Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 380 (2007) 401–410
Theoretical Computer Science
www.elsevier.com/locate/tcs

Infinite unfair shuffles and associativity

Maurice H. ter Beek[a,*], Jetty Kleijn[b]

[a] *Istituto di Scienza e Tecnologie dell'Informazione, CNR, Via G. Moruzzi 1, 56124 Pisa, Italy*
[b] *LIACS, Universiteit Leiden, PO Box 9512, 2300 RA Leiden, The Netherlands*

Abstract

We consider a general shuffling operation for finite and infinite words which is not necessarily fair. This means that it may be the case that in a shuffle of two words, from some point onwards, one of these words prevails *ad infinitum* even though the other word still has letters to contribute. Prefixes and limits of shuffles are investigated, leading to a characterization of general shuffles in terms of shuffles of finite words, a result which does not hold for fair shuffles. Associativity of shuffling is an immediate corollary.

Keywords: Shuffling; Infinite words; Fairness; Associativity

1. Introduction

The shuffling of two words is usually defined as arbitrarily interleaving subwords in such a way that the resulting word contains all letters of both words, like shuffling two decks of cards. Shuffling is a well known operation – sometimes referred to as interleaving, weaving, or merging – that, in many variants, has been extensively studied. Its popularity comes from its purely mathematical interest [6,8,10,12–15,17,19,20,23,26] and from its significance as a semantics for concurrent systems consisting of several components [2,3,5,7,16,21,22,24,27,28].

When systems may be iteratively composed, the modularity of the chosen semantics becomes important. In particular, when a form of shuffling is used to combine behaviours, this operation should be commutative and associative. In addition, systems – in particular reactive systems – may exhibit ongoing, infinite behaviours, represented by infinite words. While it is in general not difficult to prove the commutativity and associativity of shuffling operations when only finite words are involved [3,5,8,12,15,20,24,26–28], this changes when infinite words are allowed or certain variants of shuffling are considered. Mostly it is still easy to prove commutativity, but it may be quite challenging to prove associativity (see, *e.g.*, [22] and also [10,17,23]). There even exist variants of shuffling for which associativity does not hold [6,10,17,19,20], contrary to the intuition.

In this paper, we consider shuffles of possibly *infinite* words which are not necessarily fair in the sense that one of the two words may be delayed indefinitely, while for each position in the shuffle, an occurrence of a letter from the other word is chosen. Note that with this definition, a shuffle of two finite words is always a standard – fair – shuffle.

* Corresponding author. Tel.: +39 050 3153471; fax: +39 050 3152810.
E-mail addresses: maurice.terbeek@isti.cnr.it (M.H. ter Beek), kleijn@liacs.nl (J. Kleijn).

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2007.03.030

The motivation for this particular shuffle operation stems from our attempts to describe the behaviour of a certain type of team automaton as a language composed of the languages of its constituting component automata [3–5]. These languages are prefix-closed and may contain infinite words. The composed behaviour as exhibited by the team is not necessarily fair, in the sense that any individual component is allowed to execute its behaviour *ad infinitum* without giving other components a fair turn to continue. This leads to a language consisting of potentially unfair shuffles of words representing behaviours of the various components. Since team automata consist in general of two or more components and may also be defined in an iterative fashion, an associativity result for this generalized form of shuffling is needed to establish the compositionality of the semantics. As demonstrated in the Ph.D. thesis [3] of the first author, this associativity result can also be used for proving the associativity of other more involved – synchronized – shuffle operations, relevant when describing the behaviour of team automata cooperating under different synchronization strategies.

Unfortunately, we were unable to find in the literature explicit results concerning the associativity of the shuffle operation considered here, even though there do exist many references to the associativity of related shuffle operations [8,12,15,20,23,24,27,28]. We could thus have tried to adapt existing results to the general case when the words that are shuffled may be finite or infinite and the shuffle does not have to be fair. However, rather than focussing on the single property of associativity, we propose to investigate here the more general issue of the relationship between shuffles of (finite or infinite) words and the shuffles of their finite prefixes. This should shed more light on the relationships between the finite and the infinite behaviours of the composed system, and contribute to the general knowledge of shuffling in the context of infinite words. The associativity of shuffling follows as a corollary. Hence it is our aim to give a self-contained exposition, elaborating the limit behaviour of shuffles with infinite words and leading to a characterization of shuffles in terms of their prefixes.

The organization of the paper is as follows. In Section 2 we introduce the necessary notations and definitions and establish some basic properties. Also proved here is the important result that the prefixes of the shuffles of two words are exactly the shuffles of the prefixes of these words. Next, in Section 3, we separately consider fair shuffles. Using an established technique, it is proved directly that fair shuffling is associative, even when the words involved may be infinite. Consequently, in the main Section 4, we consider general shuffles. As a main result we demonstrate that a word must be a shuffle of two given words whenever all its prefixes are shuffles of the prefixes of these two words. This result does not hold if only fair shuffles are allowed. Together with the earlier result from Section 2 this leads to a characterization of shuffles, and associativity follows.

2. Basic definitions and observations

Let Δ be an alphabet, *i.e.* a (possibly empty, possibly infinite) set of symbols or letters. A word over Δ is a sequence $a_1a_2\cdots$ with each $a_i \in \Delta$. A word may be finite or infinite. The empty word is denoted by λ. For a finite word w, we use the notation $|w|$ to denote its length. Hence $|\lambda| = 0$, and if $w = a_1a_2\cdots a_n$, with $n \geq 1$ and $a_i \in \Delta$, for all $1 \leq i \leq n$, then $|w| = n$. For a word w and an integer $j \geq 1$ such that $j \leq |w|$ if w is finite, we use $w(j)$ to denote the symbol occurring at the jth position in w; by $alph(w)$ we denote the alphabet of w, *i.e.* the set of all symbols that actually occur in w.

The set of all finite words over Δ (including λ) is denoted by Δ^*. The set $\Delta^+ = \Delta^* \setminus \{\lambda\}$ consists of all nonempty finite words. The set of all infinite words over Δ is denoted by Δ^ω. By Δ^∞, we denote the set of all words over Δ. Hence $\Delta^\infty = \Delta^* \cup \Delta^\omega$. A language (over Δ) is a set of words (over Δ). A language consisting solely of finite words is called finitary. If $L \subseteq \Delta^\omega$, *i.e.* all words of L are infinite, then L is called an infinitary language. When dealing with singleton languages, we often omit brackets and write w rather than $\{w\}$.

Given two words $u, v \in \Delta^\infty$, their concatenation $u \cdot v$ 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 $i \geq 1$. Note that $u \cdot \lambda = \lambda \cdot u = u$, for all $u \in \Delta^\infty$. The concatenation of two languages K and L is the language $K \cdot L = \{u \cdot v : u \in K,\ v \in L\}$. We will mostly write uv and KL rather than $u \cdot v$ and $K \cdot L$, respectively.

A word $u \in \Delta^*$ is a (finite) prefix of a word $w \in \Delta^\infty$ if there exists a $v \in \Delta^\infty$ such that $w = uv$. In that case, we write $u \leq w$. If $u \leq w$ and $u \neq w$, then we may use the notation $u < w$. Moreover, if $|u| = n$, for some $n \geq 0$, then u is the prefix of length n of w, denoted by $w[n]$. Note that $w[0] = \lambda$. The set of all prefixes of a word w is $pref(w) = \{ u \in \Delta^* : u \leq w \}$. For a language K, $pref(K) = \bigcup\{ pref(w) : w \in K \}$ and $alph(K) = \bigcup\{ alph(w) : w \in K \}$.

Both finite and infinite words can be defined 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 $w(i) = v_j(i)$, for all $i, j \in \mathbb{N}$ such that $i \leq |v_j|$. 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 the notation $u_1 u_2 \cdots$ to denote the word $\lim_{n\to\infty} u_1 u_2 \cdots u_n$.

We now move to shuffles. We define a *shuffle* of two words as an interleaving of consecutive finite subwords of these words which stops (is finite) only if both words have been used completely. Furthermore, one (infinite) word may prevail when the other word, from some point onwards, contributes nothing but the trivial subword λ.

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) $w = u_1 v_1 u_2 v_2 \cdots$, where $u_i, v_i \in \Delta^*$, for all $i \geq 1$, and either $u_1 u_2 \cdots \in pref(u)$ and $v = v_1 v_2 \cdots \in \Delta^\omega$, or $u = u_1 u_2 \cdots \in \Delta^\omega$ and $v_1 v_2 \cdots \in pref(v)$.

For $u, v \in \Delta^\infty$, the set of all fair shuffles of u and v is denoted by $u \mathbin{|||} v$ and the set of all shuffles of u and v is denoted by $u \mathbin{||} v$. Thus, $u \mathbin{|||} v = \{ w \in \Delta^\infty : w$ is a fair shuffle of u and $v \}$ and $u \mathbin{||} v = \{ w \in \Delta^\infty : w$ is a shuffle of u and $v \}$. Note that, as defined by the fair shuffle operator $|||$ and the shuffle operator $||$, both fair shuffling and shuffling yield languages.

The shuffling of two languages is defined element-wise: The *fair shuffle* of languages L_1 and L_2 is denoted by $L_1 \mathbin{|||} L_2$ and is defined as the set of all words which are a fair shuffle of a word from L_1 and a word from L_2. Hence $L_1 \mathbin{|||} L_2 = \{ w \in u \mathbin{|||} v : u \in L_1,\ v \in L_2 \}$. Similarly, the *shuffle* of L_1 and L_2 is denoted by $L_1 \mathbin{||} L_2$ and is defined as $L_1 \mathbin{||} L_2 = \{ w \in u \mathbin{||} v : u \in L_1,\ v \in L_2 \}$.

Note that by definition, a shuffle of two finite words is always fair: $u \mathbin{||} v = u \mathbin{|||} v$ whenever u and v are finite words. On the other hand, if at least one among u and v is infinite, then $u \mathbin{|||} v \subseteq u \mathbin{||} v$ and this inclusion may be strict, as can be concluded from the following example.

Example 2. The word ab is a shuffle of a and b and $a \mathbin{||} b = \{ab, ba\}$, $a^2 \mathbin{||} b = \{a^2b, aba, ba^2\}$; in general $a^n \mathbin{||} b = \{ a^i b a^j : i, j \geq 0,\ i + j = n \}$. Note that every shuffle in $a^n \mathbin{||} b$ is fair. Also $a^\omega \mathbin{|||} b = \{ a^i b a^\omega : i \geq 0 \}$ consists of fair shuffles only, but $a^\omega \mathbin{||} b = (a^\omega \mathbin{|||} b) \cup a^\omega$. Note that also for infinite words it may be the case that all shuffles are fair shuffles: $a^\omega \mathbin{|||} a = a^\omega \mathbin{||} a = a^\omega$.

It follows immediately from Definition 1 that both fair shuffling and shuffling are commutative operations.

Theorem 3. *Let $u, v \in \Delta^\infty$. Then $u \mathbin{|||} v = v \mathbin{|||} u$ and $u \mathbin{||} v = v \mathbin{||} u$.*

The next observation is also easily proved. It describes the structure of (fair) shuffles and it can be used as a recursive definition for shuffles of finite words (see, *e.g.*, [6,20]).

Lemma 4. *Let $u, v \in \Delta^\infty$ and let $a, b \in \Delta$. Then*

(1) $u \mathbin{||} \lambda = u \mathbin{|||} \lambda = u = \lambda \mathbin{|||} u = \lambda \mathbin{||} u$ *and*
(2) $au \mathbin{|||} bv = a(u \mathbin{|||} bv) \cup b(au \mathbin{|||} v)$ *and* $au \mathbin{||} bv = a(u \mathbin{||} bv) \cup b(au \mathbin{||} v)$.

As an intermediate result, we obtain that any concatenation of (fair) shuffles is a (fair) shuffle of concatenations. In particular, any shuffle of prefixes of two words is a prefix of a (fair) shuffle of these words.

Lemma 5. *Let $u, v \in \Delta^\infty$ and let $z, u', v' \in \Delta^*$. Then*

(1) $z(u \mathbin{|||} v) \subseteq zu \mathbin{|||} v$ *and* $z(u \mathbin{||} v) \subseteq zu \mathbin{||} v$, *and*
(2) $(u' \mathbin{||} v')(u \mathbin{|||} v) \subseteq u'u \mathbin{|||} v'v$ *and* $(u' \mathbin{||} v')(u \mathbin{||} v) \subseteq u'u \mathbin{||} v'v$.

Proof. (1) We only prove the first inclusion. The other proof is analogous. Let $w \in z(u \mathbin{|||} v)$. Then $w = zw'$ for some $w' \in u \mathbin{|||} v$. By Definition 1(1), $w' = u_1 v_1 u_2 v_2 \cdots$, with $u_i, v_i \in \Delta^*$ for all $i \geq 1$, $u = u_1 u_2 \cdots$, and $v = v_1 v_2 \cdots$. Thus $w = zw' = zu_1 v_1 u_2 v_2 \cdots$ with $zu_1 u_2 \cdots = zu$. Hence $w \in zu \mathbin{|||} v$.

(2) We only prove the first inclusion. The other proof is analogous. First assume $u' = \lambda$. Then $u' \mathbin{||} v' = v'$ by Lemma 4(1). From Theorem 3 and (1) we have $v'(u \mathbin{|||} v) \subseteq u \mathbin{|||} v'v$. The case that $v' = \lambda$ is symmetric. We proceed by induction on $|u'| + |v'|$. The cases that $|u'| = 0$ or $|v'| = 0$ have already been dealt with. We thus assume that $u' = au_1$ and $v' = bv_1$ with $a, b \in \Delta$ and $u_1, v_1 \in \Delta^*$. Then, by Lemma 4(2), $u' \mathbin{||} v' = au_1 \mathbin{||} bv_1 = a(u_1 \mathbin{||} bv_1) \cup b(au_1 \mathbin{||} v_1)$. This yields $(u' \mathbin{||} v')(u \mathbin{|||} v) = a(u_1 \mathbin{||} bv_1)(u \mathbin{|||} v) \cup b(au_1 \mathbin{||} v_1)(u \mathbin{|||} v) \subseteq a(u_1 u \mathbin{|||} bv_1 v) \cup b(au_1 u \mathbin{|||} v_1 v) \subseteq (au_1 u \mathbin{|||} bv_1 v) = (u'u \mathbin{|||} v'v)$ by applying the induction hypothesis and Lemma 4(2) twice. □

Note that the converses of the inclusions in the statement of this lemma do not hold. As an example, consider $cab \in (ab \mathbin{||} c) \setminus a(b \mathbin{||} c)$.

In addition, as we prove next, every prefix of a shuffle of two words is a fair shuffle of prefixes of these words. Consequently, the shuffles and the fair shuffles of two words determine the same set of prefixes.

Theorem 6. *Let $u, v \in \Delta^\infty$. Then*

$$pref(u) \mathbin{||} pref(v) = pref(u \mathbin{|||} v) = pref(u \mathbin{||} v) = pref(u) \mathbin{|||} pref(v).$$

Proof. From Lemma 5(2), we know that $pref(u) \mathbin{||} pref(v) \subseteq pref(u \mathbin{|||} v)$. Since $u \mathbin{|||} v \subseteq u \mathbin{||} v$ by Definition 1, it follows that $pref(u \mathbin{|||} v) \subseteq pref(u \mathbin{||} v)$ and $pref(u) \mathbin{|||} pref(v) \subseteq pref(u) \mathbin{||} pref(v)$. Hence the proof is complete once we have shown that $pref(u \mathbin{||} v) \subseteq pref(u) \mathbin{|||} pref(v)$. Let $z \in pref(u \mathbin{||} v)$. This implies that there exist an $n \geq 1$ and $u_1, u_2, \ldots, u_n, v_1, v_2, \ldots, v_n \in \Delta^*$ such that $z = u_1 v_1 u_2 v_2 \cdots u_{n-1} v_{n-1} x$ with $x \in pref(u_n v_n)$, $u_1 u_2 \cdots u_n \in pref(u)$, and $v_1 v_2 \cdots v_n \in pref(v)$. Thus clearly $z \in pref(u) \mathbin{|||} pref(v)$. □

Example 7. While $a^\omega \mathbin{|||} b \neq a^\omega \mathbin{||} b$, we have $pref(a^\omega \mathbin{|||} b) = pref(a^\omega \mathbin{||} b) = \{\, a^i b a^\omega : i \geq 0 \,\} \cup a^*$.

3. Associativity of fair shuffling

In this section, the associativity of fair shuffling is proved: $u \mathbin{|||} (v \mathbin{|||} w) = (u \mathbin{|||} v) \mathbin{|||} w$ for all words u, v, and w. Extending a technique known from, *e.g.* [15,20,25,26], to infinite words makes it possibly to prove rather directly that fair shuffling is associative. This technique is based on renaming and inserting: with each word we associate its own (indexed) alphabet and rename its letters accordingly. Next arbitrary (finite) subwords over the other indexed alphabet are inserted to simulate shuffles with arbitrary words over the other indexed alphabet. Then we intersect the resulting sets: all words in the intersection are (fair) shuffles of the renamed words. Hence, to obtain all (fair) shuffles, it is sufficient to ultimately simply go back to the original alphabets.

To formalize all this, we use homomorphisms and their extension to infinite words. Let $h : \Sigma \to \Gamma^*$ be a function assigning to each letter of alphabet 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 $x, y \in \Sigma^*$. 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 for all $i \geq 1$, $v_i \leq v_{i+1}$. Note that this is well-defined, since $v_i \leq v_{i+1}$ implies $h(v_i) \leq h(v_{i+1})$. Moreover, observe that for a word $u \in \Delta^\infty$, $h(pref(u)) = pref(h(u))$ whenever $h(a) \in \Gamma \cup \{\lambda\}$ for all $a \in alph(u)$.

Let Δ be an alphabet. For each integer $i \in \mathbb{N}$ and each $a \in \Delta$ we let $[a, i]$ be a distinct symbol. Let $[\Delta, i] = \{[a, i] : a \in \Delta\}$. Thus for all $i, j \in \mathbb{N}$ such that $i \neq j$, $[\Delta, i]$ and $[\Delta, j]$ are disjoint. We moreover assume that Δ and $[\Delta, i]$ are disjoint for all i. The homomorphisms $\beta_i : \Delta^* \to [\Delta, i]^*$ and $\overline{\beta}_i : [\Delta, i]^* \to \Delta^*$ are defined by $\beta_i(a) = [a, i]$ and $\overline{\beta}_i([a, i]) = a$, respectively. Note that β_i and $\overline{\beta}_i$ are renamings (bijections): β_i uniquely labels every letter in a word with i and $\overline{\beta}_i$ can be used to remove this label again. Now let $i \in \mathbb{N}$ and $J \subseteq \mathbb{N}$ be such that $i \notin J$. We define $\varphi_{i,J} : (\bigcup\{\,[\Delta, j] : j \in \{i\} \cup J\,\})^* \to \Delta^*$ by $\varphi_{i,J}([a, i]) = a$ and $\varphi_{i,J}([a, j]) = \lambda$, for all $j \in J$. Furthermore, we have $\psi_J : (\bigcup\{\,[\Delta, j] : j \in J\,\})^* \to \Delta^*$ defined by $\psi_J([a, j]) = a$, for all $j \in J$. Note that $\varphi_{i,\varnothing} = \overline{\beta}_i$ and $\psi_{\{j\}} = \overline{\beta}_j$. Intuitively, $\varphi_{i,J}$ is used to remove the label i from every letter in a word that is labelled by i and to erase every other symbol from that word, whereas ψ_J simply removes all labels in J from every letter in a word that is labelled by such a label from J.

We begin with the result announced above, which provides an alternative definition of the fair shuffle.

Theorem 8. *Let $u, v \in \Delta^\infty$. Then, for all $i, j \in \mathbb{N}$ such that $i \neq j$, $u \mid\mid\mid v = \psi_{\{i,j\}}(\varphi^{-1}_{i,\{j\}}(u) \cap \varphi^{-1}_{j,\{i\}}(v))$.*

Proof. The inclusion from left to right is easy. To prove the reverse inclusion, we only consider the case that $u, v \in \Delta^\omega$. The proofs of the other cases are similar. Without loss of generality, we assume that $i = 1$ and $j = 2$.

Let $w \in \psi_{\{1,2\}}(\varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v))$ and $\overline{w} \in \varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v)$ be such that $\psi_{\{1,2\}}(\overline{w}) = w$. As $\varphi_{1,\{2\}}(\overline{w}) = u$, there exist $x_1, x_2, \ldots \in \Delta^*$ and $u_1, u_2, \ldots \in \Delta^+$ such that $\overline{w} = \beta_2(x_1)\beta_1(u_1)\beta_2(x_2)\beta_1(u_2)\cdots$ and $u = u_1u_2\cdots$. Similarly, $\varphi_{2,\{1\}}(\overline{w}) = v$ implies that there exist $y_1, y_2, \ldots \in \Delta^*$ and $v_1, v_2, \ldots \in \Delta^+$ such that $\overline{w} = \beta_1(y_1)\beta_2(v_1)\beta_1(y_2)\beta_2(v_2)\cdots$ and $v = v_1v_2\cdots$. Hence $\beta_2(x_1)\beta_1(u_1)\beta_2(x_2)\beta_1(u_2)\cdots = \beta_1(y_1)\beta_2(v_1)\beta_1(y_2)\beta_2(v_2)\cdots$. Because we have $[\Delta, 1] \cap [\Delta, 2] = \varnothing$, it must be the case that either $\beta_2(x_1) = \lambda$ or $\beta_1(y_1) = \lambda$.

First assume $\beta_2(x_1) = \lambda$, *i.e.* $x_1 = \lambda$. Hence $\beta_1(u_1)\beta_2(x_2)\beta_1(u_2)\beta_2(x_3)\cdots = \beta_1(y_1)\beta_2(v_1)\beta_1(y_2)\beta_2(v_2)\cdots$. Again by $[\Delta, 1] \cap [\Delta, 2] = \varnothing$, and from the fact that $u_i, v_i \in \Delta^+$ for all $i \geq 1$, we know that $\beta_1(u_i) = \beta_1(y_i)$ and $\beta_2(v_i) = \beta_2(x_{i+1})$ for all $i \geq 1$. Thus $w = \psi_{\{1,2\}}(\overline{w}) = u_1v_1u_2v_2\cdots \in u \mid\mid\mid v$.

The case that $\beta_1(y_1) = \lambda$ is treated analogously. □

This alternative definition makes it possible to derive a symmetric description for the case that a word u is fairly shuffled with the fair shuffles $v \mid\mid\mid w$ of words v and w.

Lemma 9. *Let $u, v, w \in \Delta^\infty$. Let $i_1, i_2, i_3 \in \mathbb{N}$ be three different integers and let $j \in \mathbb{N}$ be such that $j \neq i_1$. Then*

$$\psi_{\{i_1,j\}}(\varphi^{-1}_{i_1,\{j\}}(u) \cap \varphi^{-1}_{j,\{i_1\}}(\psi_{\{i_2,i_3\}}(\varphi^{-1}_{i_2,\{i_3\}}(v) \cap \varphi^{-1}_{i_3,\{i_2\}}(w))))$$
$$= \psi_{\{i_1,i_2,i_3\}}(\varphi^{-1}_{i_1,\{i_2,i_3\}}(u) \cap \varphi^{-1}_{i_2,\{i_1,i_3\}}(v) \cap \varphi^{-1}_{i_3,\{i_1,i_2\}}(w)).$$

Proof. Without loss of generality, we assume that $i_k = k$ for $1 \leq k \leq 3$, and $j \neq 1$.

($\subseteq$) Let $z \in \psi_{\{1,j\}}(\varphi^{-1}_{1,\{j\}}(u) \cap \varphi^{-1}_{j,\{1\}}(\psi_{\{2,3\}}(\varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w))))$ and $\overline{z} \in \varphi^{-1}_{1,\{j\}}(u) \cap \varphi^{-1}_{j,\{1\}}(\psi_{\{2,3\}}(\varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w)))$ be such that $\psi_{\{1,j\}}(\overline{z}) = z$. Let $x \in \psi_{\{2,3\}}(\varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w))$ be such that $\overline{z} \in \varphi^{-1}_{1,\{j\}}(u) \cap \varphi^{-1}_{j,\{1\}}(x)$. Let $\overline{x} \in \varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w)$ be such that $\psi_{\{2,3\}}(\overline{x}) = x$. Hence $\overline{x}$ is of the form $\overline{x} = b_1c_1b_2c_2\cdots$ such that for all $i \geq 1$, $b_i \in [\Delta, 2] \cup \{\lambda\}$ and $c_i \in [\Delta, 3] \cup \{\lambda\}$, $\overline{\beta}_2(b_1b_2\cdots) = v$, and $\overline{\beta}_3(c_1c_2\cdots) = w$. Furthermore $\overline{z}$ is of the form $\overline{z} = a_1\overline{b}_1\overline{c}_1a_2\overline{b}_2\overline{c}_2\cdots$ such that for all $i \geq 1$, $a_i \in [\Delta, 1] \cup \{\lambda\}$ and $\overline{b}_i, \overline{c}_i \in [\Delta, j] \cup \{\lambda\}$, $\overline{\beta}_1(a_1a_2\cdots) = u$, and $\overline{\beta}_j(\overline{b}_1\overline{c}_1\overline{b}_2\overline{c}_2\cdots) = \psi_{\{2,3\}}(b_1c_1b_2c_2\cdots)$ is such that $\overline{\beta}_j(\overline{b}_1\overline{b}_2\cdots) = \overline{\beta}_2(b_1b_2\cdots) = v$ and $\overline{\beta}_j(\overline{c}_1\overline{c}_2\cdots) = \overline{\beta}_3(c_1c_2\cdots) = w$. Now consider that $\overline{\overline{z}} = a_1\beta_2(\overline{\beta}_j(\overline{b}_1))\beta_3(\overline{\beta}_j(\overline{c}_1))a_2\beta_2(\overline{\beta}_j(\overline{b}_2))\beta_3(\overline{\beta}_j(\overline{c}_2))\cdots$. Since $\overline{\beta}_1(a_1a_2\cdots) = u$, $\overline{\beta}_2(\beta_2(\overline{\beta}_j(\overline{b}_1))\beta_2(\overline{\beta}_j(\overline{b}_2))\cdots) = \overline{\beta}_j(\overline{b}_1\overline{b}_2\cdots) = v$, and $\overline{\beta}_3(\beta_3(\overline{\beta}_j(\overline{c}_1))\beta_3(\overline{\beta}_j(\overline{c}_2))\cdots) = \overline{\beta}_j(\overline{c}_1\overline{c}_2\cdots) = w$, we know that $\varphi_{1,\{2,3\}}(\overline{\overline{z}}) = u$, $\varphi_{2,\{1,3\}}(\overline{\overline{z}}) = v$, and $\varphi_{3,\{1,2\}}(\overline{\overline{z}}) = w$. Hence $\overline{\overline{z}} \in \varphi^{-1}_{1,\{2,3\}}(u) \cap \varphi^{-1}_{2,\{1,3\}}(v) \cap \varphi^{-1}_{3,\{1,2\}}(w)$ and $\psi_{\{1,2,3\}}(\overline{\overline{z}}) = \psi_{\{1,j\}}(\overline{z}) = z$.

($\supseteq$) Let $z \in \psi_{\{1,2,3\}}(\varphi^{-1}_{1,\{2,3\}}(u) \cap \varphi^{-1}_{2,\{1,3\}}(v) \cap \varphi^{-1}_{3,\{1,2\}}(w))$ and $\overline{z} \in \varphi^{-1}_{1,\{2,3\}}(u) \cap \varphi^{-1}_{2,\{1,3\}}(v) \cap \varphi^{-1}_{3,\{1,2\}}(w)$ be such that $\psi_{\{1,2,3\}}(\overline{z}) = z$. Hence $\overline{z}$ is of the form $\overline{z} = a_1b_1c_1a_2b_2c_2\cdots$ such that for all $i \geq 1$, $a_i \in [\Delta, 1] \cup \{\lambda\}$, $b_i \in [\Delta, 2] \cup \{\lambda\}$, and $c_i \in [\Delta, 3] \cup \{\lambda\}$, $\overline{\beta}_1(a_1a_2\cdots) = u$, $\overline{\beta}_2(b_1b_2\cdots) = v$, and $\overline{\beta}_3(c_1c_2\cdots) = w$. Let $\overline{u} = a_1\alpha_1a_2\alpha_2\cdots$, with $\alpha_i \in ([\Delta, j] \cup \{\lambda\})^*$, be such that for all $i \geq 1$, $\overline{\beta}_j(\alpha_i) = \psi_{\{2,3\}}(b_ic_i)$. Then clearly $\overline{u} \in \varphi^{-1}_{1,\{j\}}(u)$. Next, let $\overline{x} = b_1c_1b_2c_2\cdots$. Then $\overline{x} \in \varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w)$. Since for all $i \geq 1$, $\varphi_{j,\{1\}}(\alpha_i) = \overline{\beta}_j(\alpha_i) = \psi_{\{2,3\}}(b_ic_i)$ and $a_i \in [\Delta, 1] \cup \{\lambda\}$, it follows that $\overline{u} \in \varphi^{-1}_{j,\{1\}}(\psi_{\{2,3\}}(\overline{x}))$. Thus $\overline{u} \in \varphi^{-1}_{1,\{j\}}(u) \cap \varphi^{-1}_{j,\{1\}}(\psi_{\{2,3\}}(\overline{x}))$. Finally, the fact that for all $i \geq 1$, $\overline{\beta}_j(\alpha_i) = \psi_{\{2,3\}}(b_ic_i)$ now implies that $\psi_{\{1,j\}}(\overline{u}) = \psi_{\{1,2,3\}}(\overline{z}) = z$. □

With this lemma, it is now straightforward to prove that the fair shuffling of possibly infinite words is associative, a result which is mentioned in [22] (where fair shuffling is called fair merging), but which is not proved there due to the complications caused by a different setting.

Theorem 10. *Let $u, v, w \in \Delta^\infty$. Then $u \mid\mid\mid (v \mid\mid\mid w) = (u \mid\mid\mid v) \mid\mid\mid w$.*

Proof. By Theorem 8 and Lemma 9, we obtain that $u \mid\mid\mid (v \mid\mid\mid w) = \psi_{\{1,4\}}(\varphi^{-1}_{1,\{4\}}(u) \cap \varphi^{-1}_{4,\{1\}}(\psi_{\{2,3\}}(\varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w)))) = \psi_{\{1,2,3\}}(\varphi^{-1}_{1,\{2,3\}}(u) \cap \varphi^{-1}_{2,\{1,3\}}(v) \cap \varphi^{-1}_{3,\{1,2\}}(w))$. Likewise $(u \mid\mid\mid v) \mid\mid\mid w = \psi_{\{3,4\}}(\varphi^{-1}_{4,\{3\}}(\psi_{\{1,2\}}(\varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v))) \cap \varphi^{-1}_{3,\{4\}}(w)) = \psi_{\{1,2,3\}}(\varphi^{-1}_{1,\{2,3\}}(u) \cap \varphi^{-1}_{2,\{1,3\}}(v) \cap \varphi^{-1}_{3,\{1,2\}}(w))$. Hence $u \mid\mid\mid (v \mid\mid\mid w) = (u \mid\mid\mid v) \mid\mid\mid w$. □

Since for finite words shuffles and fair shuffles are the same, this theorem implies that shuffling is associative for finite words. This is a well-known fact (see, *e.g.*, [8,12,15,20,24,26,27]) which we state here explicitly for completeness' sake and for future reference.

Corollary 11. *Let* $u, v, w \in \Delta^*$. *Then* $u \mid\mid (v \mid\mid w) = (u \mid\mid v) \mid\mid w$.

Theorem 8 supplies an alternative definition for *fair* shuffles only, since the inverse homomorphisms used to insert subwords are applied to the complete words to be shuffled. To extend this theorem to the general case, we would have to consider also the prefixes of one word in case the other word is infinite. Because of this case distinction, this would lead to a less uniform description for shuffles than we now have for fair shuffles. Rather than proving associativity on basis of such an alternative definition or by further investigating the implications of the associativity of fair shuffling, we will present in the next section a more general approach based on prefix properties. We will express shuffles as limits of shuffles of finite words, which should then allow us to apply the associativity of the shuffling of finite words (Corollary 11).

4. General shuffles

In this section we will prove that a word is a shuffle of two given words if and only if each of its prefixes is a shuffle of prefixes of these two words. One direction is a consequence of Theorem 6. We now set out to prove the other direction, namely that $w \in u \mid\mid v$ whenever $\mathit{pref}(w) \subseteq \mathit{pref}(u) \mid\mid \mathit{pref}(v)$.

As before, for $i \in \{1, 2\}$, the homomorphisms $\beta_i : \Delta^* \to [\Delta, i]^*$ and $\psi_{\{1,2\}} : ([\Delta, 1] \cup [\Delta, 2])^* \to \Delta^*$ are defined by $\beta_i(a) = [a, i]$ and $\psi_{\{1,2\}}([a, i]) = a$, for all $a \in \Delta$. Let $w \in \Delta^*$. We will refer to any word $d \in \psi^{-1}_{\{1,2\}}(w)$ as a *decomposition* of w (as a shuffle of the words u and v) if $d \in \beta_1(u) \mid\mid \beta_2(v)$. This terminology is justified by the following observation, which is an immediate consequence of Theorem 8.

Corollary 12. *Let* $u, v \in \Delta^*$. *Then* $u \mid\mid v = \psi_{\{1,2\}}(\beta_1(u) \mid\mid \beta_2(v))$.

Proof. This follows directly from Theorem 8 because $\varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v) = \beta_1(u) \mid\mid\mid \beta_2(v) = \beta_1(u) \mid\mid \beta_2(v)$. □

In other words, $w \in u \mid\mid v$ if and only if $\psi^{-1}_{\{1,2\}}(w) \cap (\beta_1(u) \mid\mid \beta_2(v)) \neq \varnothing$. Hence every shuffle $w \in u \mid\mid v$ has at least one decomposition, *i.e.* an explicit description of how w can be obtained as a shuffle of u and v. It is not difficult to see that a shuffle may have several decompositions. In a series of papers (see, *e.g.*, [19,20]) Mateescu et al. use so-called 'trajectories' to describe shuffles. A trajectory defines, in a binary fashion, when to switch from one word to another. When applied, a trajectory thus defines a unique decomposition. Properties like associativity are consequently discussed per set of trajectories.

We would now like to show that whenever every prefix of a (possibly infinite) word w can be obtained as a shuffle of a prefix of a word u and a prefix of a word v, then w is indeed a shuffle of u and v. To prove this, it would be convenient if every decomposition describing a prefix of w as a shuffle of prefixes of u and v could always be prolonged and ultimately lead to w as a shuffle of u and v. Unfortunately, in general, this is not true. The following example illustrates this. It even shows that an infinite word may have infinitely many prefixes with non-prolongable decompositions.

Example 13. Let $u = (a^3b)^\omega$ and let $v = b^\omega$. Clearly $\{a^3, a^3b\} \subseteq \mathit{pref}(u)$, $\{b^2, b^3\} \subseteq \mathit{pref}(v)$, and $w = a^3b^3 \in \mathit{pref}(u) \mid\mid \mathit{pref}(v)$. Then $d = a_1a_1a_1b_2b_2b_2$ and $d' = a_1a_1a_1b_1b_2b_2$ are two decompositions of w.

Next consider $w' = wa = a^3b^3a \in \mathit{pref}(u) \mid\mid \mathit{pref}(v)$. The only decompositions of w' as a shuffle of prefixes of u and v are $e = a_1a_1a_1b_1b_2b_2a_1$ and $e' = a_1a_1a_1b_2b_2b_1a_1$. Clearly, d neither precedes e nor e'. Note, however, that d' can be prolonged to e.

Finally, let $j \geq 0$, $u_j = a^3(ba^3)^j \in \mathit{pref}(u)$, and $v_j = b^3(b^3)^j \in \mathit{pref}(v)$. Then clearly $w_j = (a^3b^4)^j a^3b^3 \in \mathit{pref}(u) \mid\mid \mathit{pref}(v)$ and $w'_j = w_ja = (a^3b^4)^j a^3b^3a \in \mathit{pref}(u) \mid\mid \mathit{pref}(v)$. Note that $d_j = (a_1a_1a_1b_1b_2b_2b_2)^j a_1a_1a_1b_2b_2b_2$ is a decomposition of w_j as a shuffle of u_j and v_j. By the same reasoning as in the case $j = 0$ above, it is however easy to see that d_j cannot be prolonged to a decomposition of w'_j.

As is immediate from this example, it is a problem that u and v may have letters in common which may give rise to multiple decompositions. Indeed, as we will demonstrate next, the disjointness of the alphabets of u and v guarantees the prolongability of the (unique) decomposition of each word in $\mathit{pref}(u) \,||\, \mathit{pref}(v)$. In fact, we can prove a more general result, by not just considering words, but *limit-closed* languages. Limit-closedness guarantees that the infinitary part of a language is characterized by its finite prefixes. This notion has been defined in many disguises throughout the literature on theoretical computer science. The oldest reference we found is [1], where the terminology used is a 'closed process', while the term *limit closure* was coined in [11] — after initially referring to the same concept as a 'König closure' in its preceding technical report.

Definition 14. Let $K \subseteq \Delta^\infty$. K is *limit-closed* if for all $w_1 \le w_2 \le \cdots \in \mathit{pref}(K)$, $\lim_{n\to\infty} w_n \in K \cup \mathit{pref}(K)$.

Example 15. All singleton languages $\{u\}$ as well as all finite languages $L_n = \{\lambda, a, \ldots, a^n\}$ over a unary alphabet are limit-closed, whereas a^* and a^*b^ω are not, because $\lim_{n\to\infty} a^n = a^\omega \notin a^* \cup a^*b^\omega$. However, each of the languages $a^*b^\omega \cup a^\omega$, $a^* \cup a^\omega$ and a^ω is limit-closed.

Lemma 16. *Let $K, L \subseteq \Delta^\infty$ be limit-closed such that $\mathit{alph}(K) \cap \mathit{alph}(L) = \varnothing$ and let $w \in \Delta^\omega$. Then $\mathit{pref}(w) \subseteq \mathit{pref}(K) \,||\, \mathit{pref}(L)$ implies that $w \in K \,||\, L$.*

Proof. Let $\Delta_1 = \mathit{alph}(K)$ and $\Delta_2 = \mathit{alph}(L)$. Define, for $i \in \{1, 2\}$, the homomorphisms $\varphi_i : \Delta^* \to \Delta_i^*$ by $\varphi_i(a) = a$ if $a \in \Delta_i$ and $\varphi_i(a) = \lambda$ otherwise.

Assume that $\mathit{pref}(w) \subseteq \mathit{pref}(K) \,||\, \mathit{pref}(L)$. Let $w' \in \mathit{pref}(w)$. Then there exist $x \in \mathit{pref}(K)$ and $y \in \mathit{pref}(L)$ such that $w' \in x \,||\, y$. Since $\mathit{alph}(K) \cap \mathit{alph}(L) = \varnothing$, it follows that x and y are unique: $x = \varphi_1(w')$ and $y = \varphi_2(w')$. Because $w \in \Delta^\omega$, it has infinitely many prefixes $w[1] < w[2] < \cdots$ and $w = \lim_{n\to\infty} w[n]$. Thus $\varphi_i(w[1]) \le \varphi_i(w[2]) \le \cdots$ for $i \in \{1, 2\}$. Hence $u = \lim_{n\to\infty} \varphi_1(w[n])$ and $v = \lim_{n\to\infty} \varphi_2(w[n])$ exist. Moreover, by the limit-closedness of K and L, we know that $u \in K$ and $v \in L$. Consequently, once $w \in u \,||\, v$ has been proved, it follows that $w \in K \,||\, L$ and we are done.

First we observe that $\mathit{pref}(w) \subseteq \mathit{pref}(u) \,||\, \mathit{pref}(v)$ because for every $w' \in \mathit{pref}(w)$, we have seen that $w' \in \varphi_1(w') \,||\, \varphi_2(w')$. Moreover, $\varphi_1(\mathit{pref}(w)) \subseteq \varphi_1(\mathit{pref}(u) \,||\, \mathit{pref}(v)) = \varphi_1(\mathit{pref}(u)) \,||\, \varphi_1(\mathit{pref}(v)) = \mathit{pref}(u) \,||\, \lambda = \mathit{pref}(u)$ and likewise $\varphi_2(\mathit{pref}(w)) \subseteq \mathit{pref}(v)$. Since $w \in \Delta^\omega$, there are only three cases to be considered. Case 1: $\varphi_1(w) \in \Delta_1^\omega$ and $\varphi_2(w) \in \Delta_2^*$. Then the fact that $\mathit{pref}(\varphi_1(w)) = \varphi_1(\mathit{pref}(w)) \subseteq \mathit{pref}(u)$ implies that $\varphi_1(w) = u$. Since $\varphi_2(\mathit{pref}(w)) \subseteq \mathit{pref}(v)$, it now follows that $w \in u \,||\, v$. Case 2: $\varphi_1(w) \in \Delta_1^*$ and $\varphi_2(w) \in \Delta_2^\omega$. Symmetrically to Case 1 it follows that $w \in u \,||\, v$. Case 3: $\varphi_1(w) \in \Delta_1^\omega$ and $\varphi_2(w) \in \Delta_2^\omega$. With similar reasoning as in Case 1, it can be seen that $\varphi_1(w) = u$ and $\varphi_2(w) = v$, which implies that $w \in u \,||\, v$. □

The notion of limit-closedness is closely related to that of *adherences* as introduced in [9]. The *adherence* of a language $L \in \Delta^*$ consists of all infinite words that can be obtained as a limit of words from L: $\mathit{adh}(L) = \{ w \in \Delta^\omega : \mathit{pref}(w) \subseteq \mathit{pref}(L) \}$. Thus $L \subseteq \Delta^\infty$ is limit-closed if and only if $\mathit{adh}(\mathit{pref}(L)) \subseteq L$. Properties of adherences can also be found in, *e.g.*, [29]. In order to extend Lemma 16 to the case of not necessarily disjoint alphabets we rely on Property 1 from [9], where it is shown – using *König's Lemma* – that any infinite language has a nonempty adherence.

Lemma 17. *Let $K, L \subseteq \Delta^\infty$ be limit-closed and let $w \in \Delta^\omega$. Then $\mathit{pref}(w) \subseteq \mathit{pref}(K) \,||\, \mathit{pref}(L)$ implies $w \in K \,||\, L$.*

Proof. Assume that $\mathit{pref}(w) \subseteq \mathit{pref}(K) \,||\, \mathit{pref}(L)$. By Corollary 12, we have that for all $x \in \mathit{pref}(w)$, there exist $u_x \in K$ and $v_x \in L$ such that $\psi_{\{1,2\}}^{-1}(x) \cap (\beta_1(\mathit{pref}(u_x)) \,||\, \beta_2(\mathit{pref}(v_x))) \neq \varnothing$. Since w has infinitely many prefixes, we see that $R = \psi_{\{1,2\}}^{-1}(\mathit{pref}(w)) \cap (\beta_1(\mathit{pref}(K)) \,||\, \beta_2(\mathit{pref}(L))) = \mathit{pref}(\psi_{\{1,2\}}^{-1}(w)) \cap (\mathit{pref}(\beta_1(K)) \,||\, \mathit{pref}(\beta_2(L)))$ is infinite. Thus (by Property 1 from [9]) there exists a $w' \in \Delta^\omega$ such that $\mathit{pref}(w') \subseteq R$. This implies that $\mathit{pref}(w') \subseteq \mathit{pref}(\psi_{\{1,2\}}^{-1}(w))$. Hence $\mathit{pref}(\psi_{\{1,2\}}(w')) = \psi_{\{1,2\}}(\mathit{pref}(w')) \subseteq \psi_{\{1,2\}}(\mathit{pref}(\psi_{\{1,2\}}^{-1}(w))) = \mathit{pref}(w)$ and so $\psi_{\{1,2\}}(w') = w$. On the other hand, it also follows that $\mathit{pref}(w') \subseteq \mathit{pref}(\beta_1(K)) \,||\, \mathit{pref}(\beta_2(L))$. From Lemma 16 we obtain $w' \in \beta_1(K) \,||\, \beta_2(L)$. Whence $w = \psi_{\{1,2\}}(w') \in \psi_{\{1,2\}}(\beta_1(K) \,||\, \beta_2(L)) = K \,||\, L$, as desired. □

The statement of this lemma in general does not hold when either K or L is not limit-closed.

Example 18. Let $K = a^*$ and let $L = \{\lambda\}$. Then $\mathit{pref}(a^\omega) = a^* = \mathit{pref}(K) \,||\, \mathit{pref}(L)$, but $a^\omega \notin a^* = K \,||\, L$.

Since singleton languages are limit-closed, we directly obtain as a corollary the desired result.

Corollary 19. *Let $u, v \in \Delta^\infty$ and let $w \in \Delta^\omega$. Then*

$$pref(w) \subseteq pref(u) \,||\, pref(v) \text{ implies } w \in u \,||\, v.$$

It must be noted here that this result does not hold for fair shuffles.

Example 20. Whereas we have $pref(a^\omega) = a^*$ and $a^* \subseteq pref(a^\omega) \,|||\, pref(b) = pref(a^\omega) \,||\, pref(b)$, we have seen in Example 2 that $a^\omega \notin a^\omega \,|||\, b$.

To conclude, Theorem 6, Lemma 17, and Corollary 19 together characterize the shuffles of two words (limit-closed languages) as exactly the limits of the shuffles of the prefixes of these words (languages).

Theorem 21. *Let $u, v \in \Delta^\infty$, let $K, L \subseteq \Delta^\infty$ be limit-closed, and let $w \in \Delta^\omega$. Then*

(1) *$w \in u \,||\, v$ if and only if $pref(w) \subseteq pref(u) \,||\, pref(v)$, and*
(2) *$w \in K \,||\, L$ if and only if $pref(w) \subseteq pref(K) \,||\, pref(L)$.*

We need one more observation in order to conclude that shuffling is associative: we now show that the shuffles of two words form a limit-closed language, or more generally, that the family of limit-closed languages is closed under shuffling.

Lemma 22. *Let $u, v \in \Delta^\infty$ and let $K, L \subseteq \Delta^\infty$ be limit-closed. Then*

(1) *$u \,||\, v$ is limit-closed and*
(2) *$K \,||\, L$ is limit-closed.*

Proof. It suffices to prove the second statement. Let $y_1 \leq y_2 \leq \cdots \in pref(K \,||\, L)$ and let $y = \lim_{n\to\infty} y_n$. Since for all $x \in pref(y)$ there exists an $i \geq 0$ such that $x \in pref(y_i) \in pref(pref(K \,||\, L)) = pref(K \,||\, L)$, it follows that $pref(y) \subseteq pref(K \,||\, L)$. We distinguish two cases. If $y \in \Delta^*$, then $y \in pref(K \,||\, L)$. If $y \in \Delta^\omega$, then by Theorem 21(2), $y \in K \,||\, L$. Hence $y \in K \,||\, L \cup pref(K \,||\, L)$, and $K \,||\, L$ is thus limit-closed. □

Theorem 23. *Let $u, v, w \in \Delta^\infty$. Then*

$$u \,||\, (v \,||\, w) = (u \,||\, v) \,||\, w.$$

Proof. If u, v, w are finite words, then we have Corollary 11.

If at least one of them is infinite, then both $u \,||\, (v \,||\, w)$ and $(u \,||\, v) \,||\, w$ consist of infinite words only. Let $x \in u \,||\, (v \,||\, w)$. Then Theorem 21(2) and Lemma 22(1) together imply that $pref(x) \subseteq pref(u) \,||\, pref(v \,||\, w)$. Hence, by Theorem 6, $pref(x) \subseteq pref(u) \,||\, (pref(v) \,||\, pref(w))$. Thus, by Corollary 11, $pref(x) \subseteq (pref(u) \,||\, pref(v)) \,||\, pref(w)$ and $pref(x) \subseteq pref(u \,||\, v) \,||\, pref(w)$ by Theorem 6. Finally, since $u \,||\, v$ and $\{w\}$ are limit-closed, Theorem 21(2) implies that $x \in (u \,||\, v) \,||\, w$. The converse inclusion follows from the above and Theorem 3. □

5. Discussion

In this paper, we have considered a general shuffling operation for possibly *infinite* words which is not necessarily fair, and we have studied its limit behaviour. This has led to a characterization of shuffles in terms of the shuffles of their prefixes, with the associativity of shuffling as an immediate corollary. This proof of the associativity of shuffling is fully self-contained and it does not rely on the sometimes vague or unsubstantiated claims made in the literature for related operations.

Associativity is of interest not only from a purely mathematical point of view. In fact, as mentioned in the Introduction, our motivation to study the associativity of shuffling stems from the use of shuffling and some of its variants to prove compositionality for different types of team automata [3,5]. Team automata consist of component automata that collaborate through synchronizations. These synchronizations can be freely chosen depending on the specific protocol of collaboration to be modelled. In [4], we have defined different strategies for choosing the synchronizations of a team automaton. To describe the behaviours of these team automata in terms of the behaviours of their components, several types of 'synchronized shuffling' have been introduced in [3,5]. The associativity of shuffling as defined in this paper, is the basis for proofs of the associativity of some variants of synchronized shuffling in the Ph.D. thesis of the first author [3]. The associativity of these variants, in their turn, is crucial to proving that several types of team automata satisfy compositionality in [3,5] (in the latter, only finitary behaviours are considered). Similarly, both shuffling and synchronized shuffling are used in [2] to express part of the compositional properties of reusable software components through explicit contracts based on behaviours.

Since the behaviours of team automata and their components are prefix-closed languages representing ongoing behaviours, we have focussed on the prefix properties of shuffles. As follows from Theorem 6, the shuffle operation is sound in the sense that indeed all prefixes of an infinite shuffle appear as shuffles of finite words (behaviours). In addition, the key Lemma 17 and its Corollary 19 show that every word which is represented through its finite prefixes in the shuffles of finite words is a shuffle of their limits (component behaviours). Together they provide a tool to investigate infinite shuffles as limits of finite shuffles. In a forthcoming paper, we intend to address similar issues for the more involved shuffles with synchronization.

Acknowledgements

This paper is dedicated to the memory of Alexandru Mateescu, the founding father of the theory of trajectories and a constant source of information on shuffling for the first author.

We thank the anonymous referees for their useful comments and additional references. We are especially grateful for the suggestions that have led to a more elegant and concise presentation of our main results.

Maurice H. ter Beek was partially supported by the EU IST-3-016004-IP-09 project SENSORIA.

References

[1] K. Abrahamson, Decidability and Expresiveness of Logics of Processes, Ph.D. Thesis, University of Washington, Seattle, 1980.
[2] A. Bailly, M. Clerbout, I. Simplot-Ryl, Component composition preserving behavioural contracts based on communication traces, in: Proceedings CIAA'05, in: LNCS, vol. 3845, Springer-Verlag, Berlin, 2006, pp. 55–66.
[3] 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.
[4] 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.
[5] M.H. ter Beek, J. Kleijn, Team automata satisfying compositionality, in: Proceedings FM'03, in: LNCS, vol. 2805, Springer-Verlag, Berlin, 2003, pp. 381–400.
[6] M.H. ter Beek, C. Martín-Vide, V. Mitrana, Synchronized shuffles, Theoretical Computer Science 341 (1–3) (2005) 263–275.
[7] J.A. Bergstra, A. Ponse, S.A. Smolka (Eds.), Handbook of Process Algebra, Elsevier Science Publishers, Amsterdam, 2001.
[8] S.L. Bloom, Z. Ésik, Free shuffle algebras in language varieties, Theoretical Computer Science 163 (1996) 55–98.
[9] L. Boasson, M. Nivat, Adherences of languages, Journal of Computer and System Sciences 20 (3) (1980) 285–309.
[10] R. De Simone, Langages infinitaires et produit de mixage, Theoretical Computer Science 31 (1984) 83–100.
[11] E.A. Emerson, Alternative semantics for temporal logics, Theoretical Computer Science 26 (1–2) (1983) 121–130. Cf. also Technical Report TR-182, Department of Computer Sciences, University of Texas, Austin, 1981.
[12] S. Ginsburg, Algebraic and Automata-Theoretic Properties of Formal Languages. Fundamental Studies in Computer Science, vol. 2, North-Holland Publishing Company, Amsterdam, 1975.
[13] S. Ginsburg, E.H. Spanier, Mappings of languages by two-tape devices, Journal of the ACM 12 (3) (1965) 423–434.
[14] J.L. Gischer, Shuffle languages, petri nets, and context sensitive grammars, Communications of the ACM 24 (1981) 597–605.
[15] M. Jantzen, The power of synchronizing operations on strings, Theoretical Computer Science 14 (1981) 127–154.
[16] T. Kimura, An algebraic system for process structuring and interprocess communication, in: Proceedings STOC'76, ACM Press, New York, 1976, pp. 92–100.
[17] M. Latteux, Y. Roos, Synchronized shuffle and regular languages, in: Jewels are Forever, Springer-Verlag, Berlin, 1999, pp. 35–44.
[18] M. Lothaire, Combinatorics on Words. Encyclopedia of Mathematics and its Applications, vol. 17, Cambridge University Press, Cambridge, 1983.
[19] A. Mateescu, G.D. Mateescu, Fair and associative infinite trajectories, in: Jewels are Forever, Springer-Verlag, Berlin, 1999, pp. 327–338.
[20] A. Mateescu, G. Rozenberg, A. Salomaa, Shuffle on trajectories: Syntactic constraints, Theoretical Computer Science 197 (1–2) (1998) 1–56.
[21] W.F. Ogden, W.E. Riddle, W.C. Rounds, Complexity of expressions allowing concurrency, in: Proceedings POPL'78, ACM Press, New York, 1978, pp. 185–194.
[22] D. Park, On the semantics of fair parallelism, in: Proceedings of the 1979 Copenhagen Winter School on Abstract Software Specifications, in: LNCS, vol. 86, Springer-Verlag, Berlin, 1980, pp. 504–526.
[23] R.R. Redziejowski, Associative Omega-products of traces, Fundamenta Informaticae 67 (1–3) (2005) 175–185.
[24] A.W. Roscoe, The Theory and Practice of Concurrency, Prentice Hall International Series in Computer Science, London, 1997.
[25] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Springer-Verlag, Berlin, 1997.

[26] J. Sakarovitch, I. Simon, Subwords, Chapter 6 in [18], 1983, pp. 105–142.
[27] A.C. Shaw, Software descriptions with flow expressions, IEEE Transactions on Software Engineering 4 (3) (1978) 242–254.
[28] J.L.A. van de Snepscheut, Trace Theory and VLSI Design. LNCS, vol. 200, Springer-Verlag, Berlin, 1985.
[29] L. Staiger, ω-languages, Chapter 6 in Volume 3 of [25], 1997, pp. 339–387.