Infinite Unfair Shuffles and Associativity*

Maurice H. ter Beek†, *Jetty Kleijn*‡

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.

1 Introduction

Shuffling 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 purely mathematical interest [5, 7, 8, 10–13, 15–17] and from its significance as a semantics for concurrent systems consisting of several components [2, 4, 6, 14, 18–20, 22, 23].

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 in case only finite words are involved [2, 4, 7, 10, 13, 17, 20, 22, 23], 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 [2, 4, 19]. There even exist variants of shuffling for which associativity does not hold [5, 8, 15–17] 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

*This work was partially supported by the EU IST-3-016004-IP-09 project SENSORIA.

†ISTI-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy, maurice.terbeek@isti.cnr.it

‡LIACS, Universiteit Leiden, P.O. Box 9512, 2300 RA Leiden, Netherlands, kleijn@liacs.nl

other word is chosen. Note that with this definition, a shuffle of two finite words is always a standard—fair—shuffle. 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 [2–4]. 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 [2] 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 as considered here, although there exist many references to the associativity of related shuffle operations [7, 10, 13, 17, 20, 22, 23]. We could thus try and 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, also 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.

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. By convention $\Delta \subseteq \Delta^+$. 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 $\mathrm{pref}(w) = \{u \in \Delta^* : u \leq w\}$. For a language K, $\mathrm{pref}(K) = \bigcup\{\mathrm{pref}(w) : w \in K\}$.

Both finite and infinite words can be defined as the limit of their prefixes. Let $v_1, v_2, \cdots \in \Delta^*$ 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 $u_1u_2\cdots$ to denote the word $\lim_{n\to\infty} u_1u_2\cdots u_n$.

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. This implies that one (infinite) word may prevail when the other word, from some point onwards, contributes nothing anymore but the trivial subword λ.

Definition 2.1 Let $u, v \in \Delta^\infty$. Then

(1) $w \in \Delta^\infty$ is a *fair shuffle* of u and v if $w = u_1v_1u_2v_2\cdots$, where $u_i, v_i \in \Delta^*$, for all $i \geq 1$, are such that $u = u_1u_2\cdots$ and $v = v_1v_2\cdots$, 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) $w = u_1v_1u_2v_2\cdots$, where $u_i, v_i \in \Delta^*$, for all $i \geq 1$, and either $u_1u_2\cdots \in \mathrm{pref}(u)$ and $v = v_1v_2\cdots \in \Delta^\omega$, or $u = u_1u_2\cdots \in \Delta^\omega$ and $v_1v_2\cdots \in \mathrm{pref}(v)$.

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

Shuffling two languages is defined element-wise: The *fair shuffle* of two languages L_1 and L_2 is denoted by $L_1 \,|||\, L_2$ 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 $L_1 \,|||\, L_2 = \{w \in u \,|||\, v : u \in L_1,\ v \in L_2\}$. Similarly, the *shuffle* of L_1 and L_2 is denoted by $L_1 \,||\, L_2$ and is defined as $L_1 \,||\, L_2 = \{w \in u \,||\, v : u \in L_1,\ v \in L_2\}$.

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

Example 2.2 The word ab is a shuffle of a and b and $a \,||\, b = \{ab, ba\}$, $a^2 \,||\, b = \{a^2b, aba, ba^2\}$; in general $a^n \,||\, b = \{a^iba^j : i, j \geq 0,\ i + j = n\}$. Note that every shuffle in $a^n \,||\, b$ is fair. Also $a^\omega \,|||\, b = \{a^iba^\omega : i \geq 0\}$ consists of fair shuffles only, but $a^\omega \,||\, b = (a^\omega \,|||\, b) \cup a^\omega$. Note that also for infinite words it may be the case that all shuffles are fair shuffles: $a^\omega \,|||\, a = a^\omega \,||\, a = a^\omega$.

It follows immediately from Definition 2.1 that both fair shuffling and shuffling are commutative operations.

Theorem 2.3 *Let $u, v \in \Delta^\infty$. Then $u \,|||\, v = v \,|||\, u$ and $u \,||\, v = v \,||\, u$.*

Also the next observation is easily proved. It describes the structure of (fair) shuffles and it can be used as a recursive definition for the shuffles of finite words (see, *e.g.*, [5, 17, 21]).

Lemma 2.4 *Let $u, v \in \Delta^\infty$ and $a, b \in \Delta$. Then*

(1) $u \,||\, \lambda = u \,|||\, \lambda = u = \lambda \,|||\, u = \lambda \,||\, u$ *and*

(2) $au \,|||\, bv = a(u \,|||\, bv) \cup b(au \,|||\, v)$ *and* $au \,||\, bv = a(u \,||\, bv) \cup b(au \,||\, v)$.

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

Lemma 2.5 *Let $u, v \in \Delta^\infty$ and $z, u', v' \in \Delta^*$. Then*

(1) *$z(u \mid\mid\mid v) \subseteq zu \mid\mid\mid v$ and $z(u \mid\mid v) \subseteq zu \mid\mid v$, and*

(2) *$(u' \mid\mid v')(u \mid\mid\mid v) \subseteq u'u \mid\mid\mid v'v$ and $(u' \mid\mid v')(u \mid\mid v) \subseteq u'u \mid\mid v'v$.*

Proof (1) We only prove the first inclusion. The other proof is analogous. Let $w \in z(u \mid\mid\mid v)$. Then $w = zw'$ for some $w' \in u \mid\mid\mid v$. By Definition 2.1(1), $w' = u_1v_1u_2v_2\cdots$, with $u_i, v_i \in \Delta^*$ for all $i \geq 1$, $u = u_1u_2\cdots$, and $v = v_1v_2\cdots$. Thus $w = zw' = zu_1v_1u_2v_2\cdots$ with $zu_1u_2\cdots = zu$. Hence $w \in zu \mid\mid\mid v$.

(2) We only prove the first inclusion. The other proof is analogous. First assume $u' = \lambda$. Then $u' \mid\mid v' = v'$ by Lemma 2.4(1). From Theorem 2.3 and (1) we have $v'(u \mid\mid\mid v) \subseteq u \mid\mid\mid v'v$. 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 2.4(2), $u' \mid\mid v' = au_1 \mid\mid bv_1 = a(u_1 \mid\mid bv_1) \cup b(au_1 \mid\mid v_1)$. This yields

$$\begin{aligned}(u' \mid\mid v')(u \mid\mid\mid v) &= a(u_1 \mid\mid bv_1)(u \mid\mid\mid v) \cup b(au_1 \mid\mid v_1)(u \mid\mid\mid v)\\ &\subseteq a(u_1u \mid\mid\mid bv_1v) \cup b(au_1u \mid\mid\mid v_1v)\\ &\subseteq (au_1u \mid\mid\mid bv_1v) \cup (au_1u \mid\mid\mid bv_1v)\\ &= (u'u \mid\mid\mid v'v)\end{aligned}$$

by applying the induction hypothesis and Lemma 2.4(2) twice. □

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

$$\mathit{pref}(u) \mid\mid \mathit{pref}(v) = \mathit{pref}(u \mid\mid\mid v) = \mathit{pref}(u \mid\mid v) = \mathit{pref}(u) \mid\mid\mid \mathit{pref}(v)\,.$$

Proof From Lemma 2.5(2) we know that $\mathrm{pref}(u) \mid\mid \mathrm{pref}(v) \subseteq \mathrm{pref}(u \mid\mid\mid v)$. Since $u \mid\mid\mid v \subseteq u \mid\mid v$ by Definition 2.1, it follows that $\mathrm{pref}(u \mid\mid\mid v) \subseteq \mathrm{pref}(u \mid\mid v)$ and $\mathrm{pref}(u) \mid\mid\mid \mathrm{pref}(v) \subseteq \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$. Hence the proof is complete once we have shown that $\mathrm{pref}(u \mid\mid v) \subseteq \mathrm{pref}(u) \mid\mid\mid \mathrm{pref}(v)$. Let $z \in \mathrm{pref}(u \mid\mid v)$. 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 $z = u_1v_1u_2v_2\cdots u_{n-1}v_{n-1}x$ with $x \in \mathrm{pref}(u_nv_n)$, $u_1u_2\cdots u_n \in \mathrm{pref}(u)$, and $v_1v_2\cdots v_n \in \mathrm{pref}(v)$. It is now immediately clear that $z \in \mathrm{pref}(u) \mid\mid\mid \mathrm{pref}(v)$. □

Example 2.7 Although $a^\omega \mid\mid\mid b \neq a^\omega \mid\mid b$, we have

$$\mathrm{pref}\,(a^\omega \mid\mid\mid b) = \mathrm{pref}\,(a^\omega \mid\mid b) = \{a^i b a^\omega : i \geq 0\} \cup a^* .$$

3 Associativity of Fair Shuffling

In this section the associativity of fair shuffling is proved: $u \mid\mid\mid (v \mid\mid\mid w) = (u \mid\mid\mid v) \mid\mid\mid w$ for all words u, v, and w. Extending a technique known from, *e.g.*, [13,17,21], 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 $h(\lambda) = \lambda$ and $h(xy) = h(x)h(y)$ for all $x, y \in \Sigma^*$. We extend h to Σ^∞ by setting $h(\lim\limits_{n\to\infty} v_n) = \lim\limits_{n\to\infty} h(v_n)$, 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 $h(v_i) \leq h(v_{i+1})$.

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 for the fair shuffle.

Theorem 3.1 *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 Without loss of generality we assume that $i = 1$ and $j = 2$.

($\subseteq$) Let $w \in u \mid\mid\mid v$. Then $w = u_1v_1u_2v_2\cdots$ with $u_1, u_2, \ldots, v_1, v_2, \ldots \in \Delta^*$ such that $u = u_1u_2\cdots$ and $v = v_1v_2\cdots$. Now consider

$$\overline{w} = \beta_1(u_1)\beta_2(v_1)\beta_1(u_2)\beta_2(v_2)\cdots .$$

It follows immediately that $\varphi_{1,\{2\}}(\overline{w}) = u$. Likewise, $\varphi_{2,\{1\}}(\overline{w}) = v$. Hence $\overline{w} \in \varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v)$. Since $\psi_{\{1,2\}}(\overline{w}) = w$, we are done.

($\supseteq$) We only prove the case that $u, v \in \Delta^\omega$. The proofs of the other cases are similar. 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

$$\beta_2(x_1)\beta_1(u_1)\beta_2(x_2)\beta_1(u_2)\cdots = \beta_1(y_1)\beta_2(v_1)\beta_1(y_2)\beta_2(v_2)\cdots .$$

Since $[\Delta, 1] \cap [\Delta, 2] = \varnothing$ it must be the case that either $\beta_2(x_1) = \lambda$ or $\beta_1(y_1) = \lambda$.

First assume that $\beta_2(x_1) = \lambda$, *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 3.2 *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*

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

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 fair shuffling of possibly infinite words is associative, a result which is mentioned in [19] (where fair shuffling is called fair merge) but which is not proved there due to the complications caused by a different setting.

Theorem 3.3 *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 3.1 and Lemma 3.1,

$$\begin{aligned} u \mid\mid\mid (v \mid\mid\mid w) &= \psi_{\{1,4\}}(\varphi^{-1}_{1,\{4\}}(u) \cap \varphi^{-1}_{4,\{1\}}(\psi_{\{2,3\}}(\varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w)))) \\ &= \psi_{\{1,2,3\}}(\varphi^{-1}_{1,\{2,3\}}(u) \cap \varphi^{-1}_{2,\{1,3\}}(v) \cap \varphi^{-1}_{3,\{1,2\}}(w))\,. \end{aligned}$$

Similarly, we have

$$\begin{aligned} (u \mid\mid\mid v) \mid\mid\mid w &= \psi_{\{3,4\}}(\varphi^{-1}_{4,\{3\}}(\psi_{\{1,2\}}(\varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v))) \cap \varphi^{-1}_{3,\{4\}}(w)) \\ &= \psi_{\{1,2,3\}}(\varphi^{-1}_{1,\{2,3\}}(u) \cap \varphi^{-1}_{2,\{1,3\}}(v) \cap \varphi^{-1}_{3,\{1,2\}}(w))\,. \end{aligned}$$

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.*, [7, 10, 13, 17, 20, 22]) which we state here explicitly for completeness' sake and for future reference.

Corollary 3.4 *Let $u, v, w \in \Delta^*$. Then $u \mid\mid (v \mid\mid w) = (u \mid\mid v) \mid\mid w$.*

Theorem 3.1 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 3.4).

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. We begin by introducing the concept of *decomposition* as an explicit description of how a shuffle is obtained from two given finite words.

Definition 4.1 Let $w \in \Delta^*$. A *decomposition* of w is a sequence $d = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$ with $n \geq 1$, $u_1 \in \Delta^*$, $u_2, u_3, \ldots, u_n, v_1, v_2, \ldots, v_{n-1} \in \Delta^+$, $v_n \in \Delta^*$, and $w = u_1 v_1 u_2 v_2 \cdots u_n v_n$. If $u_1 u_2 \cdots u_n = u$ and $v_1 v_2 \cdots v_n = v$, then d is called a (u, v)-*decomposition* of w. The *norm* of d, denoted by $|| d ||$, is n.

Note that decompositions — apart from the first and the last subword mentioned — only refer to nonempty subwords of the words that are shuffled. This provides us with a normal form for the description of finite shuffles.

Lemma 4.2 *Let $u, v, w \in \Delta^*$. Then there exists a (u, v)-decomposition of w if and only if $w \in u \mid\mid v$.*

Proof (Only if) Immediate from Definitions 2.1 and 4.1.

(If) Let $w \in u \mid\mid v$. Then by Definition 2.1 we have $w = u_1 v_1 u_2 v_2 \cdots$, with $u_i, v_i \in \Delta^*$ for all $i \geq 1$, $u = u_1 u_2 \cdots$, and $v = v_1 v_2 \cdots$. Let $\rho_1 = (u_1, v_1)$ and if $\rho_k = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell)$ for some $\ell \geq 1$ and $\alpha_j, \beta_j \in \Delta^*$, for all $1 \leq j \leq \ell$, then

$$\rho_{k+1} = \begin{cases} (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell u_{k+1}, v_{k+1}) & \text{if } \beta_\ell = \lambda, \\ (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell v_{k+1}) & \text{if } \beta_\ell \neq \lambda \text{ and } u_{k+1} = \lambda, \text{ and} \\ (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell, u_{k+1}, v_{k+1}) & \text{if } \beta_\ell \neq \lambda \text{ and } u_{k+1} \neq \lambda. \end{cases}$$

Thus ρ_{k+1} is obtained from ρ_k by adding the words u_{k+1} and v_{k+1}. These are added in such a way that only the first and the last element of ρ_{k+1} are allowed to equal λ. In general, if $\rho_k = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell)$, then $\alpha_1, \beta_\ell \in$

Δ^*, $\alpha_j \in \Delta^+$, for all $1 < j \leq \ell$, and $\beta_j \in \Delta^+$, for all $1 \leq j < \ell$. Furthermore, $\alpha_1\beta_1\alpha_2\beta_2\cdots\alpha_\ell\beta_\ell = u_1v_1u_2v_2\cdots u_kv_k$, $\alpha_1\alpha_2\cdots\alpha_\ell = u_1u_2\cdots u_k$, and $\beta_1\beta_2\cdots\beta_\ell = v_1v_2\cdots v_k$. Since w is finite, there must exist an $m \geq 1$ such that for all $n > m$, $u_n = v_n = \lambda$. Then $\rho_m = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell)$ is such that $\alpha_1\beta_1\alpha_2\beta_2\cdots\alpha_\ell\beta_\ell = w$, $\alpha_1 \in \Delta^*$, $\beta_1, \alpha_2, \beta_2, \alpha_3, \ldots, \beta_{\ell-1}, \alpha_\ell \in \Delta^+$, $\beta_\ell \in \Delta^*$, $\alpha_1\alpha_2\cdots\alpha_\ell = u$, and $\beta_1\beta_2\cdots\beta_\ell = v$. Hence ρ_m is a (u, v)-decomposition of w. □

It is not difficult to see that a shuffle may have several decompositions. In a series of papers (see, *e.g.*, [16, 17]) 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. Associativity is consequently discussed per set of trajectories. However, associativity of the shuffle as investigated here is not considered.

To be able to describe extensions of shuffles explicitly, we introduce a precedence relation for decompositions.

Definition 4.3 Let $d = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$ and $d' = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$ be two decompositions of $x_1y_1x_2y_2\cdots x_ky_k \in \Delta^*$ and $u_1v_1u_2v_2\cdots u_nv_n \in \Delta^*$, respectively. Then

(1) *d directly precedes d′* if $k \leq n$ and for all $1 \leq j \leq k-1$, $x_j = u_j$ and $y_j = v_j$, and—moreover—either

   (a) $k = n$, $x_k = u_k$, and $y_ka = v_k$, for some $a \in \Delta$, or

   (b) $k = n$, $y_k = v_k = \lambda$, and $x_ka = u_k$, for some $a \in \Delta$, or

   (c) $k = n-1$, $y_k \neq \lambda$, $v_{k+1} = \lambda$, and $u_{k+1} = a$, for some $a \in \Delta$, and

(2) *d precedes d′* if there exist decompositions $d_0, d_1, \ldots, d_\ell$ such that $\ell \geq 0$, $d = d_0$, $d' = d_\ell$, and for all $0 \leq j \leq \ell - 1$, d_j directly precedes d_{j+1}.

Note that if d and d' are two decompositions such that d directly precedes d', then $||\, d'|| = ||\, d\,||$ or $||\, d'|| = ||\, d\,|| + 1$. Hence if d precedes d', then $||\, d'|| \geq ||\, d\,||$.

It is easy to see that whenever a decomposition d precedes a decomposition d', then d decomposes a prefix of the word that d' decomposes. In fact, we have the following result.

Lemma 4.4 *Let* $d = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$ *and* $d' = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$ *be two decompositions such that d precedes d′. Then*

$$x_1x_2\cdots x_k \in \mathit{pref}(u_1u_2\cdots u_n)\,,$$

$$y_1y_2\cdots y_k \in \mathit{pref}(v_1v_2\cdots v_n)\,,$$

and

$$x_1y_1x_2y_2\cdots x_ky_k \in \mathit{pref}(u_1v_1u_2v_2\cdots u_nv_n)\,.$$

Proof If $d = d'$ there is nothing to prove, so let us assume that $d \neq d'$. From Definition 4.3 it is clear that the statement holds in case d immediately precedes d'.

If d precedes d', then there exist (s_j, t_j)-decompositions d_j of words $w_j \in \Delta^*$ with $0 \leq j \leq \ell$, for some $\ell \geq 1$, such that $d_0 = d$, $d_\ell = d'$, and d_j immediately precedes d_{j+1}, for all $0 \leq j < \ell$. Hence, for all $0 \leq j < \ell - 1$, $s_j \in \mathrm{pref}(s_{j+1})$, $t_j \in \mathrm{pref}(t_{j+1})$, and $w_j \in \mathrm{pref}(w_{j+1})$. Thus $s_0 = x_1 x_2 \cdots x_k \in \mathrm{pref}(s_\ell) = \mathrm{pref}(u_1 u_2 \cdots u_n)$, $t_0 = y_1 y_2 \cdots y_k \in \mathrm{pref}(t_\ell) = \mathrm{pref}(v_1 v_2 \cdots v_n)$, and $w_0 = x_1 y_1 x_2 y_2 \cdots x_k y_k \in \mathrm{pref}(w_\ell) = \mathrm{pref}(u_1 v_1 u_2 v_2 \cdots u_n v_n)$. □

Given this lemma it can be proved that the limit of the shuffles defined by an ordered sequence of (u_i, v_i)-decompositions is a shuffle of the limits of the u_i and the v_i.

Lemma 4.5 *For all $i \geq 0$, let d_i be a (u_i, v_i)-decomposition of a word w_i over Δ such that d_i precedes d_{i+1}. Then $u = \lim_{i\to\infty} u_i$, $v = \lim_{i\to\infty} v_i$, and $w = \lim_{i\to\infty} w_i$ exist, and $w \in u \mid\mid v$.*

Proof By Lemma 4.4 it follows that $u_i \leq u_{i+1}$, $v_i \leq v_{i+1}$, and $w_i \leq w_{i+1}$, for all $i \geq 0$, so indeed u, v, and w exist and we only have to prove that $w \in u \mid\mid v$. We distinguish two cases.

First we consider the case that there exists an $N \in \mathbb{N}$ such that $||\, d_i|| = ||\, d_N||$ for all $i \geq N$. Let $N_0 \in \mathbb{N}$ be such an N. Again we distinguish two cases.

Let us assume first that, for all $i \geq N_0$, if $d_i = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$, then $y_n = \lambda$. Consequently, for all $i \geq N_0$, $v_i = v_{N_0}$. From $u_i \leq u_{i+1}$, for all $i \geq 0$, we infer that for all $i > N_0$ there exist $z_{i-N_0} \in \Delta^*$ such that $u_{i+1} = u_i z_{i-N_0}$. Observe that $u = \lim_{i\to\infty} u_i = u_{N_0} \lim_{i\to\infty} z_1 z_2 \cdots z_{i-N_0}$. We thus obtain that for all $i > N_0$ we have $w_i = w_{N_0} z_1 z_2 \cdots z_{i-N_0}$. Since $w_{N_0} \in u_{N_0} \mid\mid v_{N_0}$ by Lemma 4.2, we conclude that $w = \lim_{i\to\infty} w_i \in (u_{N_0} \mid\mid v_{N_0}) \lim_{i\to\infty} z_1 z_2 \cdots z_{i-N_0} = (u_{N_0} \mid\mid v_{N_0})(\lim_{i\to\infty} z_1 z_2 \cdots z_{i-N_0} \mid\mid \lambda) \subseteq u \mid\mid v_{N_0} \subseteq u \mid\mid v$ by Lemma 2.5(2) and the definition of u.

Next assume there exist an $i \geq N_0$ such that $d_i = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n)$ with $y_n \neq \lambda$. Let ℓ_0 be the smallest such i. Thus, for all $i \geq \ell_0$, $u_i = u_{\ell_0}$. From $v_i \leq v_{i+1}$, for all $i \geq 0$, we infer that for all $i > \ell_0$ there exist $z_{i-\ell_0} \in \Delta^*$ such that $v_{i+1} = v_i z_{i-\ell_0}$. Observe that $v = \lim_{i\to\infty} v_i = v_{\ell_0} \lim_{i\to\infty} z_1 z_2 \cdots z_{i-\ell_0}$. Thus for all $i > \ell_0$ we have $w_i = w_{\ell_0} z_1 z_2 \cdots z_{i-\ell_0}$. Since $w_{\ell_0} \in u_{\ell_0} \mid\mid v_{\ell_0}$ by Lemma 4.2, we conclude that $w = \lim_{i\to\infty} w_i \in (u_{\ell_0} \mid\mid v_{\ell_0}) \lim_{i\to\infty} z_1 z_2 \cdots z_{i-\ell_0} = (u_{\ell_0} \mid\mid v_{\ell_0})(\lambda \mid\mid \lim_{i\to\infty} z_1 z_2 \cdots z_{i-\ell_0}) \subseteq u_{\ell_0} \mid\mid v \subseteq u \mid\mid v$ by Lemma 2.5(2) and the definition of u.

Now we move to the case that for all $N \in \mathbb{N}$ there exists a $k \in \mathbb{N}$ such that $||\, d_k|| \geq N$. Let $j_1, j_2, \ldots \in \mathbb{N}$ be the (unique) infinite sequence of integers such that for all $i \in \mathbb{N}$, $||\, d_{j_i}|| < ||\, d_{j_{i+1}}||$ and $||\, d_\ell|| = ||\, d_{j_i}||$ for all $j_i \leq \ell < j_{i+1}$. Since $||\, d_0|| \leq ||\, d_1|| \leq \cdots$ is an unbounded sequence of integers we know

that the j_i as just described exist. Since each d_{j_i} precedes $d_{j_{i+1}}$, Definition 4.3 implies that there exist $x_1, x_2, \ldots, y_1, y_2, \ldots, s_1, s_2, \ldots, t_1, t_2, \cdots \in \Delta^*$ such that $d_{j_i} = (x_1, y_1, x_2, y_2, \ldots, x_{||\, d_{j_i}||-1}, y_{||\, d_{j_i}||-1}, s_i, t_i)$, for all $i \geq 1$. By Lemma 4.4, $u_{j_i} = x_1 x_2 \cdots x_{||\, d_{j_i}||-1} s_i \in \mathrm{pref}\,(u_{j_{i+1}}) = \mathrm{pref}\,(x_1 x_2 \cdots x_{||\, d_{j_{i+1}}||-1} s_{i+1})$, for all $i \geq 1$, and thus $u = \lim\limits_{n\to\infty} x_1 x_2 \cdots x_n$. Analogously, $v = \lim\limits_{n\to\infty} y_1 y_2 \cdots y_n$, and $w = \lim\limits_{n\to\infty} x_1 y_1 x_2 y_2 \cdots x_n y_n$. Thus $w = x_1 y_1 x_2 y_2 \cdots$ with $x_1 \in \Delta^*$, $x_i \in \Delta^+$ for all $i \geq 2$, $y_i \in \Delta^+$ for all $i \geq 1$, $u = x_1 x_2 \cdots$, and $v = y_1 y_2 \cdots$. Hence $w \in u \mid\mid v$. □

On the other hand, we would now like to show that whenever every prefix of a 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 the decompositions describing the prefixes of w as shuffles of prefixes of u and v would precede each other and ultimately lead to w as a shuffle of u and v. As the next lemma demonstrates, this can be achieved by requiring that u and v have no letters in common. We write $\mathrm{alph}(w)$ to denote the alphabet of a word w, *i.e.* the set of all letters that actually occur in w.

Lemma 4.6 *Let $u, v \in \Delta^\infty$ be such that $alph(u) \cap alph(v) = \varnothing$ and let $w \in \Delta^\omega$. Then $pref(w) \subseteq pref(u) \mid\mid pref(v)$ implies $w \in u \mid\mid v$.*

Proof Let $\mathrm{pref}\,(w) \subseteq \mathrm{pref}\,(u) \mid\mid \mathrm{pref}\,(v)$. Now consider two arbitrary consecutive prefixes of w. Thus for some $n \geq 0$ we have $w[n]$ and $w[n+1] = w[n]a$ with $a \in \mathrm{alph}(u)$ or $a \in \mathrm{alph}(v)$. Since $\mathrm{pref}\,(w) \subseteq \mathrm{pref}\,(u) \mid\mid \mathrm{pref}\,(v)$, there are prefixes u_n and u_{n+1} of u, and prefixes v_n and v_{n+1} of v such that $w[n] \in u_n \mid\mid v_n$ and $w[n+1] \in u_{n+1} \mid\mid v_{n+1}$. Consequently, $u_{n+1} = u_n a$ and $v_{n+1} = v_n$ if $a \in \mathrm{alph}(u)$, and $v_{n+1} = v_n a$ and $u_{n+1} = u_n$ if $a \in \mathrm{alph}(v)$. Now let d_n be a (u_n, v_n)-decomposition of $w[n]$ with $d_n = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$ for some $k \geq 0$. Then we obtain a (u_{n+1}, v_{n+1})-decomposition of $w[n+1]$ as follows.

First assume that $a \in \mathrm{alph}(u)$. If $y_k = \lambda$, then $d_{n+1} = (x_1, y_1, x_2, y_2, \ldots, x_k a, y_k)$, whereas if $y_k \neq \lambda$, then $d_{n+1} = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k, a, \lambda)$. In both cases we have $x_1 x_2 \cdots x_k a = u_n a = u_{n+1}$ and $y_1 y_2 \cdots y_k = v_n = v_{n+1}$. Moreover $x_1 y_1 x_2 y_2 \cdots x_k y_k a = w[n]a = w[n+1]$. Thus d_{n+1} is a (u_{n+1}, v_{n+1})-decomposition of $w[n+1]$ and d_n precedes d_{n+1}.

Secondly, let $a \in \mathrm{alph}(v)$. Now $d_{n+1} = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k a)$. Since $x_1 x_2 \cdots x_k = u_n = u_{n+1}$ and $y_1 y_2 \cdots y_k a = v_n a = v_{n+1}$ are such that $x_1 y_1 x_2 y_2 \cdots x_k y_k a = w[n]a = w[n+1]$ we thus know that d_{n+1} is a (u_{n+1}, v_{n+1})-decomposition of $w[n+1]$, which is preceded by d_n.

Observe that the only decomposition of $w[0] = \lambda$ is $d_0 = (\lambda, \lambda)$. Hence we have defined an infinite (and unique) sequence of (u_i, v_i)-decompositions d_i of $w[i]$, with $i \geq 0$, such that d_i precedes d_{i+1} for all $i \geq 0$. From Lemma 4.5 it thus follows that $w = \lim\limits_{n\to\infty} w[n] \in (\lim\limits_{n\to\infty} u_n) \mid\mid (\lim\limits_{n\to\infty} v_n) = u \mid\mid v$. □

Note that this proof uses the observation that—thanks to the disjointness of the alphabets—any decomposition of a prefix of w into prefixes of u and v, has a

(unique) successor describing a decomposition of the next prefix. This ultimately leads to a description of w as a shuffle of u and v. Unfortunately, in general, it is not true that decompositions of prefixes can be extended to decompositions of the next prefix. This is shown in the following example, which even shows that an infinite word may have infinitely many prefixes with non-extendable prefixes.

Example 4.7 Let $u = (a^3b)^\omega$ and $v = b^\omega$. Clearly $\{a^3, a^3b\} \subseteq \mathrm{pref}(u)$, $\{b^2, b^3\} \subseteq \mathrm{pref}(v)$, and $w = a^3b^3 \in \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$. Note that $d_1 = (a^3, b^3)$ and $d_2 = (a^3b, b^2)$ are two decompositions of w.

Next consider $w' = wa = a^3b^3a \in \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$. The only decompositions of w' which are directly preceded by a decomposition of prefixes of u and v are $d' = (a^3b, b^2, a, \lambda)$ and $d'' = (a^3, b^2, ba, \lambda)$. Clearly, d_1 neither precedes d' nor d''. Note, however, that d_2 precedes d'.

Finally, let $j \geq 0$, $u_j = a^3(ba^3)^j \in \mathrm{pref}(u)$, and $v_j = b^3(b^3)^j \in \mathrm{pref}(v)$. Then clearly both $w_j = (a^3b^4)^j a^3b^3 \in \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$ and $w'_j = w_j a = (a^3b^4)^j a^3b^3a \in \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$. Note that $d_j = (x_0, y_0, x_1, y_1, \ldots, x_j, y_j, a^3, b^3)$, where $x_i = a^3b$ and $y_i = b^3$ for all $0 \leq i \leq j$, is a (u_j, v_j)-decomposition of w_j. Reasoning as for $j = 0$ it is however clear that there does not exist a decomposition of w'_j based on prefixes of u and v that is preceded by d_j.

Despite this example, it can however be shown that for all words $u, v \in \Delta^\infty$ and $w \in \Delta^\omega$, whenever $\mathrm{pref}(w) \subseteq \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$ then $w \in u \mid\mid v$, even when u and v have letters in common. We do this by establishing the existence of an infinite sequence of (u_n, v_n)-decompositions of $w[n]$, with $n \geq 0$, preceding each other. With this in mind we now recall *König's Lemma*.

Lemma 4.8 (König's Lemma) *If G is an infinite finitely-branching rooted tree, then there exists an infinite path through G, starting in the root.*

For later use we 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 [9]—after initially referring to the same concept as 'König closure' in its preceding technical report.

Definition 4.9 Let $K \subseteq \Delta^\infty$. K is *limit-closed* if for all $w_1 \leq w_2 \leq \cdots \in \mathrm{pref}(K)$, $\lim_{n\to\infty} w_n \in K \cup \mathrm{pref}(K)$.

Example 4.10 All singleton languages $\{u\}$ and all finitary languages $L = \{\lambda, a, \ldots, a^n : n \geq 1\}$ over a unary alphabet are limit-closed, whereas a^* is not as $\lim_{n\to\infty} a^n = a^\omega \notin a^* \cup L$. However, $a^* \cup a^\omega$ and a^ω are limit-closed.

Lemma 4.11 *Let $K, L \subseteq \Delta^\infty$ be limit-closed and let $w \in \Delta^\omega$. Then $pref(w) \subseteq pref(K) \mid\mid pref(L)$ implies $w \in K \mid\mid L$.*

Proof Let $\mathrm{pref}(w) \subseteq \mathrm{pref}(K) \mid\mid \mathrm{pref}(L)$. For $n \geq 0$, let

$$\begin{aligned} V_n \;=\; \{d : d \text{ is a } (u_n, v_n)\text{-decomposition of } w[n], \\ u_n \in \mathrm{pref}(K), \text{ and } v_n \in \mathrm{pref}(L)\} \end{aligned}$$

be the set of all possible decompositions of the prefix $w[n]$ of w. Note that $V_0 = \{(\lambda, \lambda)\}$. Note furthermore that each V_n is finite, for $n \geq 0$, and that $V_n \cap V_{n'} = \varnothing$, for all $n > n' \geq 0$.

Consider the directly precedes relation $E = \{(d, d') : d \text{ directly precedes } d'\}$. Thus $E \subseteq \bigcup_{n \geq 1}(V_{n-1} \times V_n)$. Note that $G = (\bigcup_{n \geq 0} V_n, E)$ is a directed acyclic graph. It is sketched in Figure 1.

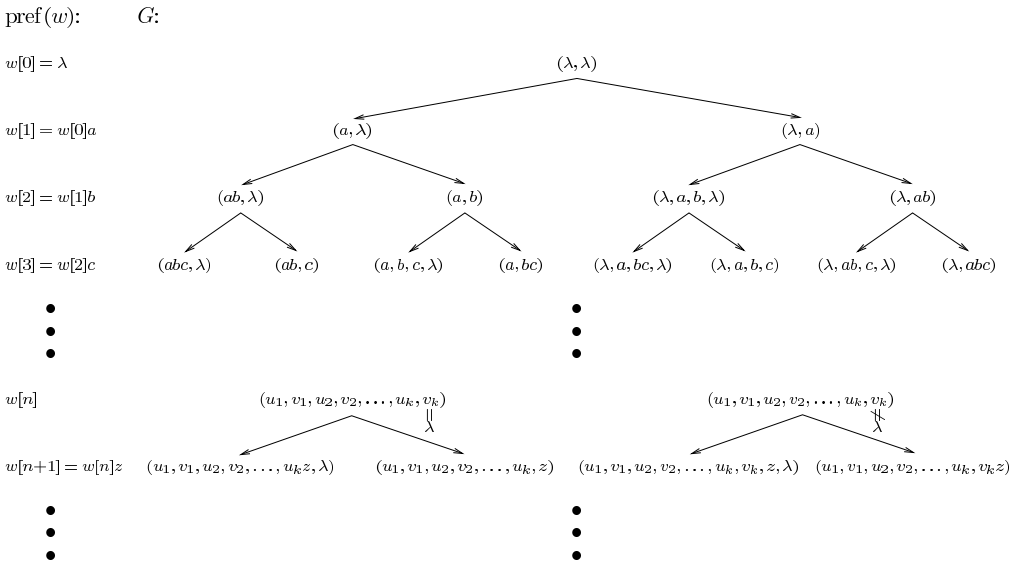

Figure 1: Sketch of tree $G = (\bigcup_{n \geq 0} V_n, E)$.

Except for (λ, λ), every vertex of G has precisely one incoming edge. This can be seen as follows. The fact that $\mathrm{pref}(w) \subseteq \mathrm{pref}(K) \mid\mid \mathrm{pref}(L)$ implies that every vertex has at least one incoming edge, whereas the fact that for every decomposition of a prefix $w[n]$, with $n \geq 1$, we can immediately distinguish the unique last symbol of $w[n]$, implies that every vertex has at most one incoming edge. Furthermore, from Definition 4.3 it follows that every vertex has at most two outgoing edges, depending on whether the symbol added to $w[n]$, with $n \geq 0$, to obtain $w[n+1]$ 'belongs' to a prefix from K or to a prefix from L. Hence G is an infinite finitely-branching rooted tree with root (λ, λ).

We can thus use König's Lemma to conclude that there exists an infinite path π through G, starting in the root (λ, λ). Let $\pi = (d_0, d_1, \dots)$. Then for all

$n \geq 0$, d_n is a (u_n, v_n)-decomposition of $w[n]$ and $(d_n, d_{n+1}) \in E$. Hence from Lemma 4.5 it follows that $u = \lim_{n\to\infty} u_n$, $v = \lim_{n\to\infty} v_n$, and $w = \lim_{n\to\infty} w_n$ exist, and $w \in u \mid\mid v$. Since K and L are limit-closed this implies that $w \in K \mid\mid L$. □

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

Example 4.12 Let $K = a^*$ and $L = \{\lambda\}$. Then

$$\mathrm{pref}\,(a^\omega) = a^* = \mathrm{pref}\,(K) \mid\mid \mathrm{pref}\,(L)\,,$$

but $a^\omega \notin a^* = K \mid\mid L$.

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

Corollary 4.13 *Let $u, v \in \Delta^\infty$ and $w \in \Delta^\omega$. Then $pref(w) \subseteq pref(u) \mid\mid pref(v)$ implies $w \in u \mid\mid v$.*

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

Example 4.14 Consider a^ω. We have $\mathrm{pref}\,(a^\omega) = a^*$ and

$$a^* \subseteq \mathrm{pref}\,(a^\omega) \mid\mid\mid \mathrm{pref}\,(b) = \mathrm{pref}\,(a^\omega) \mid\mid \mathrm{pref}\,(b)\,.$$

However, as we have seen in Example 2.2, $a^\omega \in a^\omega \mid\mid b$, but $a^\omega \notin a^\omega \mid\mid\mid b$.

Theorem 2.6 and Lemma 4.11 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 4.15 *Let $u, v \in \Delta^\infty$, let $K, L \subseteq \Delta^\infty$ be limit-closed, and let $w \in \Delta^\omega$. Then*
(1) $w \in u \mid\mid v$ if and only if $pref(w) \subseteq pref(u) \mid\mid pref(v)$, and
(2) $w \in K \mid\mid L$ if and only if $pref(w) \subseteq pref(K) \mid\mid pref(L)$.

We need one more observation in order to conclude that shuffling is associative.

Corollary 4.16 *Let $v, w \in \Delta^\infty$. Then $v \mid\mid w$ is limit-closed.*

Proof Let $y_1 \leq y_2 \leq \cdots \in \mathrm{pref}\,(v \mid\mid w)$ and let $y = \lim_{n\to\infty} y_n$. Since for all $x \in \mathrm{pref}\,(y)$, there exists an $i \geq 0$ such that $x \in \mathrm{pref}\,(y_i) \in \mathrm{pref}\,(\mathrm{pref}\,(v \mid\mid w)) = \mathrm{pref}\,(v \mid\mid w)$, it follows that $\mathrm{pref}\,(y) \subseteq \mathrm{pref}\,(v \mid\mid w)$. We distinguish two cases. If $y \in \Delta^*$, then $y \in \mathrm{pref}\,(v \mid\mid w)$. If $y \in \Delta^\omega$, then by Theorem 4.15(1), $y \in v \mid\mid w$. Hence $y \in v \mid\mid w \cup \mathrm{pref}\,(v \mid\mid w)$ and $v \mid\mid w$ is thus limit-closed. □

Theorem 4.17 *Let $u, v, w \in \Delta^\infty$. Then $u \mid\mid (v \mid\mid w) = (u \mid\mid v) \mid\mid w$.*

Proof If u, v, w are finite words, we have Corollary 3.4. If at least one of them is infinite, then both $u \mid\mid (v \mid\mid w)$ and $(u \mid\mid v) \mid\mid w$ consist of infinite words only. Let $x \in u \mid\mid (v \mid\mid w)$. Then Theorem 4.15(2) implies that $\mathrm{pref}(x) \subseteq \mathrm{pref}(u) \mid\mid \mathrm{pref}(v \mid\mid w)$. Thus, by Theorem 2.6,

$$\mathrm{pref}(x) \subseteq \mathrm{pref}(u) \mid\mid (\mathrm{pref}(v) \mid\mid \mathrm{pref}(w)) .$$

Consequently $\mathrm{pref}(x) \subseteq (\mathrm{pref}(u) \mid\mid \mathrm{pref}(v)) \mid\mid \mathrm{pref}(w)$ by Corollary 3.4 and $\mathrm{pref}(x) \subseteq \mathrm{pref}(u \mid\mid v) \mid\mid \mathrm{pref}(w)$ by Theorem 2.6. Finally, since $u \mid\mid v$ and $\{w\}$ are limit-closed, Theorem 4.15(2) implies that $x \in (u \mid\mid v) \mid\mid w$. The converse inclusion follows from the above and Theorem 2.3. □

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 not substantiated 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 [2, 4]. 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 [3] 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 [2, 4]. 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 [2]. The associativity of these variants, in their turn, is crucial to prove that several types of team automata satisfy compositionality in [2, 4] (in the latter only finitary behaviours are considered).

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 2.6, 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 4.11 and its Corollary 4.13 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.

Acknowledgement

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.

References

[1] K. Abrahamson, *Decidability and Expresiveness of Logics of Processes*. Ph.D. thesis, University of Washington, Seattle, 1980.

[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, and 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 and J. Kleijn, Team Automata Satisfying Compositionality. In *Proceedings of FME 2003: Formal Methods—the Twelfth International Symposium of Formal Methods Europe, Pisa, Italy* (K. Araki, S. Gnesi, and D. Mandrioli, eds.), *Lecture Notes in Computer Science* 2805, Springer-Verlag, Berlin, 2003, 381–400.

[5] M.H. ter Beek, C. Martín-Vide, and V. Mitrana, Synchronized Shuffles. Accepted for publication in *Theoretical Computer Science*, 2005. *Cf.* also Technical Report 2003-TR-40, Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, 2003.

[6] J.A. Bergstra, A. Ponse, and S.A. Smolka (eds.), *Handbook of Process Algebra*. Elsevier Science Publishers, Amsterdam, 2001.

[7] S.L. Bloom and Z. Ésik, Free Shuffle Algebras in Language Varieties. *Theoretical Computer Science* 163 (1996), 55–98.

[8] R. De Simone, Langages Infinitaires et Produit de Mixage. *Theoretical Computer Science* 31 (1984), 83–100.

[9] 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.

[10] S. Ginsburg, *Algebraic and Automata-Theoretic Properties of Formal Languages*. *Fundamental Studies in Computer Science* 2, North-Holland Publishing Company, Amsterdam, 1975.

[11] S. Ginsburg and E.H. Spanier, Mappings of Languages by Two-Tape Devices. *Journal of the ACM* 12, 3 (1965), 423–434.

[12] J.L. Gischer, Shuffle Languages, Petri Nets, and Context Sensitive Grammars. *Communications of the ACM* 24 (1981), 597–605.

[13] M. Jantzen, The Power of Synchronizing Operations on Strings. *Theoretical Computer Science* 14 (1981), 127–154.

[14] T. Kimura, An Algebraic System for Process Structuring and Interprocess Communication. In *Proceedings of the 8th ACM SIGACT Symposium on Theory of Computing, Hershey, Pennsylvania*, ACM Press, New York, 1976, 92–100.

[15] M. Latteux and Y. Roos, Synchronized Shuffle and Regular Languages. In *Jewels are Forever—Contributions on Theoretical Computer Science in Honor of Arto Salomaa* (J. Karhumäki, H.A. Maurer, Gh. Păun, and G. Rozenberg, eds.), Springer-Verlag, Berlin, 1999, 35–44.

[16] A. Mateescu and G.D. Mateescu, Fair and Associative Infinite Trajectories. In *Jewels are Forever—Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, Springer-Verlag, Berlin, 1999, 327–338.

[17] A. Mateescu, G. Rozenberg, and A. Salomaa, Shuffle on Trajectories: Syntactic Constraints. *Theoretical Computer Science* 197, 1-2 (1998), 1–56.

[18] W.F. Ogden, W.E. Riddle, and W.C. Rounds, Complexity of Expressions Allowing Concurrency. In *Conference Record of the 5th Annual ACM Symposium on Principles of Programming Languages, Tucson, Arizona*, ACM Press, New York, 1978, 185–194.

[19] D. Park, On the semantics of fair parallelism. In *Proceedings of the Copenhagen Winter School on Abstract Software Specifications* (D. Bjørner, ed.), *Lecture Notes in Computer Science* 86, Springer-Verlag, Berlin, 1979, 504–526.

[20] A.W. Roscoe, *The Theory and Practice of Concurrency.* Prentice Hall International Series in Computer Science, London, 1997.

[21] G. Rozenberg and A. Salomaa (eds.), *Handbook of Formal Languages.* Springer-Verlag, Berlin, 1997.

[22] A.C. Shaw, Software Descriptions with Flow Expressions. *IEEE Transactions on Software Engineering* SE-4, 3 (1978), 242–254.

[23] J.L.A. van de Snepscheut, *Trace Theory and VLSI Design. Lecture Notes in Computer Science* 200, Springer-Verlag, Berlin, 1985.