatenations in any order)

Formal Properties

Formal Properties: Let $L_1, \ldots, L_n \subseteq \Sigma^*$ and $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ a permutation. Then

1. $L_{\pi(1)} \cdot \ldots \cdot L_{\pi(n)} \subseteq ⧢_2^n(L_1, \ldots, L_n)$

   $L_1^* \subseteq L_1^{⧢_2, \circledast}$; (concatenations in any order)

2. Let $k \in \mathbb{N}_0$. Then
   $⧢_2^n(L_1, \ldots, L_n) = ⧢_2^n(L_{((1+k-1) \bmod n)+1}, \ldots, L_{((n+k-1) \bmod n)+1})$
   (cyclic permutation of arguments)

Formal Properties

Formal Properties: Let $L_1, \ldots, L_n \subseteq \Sigma^*$ and $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ a permutation. Then

1. $L_{\pi(1)} \cdot \ldots \cdot L_{\pi(n)} \subseteq ⧢_2^n(L_1, \ldots, L_n)$

   $L_1^* \subseteq L_1^{⧢_2, \circledast}$; (concatenations in any order)

2. Let $k \in \mathbb{N}_0$. Then
   $⧢_2^n(L_1, \ldots, L_n) = ⧢_2^n(L_{((1+k-1) \bmod n)+1}, \ldots, L_{((n+k-1) \bmod n)+1})$
   (cyclic permutation of arguments)

3. $⧢_1^n(L_1, \ldots, L_n) \subseteq ⧢_2^n(L_1, \ldots, L_n) \subseteq L_1 ⧢ \ldots ⧢ L_n$;
   $L_1^{⧢_1, \circledast} \subseteq L_1^{⧢_2, \circledast} \subseteq L_1^{⧢, *}$; (inclusion relations)

4. for $u_1, \ldots, u_n, u \in \Sigma^*$, if $u \in ⧢_i^n(\{u_1\}, \ldots, \{u_n\})$, then
   $|u| = |u_1| + \ldots + |u_n|$ (length preserving)

Closure Properties

A full trio (Ginsburg & Greibach, 1967) is a family of languages closed under

1. homomorphisms,
2. inverse homomorphisms,
3. and intersection with regular languages.

Closure Properties

A full trio (Ginsburg & Greibach, 1967) is a family of languages closed under

1. homomorphisms,
2. inverse homomorphisms,
3. and intersection with regular languages.

Let $\mathcal{L}$ be a full trio. The following are equivalent:

1. $\mathcal{L}$ is closed under shuffle.
2. $\mathcal{L}$ is closed under literal shuffle.
3. $\mathcal{L}$ is closed under initial literal shuffle.

Closure Properties

A full trio (Ginsburg & Greibach, 1967) is a family of languages closed under

1. homomorphisms,
2. inverse homomorphisms,
3. and intersection with regular languages.

Let $\mathcal{L}$ be a full trio. The following are equivalent:

1. $\mathcal{L}$ is closed under shuffle.
2. $\mathcal{L}$ is closed under literal shuffle.
3. $\mathcal{L}$ is closed under initial literal shuffle.
4. $\mathcal{L}$ is closed under the n-ary initial literal shuffle for a fixed $n \geq 2$.
5. $\mathcal{L}$ is closed under n-ary literal shuffle for a fixed $n \geq 2$.

Further, for $L \subseteq \Sigma^*$ and $i \in \{1, 2\}$,

$$L^{\sqcup\!\sqcup,*} = \pi_{\Sigma \cup \{\$\},\Sigma}(\pi^{-1}_{\Sigma \cup \{\$\},\Sigma}(L)^{\sqcup\!\sqcup_i,\circledast}).$$

Closure Properties

Closed under $\sqcup_i^n$ for $i \in \{1, 2\}$ and every fixed $n \geq 2$:

1. regular languages
2. context-sensitive
3. recursive
4. recursively enumerable

Closed under both iterated versions:

1. context-sensitive
2. recursive
3. recursively enumerable

Closure Properties

Class of context-free languages not closed.

Let

1. $U = \{a^{2n}b^n c \mid n \geq 1\}$,
2. $V = \{a^n b^{2n} c \mid n \geq 1\}$.

Then,

$$\sqcup\!\sqcup_1 (U, V) \cap \{a, b\}^* cc = \sqcup\!\sqcup_2(U, V) \cap \{a, b\}^* cc = \{a^n (ba)^n b^n cc \mid n \geq 1\}.$$

In general, a full trio is closed under a shuffle variant iff it is closed under intersection (Ginsburg, 1975). Even for finite languages, for the iterated versions we have:

$$(abc)^{\sqcup\!\sqcup_1,\circledast} = (abc)^{\sqcup\!\sqcup_2,\circledast} \cap a^* b^* c^* = \{a^m b^m c^m \mid m \geq 0\}.$$

Computational Complexity

The following problems are in P:

1. Input: $L \subseteq \Sigma^*$ represented by NFA, $w_1, \ldots, w_n \in \Sigma^*$.
   Question: $\sqcup\!\sqcup_1^n(w_1, \ldots, w_n) \in L$?
2. Input: Words $w, v \in \Sigma^*$
   Question: $w \in \{v\}^{\sqcup\!\sqcup_1, \circledast}$?

The following problem is NP-complete if $|\Sigma| \geq 3$:

1. Input: $L \subseteq \Sigma^*$ represented by DFA, $w_1, \ldots, w_n \in \Sigma^*$.
   Question: $\sqcup\!\sqcup_2^n(w_1, \ldots, w_n) \cap L \neq \emptyset$.

Computational Complexity

Problem

Input: $L \subseteq \Sigma^*$ represented by DFA, $w_1, \ldots, w_n \in \Sigma^*$.
Question: $\sqcup_2^n(w_1, \ldots, w_n) \cap L \neq \emptyset$.

Reduction from 3-Partition. Strongly NP-complete, i.e., it is NP-complete even when the input numbers are encoded in unary.

Input: Multiset $S = \{n_1, \ldots, n_{3m}\}$ *with* $B = (\sum_{i=1}^{3m} n_i)/m \in \mathbb{N}_0$.
Question: Partition S *into* m *disjoint* $S_1, \ldots, S_m$ *such that for each* $1 \leq k \leq m$, $|S_k| = 3$ *and* $\sum_{n \in S_k} n = B$.

Problem

Input: $L \subseteq \Sigma^*$ represented by DFA, $w_1, \ldots, w_n \in \Sigma^*$.
Question: $\sqcup_2^n(w_1, \ldots, w_n) \cap L \neq \emptyset$.

Reduction from 3-PARTITION. Strongly NP-complete, i.e., it is NP-complete even when the input numbers are encoded in unary.

Input: Multiset $S = \{n_1, \ldots, n_{3m}\}$ with $B = (\sum_{i=1}^{3m} n_i)/m \in \mathbb{N}_0$.
Question: Partition S into m disjoint $S_1, \ldots, S_m$ such that for each $1 \leq k \leq m$, $|S_k| = 3$ and $\sum_{n \in S_k} n = B$.

Reduction: Let $S = \{n_1, \ldots, n_{3m}\}$ be an instance of 3-PARTITION. Set

$$L = \{aaauc \in \{a, b, c\}^* \mid |u|_b = B, |u|_a = 0, |u|_c = 2\}^m.$$

We can construct a deterministic automaton for L in polynomial time. Then, the given instance of 3-PARTITION has a solution if and only if

$$L \cap \sqcup_2^n(ab^{n_1}c, ab^{n_2}c, \ldots, ab^{n_{3m}}c) \neq \emptyset. \quad \square$$

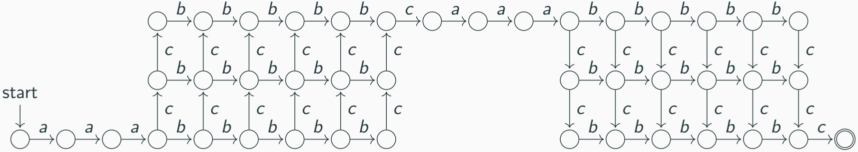
Computational Complexity

Input: $S = \{1, 1, 2, 1, 3, 2\}$.

Solution: $\{1, 1, 3\}, \{1, 2, 2\}$

Reduction: $L = U \cdot U$ with

$$U = \{aaabbbbbccc, aaabbbbcbcc, \ldots, aaaccbbbbbc\}.$$



$$L \cap \sqcup_2^6(abc, abc, abbc, abc, abbbc, abbc) \neq \emptyset$$

Computational Complexity

Ogden, Riddle & Round (1978) have shown that there exist deterministic context-free languages $U, V \subseteq \Sigma^*$ such that

Input: $w \in \Sigma^*$
Question: $w \in U \sqcup\!\sqcup V$

is NP-complete. This result was improved by Berglund, Björklund & Johanna Björklund (2013) to linear deterministic context-free languages.

Computational Complexity

Ogden, Riddle & Round (1978) have shown that there exist deterministic context-free languages $U, V \subseteq \Sigma^*$ such that

Input: $w \in \Sigma^*$
Question: $w \in U \sqcup\!\sqcup V$

is NP-complete. This result was improved by Berglund, Björklund & Johanna Björklund (2013) to linear deterministic context-free languages.

If $U, V \subseteq \Sigma^*$ are context-free, then the following problem are in P:

Input: $w \in \Sigma^*$
Question: $w \in U \sqcup\!\sqcup_1 V$

Input: $w \in \Sigma^*$
Question: $w \in U \sqcup\!\sqcup_2 V$

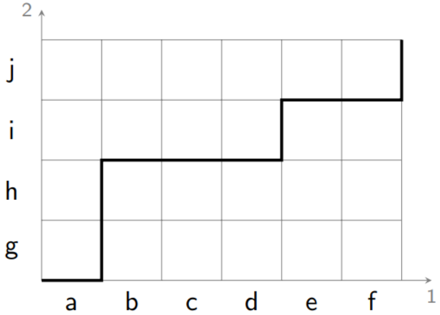
Relation to Shuffle on Trajectories

Shuffle on Trajectories

Mateescu et al.: “Shuffle on trajectories” (1998)

Trajectory

$\sqcup\!\sqcup^{2}_{1221112112}(\text{abcdef}, \text{ghij})$
$= \text{aghbcdiefj}$



Only defined if the trajectory fits the operands; generalizes to languages.

(Example due to Edixhoven & Jongmans, DLT 2021 Talk)

$\sqcup\!\sqcup^{n}_{T}(L_1, \ldots, L_n) = h(L'_1 \sqcup\!\sqcup \ldots \sqcup\!\sqcup L'_n \cap T')$ where

1. $L'_i \subseteq \Sigma_i^*$ copies of $L_i \subseteq \Sigma$ over pairwise disjoint alphabets
2. T' result from T by replacing i by Σ_i
3. $h : (\Sigma_1 \cup \ldots \cup \Sigma_n)^* \to \Sigma^*$ identifies letters again

Shuffle on Trajectories

1. $\sqcup\!\sqcup_1^3(w_1, w_2, w_3) = \sqcup\!\sqcup_T^3(w_1, w_2, w_3)$ with

$$T = (123)^*(12)^*1^* \cup (123)^*(12)^*2^* \cup (123)^*(23)^*2^* \cup (123)^*(23)^*3^*$$

2. $\sqcup\!\sqcup_2^3(w_1, w_2, w_3) = \sqcup\!\sqcup_T^3(w_1, w_2, w_3)$ with

$$\begin{aligned} T = & 1^*((12)^* + (13)^*)(123)^*((12)^* + (13)^*)1^* \\ & 2^*((12)^* + (23)^*)(123)^*((12)^* + (23)^*)2^* \\ & 3^*((13)^* + (23)^*)(123)^*((13)^* + (23)^*)3^* \\ & \quad \cdots \\ & 1^*2^*3^* + 2^*3^*1^* + 3^*2^*1^* + 1^*3^*2^* + 2^*1^*3^* + 3^*2^*1^*. \end{aligned}$$

Shuffle on Trajectories

1. Mateescu et al. (JALC 2000) extended the binary shuffle on trajectories to an n-ary version similar to the one presented here and investigated decision problems related to a fairness condition.
2. Edixhoven & Jongmans (DLT 2021) used these n-ary operations to characterize special Dyck languages were parentheses of different type freely commute by regular-like expressions.

Conclusion

Further Results & Outlook

1. Similar commutative operations with analogous closure and complexity results (see the paper)
2. The NP-complete decision problem needed an alphabet of size three. What about alphabets of size two? (in case of an unary alphabet, all n-ary shuffle variants reduce to n times concatenation and the problem is in P).
3. Investigate further, or related, decision problems for the shuffle operation from the literature for the n-ary (iterated) variants (see for example work of Berglund et al. (2013), or Eremondi et al. (2021))
4. Expand investigation to arbitrary trajectories: For example, for which trajectories are certain decision problems in P and for which NP-complete.

Thank you for your attention!