Shuffles and Synchronized Shuffles: A Survey

Maurice H. ter Beek[1], Jetty Kleijn[2]

[1] Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", CNR
Via G. Moruzzi 1, 56124 Pisa, Italy
maurice.terbeek@isti.cnr.it
[2] Leiden Institute of Advanced Computer Science, Universiteit Leiden
P.O. Box 9512, 2300 RA Leiden, The Netherlands
h.c.m.kleijn@liacs.leidenuniv.nl

Summary. We present an overview of different approaches to define shuffles and synchronized shuffles of words. The shuffle operations considered are distinguished by conditions according to which certain occurrences of symbols common to the original words may or must be identified (synchronized). The words that are shuffled may be infinite which leads to the possibility of unfair shuffling. In addition to illustrating the conceptual differences, we survey and extend known results.

"Our conclusion is, in brief, that shuffling is complex."

Ogden, Riddle, and Round in [27], POPL'78

1 Introduction

A *shuffle* of two words is any interleaving of the occurrences of symbols in these words. The operation of shuffling two words can thus be compared to the shuffling of two decks of playing cards. The notion of shuffle was first introduced as a mathematical operation in [15] in the context of abelian groups. By now it is a well-known language-theoretic operation with a long history in theoretical computer science, in particular in formal language theory [17, 16, 18, 23, 31]. The underlying idea also appears in disguise in other branches of computer science. In concurrency theory, *e.g.*, as a semantics for parallel operators modelling communication among processes [27, 30, 8]. Shuffling and variants thereof appear in the literature also under the names of interleaving, weaving, blending, merging, join, infiltration, (scattered) insertion, etc.

A generalized notion of shuffle, in which some occurrences of symbols are *synchronized* rather than interleaved, was first introduced as a mathematical operation in [11] in the context of free groups. Synchronization means that two occurrences of a symbol, each from a different word, are identified as a single occurrence in the shuffle. Like shuffling, also the concept of synchronized shuffling appears in many disguises in computer science. To the best of our knowledge, one of its earliest uses was to define the concurrent composition of

synchronizing processes in [21, 27]. In concurrency theory, the construct was used, *e.g.*, for composition operators on trace structures in (a)synchronous circuit design in [34], and in alphabetized parallel composition of (communicating sequential) processes in [30]. In formal language theory, the idea that common symbols of words being shuffled may be synchronized, first appeared in [32]. In [12], a version requiring that all occurrences of common symbols of the words being shuffled should be synchronized with a counterpart, was introduced as 'produit de mixage' or 'shuffle à l'envers', renamed synchronized shuffle in [22]. Contrary to ordinary shuffling, the result of shuffling two words may be shorter than the total length of both words. In [2, 4], this synchronization requirement was generalized to the case that all occurrences of specific symbols must be synchronized. Two variants were defined, viz. the *full synchronized shuffle* and the *relaxed synchronized shuffle*. These operations were motivated by the behavioral specification of composite systems like team automata [3], on the basis of the behavior of their components [2, 4, 1].

Three further variants were introduced in [7, 24] in the context of systems biology, as formal representations of various forms of gene linkage during so-called genome shuffling: *arbitrary synchronized shuffle* and *weak synchronized shuffle*, in which synchronization is an option rather than a requirement, and *synchronized shuffle on backbones*. The latter operation distinguishes itself from all other operations because of the predefined order and number of occurrences of symbols to be synchronized. The motivation is that each this operation should preserve the genetic backbone of the organisms involved while all other genes can be shuffled in an arbitrary fashion.

Until now we did not refer to the length of the words to be shuffled. In some applications however, ongoing behavior is described in the form of infinite sequences of occurrences of symbols (actions executed). Such sequences are obtained as limits of their finite prefixes and a similar operational approach can be followed to define (synchronized) shuffles of infinite words: a (synchronized) shuffle of two potentially infinite words is obtained by — left to right — interleaving (or synchronizing) of occurrences of symbols from the two words while preserving the original order of these occurrences within each word. In case at least one of the original words is infinite the result may be *unfair*: from some point on, occurrences from an infinite word may prevail ad infinitum without further contributions from the other word even when it still has occurrences to contribute. In general, it is not difficult to prove associativity of shuffling variants in case of only finite words (see, *e.g.*, [2, 4, 9, 16, 19, 26, 30, 32, 33, 34]). This changes when infinite words or specific variants of shuffling are considered. It may be quite challenging to prove associativity (see, *e.g.*, [12, 22, 24, 28, 29]). There exist in fact variants for which, contrary to one's intuition, associativity does not even hold [1, 7, 12, 22, 25, 24, 26].

In this paper we provide a systematic overview and comparison, including observations on properties like associativity, for all (synchronized) shuffle operations discussed above. We do this by combining results from [5] and [6] and extending concepts and results from [24] for the synchronized shuffle on backbones to the setting of possibly infinite words. In the final concluding section we discuss related work on trajectories.

2 Preliminaries

Let Δ be a fixed but arbitrary alphabet, *i.e.*, a possibly empty, possibly infinite set of symbols (or letters). A word over Δ is a (finite or infinite) string $a_1 a_2 \cdots$ of symbols $a_i \in \Delta$. The empty word is denoted by λ. The set of all finite words over Δ (including λ) is denoted by Δ^* and $\Delta^+ = \Delta^* \setminus \{\lambda\}$ contains all nonempty finite words over Δ. The set of all infinite words over Δ is denoted by Δ^ω and $\Delta^\infty = \Delta^* \cup \Delta^\omega$ denotes the set of all finite and infinite

words over Δ. A language (over Δ) is any subset of Δ^∞. A language L of only finite words ($L \subseteq \Delta^*$) is called finitary. If L contains no finite words ($L \subseteq \Delta^\omega$), then it is called infinitary. When dealing with a singleton language $L = \{w\}$, we may omit the brackets and write w rather than $\{w\}$.

For a finite word w over Δ, let $|w|$ denote its length, *i.e.*, $|\lambda| = 0$ and $|a_1 \cdots a_n| = n$, if $a_i \in \Delta$ for all $1 \leq i \leq n$. For a word $w \in \Delta^\infty$ and $j \geq 1$, we use $w(j)$ to denote the symbol from Δ which occurs at the jth position in w, provided that either w is infinite or $j \leq |w|$. The set of all symbols occurring in w is called the alphabet of w, denoted as $alph(w)$. Thus $alph(w) = \{\, a \in \Delta : \exists\, j \in \mathbb{N} \text{ such that } w(j) = a \,\}$.

The concatenation $u \cdot v \in \Delta^\infty$ of two words $u, v \in \Delta^\infty$ is defined as follows. If $u, v \in \Delta^*$, then $(u \cdot v)(i) = u(i)$ for $1 \leq i \leq |u|$ and $(u \cdot v)(|u| + i) = v(i)$ for $1 \leq i \leq |v|$. If $u \in \Delta^*$ and $v \in \Delta^\omega$, then $(u \cdot v)(i) = u(i)$ for $1 \leq i \leq |u|$ and $(u \cdot v)(|u| + i) = v(i)$ for $i \geq 1$. If $u \in \Delta^\omega$ and $v \in \Delta^\infty$, then $(u \cdot v)(i) = u(i)$ for all $i \geq 1$.

A word $u \in \Delta^*$ is a prefix of a word $w \in \Delta^\infty$, written $u \leq w$, if there exists a word $v \in \Delta^\infty$ such that $w = uv$. The set of all prefixes of w is $pref(w) = \{\, u \in \Delta^* : u \leq w \,\}$. For a language L, $pref(L) = \bigcup\{\, pref(w) : w \in L \,\}$ contains all prefixes of all words in L. A word $u = u_1 \cdots u_n \in \Delta^*$, with $u_i \in \Delta$, is a subword (sparse subword) of a word $w \in \Delta^\infty$ if $w = w_1 u w_2$ ($w \in \Delta^* u_1 \Delta^* \cdots \Delta^* u_n \Delta^\infty$, respectively).

Finite and infinite words can be obtained as the limit of their prefixes. Let $v_1, v_2, \ldots \in \Delta^*$ be an infinite sequence of words such that $v_i \leq v_{i+1}$, for all $i \geq 1$. Then $\lim_{n\to\infty} v_n$ is the unique word $w \in \Delta^\infty$ defined by $w(j) = v_i(j)$, for all $i, j \in \mathbb{N}$ such that $j \leq |v_i|$. Hence $v_i \leq w$ for all $i \geq 1$ and $w = v_k$ whenever there exists a $k \geq 1$ such that $v_n = v_{n+1}$ for all $n \geq k$. For an infinite sequence of finite words $u_1, u_2, \ldots \in \Delta^*$, we use $u_1 u_2 \cdots$ to denote the word $\lim_{n\to\infty} u_1 \cdots u_n$. A language $K \subseteq \Delta^\infty$ is limit-closed (see [6]) if $\lim_{n\to\infty} w_n \in K \cup pref(K)$ whenever $w_1 \leq w_2 \leq \cdots \in pref(K)$. Limit-closedness guarantees that the finite prefixes of a language determine its infinitary part and resembles the concept of adherence of a language [10]. Note that all singleton languages and all finite languages in fact, are limit-closed. In contrast, the language a^* is not limit-closed because $\lim_{n\to\infty} a^n = a^\omega \notin a^*$.

Let Σ and Γ be two alphabets and let $h : \Sigma \to \Gamma^*$ be a function that assigns to each symbol of Σ a finite word over Γ. The homomorphic extension of h to Σ^*, also denoted by h, is defined in the usual way by $h(xy) = h(x)h(y)$ for all $x, y \in \Sigma^*$, and we extend h to Σ^∞ by setting $h(\lim_{n\to\infty} v_n) = \lim_{n\to\infty} h(v_n)$, for all $v_1, v_2, \ldots \in \Sigma^*$ such that $v_i \leq v_{i+1}$ for all $i \geq 1$. Note that this is well-defined, since $v_i \leq v_{i+1}$ implies $h(v_i) \leq h(v_{i+1})$. The function $pres_{\Sigma,\Gamma} : \Sigma \to \Gamma \cup \lambda$ preserves symbols from Γ while erasing all other symbols in Σ. It is defined by $pres_{\Sigma,\Gamma}(a) = \lambda$ if $a \in \Sigma \setminus \Gamma$ and $pres_{\Sigma,\Gamma}(a) = a$ if $a \in \Sigma \cap \Gamma$. In the sequel, when the domain alphabet Σ is understood, we may omit the subscript Σ from $pres_{\Sigma,\Gamma}$.

3 Shuffles

Shuffling two words means combining these words into one by interleaving their symbol occurrences while preserving the original order within each word.

Definition 1. *Let $u, v \in \Delta^\infty$. Then*

(1) $w \in \Delta^\infty$ is a fair shuffle *of u and v if $w = u_1 v_1 u_2 v_2 \cdots$, where $u_i, v_i \in \Delta^*$, for all $i \geq 1$, are such that $u = u_1 u_2 \cdots$ and $v = v_1 v_2 \cdots$, and*

(2) $w \in \Delta^\infty$ is a shuffle *of u and v if either*

a) w is a fair shuffle of u and v, or

b) $w = u_1v_1u_2v_2\cdots$, *with* $u_i, v_i \in \Delta^*$ *for all* $i \geq 1$ *and either* $u_1u_2\cdots \in pref(u)$ *and* $v = v_1v_2\cdots \in \Delta^\omega$ *or* $u = u_1u_2\cdots \in \Delta^\omega$ *and* $v_1v_2\cdots \in pref(v)$.

The set of all fair shuffles of two words u and v is denoted by $u \,|||\, v$, and $u \,||\, v$ is the set consisting of all shuffles of u and v. Shuffling two languages is defined element-wise. Hence, given two languages L_1 and L_2, their *fair shuffle* is $L_1 \,|||\, L_2 = \{\, w \in u \,|||\, v : u \in L_1,\ v \in L_2 \,\}$ and their *shuffle* is $L_1 \,||\, L_2 = \{\, w \in u \,||\, v : u \in L_1,\ v \in L_2 \,\}$.

Note that, by definition, shuffles of two finite words are always fair: $u \,||\, v = u \,|||\, v$ whenever u and v are finite words. Moreover, also by definition, $u \,|||\, v \subseteq u \,||\, v$ also holds if u and/or v are infinite. This relationship between fair and unfair is a property that in fact will hold for all synchronized shuffles considered in the rest of this paper. In general however, $u \,||\, v = u \,|||\, v$ does not hold as there may be more unfair than fair shuffles when infinite words are taken into account. In fact, any unfair shuffle u and v which is not a fair shuffle, can be obtained as a fair shuffle of u and a prefix of v or as a fair shuffle of v and a prefix of u.

Example 1. $a \,||\, b = \{ab, ba\}$ and $a^n \,||\, b = \{\, a^iba^j : i, j \geq 0,\ i+j = n \,\}$. Note that every shuffle in $a^n \,||\, b$ is fair. While $a^\omega \,|||\, b = \{\, a^iba^\omega : i \geq 0 \,\}$, we have $a^\omega \,||\, b = (a^\omega \,|||\, b) \cup a^\omega$. with a^ω an unfair shuffle of a^ω and b. Note however that also for infinite words it may be the case that all shuffles are fair shuffles: $a^\omega \,|||\, a = a^\omega \,||\, a = a^\omega$.

An alternative definition of fair shuffling (of words over disjoint alphabets) is provided by the following observation.

Theorem 1. *Let* Δ_1 *and* Δ_2 *be such that* $\Delta_1 \cap \Delta_2 = \varnothing$. *Let* $u \in \Delta_1^\infty$ *and* $v \in \Delta_2^\infty$. *Then* $u \,|||\, v = \{\, w \in (\Delta_1 \cup \Delta_2)^\infty : pres_{\Delta_1}(w) = u,\ pres_{\Delta_2}(w) = v \,\}$.

It is immediate that $u \,|||\, v$ and $u \,||\, v$ are never empty. Furthermore, it is easy to see that both fair shuffling and shuffling are commutative operations. Proving that shuffling is always associative requires however some additional observations. First of all, we note that the shuffles of the prefixes of two words exactly comprise all prefixes of the shuffles of these words.

Theorem 2. [5] *Let* $u, v \in \Delta^\infty$. *Then* $pref(u) \,||\, pref(v) = pref(u \,||\, v) = pref(u \,|||\, v) = pref(u) \,|||\, pref(v)$.

Moreover, a word is a shuffle of two given words whenever all its prefixes are shuffles of prefixes of these two words.

Theorem 3. [5] *Let* $u, v \in \Delta^\infty$ *and let* $w \in \Delta^\omega$. *Then* $w \in u \,||\, v$ *if and only if* $pref(w) \subseteq pref(u) \,||\, pref(v)$.

In general, this property does not hold for fair shuffles.

Example 2. $pref(a^\omega) = a^* \subseteq pref(a^\omega) \,|||\, pref(b) = pref(a^\omega) \,||\, pref(b)$, but $a^\omega \notin a^\omega \,|||\, b = \{\, a^iba^\omega : i \geq 0 \,\}$.

The property of defining shuffles through prefixes can now be lifted to limit-closed languages, since the infinitary part of such a language is defined by its finite prefixes.

Theorem 4. [5] *Let* $K, L \subseteq \Delta^\infty$ *be limit-closed and let* $w \in \Delta^\omega$. *Then* $w \in K \,||\, L$ *if and only if* $pref(w) \subseteq pref(K) \,||\, pref(L)$.

This property does not hold when K or L is not limit-closed.

Example 3. Let $K = a^*$ and let $L = \{\lambda\}$. Then L is limit-closed, but K is not. $pref(a^\omega) = a^* = pref(K) \,||\, pref(L)$, but $a^\omega \notin a^* = K \,||\, L$.

Using Theorems 2, 3, and 4, it could be proved in [5] that shuffling is associative.

Theorem 5. [5] *Let* $u, v, w \in \Delta^\infty$. *Then* $u \,|||\, (v \,|||\, w) = (u \,|||\, v) \,|||\, w$ *and* $u \,||\, (v \,||\, w) = (u \,||\, v) \,||\, w$.

4 Synchronized Shuffles

In a synchronized shuffle of two words, each occurrence of a symbol either originates from one of these words, similar to the normal shuffle, or it is an occurrence on which these two words synchronize (or, to put it differently, it is an occurrence shared by these words). In [4, 2], we coined the term *backbone* for the sparse subwords defined by the synchronized occurrences.

Let Δ and Γ be two alphabets and w a word over Δ. Then $pres_\Gamma(w)$ is the *Γ-backbone* of w. When Γ is clear from the context we may omit it and speak simply of the *backbone* of w.

We will now first formally introduce synchronized shuffles on a given set of symbols and discuss some of their properties. Next, in four separate subsections, we consider variants. First the case that the synchronized symbols are exactly the symbols common to the explicitly defined alphabets of the words involved, after which we discuss the more general case that the symbols used for synchronization are a subset of that set in the second subsection. While in both of these cases synchronization on all occurrences of the chosen symbols is mandatory, in the succeeding two subsections the synchronization requirement is less strict. In the third subsection it will be no more than an option, whereas in the final subsection synchronization is required only when it is possible.

For the remainder of this paper, Δ, Γ, Δ_1, Δ_2, and Δ_3 are arbitrary alphabets.

Definition 2. *Let* $u, v \in \Delta^\infty$. *Then a word* $w \in \Delta^\infty$ *is a* synchronized shuffle *(*S-shuffle *for short)* on Γ *of* u *and* v *if*

(1) either $u = u_1 x_1 \cdots x_{n-1} u_n$ *and* $v = v_1 x_1 \cdots x_{n-1} v_n$, *with* $u_i, v_i \in (\Delta \setminus \Gamma)^*$ *and* $x_i \in \Gamma$ *for* $1 \le i \le n-1$, *and* $u_n, v_n \in (\Delta \setminus \Gamma)^\infty$, *in which case* $w = w_1 x_1 \cdots x_{n-1} w_n$ *with* $w_i \in u_i \,||\, v_i$ *for all* $1 \le i \le n$,

(2) or $u = u_1 x_1 u_2 x_2 \cdots$ *and* $v = v_1 x_1 v_2 x_2 \cdots$, *with* $u_i, v_i \in (\Delta \setminus \Gamma)^*$ *and* $x_i \in \Gamma$ *for all* $i \ge 1$, *in which case* $w = w_1 x_1 w_2 x_2 \cdots$ *with* $w_i \in u_i \,||\, v_i$ *for all* $i \ge 1$.

In (2) w *is a* fair *S-shuffle of* u *and* v; *in (1)* w *is* fair *whenever* $w_n \in (u_n \,|||\, v_n)$.

We use the following notation for S-shuffles. Given two words $u, v \in \Delta^\infty$, we write $u \,||^\Gamma v = \{\, w \in \Delta^\infty : w \text{ is an S-shuffle on } \Gamma \text{ of } u \text{ and } v \,\}$ and $u \,|||^\Gamma v = \{\, w \in \Delta^\infty : w \text{ is a fair S-shuffle on } \Gamma \text{ of } u \text{ and } v \,\}$. For $L_1, L_2 \subseteq \Delta^\infty$, the *S-shuffle* on Γ of L_1 and L_2 is $L_1 \,||^\Gamma L_2 = \{\, w \in u \,||^\Gamma v : u \in L_1,\ v \in L_2 \,\}$ and $L_1 \,|||^\Gamma L_2 = \{\, w \in u \,|||^\Gamma v : u \in L_1,\ v \in L_2 \,\}$ is the *fair S-shuffle* on Γ of L_1 and L_2.

Note that point (1) in Definition 2 deals with finite backbones and point (2) with the case of infinite backbones.

Obviously, shuffling is synchronized shuffling on an empty alphabet: $u \,||| v = u \,|||^\varnothing v$ and $u \,|| v = u \,||^\varnothing v$ for all $u, v \in \Delta^\infty$.

Example 4. $abc \,||^{\{c\}}\, cd = \{abcd\}$, but $abcd \notin abc \,||\, cd$ and $abc \,||^{\{b,c\}}\, cd = \varnothing$. We know from Example 1 that $a^\omega \,|||\, a = a^\omega \,||\, a = a^\omega$, but $a^\omega \,|||^{\{a\}}\, a = a^\omega \,||^{\{a\}}\, a = \varnothing$.

Hence, a (fair) S-shuffle of two nonempty words need not exist. In fact, an S-shuffle of two words $u, v \in \Delta^\infty$ exists if and only if they have the same backbone: $u \,||^\Gamma\, v \neq \varnothing$ if and only if $pres_\Gamma(u) = pres_\Gamma(v)$. Moreover, for all $w \in u \,||^\Gamma\, v$, we have $pres_\Gamma(w) = pres_\Gamma(u) = pres_\Gamma(v)$.

The following result provides an alternative definition for fair S-shuffling similar to Theorem 1 for fair shuffles.

Theorem 6. [6] *Let $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$ be such that $(\Delta_1 \setminus \Gamma) \cap (\Delta_2 \setminus \Gamma) = \varnothing$. Then $u \,|||^\Gamma\, v = \{\, w \in (\Delta_1 \cup \Delta_2)^\infty : pres_\Gamma(w) = pres_\Gamma(u) = pres_\Gamma(v),\ pres_{\Delta_1}(w) = u,\ pres_{\Delta_2}(w) = v \,\}$.*

Since (fair) S-shuffling is defined in terms of (fair) shuffling, the commutativity of (fair) shuffling implies that also (fair) S-shuffling is commutative. Similarly, the associativity of (fair) S-shuffling can be proven by referring to the associativity of (fair) shuffling.

Theorem 7. [6] *Let $u, v, w \in \Delta^\infty$. Then $u \,|||^\Gamma\, (v \,|||^\Gamma\, w) = (u \,|||^\Gamma\, v) \,|||^\Gamma\, w$ and $u \,||^\Gamma\, (v \,||^\Gamma\, w) = (u \,||^\Gamma\, v) \,||^\Gamma\, w$.*

It is evident that this would fail to hold if the synchronizing symbols could vary.

Example 5. $ca \,||^{\{a\}} (ab \,||^{\{c\}} c) = ca \,||^{\{a\}} \varnothing = \varnothing \neq cab = cab \,||^{\{c\}} c = (ca \,||^{\{a\}} ab) \,||^{\{c\}} c$.

Finally, also the S-shuffles of two words can be obtained as limits of the S-shuffles of their prefixes, and also in case of S-shuffles this property can be lifted to limit-closed languages (cf. Theorem 4).

Theorem 8. [6] *Let $u, v \in \Delta^\infty$, let $K, L \subseteq \Delta^\infty$ be limit-closed, and let $w \in \Delta^\omega$. Then $w \in u \,||^\Gamma\, v$ if and only if $pref(w) \subseteq pref(u) \,||^\Gamma\, pref(v)$, and $w \in K \,||^\Gamma\, L$ if and only if $pref(w) \subseteq pref(K) \,||^\Gamma\, pref(L)$.*

4.1 Full Synchronized Shuffles

A *full S-shuffle* of two words (over explicitly specified alphabets) is an S-shuffle in which all occurrences of all symbols common to their alphabets are synchronized.

Definition 3. *Let $u \in \Delta_1^\infty$, $v \in \Delta_2^\infty$, and $w \in (\Delta_1 \cup \Delta_2)^\infty$. Then w is a* full S-shuffle *(*fS-shuffle *for short) of u and v w.r.t. Δ_1 and Δ_2 if w is an S-shuffle on $\Delta_1 \cap \Delta_2$ of u and v. This fS-shuffle of u and v w.r.t. Δ_1 and Δ_2 is called* fair *if w is a fair S-shuffle on $\Delta_1 \cap \Delta_2$ of u and v.*

Given $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$, we write $u \,{}_{\Delta_1}\underline{|||}_{\Delta_2}\, v = \{\, w \in (\Delta_1 \cup \Delta_2)^\infty :$ w is a fair fS-shuffle of u and v w.r.t. Δ_1 and $\Delta_2 \,\}$ and $u \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, v = \{\, w \in (\Delta_1 \cup \Delta_2)^\infty :$ w is an fS-shuffle of u and v w.r.t. Δ_1 and $\Delta_2 \,\}$.

For $L_1 \subseteq \Delta_1^\infty$ and $L_2 \subseteq \Delta_2^\infty$, the *fS-shuffle* of L_1 and L_2 w.r.t. Δ_1 and Δ_2 is $L_1 \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, L_2 = \{\, w \in u \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, v : u \in L_1,\ v \in L_2 \,\}$ and $L_1 \,{}_{\Delta_1}\underline{|||}_{\Delta_2}\, L_2 = \{\, w \in u \,{}_{\Delta_1}\underline{|||}_{\Delta_2}\, v :$ $u \in L_1,\ v \in L_2 \,\}$ is the *fair fS-shuffle* of L_1 and L_2 w.r.t. Δ_1 and Δ_2.

Example 6. If $\Delta = \{a, b, c, d\}$, then $abc \,{}_{\Delta}\underline{||}_{\Delta}\, cd = \varnothing$. With $\Delta' = \{a, b, c\}$ and $\Delta'' = \{c, d\}$, on the other hand, $abc \,{}_{\Delta'}\underline{||}_{\Delta''}\, cd = abc \,||^{\{c\}}\, cd = \{abcd\}$. Also, $a^\omega \,{}_{\Delta}\underline{||}_{\Delta}\, b = a^\omega \,{}_{\Delta}\underline{|||}_{\Delta}\, b = \varnothing$, while $a^\omega \,{}_{\{a\}}\underline{|||}_{\{b\}}\, b = a^\omega \,||\, b = \{\, w \in \{a, b\}^\omega : w$ has one occurrence of $b \,\}$ and $a^\omega \,{}_{\{a\}}\underline{||}_{\{b\}}\, b = a^\omega \,||\, b = (a^\omega \,{}_{\{a\}}\underline{|||}_{\{b\}}\, b) \cup \{a^\omega\}$.

This example shows that a (fair) fS-shuffle of two nonempty words may be empty.

For all pairs of words (or languages) over Δ_1 and Δ_2, we have that ${}_{\Delta_1}|||\underline{\ }_{\Delta_2} = |||^{\Delta_1\cap\Delta_2}$ and ${}_{\Delta_1}||\underline{\ }_{\Delta_2} = ||^{\Delta_1\cap\Delta_2}$. Hence, in case of disjoint alphabets $\Delta_1 \cap \Delta_2 = \varnothing$, the fS-shuffle is the ordinary shuffle for words and languages over Δ_1 and Δ_2: ${}_{\Delta_1}|||\underline{\ }_{\Delta_2} = |||^{\varnothing} = |||$ and ${}_{\Delta_1}||\underline{\ }_{\Delta_2} = ||^{\varnothing} = ||$.

Also the alternative definition of fair S-shuffling is directly carried over to fair fS-shuffling, because non-synchronizing symbols are by definition not shared in fS-shuffles and the backbone of synchronizing symbols is an invariant of the words involved.

Theorem 9. *Let $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$. Then $u\ {}_{\Delta_1}|||\underline{\ }_{\Delta_2}\ v = \{\, w \in (\Delta_1\cup\Delta_2)^\infty : pres_{\Delta_1}(w) = u, pres_{\Delta_2}(w) = v \,\}$.*

Since (fair) S-shuffling is commutative, also (fair) fS-shuffling is commutative. Note however that the words commute together with their alphabets that act as parameters to the operation: $u\ {}_{\Delta_1}|||\underline{\ }_{\Delta_2}\ v = v\ {}_{\Delta_2}|||\underline{\ }_{\Delta_1}\ u$ and $u\ {}_{\Delta_1}||\underline{\ }_{\Delta_2}\ v = v\ {}_{\Delta_2}||\underline{\ }_{\Delta_1}\ u$. Fair fS-shuffling is also associative, again with a natural distribution of the alphabets of the words.

Theorem 10. [6] *Let $u \in \Delta_1^\infty$, let $v \in \Delta_2^\infty$, and let $w \in \Delta_3^\infty$. Then $u\ {}_{\Delta_1}|||\underline{\ }_{\Delta_2\cup\Delta_3}\ (v\ {}_{\Delta_2}|||\underline{\ }_{\Delta_3}\ w) = (u\ {}_{\Delta_1}|||\underline{\ }_{\Delta_2}\ v)\ {}_{\Delta_1\cup\Delta_2}|||\underline{\ }_{\Delta_3}\ w$.*

However, unfair fS-shuffling is *not* associative.

Example 7. Obviously, $a\ {}_{\{a,b\}}||\underline{\ }_{\{b\}}\ b = \varnothing$ and thus $(a\ {}_{\{a,b\}}||\underline{\ }_{\{b\}}\ b)\ {}_{\{a,b\}}||\underline{\ }_{\{c\}}\ c^\omega = \varnothing$. However, $a\ {}_{\{a,b\}}||\underline{\ }_{\{b,c\}}\ (b\ {}_{\{b\}}||\underline{\ }_{\{c\}}\ c^\omega) = a\ {}_{\{a,b\}}||\underline{\ }_{\{b,c\}}\ (\{\, c^n bc^\omega : n \geq 0 \,\} \cup \{c^\omega\}) = a\ {}_{\{a,b\}}||\underline{\ }_{\{b,c\}}\ \{c^\omega\} = \{\, c^n ac^\omega : n \geq 0 \,\} \cup \{c^\omega\}$.

The solution adopted in [6] is to only consider the case that all prefixes of the words involved in the fS-shuffles are present proving the associativity of (general, unfair) fS-shuffling when applied to prefix-closed languages.

Theorem 11. *Let $u \in \Delta_1^\infty$, $v \in \Delta_2^\infty$, and $w \in \Delta_3^\infty$. Then $(\{u\} \cup pref(u))\ {}_{\Delta_1}||\underline{\ }_{\Delta_2\cup\Delta_3}\ ((\{v\} \cup pref(v))\ {}_{\Delta_2}||\underline{\ }_{\Delta_3}\ (\{w\} \cup pref(w))) = ((\{u\} \cup pref(u))\ {}_{\Delta_1}||\underline{\ }_{\Delta_2}\ (\{v\} \cup pref(v)))\ {}_{\Delta_1\cup\Delta_2}||\underline{\ }_{\Delta_3}\ (\{w\} \cup pref(w))$.*

4.2 Relaxed Synchronized Shuffles

A *relaxed S-shuffle* of two words (over explicitly specified alphabets) is an S-shuffle in which all occurrences of specific symbols selected from the intersection of their alphabets are synchronized.

Definition 4. *Let $u \in \Delta_1^\infty$, $v \in \Delta_2^\infty$, and $w \in (\Delta_1 \cup \Delta_2)^\infty$. Then w is a* relaxed S-shuffle *(*rS-shuffle *for short)* on *Γ of u and v w.r.t. Δ_1 and Δ_2 if w is an S-shuffle on $\Gamma \cap \Delta_1 \cap \Delta_2$ of u and v. This rS-shuffle on Γ of u and v w.r.t. Δ_1 and Δ_2 is called* fair *if w is a fair S-shuffle on $\Gamma \cap \Delta_1 \cap \Delta_2$ of u and v.*

Given $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$, we write $u\ {}_{\Delta_1}|||^{\Gamma}_{\ \Delta_2}\ v = \{\, w \in (\Delta_1 \cup \Delta_2)^\infty : w$ is a fair rS-shuffle on Γ of u and v w.r.t. Δ_1 and $\Delta_2 \,\}$ and $u\ {}_{\Delta_1}||^{\Gamma}_{\ \Delta_2}\ v = \{\, w \in (\Delta_1 \cup \Delta_2)^\infty :$ w is an rS-shuffle on Γ of u and v w.r.t. Δ_1 and $\Delta_2 \,\}$.

For $L_1 \subseteq \Delta_1^\infty$ and $L_2 \subseteq \Delta_2^\infty$, the *fair rS-shuffle* on Γ of L_1 and L_2 w.r.t. Δ_1 and Δ_2 is $L_1\ {}_{\Delta_1}|||^{\Gamma}_{\ \Delta_2}\ L_2 = \{\, w \in u\ {}_{\Delta_1}|||^{\Gamma}_{\ \Delta_2}\ v : u \in L_1,\ v \in L_2 \,\}$ and $L_1\ {}_{\Delta_1}||^{\Gamma}_{\ \Delta_2}\ L_2 = \{\, w \in u\ {}_{\Delta_1}||^{\Gamma}_{\ \Delta_2}\ v : u \in L_1,\ v \in L_2 \,\}$ is the *rS-shuffle* on Γ of L_1 and L_2 w.r.t. Δ_1 and Δ_2.

Note that the (fair) rS-shuffles thus defined are (fair) fS-shuffles whenever $\Gamma \supseteq \Delta_1 \cap \Delta_2$. As a result, also the (fair) rS-shuffle of two nonempty words may be empty.

In fact, if $\Gamma \supseteq \Delta_1 \cap \Delta_2$, then for all pairs of words (or languages) over Δ_1 and Δ_2, we have ${}_{\Delta_1}|||^{\Gamma}_{\Delta_2} = {}_{\Delta_1}|||_{\Delta_2} = |||^{\Delta_1 \cap \Delta_2}$ and ${}_{\Delta_1}||^{\Gamma}_{\Delta_2} = {}_{\Delta_1}||_{\Delta_2} = ||^{\Delta_1 \cap \Delta_2}$. In case $\Gamma = \varnothing$, then there are no synchronizing symbols and rS-shuffling degenerates to ordinary shuffling. Hence, for all pairs of words (languages) over Δ_1 and Δ_2, we have ${}_{\Delta_1}|||^{\varnothing}_{\Delta_2} = |||^{\varnothing} = |||$ and ${}_{\Delta_1}||^{\varnothing}_{\Delta_2} = ||^{\varnothing} = ||$.

Note that this time we do not provide an alternative definition in terms of preserving symbols, since with the current operation common symbols occur without synchronization and can thus come from either of the two words.

As for the fS-shuffle, also the (fair) rS-shuffle is commutative: $u \,{}_{\Delta_1}|||^{\Gamma}_{\Delta_2}\, v = v \,{}_{\Delta_2}|||^{\Gamma}_{\Delta_1}\, u$ and $u \,{}_{\Delta_1}||^{\Gamma}_{\Delta_2}\, v = v \,{}_{\Delta_2}||^{\Gamma}_{\Delta_1}\, u$, for all $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$.

Also fair rS-shuffling is associative (again taking into account the alphabet parameters).

Theorem 12. [6] *Let $u \in \Delta_1^\infty$, let $v \in \Delta_2^\infty$, and let $w \in \Delta_3^\infty$. Then $u \,{}_{\Delta_1}|||^{\Gamma}_{\Delta_2 \cup \Delta_3} (v \,{}_{\Delta_2}|||^{\Gamma}_{\Delta_3} w) = (u \,{}_{\Delta_1}|||^{\Gamma}_{\Delta_2} v) \,{}_{\Delta_1 \cup \Delta_2}|||^{\Gamma}_{\Delta_3} w$.*

Obviously, this would not hold if the synchronizing symbols could vary.

Example 8. $(a \,{}_{\{a\}}|||^{\{a\}}_{\{b\}}\, b) \,{}_{\{a,b\}}|||^{\{b\}}_{\{a,b\}}\, ab = \{ab, ba\} \,{}_{\{a,b\}}||^{\{b\}}_{\{a,b\}}\, ab = \{aab, aba\}$, which differs from $a \,{}_{\{a\}}|||^{\{a\}}_{\{a,b\}} (b \,{}_{\{b\}}|||^{\{b\}}_{\{a,b\}}\, ab) = a \,{}_{\{a\}}|||^{\{a\}}_{\{a,b\}}\, ab = \{ab\}$.

Exactly as for the fS-shuffle, associativity is not a property of (possibly unfairly) rS-shuffling words. This can again be concluded from Example 7 by noting that rS-shuffles are fS-shuffles whenever Γ includes the alphabets involved. However, as for fS-shuffling, when applied to prefix-closed languages, rS-shuffling is associative.

Theorem 13. [6] *Let $u \in \Delta_1^\infty$, $v \in \Delta_2^\infty$, and $w \in \Delta_3^\infty$. Then $(\{u\} \cup \mathit{pref}(u)) \,{}_{\Delta_1}||^{\Gamma}_{\Delta_2 \cup \Delta_3} ((\{v\} \cup \mathit{pref}(v)) \,{}_{\Delta_2}||^{\Gamma}_{\Delta_3} (\{w\} \cup \mathit{pref}(w))) = ((\{u\} \cup \mathit{pref}(u)) \,{}_{\Delta_1}||^{\Gamma}_{\Delta_2} (\{v\} \cup \mathit{pref}(v))) \,{}_{\Delta_1 \cup \Delta_2}||^{\Gamma}_{\Delta_3} (\{w\} \cup \mathit{pref}(w))$.*

4.3 Arbitrary Synchronized Shuffles

An *arbitrary S-shuffle* of two words is an S-shuffle in which synchronization on occurrences of certain specified symbols is an option rather than a requirement.

Definition 5. *Let $u, v \in \Delta^\infty$. Then a word $w \in \Delta^\infty$ is an* arbitrary S-shuffle *(*aS-shuffle *for short)* on *Γ of u and v if*

(1) either $u = u_1 x_1 u_2 x_2 \cdots x_{n-1} u_n$ and $v = v_1 x_1 v_2 x_2 \cdots x_{n-1} v_n$, with $u_i, v_i \in \Delta^*$ and $x_i \in \Gamma$, for $1 \leq i \leq n-1$, and $u_n, v_n \in \Delta^\infty$, in which case $w = w_1 x_1 w_2 x_2 \cdots x_{n-1} w_n$ with $w_i \in u_i \,||\, v_i$ for all $1 \leq i \leq n$,

(2) or $u = u_1 x_1 u_2 x_2 \cdots$ and $v = v_1 x_1 v_2 x_2 \cdots$, with $u_i, v_i \in \Delta^*$ and $x_i \in \Gamma$, for all $i \geq 1$, in which case $w = w_1 x_1 w_2 x_2 \cdots$ with $w_i \in u_i \,||\, v_i$ for all $i \geq 1$.

In (2) w is a fair *aS-shuffle; in (1) w is* fair *whenever $w_n \in (u_n \,|||\, v_n)$.*

Given $u, v \in \Delta^\infty$, we write $u \between^\Gamma v = \{ w \in \Delta^\infty : w$ is a fair aS-shuffle on Γ of u and $v \}$ and $u \wr^\Gamma v = \{ w \in \Delta^\infty : w$ is an aS-shuffle on Γ of u and $v \}$.

For $L_1, L_2 \subseteq \Delta^\infty$, the *fair aS-shuffle* on Γ of L_1 and L_2 is $L_1 \between^\Gamma L_2 = \{ w \in u \between^\Gamma v : u \in L_1,\ v \in L_2 \}$ and $L_1 \wr^\Gamma L_2 = \{ w \in u \wr^\Gamma v : u \in L_1,\ v \in L_2 \}$ is the *aS-shuffle* on Γ of L_1 and L_2.

Note that point (1) in Definition 5 deals with the case of a finite number of synchronized occurrences and point (2) with the case of an infinite number of synchronized occurrences.

Comparing Definition 5 with Definition 2, we see that the subwords u_i and v_i may have occurrences of synchronizing symbols (from Γ) which are thus not used for synchronization. In fact, like shuffling but contrary to S-shuffling, fS-shuffling, and rS-shuffling, both $u \between^\Gamma v$ and $u \wr^\Gamma v$ are never empty. Again, we do not provide an alternative definition in terms of preserving symbols, since common symbols may occur without synchronization and can thus originate from either of the two words. Furthermore, $u \,||^\Gamma v \subseteq u \wr^\Gamma v$ always holds, but u, v, and their aS-shuffles may have different Γ-backbones. Note that the (fair) aS-shuffles thus defined are the same as the generalized shuffles defined in [11] (and renamed infiltrations in [32]) whenever $\Gamma = \Delta$. In case $\Gamma = \varnothing$, then there are no synchronizing symbols and, similar to the S-shuffle, rS-shuffling degenerates to ordinary shuffling: $\between^\varnothing = |||$ and $\wr^\varnothing = ||$.

As for S-shuffling, the commutativity of shuffling implies that also (fair) aS-shuffling is commutative. aS-shuffling is moreover associative.

Theorem 14. [6] *Let* $u, v, w \in \Delta^\infty$. *Then* $u \between^\Gamma (v \between^\Gamma w) = (u \between^\Gamma v) \between^\Gamma w$ *and* $u \wr^\Gamma (v \wr^\Gamma w) = (u \wr^\Gamma v) \wr^\Gamma w$.

This would no longer hold if the set of synchronizing symbols were not fixed.

Example 9. $cab \in ((ca \wr^{\{a\}} ab) \wr^{\{c\}} c)$, whereas all the words in $(ca \wr^{\{a\}} (ab \wr^{\{c\}} c))$ contain two occurrences of c.

4.4 Weak Synchronized Shuffles

A *weak S-shuffle* of two words is an S-shuffle in which synchronization on occurrences of certain specified symbols is required, but only if it is possible to do so.

Definition 6. *Let* $u, v \in \Delta^\infty$. *Then a word* $w \in \Delta^\infty$ *is a* weak S-shuffle *(*wS-shuffle *for short)* on Γ *of* u *and* v *if*

(1) either $u = u_1 x_1 u_2 x_2 \cdots x_{n-1} u_n$ *and* $v = v_1 x_1 v_2 x_2 \cdots x_{n-1} v_n$, *with* $u_i, v_i \in \Delta^*$, $alph(u_i) \cap alph(v_i) \cap \Gamma = \varnothing$ *and* $x_i \in \Gamma$ *for* $1 \le i \le n-1$, *and* $u_n, v_n \in \Delta^\infty$, *in which case* $w = w_1 x_1 w_2 x_2 \cdots x_{n-1} w_n$ *with* $w_i \in u_i \,||\, v_i$ *for all* $1 \le i \le n$,

(2) or $u = u_1 x_1 u_2 x_2 \cdots$ *and* $v = v_1 x_1 v_2 x_2 \cdots$, *with* $u_i, v_i \in \Delta^*$, $alph(u_i) \cap alph(v_i) \cap \Gamma = \varnothing$ *and* $x_i \in \Gamma$, *for all* $i \ge 1$, *in which case* $w = w_1 x_1 w_2 x_2 \cdots$ *with* $w_i \in u_i \,||\, v_i$ *for all* $i \ge 1$.

In (2) w *is a* fair *wS-shuffle; in (1)* w *is* fair *whenever* $w_n \in (u_n \,|||\, v_n)$.

Given $u, v \in \Delta^\infty$, we write $u \,\widetilde{||}^\Gamma v = \{ w \in \Delta^\infty : w$ is a wS-shuffle on Γ of u and $v \}$ and $u \,\widetilde{|||}^\Gamma v = \{ w \in \Delta^\infty : w$ is a fair wS-shuffle on Γ of u and $v \}$.

For $L_1, L_2 \subseteq \Delta^\infty$, the *fair wS-shuffle* on Γ of L_1 and L_2 is $L_1 \,\widetilde{|||}^\Gamma L_2 = \{ w \in u \,\widetilde{|||}^\Gamma v : u \in L_1,\ v \in L_2 \}$ and $L_1 \,\widetilde{||}^\Gamma L_2 = \{ w \in u \,\widetilde{||}^\Gamma v : u \in L_1,\ v \in L_2 \}$ is the *wS-shuffle* on Γ of L_1 and L_2.

As before, point (1) in Definition 6 deals with the case of a finite number of synchronized occurrences and point (2) with the case of an infinite number of synchronized occurrences.

Comparing Definition 6 with Definition 2, and with Definition 5, we see that also in this case not all occurrences of synchronizing symbols (from Γ) in the subwords u_i and v_i have to be used for synchronization. In fact, like shuffling and aS-shuffling but contrary to S-shuffling, fS- and rS-shuffling, both $u \,|||^{\Gamma}\, v$ and $u \,||^{\Gamma}\, v$ are never empty.

Again, we do not provide an alternative definition in terms of preserving symbols, since common symbols may occur without synchronization and can thus originate from either of the two words.

To some extent similar to aS-shuffling, $u \,||^{\Gamma}\, v \subseteq u \,\widetilde{||}^{\Gamma}\, v \subseteq u \,\wr^{\Gamma}\, v$ always holds, but u, v, and their wS-shuffles may have different Γ-backbones. Moreover, similar to the (a)S-shuffle, also the wS-shuffle degenerates to the ordinary shuffle in the absence of symbols to synchronize on: $\widetilde{|||}^{\varnothing} = |||$ and $\widetilde{||}^{\varnothing} = ||$.

Example 10. $abc \,||^{\{c\}}\, cd = \{abcd\}$. Moreover, $a^{\omega} \,\widetilde{||}^{\{a\}}\, a = a^{\omega} \,\widetilde{|||}^{\{a\}}\, a = a^{\omega} \,||\, a = a^{\omega} \,|||\, a = a^{\omega}$, while $a^{\omega} \,||^{\{a\}}\, a = \widetilde{a^{\omega}} \,|||^{\{a\}}\, a = \varnothing$.

As for S-shuffling, it follows from the commutativity of shuffling that also (fair) wS-shuffling is commutative. However, contrary to the other S-shuffles, we know from [7] that (fair) wS-shuffling is *not* associative, as is confirmed by the following example.

Example 11. Clearly $(ab \,||^{\{a,b\}}\, (ba \,||^{\{a,b\}}\, ba)) = ab \,||^{\{a,b\}}\, ba = \{aba, bab\}$, whereas $((ab \,\widetilde{||}^{\{a,b\}}\, ba) \,\widetilde{||}^{\{a,b\}}\, ba) = \widetilde{\{aba, bab\}} \,\widetilde{||}^{\{a,b\}}\, ba = \{aba, \widetilde{bab}, baba\}$.

5 Synchronized Shuffles on Backbones

An *S-shuffle* on a backbone of two words is an S-shuffle in which synchronization must adhere to a predefined backbone.

Definition 7. *Let $u, v \in \Delta^{\infty}$. Then a word $w \in \Delta^{\infty}$ is an* S-shuffle on backbone $x \in \Delta^{\infty}$ *(*bS-shuffle *for short) of u and v if*

(1) either $u = u_1 x_1 \cdots x_{n-1} u_n$, $v = v_1 x_1 \cdots x_{n-1} v_n$, and $x = x_1 x_2 \cdots x_{n-1}$, with $u_i, v_i \in \Delta^*$ and $u_n, v_n \in \Delta^{\infty}$, in which case $w = w_1 x_1 \cdots x_{n-1} w_n$ with $w_i \in u_i \,||\, v_i$ for all $1 \leq i \leq n$,
(2) or $u = u_1 x_1 u_2 x_2 \cdots$, $v = v_1 x_1 v_2 x_2 \cdots$, and $x = x_1 x_2 \cdots$, with $u_i, v_i \in \Delta^*$, in which case $w = w_1 x_1 w_2 x_2 \cdots$ with $w_i \in u_i \,||\, v_i$ for all $i \geq 1$.

In (2) w is a fair *bS-shuffle of u and v; in (1) w is* fair *whenever $w_n \in (u_n \,|||\, v_n)$.*

The bS-shuffle is parameterized by a backbone, which defines a sequence of occurrences for synchronizing rather than a set of symbols: not all symbols from the alphabet of the backbone must synchronize, *i.e.*, $pres_{alph\,(x)}(u) = pres_{alph\,(x)}(v)$ need not hold.

For $u, v, x \in \Delta^{\infty}$, we let $u \,||_x\, v = \{\, w \in \Delta^{\infty} : w \text{ is a bS-shuffle on } x \text{ of } u \text{ and } v \,\}$ and $u \,|||_x\, v = \{\, w \in \Delta^{\infty} : w \text{ is a fair bS-shuffle on } x \text{ of } u \text{ and } v \,\}$. For $L_1, L_2 \subseteq \Delta^{\infty}$ and $x \in \Delta^{\infty}$, the *bS-shuffle* on x of L_1 and L_2 is $L_1 \,||_x\, L_2 = \{\, w \in u \,||_x\, v : u \in L_1,\ v \in L_2 \,\}$ and $L_1 \,|||_x\, L_2 = \{\, w \in u \,|||_x\, v : u \in L_1,\ v \in L_2 \,\}$ is the *fair bS-shuffle* on x of L_1 and L_2.

Note that point (1) in Definition 7 deals with finite backbones and point (2) with the case of infinite backbones.

By definition, $u \,|||_x\, v$ and $u \,||_x\, v$ are empty whenever x is not a sparse subword of both u and v.

Using Theorem 6 we can generalize the observation from [24] that bS-shuffling yields an alternative definition of fair S-shuffling to the case of infinite words.

Theorem 15. *Let $u, v \in \Delta^\infty$ and let $\Gamma \subseteq \Delta$. Then $u \,|||^\Gamma\, v = (u \,|||_{pres_\Gamma(u)}\, v) \cap (u \,|||_{pres_\Gamma(v)}\, v)$.*

The definition of bS-shuffling can be extended to *sets* of backbones in the following way [24]. Given $u, v \in \Delta^\infty$ and $L_1, L_2, X \subseteq \Delta^\infty$, we define $u \,||_X\, v = \bigcup_{x \in X} u \,||_x\, v$, $u \,|||_X\, v = \bigcup_{x \in X} u \,|||_x\, v$, $L_1 \,||_X\, L_2 = \bigcup_{x \in X} L_1 \,||_x\, L_2$, and $L_1 \,|||_X\, L_2 = \bigcup_{x \in X} L_1 \,|||_x\, L_2$.

Clearly, if X is an alphabet, then $|||_{X^\infty}$ and $||_{X^\infty}$ describe the option to synchronize on any backbone consisting of (possibly infinitely many) shared occurrences of symbols from X. Hence the following relationship with full, weak, and arbitrary synchronized shuffling: for $L_1, L_2 \subseteq \Delta^\infty$ and $X \subseteq \Delta$, it is always the case that $L_1 \,|||_{X^\infty}\, L_2 \supset L_1 \,|||^X\, L_2$ and $L_1 \,||_{X^\infty}\, L_2 \supset L_1 \,||^X\, L_2$ and also $L_1 \,|||_{X^\infty}\, L_2 \supset L_1 \,|||^X\, L_2$ and $L_1 \,||_{X^\infty}\, L_2 \supset L_1 \,||^X\, L_2$, whereas $|||_{X^\infty} = \wr\wr^X$ and $||_{X^\infty} = \wr^X$. This extends observations from [24] to the case of possibly infinite words and unfair shuffling.

The commutativity of shuffling implies that (fair) bS-shuffling is commutative, for any set of backbones. From [24] we know however that (fair) bS-shuffling is *not* associative. Here we provide an example showing that even for a single backbone consisting of a single symbol, bS-shuffling is not associative.

Example 12. We have $(aa \,||_a\, (ab \,||_a\, ba)) = aa \,||_a\, bab = \{abab, baba, baab\}$, whereas $((aa \,||_a\, ab) \,||_a\, ba) = \{aab, aba\} \,||_a\, ba = \{abab, baba, baab, abba\}$.

Still, bS-shuffling is associative if the set of backbones is X^∞, for some $X \subseteq \Delta$, simply because in that case, as mentioned above, bS-shuffling equals aS-shuffling.

Example 13. In Example 12 we have seen that $abba \notin (aa \,||_a\, (ab \,||_a\, ba))$. However, it is not difficult to see that $abba \in (aa \,||_{a^*}\, (ab \,||_{a^*}\, ba))$.

In [24], it was asked for what other languages X would $||_X$ (and $|||_X$) be associative. Here, we contribute to the answer by showing that in the unary case $|||_x$ and $||_x$ are associative, for all backbones x and hence for any combination of backbones. In case of one-letter-alphabets, the bS-shuffle of words u and v on backbone x results in the single word y that is infinite whenever u or v is infinite and otherwise $|y| = |u| + |v| - |x|$ under the condition that both u and v are not a proper prefix of x.

Lemma 1. *Let $u, v, w, x, y \in \{a\}^\infty$. Then*

1. $u \,|||_x\, (v \,|||_y\, w) = (u \,|||_x\, v) \,|||_y\, w$ *if* $(u \,|||_x\, (v \,|||_y\, w)) \neq \varnothing$ *and* $((u \,|||_x\, v) \,|||_y\, w) \neq \varnothing$.
2. $u \,||_x\, (v \,||_y\, w) = (u \,||_x\, v) \,||_y\, w$ *if* $(u \,||_x\, (v \,||_y\, w)) \neq \varnothing$ *and* $((u \,||_x\, v) \,||_y\, w) \neq \varnothing$.

Example 14. $(a^2 \,||_a\, (a^3 \,||_{a^4}\, a^5)) = a^2 \,||_a\, \varnothing = \varnothing \neq \{a^5\} = \{a^4\} \,||_{a^4}\, a^5 = (a^2 \,||_a\, a^3) \,||_{a^4}\, a^5$. However, $(a^2 \,||_{a^2}\, (a^3 \,||_{a^3}\, a^5)) = a^2 \,||_{a^2}\, \{a^5\} = \{a^5\} = \{a^3\} \,||_{a^3}\, a^5 = (a^2 \,||_{a^2}\, a^3) \,||_{a^3}\, a^5$.

As an immediate consequence we obtain the following result.

Theorem 16. *Let $u, v, w, x, y \in \{a\}^\infty$. Then*

1. $u \,|||_x\, (v \,|||_x\, w) = (u \,|||_x\, v) \,|||_x\, w$.
2. $u \,||_x\, (v \,||_x\, w) = (u \,||_x\, v) \,||_x\, w$.

When more symbols are involved, the backbone x in $|||_x$ and $||_x$ should use all symbols, otherwise associativity will be violated as the following generalized version of Example 12 shows.

Theorem 17. *If $\# \in \Delta$ and $w \in (\Delta \setminus \{\#\})^+$, then $||_w$ is not associative (on Δ^∞).*

Proof. It suffices to show that $((ww\,||_w\,w\#)\,||_w\,\#w) \setminus (ww\,||_w\,(w\#\,||_w\,\#w)) \neq \emptyset$. Clearly $(ww\,||_w\,(w\#\,||_w\,\#w)) = ww\,||_w\,\{\#w\#\} = \{\#w\#w, \#ww\#, w\#w\#\}$, but $((ww\,||_w\,w\#)\,||_w\,\#w) = \{w\#w, ww\#\}\,||_w\,\#w = \{\#w\#w, w\#\#w, \#ww\#, w\#w\#\}$. Recall $w \neq \lambda$. Hence $w\#\#w \in ((ww\,||_w\,w\#)\,||_w\,\#w) \setminus (ww\,||_w\,(w\#\,||_w\,\#w))$. □

This however does not yet provide a definite answer to the question posed. Part of the problem is that associativity of bS-shuffling on a set of backbones may follow from combining different backbones rather than from combining bS-shuffles of three words on single backbones.

6 Discussion and Related Work

In this paper, we have recalled from the literature the general notion of synchronized shuffling (see, *e.g.*, [11, 32, 12, 22]) and variants of synchronized shuffling introduced during the last decade [2, 4, 7, 24], allowing infinite words and the possibility of unfair shuffling. For all six operations, we have given an overview of the results from [2, 7, 24, 6] on their status with respect to various properties and in particular associativity.

As the next example shows, the S-shuffle and the variants considered here may yield languages lying strictly within each other.

Example 15. Let $\Delta_1 = \{a, b\}$, $\Delta_2 = \{a, b, c\}$, $\Gamma = \{a, c\}$, $X = \{aa, ba\}$, $K = \{aba\}$, and $L = \{aa, aac, aba\}$. Then $K \,\underline{||}_{\Delta_1}\, \underline{||}_{\Delta_2}\, L = \{aba\}$, $K\,||^\Gamma\,L = (K \,\underline{||}_{\Delta_1}\, \underline{||}_{\Delta_2}\, L) \cup \{abba\}$, $K \,\underline{||}_{\Delta_1}^{\Gamma}\, \underline{||}_{\Delta_2}\, L = (K\,||^\Gamma\,L) \cup \{abaa\}$, $K\,||_X\,L = (K \,\underline{||}_{\Delta_1}^{\Gamma}\, \underline{||}_{\Delta_2}\, L) \cup \{aaba\}$, $K\,\widetilde{||}^\Gamma L = (K\,||_X\,L) \cup \{abaa, abaac, aabac, ababa\}$, and $K\,\wr^\Gamma L = (K\,\widetilde{||}^\Gamma L) \cup (K\,||\,L) \cup \{abbaa, aabba, abaca, aacba, aabca\}$, where $K\,||\,L = \{abaaa, \widetilde{a}abaa, aaaba, abaaac, abaaca, aabaca, aabaac, aaabac, aaabca, aaacba, aacaba, aababa, aabbaa, abaaba, ababaa\}$.

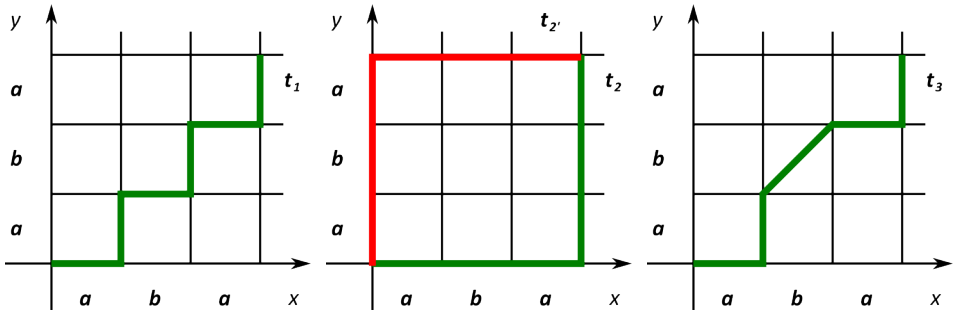
Note that even if $pres_\Gamma(u) = pres_\Gamma(v)$ and $pres_{alph\,(x)}(u) = pres_{alph\,(x)}(v)$, for all $x \in X$, at least the four S-shuffle variants that are not parameterized by the alphabets of the words involved (*i.e.*, u and v) may yield languages of increasing size.

Example 16. Let $u = v = aba \in \{a, b\}^*$, $\Gamma = \{a\}$, and $X = \{a, aa\}$. Then $u\,||^\Gamma\,v = \{abba\}$, $u\,\widetilde{||}^\Gamma\,v = (u\,||^\Gamma\,v) \cup \{ababa\}$, $u\,||_X\,v = (u\,\widetilde{||}^\Gamma\,v) \cup \{abbaa, aabba\}$, and $u\,\wr^\Gamma\,v = (u\,\widetilde{||}^\Gamma\,v) \cup \{\widetilde{a}ababa, aabbaa, abaaba, ababaa\}$.

In a series of papers (see, *e.g.*, [26, 25, 14]) Mateescu et al. use *trajectories* to describe shuffles. A trajectory defines, in a binary fashion, when to switch from one word to another, thus defining a unique way to (de)compose the shuffle of two words, independently of the structure of the words. Mateescu et al. call them shuffles on trajectories defined by *syntactic* constraints, as opposed to (synchronized) shuffles on trajectories defined by *semantic* constraints that do consider the word structure [13]. Algebraic properties like commutativity and associativity are consequently discussed per set of trajectories. In [24], the notion of trajectory is relaxed by fixing only the string of synchronizing symbols that a synchronized shuffle must adhere to.

The least restrictive set of trajectories, $\{0, 1\}^*$, defines the ordinary shuffle. An example is depicted in Fig. 1(a), viz. the shuffle of *aba* and *aba* on trajectory 010101 resulting in *aabbaa*.

Note that we make use of the fact that a trajectory has a natural geometric interpretation as a 'walk' on the nonnegative Cartesian plane. The trajectory sets 0^*1^* and 1^*0^* instead define concatenations (special cases of shuffling). Two examples are depicted in Fig. 1(b), viz. the concatenations of *aba* and *aba* on trajectories 0^31^3 and 1^30^3 resulting in *abaaba*. Inspired by the semantic shuffle on trajectories with synchronization constraint σ defined in [13], we note that the set of trajectories $\{0, 1, \sigma\}^*$ defines a synchronized shuffle. Pictorially, we visualize σ as a diagonal 'step' on the plane. An example is depicted in Fig. 1(c), viz. the synchronized shuffle of *aba* and *aba* on trajectory $01\sigma01$ resulting in *aabaa*.



(a) $t_1 = 010101 \in \{0, 1\}^*$ **(b)** $t_2 = 0^31^3$, $t_2' = 1^30^3$ **(c)** $t_3 = 01\sigma01 \in \{0,1, \sigma\}^*$

Fig. 1. Examples of (synchronized) shuffles on trajectories

References

1. A. Bailly, M. Clerbout, I. Simplot-Ryl, Component composition preserving behavioural contracts based on communication traces. *Proc. 10th International Conference on Implementation and Application of Automata (CIAA'05)* (J. Farré, I. Litovsky, S. Schmitz, eds.), LNCS, 3845, Springer, 2006, 55–66.
2. M.H. ter Beek, *Team Automata: A Formal Approach to the Modeling of Collaboration Between System Components*. Ph.D. Thesis, Leiden Institute of Advanced Computer Science, Leiden University, 2003.
3. M.H. ter Beek, C.A. Ellis, J. Kleijn, G. Rozenberg, Synchronizations in team automata for groupware systems. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 12, 1 (2003), 21–69.
4. M.H. ter Beek, J. Kleijn, Team automata satisfying compositionality. *Proc. 12th International Symposium of Formal Methods Europe (FME'03)* (K. Araki, S. Gnesi, D. Mandrioli, eds.), LNCS, 2805, Springer, 2003, 381–400.
5. M.H. ter Beek, J. Kleijn, Infinite unfair shuffles and associativity. *Theoretical Computer Sci.*, 380, 3 (2007), 401–410.
6. M.H. ter Beek, J. Kleijn, Associativity of infinite synchronized shuffles and team automata. *Fundamenta Informaticae*, 91, 3-4 (2009), 437–461.
7. M.H. ter Beek, C. Martín-Vide, V. Mitrana, Synchronized shuffles. *Theoretical Computer Sci.*, 341, 1–3 (2005), 263–275.
8. J.A. Bergstra, A. Ponse, S.A. Smolka, eds., *Handbook of Process Algebra*. Elsevier Science Publishers, Amsterdam, 2001.

9. S.L. Bloom, Z. Ésik, Free shuffle algebras in language varieties. *Theoretical Computer Sci.*, 163, 1–2 (1996), 55–98.
10. L. Boasson, M. Nivat, Adherences of languages. *Journal of Computer and System Sciences*, 20, 3 (1980), 285–309.
11. K.T. Chen, R.H. Fox, R.C. Lyndon, Free differential calculus, IV. The quotient groups of the lower central series. *Annals of Mathematics*, 68, 1 (1958), 81–95.
12. R. De Simone, Langages infinitaires et produit de mixage. *Theoretical Computer Sci.*, 31 (1984), 83–100.
13. M. Domaratzki, Semantic shuffle on and deletion along trajectories. *Proc. 8th International Conference on Developments in Language Theory (DLT'04)* (C. Calude, E. Calude, M.J. Dinneen, eds.), LNCS, 3340, Springer, 2004, 163–174.
14. M. Domaratzki, More words on trajectories. *Bulletin of the EATCS*, 86 (2005), 107–145.
15. S. Eilenberg, S. Mac Lane, On the groups $H(\pi, n)$, I. *Annals of Mathematics*, 58, 1 (1953), 55–106.
16. S. Ginsburg, *Algebraic and Automata-Theoretic Properties of Formal Languages*. Fundamental Studies in Computer Science 2, North-Holland Publishing Company, Amsterdam, 1975.
17. S. Ginsburg, E.H. Spanier, Mappings of languages by two-tape devices. *Journal of the ACM*, 12, 3 (1965), 423–434.
18. J.L. Gischer, Shuffle languages, Petri nets, and context sensitive grammars. *Communications of the ACM*, 24 (1981), 597–605.
19. M. Jantzen, The power of synchronizing operations on strings. *Theoretical Computer Science*, 14 (1981), 127–154.
20. J. Karhumäki, H.A. Maurer, Gh. Păun, G. Rozenberg, eds., *Jewels are Forever—Contributions on Theoretical Computer Science in Honor of Arto Salomaa*. Springer, Berlin, 1999.
21. T. Kimura, An algebraic system for process structuring and interprocess communication. *Proceedings of the 8th ACM SIGACT Symposium on Theory of Computing (STOC'76)*, ACM Press, New York, 1976, 92–100.
22. M.Latteux, Y. Roos, Synchronized shuffle and regular languages. In [20], 1999, 35–44.
23. M. Lothaire, *Combinatorics on Words*. Encyclopedia of Mathematics and its Applications 17, Cambridge University Press, Cambridge, 1983.
24. F. Manea, V. Mitrana, D.-C. Voinescu, Synchronized shuffle on backbones. *Fundamenta Informaticae*, 73, 1–2 (2006), 191–202.
25. A. Mateescu, G.D. Mateescu, Fair and associative infinite trajectories. In [20], 1999, 327–338.
26. A. Mateescu, G. Rozenberg, A. Salomaa, Shuffle on trajectories: Syntactic constraints, *Theoretical Computer Sci.*, 197, 1–2 (1998), 1–56.
27. W.F. Ogden, W.E. Riddle, W.C. Rounds, Complexity of expressions allowing concurrency. *Proc. 5th Annual ACM Symposium on Principles of Programming Languages (POPL'78)*, ACM Press, New York, 1978, 185–194.
28. D. Park, On the semantics of fair parallelism. *Proc. Copenhagen Winter School on Abstract Software Specifications* (D. Bjørner, ed.), LNCS, 86, Springer, 1979, 504–526.
29. R.R. Redziejowski, Associative omega-products of traces. *Fundamenta Informaticae*, 67, 1–3 (2005), 175–185.
30. A.W. Roscoe, *The Theory and Practice of Concurrency*. Prentice Hall, London, 1997.
31. G. Rozenberg, A. Salomaa, eds., *Handbook of Formal Languages*. Springer, Berlin, 1997.
32. J. Sakarovitch, I. Simon, Subwords. Chapter 6 in [23], 1983, 105–142.
33. A.C. Shaw, Software descriptions with flow expressions. *IEEE Transactions on Software Engineering*, **4**(3), 1978, 242–254.
34. J.L.A. van de Snepscheut, *Trace Theory and VLSI Design*. LNCS, 200, Springer, 1985.