Team automata : a formal approach to the modeling of collaboration between system components

Beek, M.H. ter

6. Behavior of Team Automata

In this chapter we study the behavior of team automata. We begin with a few elementary observations on the computational power for the case of finite component automata, i.e. component automata with a finite set of states and a finite alphabet (of input, output, and internal actions). For the rest of this chapter we then turn to component automata and team automata with possibly infinite sets of states and actions. We study the relation between the computations and behavior of team automata on the one hand, and those of their constituting component automata on the other hand. Since a composable system does not uniquely define a team automaton, the relation between the computations and behavior of a team automaton and those of its constituting component automata depends on the allowed synchronizations.

We are particularly interested in conditions which guarantee that a team automaton satisfies *compositionality*. This means that the behavior of a team automaton can be described as a function of the behavior of its constituting component automata. Since component automata and team automata have languages as behavior, we use language-theoretic operations — so called *shuffles* — to describe the combination of words into new words. In order to be able to apply these shuffles in the context of team automata, we extensively investigate their properties in two separate sections. This eventually enables us to identify several types of team automata satisfying compositionality.

6.1 Behavior of Finite Component Automata

Most types of automata considered in this thesis may have an infinite set of states and an infinite set of actions. As already discussed in Section 3.1, by allowing the automata in our framework to have an infinite set of states we end up with automata that have Turing machine power. In this section we study the behavior of *finite component automata*, i.e. of component automata with a finite set of states and a finite alphabet, and — subsequently — the influence that the distinction between input, output, and internal actions has on their behavior (cf. Section 7.1.4).

In the remainder of this section, all component automata have a finite set of states and a finite alphabet. Moreover, we restrict our study to an investigation of their finite computations, and the resulting finitary behavior.

Component automata differ from automata only by the distinction of their set of actions into input, output, and internal actions. In fact, by ignoring this distinction, every finite component automaton $\mathcal{C} = (Q, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{int}), \delta, I)$ can be viewed as an automaton $\mathcal{A} = (Q, \Sigma, \delta, I)$ such that $\Sigma = \Sigma_{inp} \cup \Sigma_{out} \cup \Sigma_{int}$, with $\mathbf{B}^{\Sigma}_{\mathcal{A}} = \mathbf{B}^{\Sigma}_{\mathcal{C}}$. Conversely, every automaton $\mathcal{A} = (Q, \Sigma, \delta, I)$ such that Q and Σ are finite can be viewed — once its alphabet is disjointly distributed over input, output, and internal actions Σ_1, Σ_2, and Σ_3 — as a component automaton $\mathcal{C} = (Q, (\Sigma_1, \Sigma_2, \Sigma_3), \delta, I)$ such that $\Sigma_1 \cup \Sigma_2 \cup \Sigma_3 = \Sigma$, with $\mathbf{B}^{\Sigma}_{\mathcal{C}} = \mathbf{B}^{\Sigma}_{\mathcal{A}}$.

The computational power of automata with a finite set of states and a finite set of actions equals that of the family of prefix-closed regular finitary languages, which we denote by pREG. The family of regular languages, denoted by REG, is precisely the family of languages accepted by the well-known model of finite (state) automata (cf. the introduction to Chapter 3). Formally, $\mathsf{pREG} = \{L \in \mathsf{REG} \mid L \text{ is prefix closed}\}$. It is known that $\mathsf{pREG} \subset \mathsf{REG}$ and $\mathsf{FIN} \subset \mathsf{REG}$, where FIN denotes the family of finite languages, while FIN and pREG are incomparable.

We denote $\mathsf{CA} = \{\mathbf{B}^{\Sigma}_{\mathcal{C}} \mid \Sigma \text{ is an alphabet and } \mathcal{C} \text{ is a finite component automaton with alphabet } \Sigma\}$. Then the above observations immediately yield the following result.

Lemma 6.1.1. $\mathsf{pREG} = \mathsf{CA}$. □

Note that the inclusion $\mathsf{pREG} \subseteq \mathsf{CA}$ can be proven by choosing any distribution of an automaton's alphabet over input, output, and internal alphabets.

Using this observation once more we now prove that all behavior collected in CA (and hence in pREG) can also be obtained as the input, output, internal, external, and locally-controlled behavior of component automata.

First we introduce some notation. Consider an arbitrary component automaton $\mathcal{C} = (Q, (\Sigma_1, \Sigma_2, \Sigma_3), \delta, I)$ and let $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3$. Consequently we set $\mathbf{B}^{inp}_{\mathcal{C}} = \mathbf{B}^{\Sigma_1}_{\mathcal{C}}$, thus $\mathbf{B}^{inp}_{\mathcal{C}} = \mathrm{pres}_{\Sigma_1}(\mathbf{B}^{\Sigma}_{\mathcal{C}})$; $\mathbf{B}^{out}_{\mathcal{C}} = \mathbf{B}^{\Sigma_2}_{\mathcal{C}}$, thus $\mathbf{B}^{out}_{\mathcal{C}} = \mathrm{pres}_{\Sigma_2}(\mathbf{B}^{\Sigma}_{\mathcal{C}})$; $\mathbf{B}^{int}_{\mathcal{C}} = \mathbf{B}^{\Sigma_3}_{\mathcal{C}}$, thus $\mathbf{B}^{int}_{\mathcal{C}} = \mathrm{pres}_{\Sigma_3}(\mathbf{B}^{\Sigma}_{\mathcal{C}})$; $\mathbf{B}^{ext}_{\mathcal{C}} = \mathbf{B}^{\Sigma_1 \cup \Sigma_2}_{\mathcal{C}}$, thus $\mathbf{B}^{ext}_{\mathcal{C}} = \mathrm{pres}_{\Sigma_1 \cup \Sigma_2}(\mathbf{B}^{\Sigma}_{\mathcal{C}})$; $\mathbf{B}^{loc}_{\mathcal{C}} = \mathbf{B}^{\Sigma_2 \cup \Sigma_3}_{\mathcal{C}}$, thus $\mathbf{B}^{loc}_{\mathcal{C}} = \mathrm{pres}_{\Sigma_2 \cup \Sigma_3}(\mathbf{B}^{\Sigma}_{\mathcal{C}})$.

Next we consider the following component automata as variants of $\mathcal{C}$: $[\mathcal{C}, inp] = (Q, (\Sigma, \varnothing, \varnothing), \delta, I)$, $[\mathcal{C}, out] = (Q, (\varnothing, \Sigma, \varnothing), \delta, I)$, and $[\mathcal{C}, int] = (Q, (\varnothing, \varnothing, \Sigma), \delta, I)$.

Lemma 6.1.2. *Let $[\mathcal{C}, inp]$, $[\mathcal{C}, out]$, and $[\mathcal{C}, int]$ be as described above. Then*

(1) $\mathbf{B}^{\Sigma}_{\mathcal{C}} = \mathbf{B}^{inp}_{[\mathcal{C},inp]} = \mathbf{B}^{out}_{[\mathcal{C},out]} = \mathbf{B}^{int}_{[\mathcal{C},int]}$,

(2) $\mathbf{B}^{\Sigma}_{[\mathcal{C},inp]} = \mathbf{B}^{inp}_{[\mathcal{C},inp]} = \mathbf{B}^{ext}_{[\mathcal{C},inp]}$,

(3) $\mathbf{B}^{\Sigma}_{[\mathcal{C},out]} = \mathbf{B}^{out}_{[\mathcal{C},out]} = \mathbf{B}^{ext}_{[\mathcal{C},out]} = \mathbf{B}^{loc}_{[\mathcal{C},out]}$, *and*

(4) $\mathbf{B}^{\Sigma}_{[\mathcal{C},int]} = \mathbf{B}^{int}_{[\mathcal{C},int]} = \mathbf{B}^{loc}_{[\mathcal{C},int]}$.

Proof. (1) Let $alph \in \{inp, out, int\}$. Then $\mathbf{B}^{\Sigma}_{\mathcal{C}} = \mathbf{B}^{\Sigma}_{[\mathcal{C},alph]} = \text{pres}_{\Sigma}(\mathbf{B}^{\Sigma}_{[\mathcal{C},alph]})$ $= \mathbf{B}^{alph}_{[\mathcal{C},alph]}$.

(2) $\mathbf{B}^{\Sigma}_{[\mathcal{C},inp]} = \text{pres}_{\Sigma}(\mathbf{B}^{\Sigma}_{[\mathcal{C},inp]}) = \mathbf{B}^{inp}_{[\mathcal{C},inp]}$ and $\mathbf{B}^{ext}_{[\mathcal{C},inp]} = \text{pres}_{\Sigma\cup\varnothing}(\mathbf{B}^{\Sigma}_{[\mathcal{C},inp]})$ $= \text{pres}_{\Sigma}(\mathbf{B}^{\Sigma}_{[\mathcal{C},inp]})$.

(3,4) Analogous to (2). □

Now we denote $\mathsf{CA}^{alph} = \{\mathbf{B}^{alph}_{\mathcal{C}} \mid \mathcal{C}$ is a finite component automaton$\}$, with $alph \in \{inp, out, int, ext, loc\}$.

All languages in CA^{alph} are the images under a weak coding pres_{Σ} of languages in $\mathsf{CA} = \mathsf{pREG}$. It is known that pREG is closed under (weak) codings, i.e. whenever $L \in \mathsf{pREG}$ and L' is a (weak) coding of L, then we know that also $L' \in \mathsf{pREG}$. Using this closure of pREG under weak codings we immediately obtain the following result.

Lemma 6.1.3. *Let* $alph \in \{inp, out, int, ext, loc\}$. *Then*

$\mathsf{CA}^{alph} \subseteq \mathsf{pREG}$. □

Combining this lemma with Lemmata 6.1.1 and 6.1.2 leads to the following result, which shows that the distinction of the set of actions into input, output, and internal actions has no influence on the behavior of finite component automata.

Theorem 6.1.4. $\mathsf{pREG} = \mathsf{CA} = \mathsf{CA}^{inp} = \mathsf{CA}^{out} = \mathsf{CA}^{int} = \mathsf{CA}^{ext} = \mathsf{CA}^{loc}$. □

6.2 Team Behavior Versus Component Behavior

For the remainder of this chapter all component automata (and thus all team automata) have a possibly infinite set of states and a possibly infinite set of actions. We investigate the relation between the computations and behavior of team automata on the one hand, and those of their constituting component automata on the other hand. Since we know that subteams of a team automaton can be viewed as components of (an iterated version of) that team automaton, it suffices to study the relation between team automata and their constituting component automata.

We first continue our study started in Section 4.2. Given the computations (behavior) of a team automaton we investigate how to extract the computations (behavior) of its constituting component automata. Later we change focus and investigate how to combine the given computations (behavior) of a composable system in such a way that the resulting computations (behavior) are those of a team automaton over that composable system.

Initially we consider team automata in which all actions are *ai*. In such a team automaton, in every synchronization on a given action always the same component automata participate. The results we obtain in this case form a satisfying picture. Consequently we move on to consider team automata with only *free* actions. In such a team automaton — although depending on the state a component (team) automaton is in — in every synchonization on a given action always only one component automaton participates. Also in this case we obtain interesting results. Finally, in a team automaton with only *si* actions, the participation of component automaton in synchronizations is fully state dependent. We argue that a drastically different approach is required to obtain results in this case.

Notation 9. *Also in this chapter we once more assume a fixed, but arbitrary and possibly infinite index set $\mathcal{I} \subseteq \mathbb{N}$, which we will use to index the component automata involved. For each $i \in \mathcal{I}$, we let $\mathcal{C}_i = (Q_i, (\Sigma_{i,inp}, \Sigma_{i,out}, \Sigma_{i,int}), \delta_i, I_i)$ be a fixed component automaton and we use Σ_i to denote its set of actions $\Sigma_{i,inp} \cup \Sigma_{i,out} \cup \Sigma_{i,int}$. Moreover, we once more let $\mathcal{S} = \{\mathcal{C}_i \mid i \in \mathcal{I}\}$ be a fixed composable system and we let $\mathcal{T} = (Q, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{int}), \delta, I)$ be a fixed team automaton over $\mathcal{S}$. Furthermore, we use Σ to denote its set of actions $\Sigma_{inp} \cup \Sigma_{out} \cup \Sigma_{int}$. Recall that $\mathcal{I} \subseteq \mathbb{N}$ implies that $\mathcal{I}$ is ordered by the usual $\leq$ relation on $\mathbb{N}$, thus inducing an ordering on $\mathcal{S}$, and that the $\mathcal{C}_i$ are not necessarily different. Finally, we let Θ be an arbitrary but fixed alphabet disjoint from Q.* □

6.2.1 From Team Automata to Component Automata

In this subsection we assume that the computations and behavior of a team automaton are given. From these we want to extract computations and behavior of its constituting component automata. We start by addressing this issue element-wise, i.e. given one particular computation (behavior) of a team automaton, we want to know whether we can extract from it the underlying computation (behavior) of one of its constituting component automata.

Notation 10. *For the remainder of this section we let $j \in \mathcal{I}$.* □

Given a team computation $\alpha \in \mathbf{C}_{\mathcal{T}}^{\infty}$ we know from Corollary 4.2.7 that $\pi_{\mathcal{C}_j}(\alpha) \in \mathbf{C}_{\mathcal{C}_j}^{\infty}$. Hence we can simply apply projections on the computations of team automata in order to obtain computations of its constituting component automata. Moreover, by definition, $\text{pres}_{\Theta}(\alpha) \in \mathbf{B}_{\mathcal{T}}^{\Theta,\infty}$ and $\text{pres}_{\Theta}(\pi_{\mathcal{C}_j}(\alpha)) \in \mathbf{B}_{\mathcal{C}_j}^{\Theta,\infty}$. This reflects the fact that behavior is obtained by filtering out state information from computations. We thus have the situation depicted by the diagram in Figure 6.1.

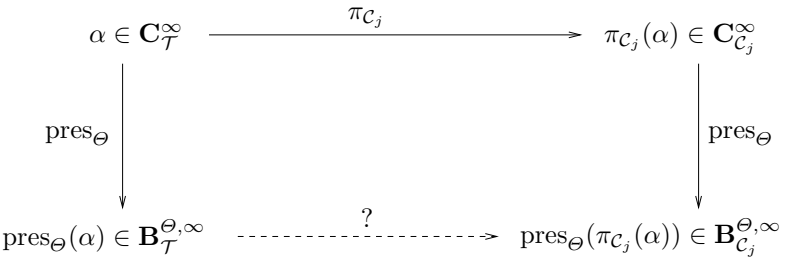

Fig. 6.1. Extracting behavior from team automata to component automata.

In addition we would like to obtain the Θ-behavior $\text{pres}_{\Theta}(\pi_{\mathcal{C}_j}(\alpha))$ of component automaton $\mathcal{C}_j$ directly from the Θ-behavior $\text{pres}_{\Theta}(\alpha)$ of team automaton $\mathcal{T}$. We thus look for an operation that makes the diagram of Figure 6.1 commute. A natural candidate is the homomorphism pres_{Σ_j} preserving only those actions from $\text{pres}_{\Theta}(\alpha)$ that belong to component automaton $\mathcal{C}_j$. Hence we wonder whether $\text{pres}_{\Sigma_j}(\text{pres}_{\Theta}(\alpha)) = \text{pres}_{\Theta}(\pi_{\mathcal{C}_j}(\alpha))$. In the following example we show that this equality in general does not hold.

Notation 11. *For the remainder of this section we may also specify our fixed component automata $\mathcal{C}_i$ as $(Q_i, \Sigma_i, \delta_i, I_i)$, $i \in \mathcal{I}$, and our fixed team automaton $\mathcal{T}$ as (Q, Σ, δ, I) whenever the distinctions of their alphabets into input, output, and internal actions are irrelevant.* □

Example 6.2.1. Let component automata $\mathcal{C}_1 = (\{q_1, q_1'\}, \{a, b\}, \{(q_1, b, q_1), (q_1, a, q_1')\}, \{q_1\})$ and $\mathcal{C}_2 = (\{q_2, q_2'\}, \{a, b\}, \{(q_2, a, q_2'), (q_2', b, q_2')\}, \{q_2\})$ be as depicted in Figure 6.2.

We assume $\{\mathcal{C}_1, \mathcal{C}_2\}$ to be a composable system and consider team automaton $\mathcal{T} = (Q, \{a, b\}, \delta, \{(q_1, q_2)\})$, with $Q = \{(q_1, q_2), (q_1, q_2'), (q_1', q_2), (q_1', q_2')\}$ and $\delta = \{((q_1, q_2), b, (q_1, q_2)), ((q_1, q_2), a, (q_1', q_2'))\}$, over this composable system. It is depicted in Figure 6.3(a).

Let $\alpha = (q_1, q_2)b(q_1, q_2)a(q_1', q_2') \in \mathbf{C}_{\mathcal{T}}$. Then $\text{pres}_{\Sigma_2}(\text{pres}_{\{a,b\}}(\alpha)) = ba \neq a = \text{pres}_{\{a,b\}}(q_2 a q_2') = \text{pres}_{\{a,b\}}(\pi_{\mathcal{C}_2}(\alpha))$. □

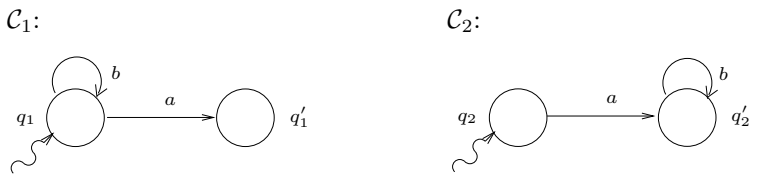

Fig. 6.2. Component automata $\mathcal{C}_1$ and $\mathcal{C}_2$.

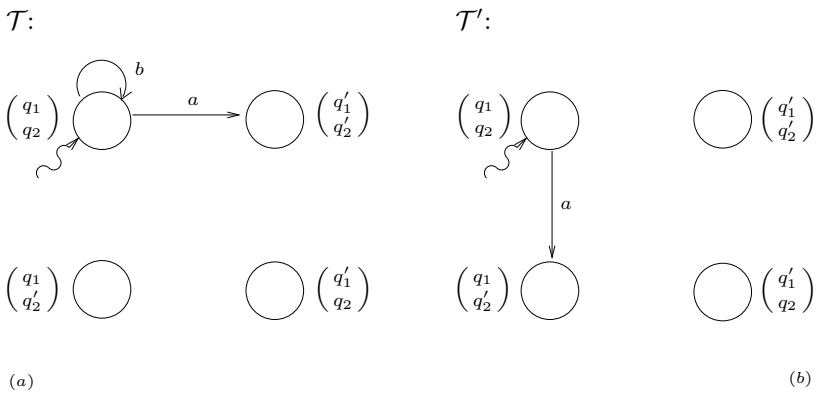

Fig. 6.3. Team automata $\mathcal{T}$ and $\mathcal{T}'$.

This example shows that in general we cannot assume that a component automaton participates in a synchronization, *just* because it has the action that is being synchronized as one of its actions. Hence there is no a priori relation between a component automaton's set of actions and its participation in synchronizations of those actions. The question we ask ourselves in this section now boils down to finding a necessary and sufficient condition which guarantees that $\text{pres}_{\Sigma_j}(\text{pres}_{\Theta}(\alpha)) = \text{pres}_{\Theta}(\pi_{\mathcal{C}_j}(\alpha))$.

As suggested by the example, we thus need to find a way to know whether or not a component automaton participates in a synchronization of the team automaton. It is therefore not surprising that the condition we present next is based on the *ai* principle, since every synchronization of an *ai* action involves all component automata that share this action. However, we obviously do not care about useless transitions as they can never be used anyway. It thus suffices to require the actions of $\mathcal{T}$ to be *ai* with respect to useful transitions only. Furthermore, for a given component $\mathcal{C}_j$ and action $a \in \Sigma_j$ it suffices to know that a is *ai with respect to* j, i.e. it is sufficient if $\mathcal{C}_j$ is required to participate in every useful a-transition of $\mathcal{T}$. This leads to the following definition.

Definition 6.2.2. *The set of* useful j-action-indispensable *actions is denoted by* $uAI_j(\mathcal{T})$ *and is defined as*

$$uAI_j(\mathcal{T}) = \{a \in \Sigma_j \mid \forall q, q' \in Q : (q, q') \in \delta_a \text{ is useful } \Rightarrow \\ proj_j{}^{[2]}(q, q') \in \delta_{j,a}\}. \ \square$$

Note that $AI(\mathcal{T}) \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$. We moreover note that whenever an action a of a component $\mathcal{C}_j$ is not active in $\mathcal{T}$, then $a \in uAI_j(\mathcal{T})$.

We can now formulate a sufficient condition under which the preserving homomorphism pres_{Σ_j} makes the diagram of Figure 6.1 commute. First we limit ourselves to finite computations.

Lemma 6.2.3. *If* $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$*, then for all* $\alpha \in \mathbf{C}_\mathcal{T}$*,* $pres_{\Sigma_j}(pres_\Theta(\alpha)) = pres_\Theta(\pi_{\mathcal{C}_j}(\alpha))$.

Proof. Let $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$ and let $\alpha = q_0 a_1 q_1 a_2 q_2 \cdots a_n q_n \in \mathbf{C}_\mathcal{T}$. By induction on n we prove $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha))$.

If $n = 0$, then $\alpha = q_0$ and thus $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(q_0)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(q_0)) = \lambda$.

Next assume that $n = k+1$, for some $k \geq 0$, and that $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\beta)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\beta))$, where $\beta = q_0 a_1 q_1 a_2 q_2 \cdots a_k q_k$. Hence $\alpha = \beta a_n q_n$. This implies that $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\beta))a_n$ if $a_n \in \Theta \cap \Sigma_j$ and $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\beta))$ if $a_n \notin \Theta \cap \Sigma_j$.

First consider that $a_n \in \Theta \cap \Sigma_j$. Then $\text{proj}_j{}^{[2]}(q_n, q_{n+1}) \in \delta_{j,a_n}$ since $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$ and thus $\text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\beta)a_n\text{proj}_j(q_{n+1})) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\beta))a_n = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\beta a_n q_n))$ by the induction hypothesis. Hence $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha))$.

Next consider that $a_n \notin \Theta \cap \Sigma_j$. Then $a_n \notin \Theta$ or $a_n \notin \Sigma_j$.
If $a_n \notin \Sigma_j$, then $\pi_{\mathcal{C}_j}(\alpha) = \pi_{\mathcal{C}_j}(\beta)$ and thus, by the induction hypothesis, $\text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\beta)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\beta))$. As $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\beta)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\beta a_n q_n))$ it follows that $\text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha))$.
If $a_n \notin \Theta$, then $\text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\beta))$ and thus, by the induction hypothesis, $\text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\beta)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha))$. $\square$

Next we allow also infinite computations.

Corollary 6.2.4. *If* $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$*, then for all* $\alpha \in \mathbf{C}^\infty_\mathcal{T}$*,* $pres_{\Sigma_j}(pres_\Theta(\alpha)) = pres_\Theta(\pi_{\mathcal{C}_j}(\alpha))$.

Proof. Let $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$ and let $\alpha \in \mathbf{C}^\infty_\mathcal{T}$. Due to Lemma 6.2.3 we only need to consider the infinite case. Hence we assume that $\alpha \in \mathbf{C}^\omega_\mathcal{T}$. Let $\alpha_1 \leq \alpha_2 \leq \cdots \in \mathbf{C}_\mathcal{T}$ be such that $\alpha = \lim_{n\to\infty} \alpha_n$. Thus $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\lim_{n\to\infty} \alpha_n))$. Then, by the definition of homomorphisms on infinite words, $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\lim_{n\to\infty} \alpha_n)) = \lim_{n\to\infty} \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha_n))$. Consequently,

by the same reason and from Lemma 6.2.3 it now follows that $\lim_{n\to\infty} \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha_n)) = \lim_{n\to\infty} \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha_n)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\lim_{n\to\infty} \alpha_n)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha))$. □

It turns out that the condition proposed above is also necessary.

Lemma 6.2.5. *If $(\Theta \cap \Sigma_j) \setminus uAI_j(\mathcal{T}) \neq \varnothing$, then there exists an $\alpha \in \mathbf{C}_\mathcal{T}$ such that $pres_{\Sigma_j}(pres_\Theta(\alpha)) \neq pres_\Theta(\pi_{\mathcal{C}_j}(\alpha))$.*

Proof. Let $(\Theta \cap \Sigma_j) \setminus uAI_j(\mathcal{T}) \neq \varnothing$. Then the following situation must exist. Let $\alpha = q_0 a_1 q_1 a_2 q_2 \cdots a_n q_n \in \mathbf{C}_\mathcal{T}$ be such that for all $1 \leq i < n$, either $a_i \notin \Theta$, or $a_i \notin \Sigma_j$, or $\text{proj}_j{}^{[2]}(q_{i-1}, q_i) \in \delta_{j,a_i}$, while $\text{proj}_j{}^{[2]}(q_{n-1}, q_n) \notin \delta_{j,a_n}$, with $a_n \in \Theta \cap \Sigma_j$. Hence $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha)) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(a_1 a_2 \cdots a_{n-1}))a_n$. Then $\text{proj}_j{}^{[2]}(q_{n-1}, q_n) \notin \delta_{j,a_n}$ however implies that $\text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(q_0 a_1 q_1 a_2 q_2 \cdots a_{n-1} q_{n-1})) \neq \text{pres}_{\Sigma_j}(\text{pres}_\Theta(a_1 a_2 \cdots a_{n-1}))a_n = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha))$. □

We thus conclude that the proposed condition is necessary and sufficient for the diagram of Figure 6.1 to commute.

Theorem 6.2.6. *For all $\alpha \in \mathbf{C}^\infty_\mathcal{T}$, $pres_{\Sigma_j}(pres_\Theta(\alpha)) = pres_\Theta(\pi_{\mathcal{C}_j}(\alpha))$ if and only if $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$.*

Proof. (Only if) This is the contrapositive of Lemma 6.2.5.
(If) Directly from Corollary 6.2.4. □

Summarizing, we thus have the following situation. Whenever $\mathcal{C}_j$ contains at least one action from Θ which is not useful j-action-indispensable in $\mathcal{T}$, then $\mathcal{T}$ can execute a computation α for which $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha))$ does not equal $\text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha))$ (cf. Lemma 6.2.5).

Until now we extracted the behavior of the component automata of a team automaton from the computations of this team automaton. The above results however also provide us with a sufficient condition for obtaining the behavior of component automaton $\mathcal{C}_j$ directly from the behavior of team automaton $\mathcal{T}$, viz. by simply applying pres_{Σ_j} to its behavior.

Theorem 6.2.7. *If $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$, then $\mathbf{B}^{\Theta\cap\Sigma_j,\infty}_\mathcal{T} \subseteq \mathbf{B}^{\Theta,\infty}_{\mathcal{C}_j}$.*

Proof. Let $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$ and let $v \in \mathbf{B}^{\Theta\cap\Sigma_j,\infty}_\mathcal{T}$. This means that $v \in \text{pres}_{\Theta\cap\Sigma_j}(\mathbf{C}^\infty_\mathcal{T})$. Now let $\alpha \in \mathbf{C}^\infty_\mathcal{T}$ be such that $\text{pres}_{\Theta\cap\Sigma_j}(\alpha) = v$. From Corollary 4.2.7 we know that $\pi_{\mathcal{C}_j}(\alpha) \in \mathbf{C}^\infty_{\mathcal{C}_j}$. Since $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$, Corollary 6.2.4 implies that $\text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha))$. Hence $v = \text{pres}_{\Theta\cap\Sigma_j}(\alpha) = \text{pres}_{\Sigma_j}(\text{pres}_\Theta(\alpha)) = \text{pres}_\Theta(\pi_{\mathcal{C}_j}(\alpha)) \in \mathbf{B}^{\Theta,\infty}_{\mathcal{C}_j}$. □

Note that Example 4.2.8 implies that it can still be the case that $\mathbf{B}_{\mathcal{T}}^{\Theta\cap\Sigma_j,\infty} \subset \mathbf{B}_{\mathcal{C}_j}^{\Theta,\infty}$, even if $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$.

Contrary to what might be expected from Theorem 6.2.6, the next example demonstrates that the statement from Theorem 6.2.7 cannot be reversed, i.e. $\Theta \cap \Sigma_j \subseteq uAI_j(\mathcal{T})$ is not a necessary condition for $\mathbf{B}_{\mathcal{T}}^{\Theta\cap\Sigma_j,\infty} \subseteq \mathbf{B}_{\mathcal{C}_j}^{\Theta,\infty}$ to hold. The reason is that the $\Theta \cap \Sigma_j$-behavior of $\mathcal{T}$ can be nonempty due to team computations in which $\mathcal{C}_j$ does not participate at all.

Example 6.2.8. (Example 6.2.1 continued) Consider team automaton $\mathcal{T}' = (Q, \{a, b\}, \{((q_1, q_2), a, (q_1, q_2'))\}, \{(q_1, q_2)\})$ over $\{\mathcal{C}_1, \mathcal{C}_2\}$. It is depicted in Figure 6.3(b).

Clearly $\mathbf{B}_{\mathcal{T}'}^{\Sigma,\infty} = \{\lambda, a\}$. Now let $\Theta = \{a, b\}$. Then $\Theta \cap \Sigma_1 = \{a, b\} \cap \{a, b\} = \{a, b\} \not\subseteq \{b\} = uAI_1(\mathcal{T}')$. However, $\mathbf{B}_{\mathcal{T}'}^{\Theta\cap\Sigma_1,\infty} = \mathbf{B}_{\mathcal{T}'}^{\{a,b\},\infty} = \{\lambda, a\} \subseteq \{b^n \mid n \geq 0\} \cup \{b^\omega\} \cup \{b^n a \mid n \geq 0\} = \mathbf{B}_{\mathcal{C}_1}^{\{a,b\},\infty} = \mathbf{B}_{\mathcal{C}_1}^{\Theta,\infty}$. □

Whereas a simple projection $\pi_{\mathcal{C}_j}$ applied to a computation of $\mathcal{T}$ suffices to obtain a computation of $\mathcal{C}_j$, a similarly simple preserving homomorphism pres_{Σ_j} applied to a behavior of $\mathcal{T}$ need not always yield a behavior of $\mathcal{C}_j$ *unless* all actions Σ_j of $\mathcal{C}_j$ are useful *j-ai*. The reason for this difference is as follows.

In a computation of $\mathcal{T}$ we still have available the information as to which components from $\mathcal{S}$ participated in each synchronization performed during this computation. When we deal with a behavior of $\mathcal{T}$, however, only the sequence of executed actions is available, i.e. we have lost all information as to which component automata from $\mathcal{S}$ participated in which execution. This implies that whenever we can be sure of a component automaton's participation in each execution of an action it has as an action itself, then we can simply apply our preserving homomorphism to the behavior of a team automaton in order to obtain the behavior of that component automaton.

Since every action of a component automaton from $\mathcal{S}$ is useful j-action-indispensable in the *maximal-ai* team automaton $\mathcal{T}$ over $\mathcal{S}$, Theorem 6.2.7 implies the following result. Slightly less general versions of this result, viz. without Θ being an arbitrary alphabet, have been formulated for other automata-based specification models with composition based on the *ai* principle (see, e.g., [Tut87] and [Jon87]). Theorems 6.2.6 and 6.2.7 however show a more precise condition guaranteeing this result and moreover exclude the existence of a similar relation in case composition is not based on the *ai* principle.

Theorem 6.2.9. *Let $\mathcal{T}$ be the $\mathcal{R}^{ai}$-team automaton over $\mathcal{S}$. Then*

$$\mathbf{B}_{\mathcal{T}}^{\Theta\cap\Sigma_j,\infty} \subseteq \mathbf{B}_{\mathcal{C}_j}^{\Theta,\infty}.$$ □

At this point it is important to recall that in case $\mathcal{S}$ is such that none of its constituting component automata shares an action, then the *maximal-free* team automaton over $\mathcal{S}$ equals the *maximal-ai* team automaton over $\mathcal{S}$ (cf. Theorem 4.5.5) — in which case this theorem can thus be applied.

This completes our display of how to obtain the computations (behavior) of component automata constituting $\mathcal{S}$ from the computations (behavior) of team automata over $\mathcal{S}$. In the next section we study the dual approach.

6.2.2 From Component Automata to Team Automata

Contrary to the previous subsection we now assume that the computations and behavior of a set of component automata are given. Consequently we want to use this information to describe computations and behavior of team automata that can be composed over that set of component automata. We start by addressing this issue element-wise, i.e. given a computation (behavior) of each component automaton in a subset of $\mathcal{S}$ we want to know whether there exists a team automaton over $\mathcal{S}$ with a computation (behavior) that *uses* this combination of computations.

Definition 6.2.10. *Let* $\alpha \in \prod_{i\in\mathcal{I}} \mathbf{C}^{\infty}_{\mathcal{C}_i}$. *Then*

> α *is* used in $\mathcal{T}$ *if there exists a* $\beta \in \mathbf{C}^{\infty}_{\mathcal{T}}$ *such that for all* $i \in \mathcal{I}$, $\pi_{\mathcal{C}_i}(\beta) = proj_i(\alpha)$. □

Note that any vector of initial states is used in $\mathcal{T}$ since $\prod_{i\in\mathcal{I}} I_i \subseteq \mathbf{C}^{\infty}_{\mathcal{T}}$. If $K \subseteq \mathcal{I}$ and $\alpha_k \in \mathbf{C}^{\infty}_{\mathcal{C}_k}$, for all $k \in K$, then we say that $\prod_{k\in K} \alpha_k$ is *used in* $\mathcal{T}$ whenever there exists a $\gamma \in \prod_{i\in\mathcal{I}} \mathbf{C}^{\infty}_{\mathcal{C}_i}$ that is used in $\mathcal{T}$ and which is such that $\mathrm{proj}_k(\gamma) = \alpha_k$, for all $k \in K$. Finally, as vectors $\prod_{\{j\}} \mathbf{C}^{\infty}_{\mathcal{C}_j}$ have one element we will also say that $\alpha \in \mathbf{C}^{\infty}_{\mathcal{C}_j}$ is used in $\mathcal{T}$ whenever $\prod_{\{j\}} \alpha$ is.

In the following example we show that in general not all vectors over computations of component automata from $\mathcal{S}$ are used in $\mathcal{T}$. It may be the case that a computation of a component automaton from $\mathcal{S}$ never participates in a team computation. Moreover, it may happen that a vector over two or more computations of component automata from $\mathcal{S}$ is not used as such in $\mathcal{T}$, even when each entry of this vector *is* used in $\mathcal{T}$.

Example 6.2.11. (Examples 6.2.1 and 6.2.8 continued) We immediately see that $\mathcal{C}_2$ has a computation $\alpha' = q_2 a q_2' b q_2' \in \mathbf{C}_{\mathcal{C}_2}$ that is not used in $\mathcal{T}$ since there exists no $\beta \in \mathbf{C}^{\infty}_{\mathcal{T}}$ such that $\pi_{\mathcal{C}_2}(\beta) = \alpha'$.

Next we consider the team automaton $\mathcal{T}''$ over $\{\mathcal{C}_1, \mathcal{C}_2\}$, which is obtained from team automaton $\mathcal{T}'$ as specified in Example 6.2.8 by adding transition $((q_1, q_2), a, (q_1', q_2))$ to its transition relation. It is depicted in Figure 6.4(a).

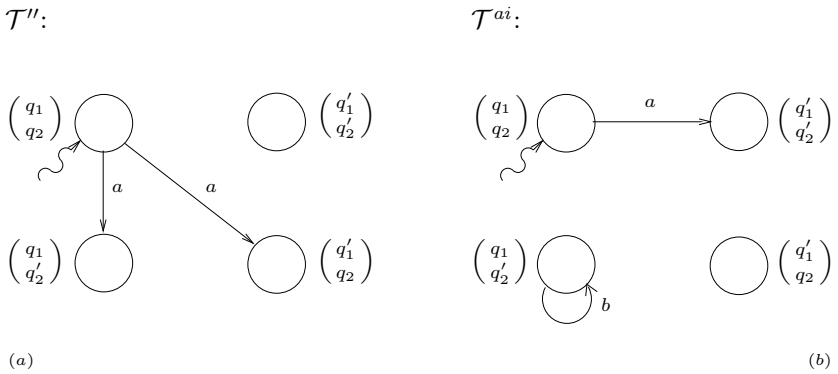

Fig. 6.4. Team automaton $\mathcal{T}''$ and *maximal-ai* team automaton $\mathcal{T}^{ai}$.

Clearly, both $\alpha_1 = q_1 a q_1' \in \mathbf{C}_{\mathcal{C}_1}$ and $\alpha_2 = q_2 a q_2' \in \mathbf{C}_{\mathcal{C}_2}$ are used in $\mathcal{T}''$ since $\beta_1 = (q_1, q_2)a(q_1', q_2) \in \mathbf{C}_{\mathcal{T}''}$ and $\beta_2 = (q_1, q_2)a(q_1, q_2') \in \mathbf{C}_{\mathcal{T}''}$ are such that $\pi_{\mathcal{C}_1}(\beta_1) = \alpha_1$ and $\pi_{\mathcal{C}_2}(\beta_2) = \alpha_2$. However, β_1 and β_2 are the only two nontrivial computations of $\mathcal{T}''$. Because $\pi_{\mathcal{C}_1}(\beta_2) = q_1$ and $\pi_{\mathcal{C}_2}(\beta_1) = q_2$ this means that there exists no $\beta \in \mathbf{C}^{\infty}_{\mathcal{T}''}$ such that $\pi_{\mathcal{C}_1}(\beta) = \alpha_1$ and $\pi_{\mathcal{C}_2}(\beta) = \alpha_2$. Hence (α_1, α_2) is not used in $\mathcal{T}''$.

Finally, note that (α_1, α_2) *is* used in team automaton $\mathcal{T}$ since $\beta = (q_1, q_2)a(q_1', q_2') \in \mathbf{C}_{\mathcal{T}}$ is such that $\pi_{\mathcal{C}_1}(\beta) = \text{proj}_1((\alpha_1, \alpha_2)) = \alpha_1$ and $\pi_{\mathcal{C}_2}(\beta) = \text{proj}_2((\alpha_1, \alpha_2)) = \alpha_2$. □

While in general not every vector over computations of component automata from $\mathcal{S}$ is used in $\mathcal{T}$, we wonder whether the situation improves in case $\mathcal{T}$ is defined in a particular way.

In analogy with the previous subsection, we first consider $\mathcal{T}$ to be such that all of its actions are *ai*. This is not yet enough, though, since whenever $\mathcal{T}$ has an empty transition relation, then all of its actions are *ai* while none of the computations of component automata from $\mathcal{S}$ is used in $\mathcal{T}$. Therefore we furthermore require $\mathcal{T}$ to be the *maximal-ai* team automaton over $\mathcal{S}$. However, in the following example we show that in general not all vectors over computations (behavior) of component automata from $\mathcal{S}$ are used in computations of the *maximal-ai* team automata over $\mathcal{S}$.

Example 6.2.12. (Example 6.2.11 continued) The *maximal-ai* team automaton over $\{\mathcal{C}_1, \mathcal{C}_2\}$ is $\mathcal{T}^{ai} = (Q, \{a, b\}, \delta^{ai}, \{(q_1, q_2)\})$, where $\delta^{ai} = \{((q_1, q_2), a, (q_1', q_2')), ((q_1, q_2'), b, (q_1, q_2'))\}$. It is depicted in Figure 6.4(b).

Trivially, $q_1 \in \mathbf{C}_{\mathcal{C}_1}$. However, since $(q_1, q_2)a(q_1', q_2')$ is the only nontrivial computation of $\mathcal{T}^{ai}$, there exists no computation $\beta' \in \mathbf{C}^{\infty}_{\mathcal{T}^{ai}}$ such that $\pi_{\mathcal{C}_1}(\beta') = q_1$ and $\pi_{\mathcal{C}_2}(\beta') = \alpha_2$. Hence (q_1, α_2) is not used in $\mathcal{T}^{ai}$. □

The fact that the *ai* type of synchronization forces component automata to synchronize on their shared actions provides us with enough information to formulate the conditions under which a vector of computations *is* used in a computation of the *maximal-ai* team automaton over $\mathcal{S}$. To this aim we define a vector α consisting of computations of the component automata from $\mathcal{S}$ — one for each such component automaton — to be *ai-consistent* if there exists a word w over Σ with the following property: whenever we preserve from w only the actions of a component automaton from $\mathcal{S}$, then we obtain exactly the behavior resulting from the computation in α that originates from that component automaton. In an *ai*-consistent vector the computations forming its entries thus "agree" with respect to the behavior of their respective components.

Definition 6.2.13. *Let $\alpha \in \prod_{i\in\mathcal{I}} \mathbf{C}^{\infty}_{\mathcal{C}_i}$. Then*

α is ai-consistent *if there exists a $w \in \Sigma^{\infty}$ such that for all $i \in \mathcal{I}$, $pres_{\Sigma_i}(w) = pres_{\Sigma_i}(proj_i(\alpha))$.* □

It turns out that each *ai*-consistent vector over computations of component automata from $\mathcal{S}$ is used in the *maximal-ai* team automaton $\mathcal{T}$ over $\mathcal{S}$.

Lemma 6.2.14. *Let $\mathcal{T}$ be the $\mathcal{R}^{ai}$-team automaton over $\mathcal{S}$ and let $\alpha \in \prod_{i\in\mathcal{I}} \mathbf{C}^{\infty}_{\mathcal{C}_i}$. Then*

if α is ai-consistent, then α is used in $\mathcal{T}$.

Proof. Let α be *ai*-consistent. Then by definition there exists a $w \in \Sigma^{\infty}$ such that $\mathrm{pres}_{\Sigma_i}(w) = \mathrm{pres}_{\Sigma_i}(\mathrm{proj}_i(\alpha))$, for all $i \in \mathcal{I}$.

First consider the case that $w \in \Sigma^*$. Let $w = a_1 a_2 \cdots a_n$ for some $n \geq 0$ and $a_k \in \Sigma$, for all $k \in [n]$. For each $i \in \mathcal{I}$, let the indices $i_1, i_2, \ldots, i_{n_i} \in [n]$ be such that $\mathrm{pres}_{\Sigma_i}(w) = a_{i_1} a_{i_2} \cdots a_{i_{n_i}}$. Hence $n_i = 0$ if $\mathrm{pres}_{\Sigma_i}(w) = \lambda$ and $1 \leq i_1 < i_2 < \cdots < i_{n_i} \leq n$ otherwise. Moreover, observe that $\bigcup_{i\in\mathcal{I}} \{i_1, i_2, \ldots, i_{n_i}\} = [n]$. Since for all $i \in \mathcal{I}$, $\mathrm{pres}_{\Sigma_i}(w) = \mathrm{pres}_{\Sigma_i}(\mathrm{proj}_i(\alpha))$ and $\mathrm{proj}_i(\alpha) \in \mathbf{C}_{\mathcal{C}_i}$, it follows that for all $i \in \mathcal{I}$, $\mathrm{proj}_i(\alpha) = q^i_0 a_{i_1} q^i_1 a_{i_2} \cdots a_{i_{n_i}} q^i_{n_i}$ with $q^i_0 \in I_i$ and $q^i_1, q^i_2, \ldots, q^i_{n_i} \in Q_i$.

Now define $\beta = q_0 a_1 q_1 a_2 \cdots a_n q_n$, with $q_k \in \prod_{i\in\mathcal{I}} Q_i$ for all $0 \leq k \leq n$, in such a way that for all $i \in \mathcal{I}$ and for all $0 \leq k \leq n$, $\mathrm{proj}_i(q_k) = q^i_\ell$ if $i_\ell \leq k < i_{\ell+1}$ with $\ell < n_i$ (by convention, $i_0 = 0$) and $\mathrm{proj}_i(q_k) = q^i_{n_i}$ if $i_{n_i} \leq k \leq n$. Consequently we prove that $\beta \in \mathbf{C}_{\mathcal{T}}$ while — in one stroke — $\pi_{\mathcal{C}_i}(\beta) = \mathrm{proj}_i(\alpha)$, for all $i \in \mathcal{I}$, follows from an inductive argument.

By its definition, $q_0 = \prod_{i\in\mathcal{I}} q^i_0 \in \prod_{i\in\mathcal{I}} I_i = I$. Next consider (q_{k-1}, a_k, q_k), for some $k \in [n]$. Let $i \in \mathcal{I}$. We distinguish the following two cases.

If $a_k \in \Sigma_i$, then $k = i_\ell$ for some $\ell \in [n_i]$ and $i_{\ell-1} \leq k-1 < k = i_\ell$. The definitions of q_{k-1} and q_k then yield $\text{proj}_i(q_{k-1}) = q^i_{\ell-1}$ and $\text{proj}_i(q_k) = q^i_\ell$. Since $\text{proj}_i(\alpha) \in \mathbf{C}_{\mathcal{C}_i}$ it follows that $(q^i_{\ell-1}, q^i_\ell) \in \delta_{i,a_{i_\ell}} = \delta_{i,a_k}$.

If $a_k \notin \Sigma_i$, then $k \neq i_\ell$ for some $\ell \in [n_i]$.
If $k < i_{n_i}$, then there exists an $\ell \geq 1$ such that $i_{\ell-1} \leq k-1 < k < i_\ell$ and thus $\text{proj}_i(q_{k-1}) = \text{proj}_i(q_k) = q^i_{\ell-1}$.
Conversely, if $k \geq i_{n_i}$, then $i_{n_i} \leq k-1 < k \leq n$ and thus again $\text{proj}_i(q_{k-1}) = \text{proj}_i(q_k)$.

Since $\bigcup_{i\in\mathcal{I}}\{i_1, i_2, \ldots, i_{n_i}\} = [n]$, it follows that $a_k \in \Sigma_i$ for at least one $i \in \mathcal{I}$ and hence $(q_{k-1}, q_k) \in \mathcal{R}^{ai}_{a_k}(\mathcal{S}) = \delta_{a_k}$. This implies that for all $k \in [n]$, $q_0a_1q_1a_2\cdots a_kq_k \in \mathbf{C}_\mathcal{T}$ and for all $i \in \mathcal{I}$, $\pi_{\mathcal{C}_i}(q_0a_1q_1a_2\cdots a_kq_k) \in \mathbf{C}_{\mathcal{C}_i}$. Hence for all $i \in \mathcal{I}$, $\pi_{\mathcal{C}_i}(\beta) = \pi_{\mathcal{C}_i}(q_0a_1q_1a_2\cdots a_nq_n) = \text{proj}_i(\alpha)$ and α is thus used in the *maximal-ai* team automaton $\mathcal{T}$.

Next consider the case that $w \in \Sigma^\omega$. Let $w = a_1a_2\cdots$, with $a_k \in \Sigma$ for all $k \geq 1$. Let $i \in \mathcal{I}$. For each $i \in \mathcal{I}$, if $\text{pres}_{\Sigma_i}(w) \in \Sigma_i^*$, then as before there are indices $i_1, i_2, \ldots, i_{n_i}$ such that $\text{pres}_{\Sigma_i}(w) = a_{i_1}a_{i_2}\cdots a_{i_{n_i}}$. Moreover, $\text{proj}_i(\alpha) = q^i_0a_{i_1}q^i_1a_{i_2}\cdots a_{i_{n_i}}q^i_{n_i}$. If $\text{pres}_{\Sigma_i}(w) \in \Sigma_i^\infty$, then there is an infinite sequence $1 \leq i_1 < i_2 < \cdots$ such that $\text{pres}_{\Sigma_i}(w) = a_{i_1}a_{i_2}\cdots$. Then because w is such that for all $i \in \mathcal{I}$, $\text{pres}_{\Sigma_i}(w) = \text{pres}_{\Sigma_i}(\text{proj}_i(\alpha))$, we can assume that $\text{proj}_i(\alpha) = q^i_0a_{i_1}q^i_1a_{i_2}\cdots$ for some $q^i_k \in Q_i$, with $k \geq 0$.

Now we define $\beta = q_0a_1q_1a_2\cdots$ such that for all $i \in \mathcal{I}$, $\pi_{\mathcal{C}_i}(q_0) = q^i_0$ and $\pi_{\mathcal{C}_i}(q_k) = q^i_\ell$, for $i_\ell \leq k < i_{\ell+1}$ and $\ell \geq 1$. Clearly $q_0 \in I$. Similar to the finitary case it can now be shown that $(q_{k-1}, a_k, q_k) \in \delta$, for all $k \geq 1$.

Hence $\beta \in \mathbf{C}^\omega_\mathcal{T}$ and, moreover, $\pi_{\mathcal{C}_i}(\beta) = \text{proj}_i(\alpha)$, for all $i \in \mathcal{I}$. □

From the proof of this lemma we immediately obtain the following result. Corresponding versions of both these results have been formulated for other automata-based specification models with composition based on the *ai* principle (see, e.g., [Tut87] and [Jon87]).

Corollary 6.2.15. *Let $\mathcal{T}$ be the $\mathcal{R}^{ai}$-team automaton over $\mathcal{S}$ and let $\alpha \in \prod_{i\in\mathcal{I}} \mathbf{C}^\infty_{\mathcal{C}_i}$. Then*

if $w \in \Sigma^\infty$ is such that for all $i \in \mathcal{I}$, $pres_{\Sigma_i}(w) = pres_{\Sigma_i}(proj_i(\alpha))$, then there exists a $\beta \in \mathbf{C}^\infty_\mathcal{T}$ such that $pres_\Sigma(\beta) = w$. □

We thus see that *ai*-consistency is a sufficient condition for a vector over computations of component automata from $\mathcal{S}$ to be used in the *maximal-ai* team automaton over $\mathcal{S}$. Next we show that this condition is also necessary.

Theorem 6.2.16. *Let $\mathcal{T}$ be the $\mathcal{R}^{ai}$-team automaton over $\mathcal{S}$ and let $\alpha \in \prod_{i\in\mathcal{I}} \mathbf{C}^\infty_{\mathcal{C}_i}$. Then*

α is used in $\mathcal{T}$ if and only if α is ai-consistent.

Proof. (If) This is Lemma 6.2.14.

(Only if) Let $\beta \in \mathbf{C}^{\infty}_{\mathcal{T}}$ be such that $\pi_{\mathcal{C}_i}(\beta) = \mathrm{proj}_i(\alpha)$, for all $i \in \mathcal{I}$. Since every action of $\mathcal{T}$ is *ai*, we can now apply Corollary 6.2.4 to obtain $\mathrm{pres}_{\Sigma_i}(\mathrm{pres}_{\Sigma}(\beta)) = \mathrm{pres}_{\Sigma}(\pi_{\mathcal{C}_i}(\beta)) = \mathrm{pres}_{\Sigma_i}(\pi_{\mathcal{C}_i}(\beta)) = \mathrm{pres}_{\Sigma_i}(\mathrm{proj}_i(\alpha))$, for all $i \in \mathcal{I}$. Hence α is *ai*-consistent. □

In order to formulate a general result relating the computations of *maximal-ai* team automata to the computations of their constituting component automata, we now define when a composable system is *ai*-consistent.

Definition 6.2.17. *$\mathcal{S}$ is* ai-consistent *if for all $i \in \mathcal{I}$ and for each $\gamma \in \mathbf{C}^{\infty}_{\mathcal{C}_i}$ there exists an ai-consistent vector $\alpha \in \prod_{i\in\mathcal{I}} \mathbf{C}^{\infty}_{\mathcal{C}_i}$ such that $proj_i(\alpha) = \gamma$.* □

Note that we have now defined *ai*-consistency both for vectors (over computations) and for composable systems. However, from the context it will always be clear whether we deal with an *ai*-consistent vector or rather with an *ai*-consistent composable system.

An *ai*-consistent composable system thus guarantees that for all computations of its constituents there exists a vector over such computations which is *ai*-consistent and thus each computation of a component automaton from $\mathcal{S}$ is used in a computation of the *maximal-ai* team automaton $\mathcal{T}$ over $\mathcal{S}$. In that case the set of computations (behavior) of a component automaton from $\mathcal{S}$ thus equals the set of computations (behavior) of the *maximal-ai* team automaton over $\mathcal{S}$ *projected* on that component automaton.

Theorem 6.2.18. *Let $\mathcal{T}$ be the $\mathcal{R}^{ai}$-team automaton over $\mathcal{S}$. Then*

(1) $\mathbf{C}^{\infty}_{\mathcal{C}_i} = \pi_{\mathcal{C}_i}(\mathbf{C}^{\infty}_{\mathcal{T}})$, for all $i \in \mathcal{I}$, if and only if $\mathcal{S}$ is ai-consistent, and

(2) if $\mathcal{S}$ is ai-consistent, then $\mathbf{B}^{\Sigma_i,\infty}_{\mathcal{C}_i} = \mathbf{B}^{\Sigma_i,\infty}_{\mathcal{T}}$, for all $i \in \mathcal{I}$.

Proof. (1) (Only if) Let $\mathbf{C}^{\infty}_{\mathcal{C}_i} = \pi_{\mathcal{C}_i}(\mathbf{C}^{\infty}_{\mathcal{T}})$, for all $i \in \mathcal{I}$. Let $\gamma \in \mathbf{C}^{\infty}_{\mathcal{C}_k}$ for some $k \in \mathcal{I}$. Since $\mathbf{C}^{\infty}_{\mathcal{C}_k} = \pi_{\mathcal{C}_k}(\mathbf{C}^{\infty}_{\mathcal{T}})$ there exists a $\beta \in \mathbf{C}^{\infty}_{\mathcal{T}}$ such that $\pi_{\mathcal{C}_k}(\beta) = \gamma$. Now let $\alpha = \prod_{i\in\mathcal{I}} \pi_{\mathcal{C}_i}(\beta)$. Since $\pi_{\mathcal{C}_i}(\mathbf{C}^{\infty}_{\mathcal{T}}) = \mathbf{C}^{\infty}_{\mathcal{C}_i}$, for all $i \in \mathcal{I}$, it follows that $\alpha \in \prod_{i\in\mathcal{I}} \mathbf{C}^{\infty}_{\mathcal{C}_i}$. Furthermore, α is used and thus, by Theorem 6.2.16, α is *ai*-consistent. Definition 6.2.17 then implies that $\mathcal{S}$ is *ai*-consistent.

(If) Due to Corollary 4.2.7 we only need to prove that whenever $\mathcal{S}$ is *ai*-consistent, then for all $i \in \mathcal{I}$, $\mathbf{C}^{\infty}_{\mathcal{C}_i} \subseteq \pi_{\mathcal{C}_i}(\mathbf{C}^{\infty}_{\mathcal{T}})$. Now let $\gamma \in \mathbf{C}^{\infty}_{\mathcal{C}_k}$ for some $k \in \mathcal{I}$. Since $\mathcal{S}$ is *ai*-consistent there exists an *ai*-consistent vector $\alpha \in \prod_{i\in\mathcal{I}} \mathbf{C}^{\infty}_{\mathcal{C}_i}$ such that $\mathrm{proj}_k(\alpha) = \gamma$. Then by Theorem 6.2.16 there exists a $\beta \in \mathbf{C}^{\infty}_{\mathcal{T}}$ such that $\pi_{\mathcal{C}_k}(\beta) = \mathrm{proj}_k(\alpha) = \gamma$. Hence $\gamma \in \pi_{\mathcal{C}_k}(\mathbf{C}^{\infty}_{\mathcal{T}})$.

(2) Let $k \in \mathcal{I}$. Since $\mathcal{T}$ is the $\mathcal{R}^{ai}$-team automaton over $\mathcal{S}$, Theorem 6.2.9 implies that $\mathbf{B}_{\mathcal{T}}^{\Sigma_k,\infty} \subseteq \mathbf{B}_{\mathcal{C}_k}^{\infty}$. Moreover, by (1) and Corollary 6.2.4, $\mathbf{B}_{\mathcal{C}_k}^{\infty} \subseteq \mathbf{B}_{\mathcal{T}}^{\Sigma_k,\infty}$. □

Next we move on to the case that our team automaton $\mathcal{T}$ under consideration is the *maximal-free* team automaton over $\mathcal{S}$. Hence $\mathcal{T}$ consists of completely independent, non-synchronizing component automata. Consequently, our first intuition might be to jump to the conclusion that *every* single computation of a component automaton from $\mathcal{S}$ is used in $\mathcal{T}$.

As we have seen in Section 4.6, however, $\mathcal{T}$ does have one tricky characteristic in case loops are present in the component automata from $\mathcal{S}$: the combination of a loop, e.g. on a, in one component automaton from $\mathcal{S}$ and an a-transition in another component automaton from $\mathcal{S}$ results in the latter of these a-transitions not being omnipresent in $\mathcal{T}$. This implies that even if this a-transition is useful in its component automaton, i.e. it is part of a computation of that component automaton, then it is not at all guaranteed that this computation is used in $\mathcal{T}$. The reason for this is the maximal interpretation of the participation of transitions from component automata in transitions of team automata that we adopted in Section 4.2.

Indeed, in the following example we show that in general not each computation of a component automata from $\mathcal{S}$ is used in the *maximal-free* team automaton over $\mathcal{S}$.

Example 6.2.19. Let component automata $\mathcal{C} = (\{p\}, \{b\}, \{(p, b, p)\}, \{p\})$ and $\mathcal{C}' = (\{q, r\}, \{b\}, \{(q, b, r)\}, \{q\})$ be as depicted in Figure 6.5(a).

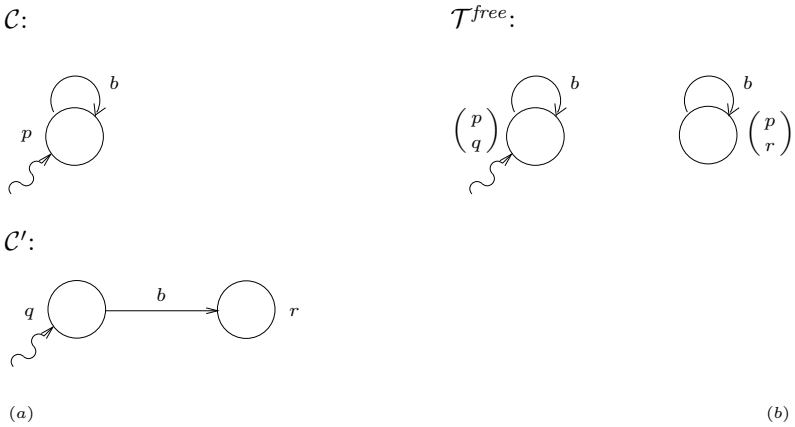

Fig. 6.5. Component automata $\mathcal{C}$ and $\mathcal{C}'$, and *maximal-free* team automaton $\mathcal{T}^{free}$.

Obviously $\{\mathcal{C}, \mathcal{C}'\}$ is a composable system. The $\mathcal{R}^{free}$-team automaton over $\{\mathcal{C}, \mathcal{C}'\}$ is $\mathcal{T}^{free} = (\{(p,q),(p,r)\}, \{b\}, \{((p,q),b,(p,q)),((p,r),b,(p,r))\}, \{(p,q)\})$. It is depicted in Figure 6.5(b).

It is clear that $\alpha = qbr \in \mathbf{C}_{\mathcal{C}'}$ and that there does not exist a computation $\beta \in \mathbf{C}^{\infty}_{\mathcal{T}^{free}}$ such that $\pi_{\mathcal{C}'}(\beta) = \alpha$. Hence α is not used in $\mathcal{T}^{free}$. □

In case we only deal with a specific type of component automata, however, we can use Theorem 4.6.10(2). Recall that, given that $\mathcal{S}$ is j-loop limited and that $\mathcal{T}$ is the *maximal-free* team automaton over $\mathcal{S}$, this theorem states that every transition of $\mathcal{C}_j$ is omnipresent in $\mathcal{T}$. This means that whenever (p,a,p') is a transition of $\mathcal{C}_j$, then for all states q of $\mathcal{T}$ for which $\text{proj}_j(q) = p$, there exists a transition (q,a,q') in $\mathcal{T}$ such that $\text{proj}_j(q') = p'$, i.e. in which (p,a,p') is participating. Since $\mathcal{T}$ is the *maximal-free* team automaton over $\mathcal{S}$ we moreover know that $\text{proj}_i(q') = \text{proj}_i(q)$, for all $i \in \mathcal{I} \setminus \{j\}$, i.e. (p,a,p') is the only participating transition. It thus comes as no surprise that in case $\mathcal{S}$ is j-loop limited, each computation of a component automaton from $\mathcal{S}$ *is* used in a computation of the *maximal-free* team automaton over $\mathcal{S}$.

Lemma 6.2.20. *Let $\mathcal{T}$ be the $\mathcal{R}^{free}$-team automaton over $\mathcal{S}$ and let $\alpha \in \mathbf{C}^{\infty}_{\mathcal{C}_j}$. Then*

if $\mathcal{S}$ is j-loop limited, then α is used in $\mathcal{T}$.

Proof. Let $\mathcal{S}$ be j-loop limited.

First consider the case that $\alpha \in \mathbf{C}_{\mathcal{C}_j}$. Let $\alpha = p_0 a_1 p_1 a_2 \cdots a_n p_n$ for some $n \geq 0$, i.e. $(p_{k-1}, p_k) \in \delta_{j,a_k}$, for all $1 \leq k \leq n$. Since $Q = \prod_{i \in \mathcal{I}} Q_i$ and $I = \prod_{i \in \mathcal{I}} I_i$, Theorem 4.6.10(2) implies that there exists a computation $\beta = q_0 a_1 q_1 a_2 \cdots a_n q_n \in \mathbf{C}_{\mathcal{T}}$ such that $\text{proj}_j{}^{[2]}(q_{k-1}, q_k) = (p_{k-1}, p_k) \in \delta_{j,a_k}$, for all $1 \leq k \leq n$. Hence $\pi_{\mathcal{C}_j}(\beta) = \alpha$ and α is thus used in $\mathcal{T}$.

Secondly, the case that $\alpha \in \mathbf{C}^{\omega}_{\mathcal{C}_j}$ is analogous to the finitary case. □

We thus see that loop limitedness is a sufficient condition for a vector over computations of component automata from $\mathcal{S}$ to be used in the *maximal-free* team automaton over $\mathcal{S}$. We will soon see that this condition is not necessary.

From Corollary 4.2.7 we know that given a computation of a team automaton $\mathcal{T}$ over $\mathcal{S}$, the projection on a component automaton from $\mathcal{S}$ is included in the set of computations of that component automaton. Together with Lemma 6.2.20 this implies that whenever $\mathcal{S}$ is j-loop limited, then the set of computations of a component automaton from $\mathcal{S}$ equals the set of computations of the *maximal-free* team automaton $\mathcal{T}$ over $\mathcal{S}$ *projected* on that component automaton. Moreover, the behavior of that component automaton is included in the behavior of $\mathcal{T}$. Like the proof of Lemma 6.2.20, also

the proof of this statement is based on the observation that in a *maximal-free* team automaton, each executed action has only one participating component automaton. This implies that the team automaton can always execute any computation of any of its constituting component automata while keeping all remaining component automata from $\mathcal{S}$ in an initial state.

Theorem 6.2.21. *Let $\mathcal{T}$ be the $\mathcal{R}^{free}$-team automaton over $\mathcal{S}$. Then*

if $\mathcal{S}$ is loop limited, then $\mathbf{C}^{\infty}_{\mathcal{C}_i} = \pi_{\mathcal{C}_i}(\mathbf{C}^{\infty}_{\mathcal{T}})$ and $\mathbf{B}^{\Sigma_i,\infty}_{\mathcal{C}_i} \subseteq \mathbf{B}^{\Sigma,\infty}_{\mathcal{T}}$, for all $i \in \mathcal{I}$.

Proof. Let $\mathcal{S}$ be loop limited. Then Lemma 6.2.20 implies that $\mathbf{C}^{\infty}_{\mathcal{C}_i} \subseteq \pi_{\mathcal{C}_i}(\mathbf{C}^{\infty}_{\mathcal{T}})$ and thus, by Corollary 4.2.7, $\mathbf{C}^{\infty}_{\mathcal{C}_i} = \pi_{\mathcal{C}_i}(\mathbf{C}^{\infty}_{\mathcal{T}})$. Now let $k \in \mathcal{I}$, let $\alpha \in \mathbf{B}^{\Sigma_k,\infty}_{\mathcal{C}_k}$ and let $\beta \in \mathbf{C}^{\infty}_{\mathcal{C}_k}$ be such that $\mathrm{pres}_{\Sigma_k}(\beta) = \alpha$. Since $\mathbf{C}^{\infty}_{\mathcal{C}_k} = \pi_{\mathcal{C}_k}(\mathbf{C}^{\infty}_{\mathcal{T}})$, there must exist a $\gamma \in \mathbf{C}^{\infty}_{\mathcal{T}}$ such that $\beta = \pi_{\mathcal{C}_k}(\gamma)$. Moreover, since by Theorem 4.6.10(2) all transitions of $\mathcal{C}_k$ are omnipresent in $\mathcal{T}$, it follows that we may assume that $\pi_{\mathcal{C}_\ell}(\gamma) \in I_\ell$, for all $\ell \in \mathcal{I} \setminus \{k\}$. Hence $\mathrm{pres}_{\Sigma}(\gamma) = \mathrm{pres}_{\Sigma}(\pi_{\mathcal{C}_k}(\gamma)) = \mathrm{pres}_{\Sigma_k}(\beta) = \alpha$ and thus $\alpha \in \mathbf{B}^{\Sigma,\infty}_{\mathcal{T}}$. □

The behavior of the *maximal-free* team automaton $\mathcal{T}$ over $\mathcal{S}$ trivially is made up of the behavior of not just one component automaton from $\mathcal{S}$, but of the behavior of all of the component automata from $\mathcal{S}$. Therefore, in general $\mathbf{B}^{\Sigma_j,\infty}_{\mathcal{C}_j} \subseteq \mathbf{B}^{\Sigma,\infty}_{\mathcal{T}}$ will be proper, even if $\mathcal{S}$ is j-loop limited. Furthermore, the fact that $\mathbf{C}^{\infty}_{\mathcal{C}_i} = \pi_{\mathcal{C}_i}(\mathbf{C}^{\infty}_{\mathcal{T}})$, for all $i \in \mathcal{I}$, need not imply that $\mathcal{S}$ is loop limited.

Example 6.2.22. (Example 6.2.11 continued) The *maximal-free* team automaton over $\{\mathcal{C}_1, \mathcal{C}_2\}$ is $\mathcal{T}^{free} = (\, Q,\ \{\, a, b \,\},\ \delta^{free},\ \{\, (q_1, q_2) \,\}\,)$, where $\delta^{free} = \{\, ((q_1,q_2),b,(q_1,q_2)\,),\ ((q_1,q_2),a,(q_1,q_2')\,),\ ((q_1,q_2),a,(q_1',q_2)\,),\ ((q_1,q_2'),a,(q_1',q_2')\,),\ ((q_1',q_2),a,(q_1',q_2')\,),\ ((q_1',q_2'),b,(q_1',q_2')\,)\,\}$. It is depicted in Figure 6.6(a).

Since $\beta = (q_1,q_2)a(q_1,q_2')a(q_1',q_2')b(q_1',q_2') \in \mathbf{C}_{\mathcal{T}^{free}}$, α' is used in $\mathcal{T}^{free}$. It is moreover not difficult to see that for all $k \in [2]$, $\mathbf{C}^{\infty}_{\mathcal{C}_k} \subseteq \pi_{\mathcal{C}_k}(\mathbf{C}^{\infty}_{\mathcal{T}^{free}})$ and thus, by Corollary 4.2.7, $\mathbf{C}^{\infty}_{\mathcal{C}_k} = \pi_{\mathcal{C}_k}(\mathbf{C}^{\infty}_{\mathcal{T}^{free}})$. However, $\{\mathcal{C}_1, \mathcal{C}_2\}$ is not loop limited because $(q_1, q_1) \in \delta_{1,b}$ and $(q_2', q_2') \in \delta_{2,b}$. □

Note that Theorem 6.2.21 relies heavily on the fact that in the *maximal-free* team automaton over a loop-limited $\mathcal{S}$, each action of a component automaton can be executed independently of the current local states that the other component automata of $\mathcal{S}$ are in, since none of these other component automata participates in such an execution. In the *maximal-ai* team automaton over $\mathcal{S}$, this situation can only occur when none of the other component automata from $\mathcal{S}$ contains any of the actions that was executed as part of the computation of the *maximal-ai* team automaton. Hence even when $\mathcal{S}$ is *ai*-consistent, then the behavior of $\mathcal{C}_j$ is in general not contained in the behavior

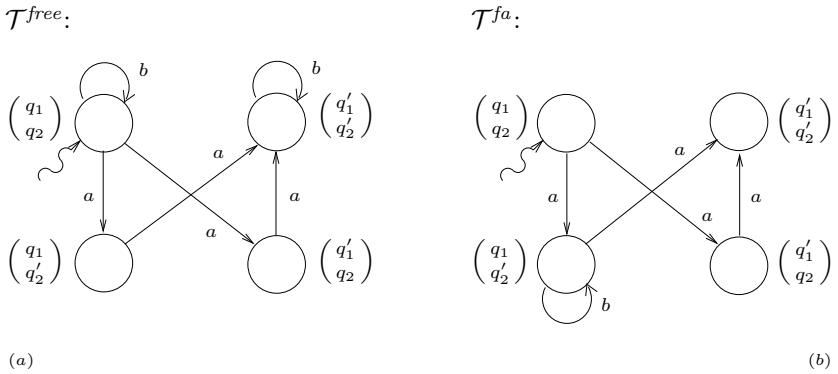

Fig. 6.6. Team automata $\mathcal{T}^{free}$ and $\mathcal{T}^{fa}$.

of the *maximal-ai* team automaton over $\mathcal{S}$. From Theorem 6.2.18(2) we however know that if $\mathcal{S}$ is *ai*-consistent, then the behavior of $\mathcal{C}_j$ *is* contained in the Σ_j-behavior of the *maximal-ai* team automaton over $\mathcal{S}$.

Both for *maximal-ai* team automata (cf. Theorem 6.2.18(1)) and for *maximal-free* team automata (cf. Theorem 6.2.21) over $\mathcal{S}$ we have thus found a sufficient condition on $\mathcal{S}$ (*ai*-consistency and loop limitedness, respectively) under which all component computations contribute to team computations. In case of *maximal-ai* team automata this condition is even necessary. As direct consequences of these results we have subsequently been able to relate the behavior of component automata to that of *maximal-ai* team automata (cf. Theorem 6.2.18(2)) and to that of *maximal-free* team automata (cf. Theorem 6.2.21).

In the remainder of this chapter we moreover define the behavior of the *maximal-ai* (*maximal-free*) team automaton over $\mathcal{S}$ in terms of the behavior of its constituting component automata. This requires establishing which combinations of words — if any — from the behavior of component automata from $\mathcal{S}$ can be combined — and in particular how — such that a word from the behavior of the *maximal-ai* (*maximal-free*) team automaton over $\mathcal{S}$ results. For this we use *shuffling* operations, known from the theory of formal languages. We will consider both "free" shuffles (to deal with *free* actions) and "synchronized" shuffles (to deal with *ai* actions).

In the succeeding two sections we formally define the different kinds of shuffles and study some of their properties. In the subsequent and final section of this chapter we then show that the behavior of team automata constructed on the basis of *maximal-ai* and/or *maximal-free* synchronizations can be expressed as a (synchronized) shuffle of the behavior of their consti-

tuting component automata, where the kind of shuffle depends on the type of synchronization.

The two sections dealing with various kinds of shuffles are rather technical and relatively extensive. One of the main reasons for this resides in the fact that our team automata framework allows for infinite computations and infinite behavior. Therefore we need to consider shuffles on finite as well as infinite words. Moreover, when dealing with composable systems consisting of two or more component automata, notions of commutativity and associativity for the various kinds of shuffles are of crucial importance. Readers interested only in the results can jump to the final section of this chapter and when necessary skim Subsections 6.3.1, 6.3.4, 6.4.1, and 6.4.4 for the definitions needed to interpret the results.

6.3 Shuffles

This section marks the beginning of our exposition on shuffles. The idea behind a shuffle of languages is the creation of a new language, the words of which consist of the words of the original languages "woven" in a particular fashion. For one, words of the original languages are part of the words of the new language. Consider, e.g., the (finite) words *eae* and *wv*. Then we can weave these words into a new (finite) word *weave*. To the best of our knowledge, the oldest reference to this way of shuffling two (finite) words is [GS65], which was presented at a conference as early as 1964.

In this simple example we described a shuffle of two finite words. We know, however, that the languages of our component (team) automata may contain infinite words. When we try to shuffle two infinite words in the abovementioned way we are forced to take some decisions concerning "fairness". Consider, e.g., the words a^ω and b^ω. Then we can weave these words into new (infinite) words of the form $(a^+b^+)^\omega$, consisting of both infinitely many a's and infinitely many b's. Hence a^ω and b^ω are woven in a fair way: finite nonempty subwords of the two words occur alternatingly, i.e. each word gets its fair turn in the new words. However, we could also decide to allow infinite subwords of the original words to appear in the new word. In that case a result of weaving a^ω and b^ω can be an (infinite) word from $(a^+b^+)^*a^\omega$. Note, however, that in this case the result does not contain an infinite number of b's. The oldest reference — again, to the best of our knowledge — to this idea of shuffling two infinite words is [Sha78], and to this idea of fair shuffling is [Par79] (where fair shuffling is called fair merging, though).

These simplified examples suggest that there is a clear need to define precisely and unambiguously what types of shuffles we shall consider. Another

reason for being more precise is to avoid the confusion that could arise from the fact that the (fair) shuffle is a well-known language-theoretic operation. It thus has a long history within theoretical computer science, in particular within formal language theory. Shuffling is sometimes called interleaving, weaving, or merging, and — given two words u and v — it may be denoted by $u \odot v$, $u \mid\mid v$, $u \sqcup\!\sqcup v$, $u \square v$, $u \otimes v$, $u \mid\mid\mid v$, or $u \diamond v$ (see, e.g., [GS65], [Sha78], [Par79], [Gis81], [Jan81], [BÉ96], [RS97]). The idea of shuffling also appears in numerous other disguises throughout the computer science literature. Within concurrency theory, e.g., as a semantics of parallel operators modeling communication between processes (see, e.g., [Ros97] and [BPS01]). In the next section we will consider also shuffles which are not merely interleavings, but which may require the synchronized occurrence of certain symbols.

The remainder of this section and the subsequent section together form a self-contained theory of shuffles. By no means do we claim that all results are new. We are aware of the fact that some results have appeared in the literature, but we have been unable to find a comprehensive theory of shuffles in the literature that would suit our approach. Due to the generic setup of the team automaton model we need to be able to deal with shuffles of infinite words and, moreover, we need several specific shuffles that are combinations of shuffling and synchronizing. Most of this has largely gone unexplored in the literature.

In this section we introduce the basic shuffle, well-known from the literature. We study its basic properties and prove its commutativity and associativity. In the subsequent section we consequently introduce three more intricate types of shuffles, built on top of the basic shuffle. We briefly study also their properties and establish notions of commutativity and associativity also for these types of shuffles. The fact that all four shuffles satisfy some sort of commutativity and associativity is crucial for applying them in the context of team automata in the final section of this chapter.

6.3.1 Definitions

We begin by introducing the basic shuffle.

Definition 6.3.1. *Let $u, v \in \Delta^\infty$. Then*

a word $w \in \Delta^\infty$ is a shuffle *of u and v if one of the following four cases holds. Either*

(1) $u, v \in \Delta^*$ and $w = u_1 v_1 u_2 v_2 \cdots u_n v_n$, where $n \geq 1$, $u_1 \in \Delta^*$, $u_2, u_3, \ldots, u_n, v_1, v_2, \ldots, v_{n-1} \in \Delta^+$, $v_n \in \Delta^*$, $u_1 u_2 \cdots u_n = u$, and $v_1 v_2 \cdots v_n = v$, or

(2) $u \in \Delta^*$, $v \in \Delta^\omega$, *and* $w = u_1v_1u_2v_2\cdots u_nv_n$, *where* $n \geq 1$, $u_1 \in \Delta^*$, $u_2, u_3, \ldots, u_n, v_1, v_2, \ldots, v_{n-1} \in \Delta^+$, $v_n \in \Delta^\omega$, $u_1u_2\cdots u_n \in pref(u)$, *and* $v_1v_2\cdots v_n = v$, *or*

(3) $u \in \Delta^\omega$, $v \in \Delta^*$, *and* w *is a shuffle of* v *and* u, *or*

(4) $u, v \in \Delta^\omega$ *and either*

(a) w *is a shuffle of* u *and a prefix of* v, *or*

(b) w *is a shuffle of* v *and a prefix of* u, *or*

(c) $w = u_1v_1u_2v_2\cdots$, *where* $u_1 \in \Delta^*$, $u_j, v_1, v_j \in \Delta^+$ *for all* $j \geq 2$, $u = \lim_{n\to\infty} u_1u_2\cdots u_n$, *and* $v = \lim_{n\to\infty} v_1v_2\cdots v_n$.

A shuffle w *of* u *and* v *is called* fair *(w.r.t.* u *and* v*) if* u *and* v *are finite (case (1)), or if in case (2)* $u_1u_2\cdots u_n = u$, *or if in case (3)* w *is a fair shuffle of* v *and* u, *or if in case (4) subcase (c) holds.* □

For $u, v \in \Delta^\infty$ the language consisting of all *(fair) shuffles* of u and v is denoted by $u \mid\mid v$ ($u \mid\mid\mid v$) and is defined as $u \mid\mid v = \{w \in \Delta^\infty \mid w \text{ is a shuffle of } u \text{ and } v\}$ and $u \mid\mid\mid v = \{w \in \Delta^\infty \mid w \text{ is a fair shuffle of } u \text{ and } v\}$, respectively.

For $L_1, L_2 \subseteq \Delta^\infty$ the *(fair) shuffle* of L_1 and L_2 is denoted by $L_1 \mid\mid L_2$ ($L_1 \mid\mid\mid L_2$) and is defined as the language consisting of all words which are a (fair) shuffle of a word from L_1 and a word from L_2. Thus $L_1 \mid\mid L_2 = \{w \in u \mid\mid v \mid u \in L_1,\ v \in L_2\} = \bigcup_{u\in L_1,\, v\in L_2}(u \mid\mid v)$ and $L_1 \mid\mid\mid L_2 = \bigcup_{u\in L_1,\, v\in L_2}(u \mid\mid\mid v)$.

Example 6.3.2. Let $\Delta = \{a, b, c, d\}$. Let $u = abc \in \Delta^*$ and let $v = cd \in \Delta^*$. Then $u \mid\mid v = \{abcd, acbcd, cabcd, abcdc, acbdc, cabdc, acdbc, cadbc, cdabc\} = u \mid\mid\mid v$.

Consequently, let $w_1 = a^\omega \in \Delta^\infty$ and let $w_2 = b^\omega \in \Delta^\infty$. Then $(ab)^\omega$ is a fair shuffle of w_1 and w_2, whereas aba^ω is a shuffle of w_1 and w_2, but not a fair shuffle.

Moreover, note that $v \mid\mid\mid w_2 = \{b^mcb^ndb^\omega \mid m, n \geq 0\}$, whereas $v \mid\mid w_2 = \{b^\omega\} \cup \{b^ncb^\omega \mid n \geq 0\} \cup v \mid\mid\mid w_2$. □

6.3.2 Basic Observations

Definition 6.3.1 immediately implies that the fair shuffle of two languages is included in the shuffle of those two languages.

Lemma 6.3.3. *Let* $u, v \in \Delta^\infty$ *and let* $L_1, L_2 \subseteq \Delta^\infty$. *Then*

(1) $u \mid\mid\mid v \subseteq u \mid\mid v$ *and*

(2) $L_1 \,|||\, L_2 \subseteq L_1 \,||\, L_2$. □

From Example 6.3.2 we conclude that both of these inclusions may be proper. In fact, the following result follows immediately from Definition 6.3.1.

Lemma 6.3.4. *(1) If* $u, v \in \Delta^*$, *then* $u \,||\, v = u \,|||\, v$,

(2) if $u \in \Delta^*$ *and* $v \in \Delta^\omega$, *then* $u \,||\, v = \bigcup_{u' \in pref(u)} (u' \,|||\, v)$, *and*

(3) if $u, v \in \Delta^\omega$, *then* $u \,||\, v = \bigcup_{u' \in pref(u)} (u' \,|||\, v) \cup \bigcup_{v' \in pref(v)} (u \,|||\, v') \cup u \,|||\, v$. □

Example 6.3.5. (Example 6.3.2 continued) We thus have that $w_1 \,||\, w_2 = (a^* \,|||\, \{w_2\}) \cup (\{w_1\} \,|||\, b^*) \cup (w_1 \,|||\, w_2)$, with $w_1 \,|||\, w_2 = (a^+ \,||\, b^+)^\omega$. □

Note furthermore that two words always define at least one (fair) shuffle, i.e. given $u, v \in \Delta^\infty$, then $u \,|||\, v \neq \varnothing$ (and thus $u \,||\, v \neq \varnothing$). Whenever both u and v equal λ, however, then $u \,||\, v = u \,|||\, v = \{\lambda\}$. Also the case that only one of the words u and v is λ exhibits no surprises.

Lemma 6.3.6. *Let* $u \in \Delta^\infty$. *Then*

$$u \,||\, \lambda = u \,|||\, \lambda = \{u\} = \lambda \,|||\, u = \lambda \,||\, u.$$ □

In Definition 6.3.1 we have defined a (fair) shuffle of two words as an (infinite) alternation of (finite) nonempty subwords of the one word with (finite) nonempty subwords of the other word. We now show that dropping the requirement that the subwords be nonempty does not alter the definition.

Lemma 6.3.7. *Let* $u, v \in \Delta^\infty$. *Then*

(1) $w \in u \,|||\, v$ *if and only if* $w = u_1 v_1 u_2 v_2 \cdots$, *with* $u_i, v_i \in \Delta^*$ *for all* $i \geq 1$, $u = u_1 u_2 \cdots$, *and* $v = v_1 v_2 \cdots$, *and*

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

Proof. (1) (Only if) Immediate from Definition 6.3.1.

(If) Let $w = u_1 v_1 u_2 v_2 \cdots$, with $u_i, v_i \in \Delta^*$ for all $i \geq 1$, $u = u_1 u_2 \cdots$, and $v = v_1 v_2 \cdots$. The proof of the statement makes use of the following construction, which provides representations ρ_k, $k \geq 1$, of prefixes of w satisfying some particular properties. Formally, the representations ρ_k, for all $k \geq 1$,

are defined by $\rho_1 = (u_1, v_1)$ and if $\rho_k = (\alpha_1, \beta_1, \alpha_2, \beta_2, \cdots, \alpha_\ell, \beta_\ell)$ for some $l \geq 1$ and $\alpha_j, \beta_j \in \Delta^*$, for all $1 \leq j \leq \ell$, then

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

The representation ρ_{k+1} is thus obtained by adding the words u_{k+1} and v_{k+1} to ρ_k. They are added to ρ_k in such a way that only the first and the last element of ρ_{k+1} are allowed to equal λ. As a result in the representation ρ_{k+1} of the prefix $u_1 v_1 u_2 v_2 \cdots u_{k+1} v_{k+1}$ all intermediate λ's have been omitted. Formally, the representations thus constructed possess the following properties that we use in this proof. Let $\rho_k = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell)$ for some $l \geq 1$ and $\alpha_j, \beta_j \in \Delta^*$, for all $j \in [\ell]$. Then $\alpha_1, \beta_\ell \in \Delta^*$, $\alpha_j \in \Delta^+$, for all $1 < j \leq \ell$, and $\beta_j \in \Delta^+$, for all $1 \leq j < \ell$. Furthermore, $\alpha_1\beta_1\alpha_2\beta_2 \cdots \alpha_\ell\beta_\ell = u_1 v_1 u_2 v_2 \cdots u_k v_k$, $\alpha_1\alpha_2 \cdots \alpha_\ell = u_1 u_2 \cdots u_k$, and $\beta_1\beta_2 \cdots \beta_\ell = v_1 v_2 \cdots v_k$. We now turn to the actual proof.

First consider the case that $u, v \in \Delta^*$. Since w is an infinite alternation of $u_i, v_i \in \Delta^*$, 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 $\alpha_1\beta_1\alpha_2\beta_2 \cdots \alpha_\ell\beta_\ell = w$, $\alpha_1, \beta_\ell \in \Delta^*$, and $\beta_1, \alpha_2, \beta_2, \alpha_3, \ldots, \beta_{\ell-1}, \alpha_\ell \in \Delta^+$. Hence $w \in u \mid\mid\mid v$.

Next consider the case that $u \in \Delta^*$ and $v \in \Delta^\omega$. Hence there must exist an $m \geq 1$ such that for all $n > m$, $u_n = \lambda$. Then with $\rho_m = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell)$ we obtain that for all $k \geq 1$, $\rho_{m+k} = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell v_{m+1} v_{m+2} \cdots v_{m+k})$, where $\alpha_1, \beta_\ell v_{m+1} v_{m+2} \cdots v_{m+k} \in \Delta^*$, $\alpha_j \in \Delta^+$, for all $1 < j \leq \ell$, $\beta_j \in \Delta^+$, for all $1 \leq j < \ell$, and $w = \alpha_1\beta_1\alpha_2\beta_2 \cdots \alpha_\ell\beta_\ell v_{m+1} v_{m+2} \cdots$. Hence $w \in u \mid\mid\mid v$.

Now consider the case that $u \in \Delta^\omega$ and $v \in \Delta^*$. Let $m \geq 1$ be the smallest index such that $u_m \neq \lambda$ and for all $n \geq m$, $v_n = \lambda$. Then with $\rho_m = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell)$, where $\beta_\ell = \lambda$ we obtain that for all $k \geq 1$, $\rho_{m+k} = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell u_{m+1} u_{m+2} \cdots u_{m+k}, \lambda)$, where $\alpha_1 \in \Delta^*$, $\alpha_j \in \Delta^+$, for all $1 < j < \ell$, $\alpha_\ell u_{m+1} u_{m+2} \cdots u_{m+k} \in \Delta^+$, $\beta_j \in \Delta^+$, for all $1 \leq j < \ell$, and $w = \alpha_1\beta_1\alpha_2\beta_2 \cdots \beta_{\ell-1}\alpha_\ell u_{m+1} u_{m+2} \cdots$. Hence $w \in u \mid\mid\mid v$.

Finally, consider the case that $u, v \in \Delta^\omega$. For every finite prefix $w' = u_1 v_1 u_2 v_2 \cdots u_m v_m \in \operatorname{pref}(w)$, for some $m \geq 1$, we know that $\rho_m = (\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots, \alpha_\ell, \beta_\ell)$ is such that $\alpha_1, \beta_\ell \in \Delta^*$, $\alpha_j \in \Delta^+$, for all $1 < j \leq \ell$, and $\beta_j \in \Delta^+$, for all $1 \leq j < \ell$. We obtain that $\lim_{\ell \to \infty} \alpha_1\beta_1\alpha_2\beta_2 \cdots \alpha_\ell\beta_\ell = \lim_{m \to \infty} u_1 v_1 u_2 v_2 \cdots u_m v_m = w$. Hence $w \in u \mid\mid\mid v$.

(2) By using Lemma 6.3.4(3) this follows from (1). □

Lemma 6.3.7 thus serves as an alternative definition of a shuffle of two (possibly infinite) words. With this alternative definition, commutativity of (fairly) shuffling two (possibly infinite) words follows immediately.

Theorem 6.3.8. *Let $u, v \in \Delta^\infty$. Then*

(1) $u \mid\mid\mid v = v \mid\mid\mid u$ *and*

(2) $u \mid\mid v = v \mid\mid u$.

Proof. (1) By symmetry it suffices to prove that $u \mid\mid\mid v \subseteq v \mid\mid\mid u$. Let $w \in u \mid\mid\mid v$. By Lemma 6.3.7(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 $v = v_1 v_2 \cdots$. Clearly we can also write w as $v_0 u_1 v_1 u_2 v_2 \cdots$, with $v_0 = \lambda$. Lemma 6.3.7(1) then implies that $w \in v \mid\mid\mid u$.

(2) Analogous. □

This theorem implies that also the (fair) shuffle of two (infinitary) languages is commutative.

Theorem 6.3.9. *Let $L_1, L_2 \subseteq \Delta^\infty$. Then*

(1) $L_1 \mid\mid\mid L_2 = L_2 \mid\mid\mid L_1$ *and*

(2) $L_1 \mid\mid L_2 = L_2 \mid\mid L_1$.

Proof. (1) By symmetry it suffices to prove that $L_1 \mid\mid\mid L_2 \subseteq L_2 \mid\mid\mid L_1$. Let $w \in L_1 \mid\mid\mid L_2$. Then there exist a $u \in L_1$ and a $v \in L_2$ such that $w \in u \mid\mid\mid v$. By Theorem 6.3.8(1) it now follows that $w \in v \mid\mid\mid u$ and hence $w \in L_2 \mid\mid\mid L_1$.

(2) Analogous. □

Recall from Lemma 6.3.4(1) that in case of finite words there is no need to distinguish shuffles from fair shuffles. The following results also follow immediately from Definition 6.3.1.

Lemma 6.3.10. *Let $u, v \in \Delta^*$ and let $w \in u \mid\mid v$. Then*

(1) $alph(w) = alph(u) \cup alph(v)$ *and*

(2) $|w| = |u| + |v|$. □

Note that in case u or v (or both) are infinite words, then a word w from the shuffle $u \mid\mid v$ does not necessarily contain all letters that are contained in u and v, unless the shuffle is fair.

Lemma 6.3.10 immediately implies that the language formed by the shuffles of two finite words is finite.

Corollary 6.3.11. *Let $u, v \in \Delta^*$. Then*

$\#(u \mid\mid v) \leq (\#(alph(u) \cup alph(v)))^{|u|+|v|}$ *and hence $u \mid\mid v$ is finite.* □

Next we wonder whether the language formed by the (fair) shuffles of two possibly infinite words can be finite. It turns out that this is the case. In fact, the series of results below leads to an exact formulation (cf. Theorem 6.3.26) of the conditions that guarantee this.

Lemma 6.3.12. *Let $u, v \in \Delta^\infty$ and let $z \in \Delta^*$. Then*

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

(2) $\{z\}(u \mid\mid v) \subseteq zu \mid\mid v$.

Proof. (1) Let $w \in \{z\}(u \mid\mid\mid v)$. Then $w = zw'$ for some $w' \in u \mid\mid\mid v$. By Lemma 6.3.7(1), $w' = u_1v_1u_2v_2\cdots$ for some $u_i, v_i \in \Delta^*$ 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) Analogous. □

Lemma 6.3.13. *Let $u, v \in \Delta^\infty$ and let $a, b \in \Delta$. Then*

(1) $au \mid\mid\mid bv = \{a\}(u \mid\mid\mid bv) \cup \{b\}(au \mid\mid\mid v)$ *and*

(2) $au \mid\mid bv = \{a\}(u \mid\mid bv) \cup \{b\}(au \mid\mid v)$.

Proof. (1) From Lemma 6.3.12(1) it follows that $\{a\}(u \mid\mid\mid bv) \subseteq au \mid\mid\mid bv$ and by Theorem 6.3.8(1) also $\{b\}(au \mid\mid\mid v) = \{b\}(v \mid\mid\mid au) \subseteq bv \mid\mid\mid au = au \mid\mid\mid bv$. Thus we are left with proving the inclusions in the statement from left to right. Let $w \in au \mid\mid\mid bv$.

By Lemma 6.3.7(1), $w = u_1v_1u_2v_2\cdots$ for some $u_i, v_i \in \Delta^*$ for all $i \geq 1$, $u_1u_2\cdots = au$, and $v_1v_2\cdots = bv$. We can distinguish the following two cases.

First let $k \geq 1$ be such that $u_k = au'_k$ and for all $1 \leq j < k$, $u_j = v_j = \lambda$. In this case $w \in \{a\}(u \mid\mid\mid bv)$.

Secondly, let $k \geq 1$ be such that $u_k = \lambda$, $v_k = bv'_k$, and for all $1 \leq j < k$, $u_j = v_j = \lambda$. In this case $w \in \{b\}(au \mid\mid\mid v)$.
Hence we conclude that $w \in \{a\}(u \mid\mid\mid bv) \cup \{b\}(au \mid\mid\mid v)$.

(2) Analogous. □

Lemma 6.3.14. *Let $u_1, v_1 \in \Delta^*$ and let $u_2, v_2 \in \Delta^\infty$. Then*

(1) $(u_1 \mid\mid v_1)(u_2 \mid\mid\mid v_2) \subseteq u_1u_2 \mid\mid\mid v_1v_2$ *and*

(2) $(u_1 \mid\mid v_1)(u_2 \mid\mid v_2) \subseteq u_1u_2 \mid\mid v_1v_2$.

Proof. (1) First assume that $u_1 = \lambda$. Then $u_1 \mid\mid v_1 = \{v_1\}$ by Lemma 6.3.6. Moreover, by the commutativity of $\mid\mid\mid$ and Lemma 6.3.12(1), we have that $\{v_1\}(u_2 \mid\mid\mid v_2) \subseteq u_2 \mid\mid\mid v_1v_2$. The case that $v_1 = \lambda$ is symmetric.

Next we proceed by induction on $|u_1| + |v_1|$. The cases $|u_1| + |v_1| = 0$ and $|u_1| + |v_1| = 1$ have already been dealt with. Thus assume that $|u_1| + |v_1| \geq 2$ with $u_1 = au_1'$ and $v_1 = bv_1'$ for some $a, b \in \Delta$ and some $u_1', v_1' \in \Delta^*$. Then by Lemma 6.3.13(2), $u_1 \mid\mid v_1 = au_1' \mid\mid bv_1' = \{a\}(u_1' \mid\mid bv_1') \cup \{b\}(au_1' \mid\mid v_1')$. This yields $(u_1 \mid\mid v_1)(u_2 \mid\mid\mid v_2) = \{a\}(u_1' \mid\mid bv_1')(u_2 \mid\mid\mid v_2) \cup \{b\}\ (au_1' \mid\mid v_1')\ (u_2 \mid\mid\mid v_2) \subseteq \{a\}\ (u_1'u_2 \mid\mid\mid bv_1'v_2) \cup \{b\}\ (au_1'u_2 \mid\mid\mid v_1'v_2) \subseteq (au_1'u_2 \mid\mid\mid bv_1'v_2) \cup (au_1'u_2 \mid\mid\mid bv_1'v_2) = (u_1u_2 \mid\mid\mid v_1v_2)$ by applying the induction hypothesis and Lemma 6.3.13 twice.

(2) Analogous. □

In the following example we show that the inclusions of this lemma can be proper.

Example 6.3.15. Let $\Delta = \{a, b\}$. Let $u = v = ab \in \Delta^*$. Then $u \mid\mid v \supseteq (a \mid\mid a)(b \mid\mid b)$ by Lemma 6.3.14(2). Since $abab \in u \mid\mid v$ and $(a \mid\mid a)(b \mid\mid b) = a^2b^2$, this inclusion is proper. □

Lemma 6.3.14 has the following direct consequences.

Corollary 6.3.16. *Let $u = u_1u_2 \cdots u_n$ and $v = v_1v_2 \cdots v_n$ be such that $u_i, v_i \in \Delta^*$, with $1 \leq i < n$, and $u_n, v_n \in \Delta^\infty$. Then*

(1) $(u_1 \mid\mid v_1)(u_2 \mid\mid v_2) \cdots (u_{n-1} \mid\mid v_{n-1})(u_n \mid\mid\mid v_n) \subseteq u \mid\mid\mid v$ *and*

(2) $(u_1 \mid\mid v_1)(u_2 \mid\mid v_2) \cdots (u_n \mid\mid v_n) \subseteq u \mid\mid v$. □

Corollary 6.3.17. *Let $u, v \in \Delta^\infty$. Then*

$$pref(u) \mid\mid pref(v) \subseteq pref(u \mid\mid\mid v).$$ □

The statement of this corollary holds also the other way around. This will be stated below as part of a more general equality. First we lift the statement of this corollary to languages.

Corollary 6.3.18. *Let $K, L \subseteq \Delta^\infty$. Then*

$$pref(K) \mid\mid pref(L) \subseteq pref(K \mid\mid\mid L).$$

Proof. Let $x \in \mathrm{pref}(K) \mid\mid \mathrm{pref}(L)$. Then by definition there exist a $u \in K$ and a $v \in L$ such that $x \in \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$, which according to Corollary 6.3.17 implies that $x \in \mathrm{pref}(u \mid\mid\mid v)$. Consequently, by definition $x \in \mathrm{pref}(K \mid\mid\mid L)$. □

Consequently, we obtain the following result and its extension to languages.

Lemma 6.3.19. *Let $u, v \in \Delta^\infty$. Then*

(1) $\mathit{pref}(u \mid\mid\mid v) \subseteq \mathit{pref}(u) \mid\mid\mid \mathit{pref}(v)$ *and*

(2) $\mathit{pref}(u \mid\mid v) \subseteq \mathit{pref}(u) \mid\mid \mathit{pref}(v)$.

Proof. (1) Let $z \in \mathrm{pref}(u \mid\mid\mid v)$. Then there exist $u_1, u_2, \ldots, u_n, v_1, v_2, \ldots, v_n$, and x such that $z = u_1 v_1 u_2 v_2 \cdots u_{n-1} v_{n-1} x$, where $u_1 \in \Delta^*$, $u_2, u_3, \ldots, u_{n-1}$, $v_1, v_2, \ldots, v_{n-1} \in \Delta^+$, and $x \in \Delta^*$ are such that $x \in \mathrm{pref}(u_n v_n)$, with $u_n, v_n \in \Delta^*$, $u_1 u_2 \cdots u_n \in \mathrm{pref}(u)$, and $v_1 v_2 \cdots v_n \in \mathrm{pref}(v)$. Hence $z \in \mathrm{pref}(u) \mid\mid\mid \mathrm{pref}(v)$.

(2) Analogous. □

Lemma 6.3.20. *Let $K, L \subseteq \Delta^\infty$. Then*

(1) $\mathit{pref}(K \mid\mid\mid L) \subseteq \mathit{pref}(K) \mid\mid\mid \mathit{pref}(L)$ *and*

(2) $\mathit{pref}(K \mid\mid L) \subseteq \mathit{pref}(K) \mid\mid \mathit{pref}(L)$.

Proof. (1) Let $x \in \mathrm{pref}(K \mid\mid\mid L)$. Then by definition there exist a $u \in K$ and a $v \in L$ such that $x \in \mathrm{pref}(u \mid\mid\mid v)$. Consequently, Lemma 6.3.19(1) implies that $x \in \mathrm{pref}(u) \mid\mid\mid \mathrm{pref}(v)$. Hence, by definition, $x \in \mathrm{pref}(K) \mid\mid\mid \mathrm{pref}(L)$.

(2) Analogous. □

Now we are ready to present the aforementioned equality and its extension to languages, including the converses of the statements of Corollaries 6.3.17 and 6.3.18.

Theorem 6.3.21. *Let $u, v \in \Delta^\infty$ and let $K, L \subseteq \Delta^\infty$. Then*

(1) $\mathit{pref}(u \mid\mid\mid v) = \mathit{pref}(u) \mid\mid\mid \mathit{pref}(v) = \mathit{pref}(u) \mid\mid \mathit{pref}(v) = \mathit{pref}(u \mid\mid v)$ *and*

(2) $\mathit{pref}(K \mid\mid\mid L) = \mathit{pref}(K) \mid\mid\mid \mathit{pref}(L) = \mathit{pref}(K) \mid\mid \mathit{pref}(L) = \mathit{pref}(K \mid\mid L)$.

Proof. (1) By Lemmata 6.3.19(1) and 6.3.3(2), Corollary 6.3.17, and Lemmata 6.3.3(2) and 6.3.19(2) we obtain $\mathrm{pref}(u \mid\mid\mid v) \subseteq \mathrm{pref}(u) \mid\mid\mid \mathrm{pref}(v) \subseteq \mathrm{pref}(u) \mid\mid \mathrm{pref}(v) \subseteq \mathrm{pref}(u \mid\mid\mid v) \subseteq \mathrm{pref}(u \mid\mid v) \subseteq \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$, which proves the statement.

(2) Analogous by Lemmata 6.3.20(1) and 6.3.3(2), Corollary 6.3.18, and Lemmata 6.3.3(2) and 6.3.20(2). □

We now continue our quest for a precise formulation of the conditions under which the language formed by the (fair) shuffles of two possibly infinite words can be finite.

We begin by isolating the case that u and v are words over the unary alphabet $\{a\}$. Recall from Lemma 6.3.10 that whenever $u = a^k$ and $v = a^\ell$, for some $k, \ell \in \mathbb{N}$, then $u \mid\mid v = \{a^{k+\ell}\}$. However, if $u = a^\omega$, then $u \mid\mid v = u \mid\mid\mid v = \{u\}$.

Lemma 6.3.22. *Let $w \in \Delta^*$, let $a \in \Delta$, and let $k \geq 0$. Then*

(1) $wa^\omega \mid\mid\mid a^k = (w \mid\mid\mid a^k)\{a^\omega\}$ and

(2) $wa^\omega \mid\mid a^k = (w \mid\mid a^k)\{a^\omega\}$.

Proof. First observe that $\{a^\omega\} = a^\omega \mid\mid\mid \lambda = a^\omega \mid\mid \lambda$. Then by Lemma 6.3.14 we have $(w \mid\mid\mid a^k)\{a^\omega\} = (w \mid\mid a^k)(a^\omega \mid\mid\mid \lambda) \subseteq wa^\omega \mid\mid\mid a^k$ and $(w \mid\mid a^k)\{a^\omega\} = (w \mid\mid a^k)(a^\omega \mid\mid \lambda) \subseteq wa^\omega \mid\mid a^k$. Hence we are done once we have proven that $wa^\omega \mid\mid a^k \subseteq (w \mid\mid\mid a^k)\{a^\omega\}$.

Let $x \in wa^\omega \mid\mid a^k$. This means that there exist $n \geq 1$, $v_1 \in \Delta^*$, $v_2, v_3, \ldots, v_n, u_1, u_2, \ldots, u_{n-1} \in \Delta^+$, and $u_n \in \Delta^\omega$ such that $v_1 v_2 \cdots v_n = a^\ell$ for some $\ell \leq k$, $u_1 u_2 \cdots u_n = wa^\omega$, and $x = v_1 u_1 v_2 u_2 \cdots v_n u_n$. Without loss of generality we may assume that $v_1 v_2 \cdots v_n = a^k$. This can be seen as follows. If $v_1 v_2 \cdots v_n = a^\ell$ and $\ell < k$, then since $u_n = w_2 a^\omega$ for some suffix w_2 of w we have $x = v_1 u_1 v_2 u_2 \cdots v_n w_2 a^{k-\ell} a^\omega$.
In case $w_2 \neq \lambda$ we have $x = v_1 u_1 v_2 u_2 \cdots v_n u'_n v_{n+1} u_{n+1}$ with $u'_n = w_2$, $v_{n+1} = a^{k-\ell}$, and $u_{n+1} = a^\omega$.
In case $w_2 = \lambda$ we have $x = v_1 u_1 v_2 u_2 \cdots u_{n-1} v'_n u_n$ with $v'_n = v_n a^{k-\ell}$.
Hence from here we assume that $x = v_1 u_1 v_2 u_2 \cdots v_n u_n$ with $u_1 u_2 \cdots u_n = wa^\omega$ and $v_1 v_2 \cdots v_n = a^k$.

Suppose that $u_1 u_2 \cdots u_{n-1} \in \mathrm{pref}(w)$. Then for some suffix w_2 of w we have $u_1 u_2 \cdots u_{n-1} w_2 = w$ and $u_n = w_2 a^\omega$. Consequently, we thus have $v_1 u_1 v_2 u_2 \cdots v_{n-1} u_{n-1} v_n w_2 \in a^k \mid\mid\mid w = w \mid\mid\mid a^k$ and thus $x \in (w \mid\mid\mid a^k)\{a^\omega\}$.

In the case that $u_1 u_2 \cdots u_{n-1} \notin \mathrm{pref}(w)$ we have $u_1 u_2 \cdots u_{n-1} \in w\{a\}^*$. Let $m = \min\{1 \leq j \leq n-1 \mid u_1 u_2 \cdots u_j \in w\{a\}^*\}$, where min applied to a set of positive integers selects the smallest number among this set of integers. Thus $u_m = u_{m,1} u_{m,2}$ with $u_1 u_2 \cdots u_{m-1} u_{m,1} = w$ and $u_{m,2} = \{a\}^*$. Hence with $v_1 v_2 \cdots v_m = a^\ell$ for some $\ell \leq k$ we have $u_{m,2} v_{m+1} u_{m+1} v_{m+2} \cdots u_{n-1} v_n = a^p$ for some $p \geq k - l$.

Now we have $x = v_1 u_1 v_2 u_2 \cdots v_m u_{m,1} a^p a^\omega = v_1 u_1 v_2 u_2 \cdots v_m u_{m,1} a^{k-\ell} a^\omega$ and thus $x \in (a^k \mid\mid\mid w)\{a^\omega\} = (w \mid\mid\mid a^k)\{a^\omega\}$. □

Whenever two nonempty words yield only one word as their shuffle, then it must be the case that those words are words over the same unary alphabet.

Lemma 6.3.23. *Let $u, v \in \Delta^\infty$ be such that both $u \neq \lambda$ and $v \neq \lambda$. Then*

(1) if $u \mid\mid\mid v = \{w\}$, for some $w \in \Delta^\infty$, then $u, v \in \{a\}^\infty$, for some $a \in \Delta$, and

(2) if $u \mid\mid v = \{w\}$, for some $w \in \Delta^\infty$, then $u, v \in \{a\}^\infty$, for some $a \in \Delta$.

Proof. (1) We prove the statement by contradiction, i.e. we assume that $\mathrm{alph}(u) \cup \mathrm{alph}(v)$ contains at least two elements.

First consider the case that $\mathrm{alph}(u) \setminus \mathrm{alph}(v) \neq \varnothing$. Let $b \in \mathrm{alph}(u) \setminus \mathrm{alph}(v)$. Hence $u = u_1 b u_2$ where $u_1 \in (\Delta \setminus \{b\})^*$ and $u_2 \in \Delta^\infty$. Let $v = cz$ for some $c \in \Delta \setminus \{b\}$ and $z \in (\Delta \setminus \{b\})^\infty$. Consider $w_1 = u_1 bcy$ and $w_2 = u_1 cby$, where $y \in u_2 \mid\mid\mid z$. Since $u_1 bc \in u_1 b \mid\mid\mid c$, Lemma 6.3.14(1) implies that $w_1 \in u \mid\mid\mid v$. Similarly $w_2 \in u \mid\mid\mid v$ because $u_1 cb \in u_1 b \mid\mid\mid c$. However, $b \neq c$ and thus $w_1 \neq w_2$, a contradiction.

Next consider the case that $\mathrm{alph}(u) = \mathrm{alph}(v)$. Hence $u = u_1 abu_2$ for some $a, b \in \Delta$, $a \neq b$, $u_1 \in \{a\}^*$, and $u_2 \in \Delta^\infty$. Let $v = cz$ for some $c \in \Delta$ and $z \in \Delta^\infty$. Consider $w_1 = u_1 abcy$ and $w_2 = cu_1 aby$, where $y \in u_2 \mid\mid\mid z$. As above both $w_1, w_2 \in u \mid\mid\mid v$ but $w_1 \neq w_2$, a contradiction.

Both cases thus lead to a contradiction and hence $\#(\mathrm{alph}(u) \cup \mathrm{alph}(v)) = 1$, i.e. $u, v \in \{a\}^\infty$ for some $a \in \Delta$.

(2) This follows from (1) and Lemma 6.3.3(1) combined with the fact that $u \mid\mid\mid v \neq \varnothing$. □

In fact, by Lemmata 6.3.6 and 6.3.23 it now follows that the (fair) shuffles of two words form a singleton language if and only if either one of those original words is empty, or both are words over the same unary alphabet.

Corollary 6.3.24. *Let $u, v \in \Delta^\infty$. Then*

(1) $u \mid\mid\mid v = \{w\}$, for some $w \in \Delta^\infty$, if and only if either $u = \lambda$, or $v = \lambda$, or $u, v \in \{a\}^\infty$, for some $a \in \Delta$, and

(2) $u \mid\mid v = \{w\}$, for some $w \in \Delta^\infty$, if and only if either $u = \lambda$, or $v = \lambda$, or $u, v \in \{a\}^\infty$, for some $a \in \Delta$. □

Next we state the conditions under which the (fair) shuffles of an infinite and a second (possibly infinite) word form a finite language.

Lemma 6.3.25. *Let $u \in \Delta^\omega$ and let $v \in \Delta^\infty \setminus \{\lambda\}$. Then*

(1) $u \mid\mid\mid v$ is finite if and only if either $u = wa^\omega$ and $v \in \{a\}^*$, or $u = v = a^\omega$, for some $w \in \Delta^*$ and $a \in \Delta$, and

(2) $u \mid\mid v$ *is finite if and only if either* $u = wa^\omega$ *and* $v \in \{a\}^*$*, or* $u = v = a^\omega$*, for some* $w \in \Delta^*$ *and* $a \in \Delta$*.*

Proof. (1) (If) Follows directly from Lemma 6.3.22(1).

(Only if) Let $u \mid\mid\mid v$ be a finite set and let $u = b_1 b_2 \cdots$ with $b_i \in \Delta$ for all $i \geq 1$. Suppose first that $\mathrm{alph}(v) \setminus \mathrm{alph}(u) \neq \varnothing$. Then $v = v_1 c v_2$ for some $v_1 \in \Delta^*$, $c \in \Delta \setminus \mathrm{alph}(u)$, and $v_2 \in \Delta^\infty$. Now set, for all $i \geq 0$, $W_i = v_1 b_1 b_2 \cdots b_i c(b_{i+1} b_{i+2} \cdots \mid\mid\mid v_2)$. Since $v_1 b_1 b_2 \cdots b_i c \in b_1 b_2 \cdots b_i \mid\mid\mid v_1 c$, Lemma 6.3.14(1) implies that $W_i \subseteq u \mid\mid\mid v$ for all $i \geq 0$. For each $i \geq 0$, all words in W_i have a c at position $|v_1| + i + 1$ and for all $k > i$, all words in W_k have b_i at position $|v_1| + i + 1$. Since $c \neq b_i$ for all $i \geq 1$, this implies that the W_i are mutually disjoint. Since they are not empty this implies that $\bigcup_{i \geq 0} W_i$ is infinite and hence $u \mid\mid\mid v$ is infinite, a contradiction.

Hence it must be the case that $\mathrm{alph}(v) \subseteq \mathrm{alph}(u)$. Now suppose that there exist $x \in \Delta^*$ and $y \in \Delta^\omega$ such that $u = xy$ and $\mathrm{alph}(v) \setminus \mathrm{alph}(y) \neq \varnothing$. Then by the same reasoning as given above we know that $y \mid\mid\mid v$ is infinite and since by Lemma 6.3.12(1) $x(y \mid\mid\mid v) \subseteq xy \mid\mid\mid v = u \mid\mid\mid v$ it follows that $u \mid\mid\mid v$ is infinite, again a contradiction.

Hence every symbol in v occurs infinitely often in u. Suppose that there are (at least) two different symbols occurring infinitely often in u. Thus for all $N \in \mathbb{N}$ there exists a $k_N \geq N$ such that $b_{k_N} \neq c$, where c is the first symbol of v. Thus we have $v = cv'$ with $c \in \Delta$ and $v' \in \Delta^\infty$. Let $u_N \in \Delta^\omega$ be such that $u = b_1 b_2 \cdots b_{k_N} u_N$. Set for all $N \geq 0$, $W_N = b_1 b_2 \cdots b_{k_N - 1} c b_{k_N}(u_N \mid\mid\mid v')$. Since $b_1 b_2 \cdots b_{k_N - 1} c b_{k_N} \in b_1 b_2 \cdots b_{k_N - 1} b_{k_N} \mid\mid\mid c$, Lemma 6.3.14(1) implies that $W_N \subseteq u \mid\mid\mid v$ for all $N \geq 0$. For each $N \geq 0$, all words in W_N have c at position k_N and for all N' such that $k_{N'} > k_N$, all words in $W_{N'}$ have b_{k_N} at position k_N. Since $c \neq b_{k_N}$ this implies that $W_N \cap W_{N'} = \varnothing$ whenever $k_{N'} > k_N$. Since $(k_N, N \geq 0)$ contains an infinite strictly increasing subsequence $k_{N_1} > k_{N_2} > \cdots$ this implies that $\bigcup_{N \in \mathbb{N}} W_N$ is infinite and hence $u \mid\mid\mid v$ is infinite, a contradiction once again.

Thus it must be the case that at most one symbol occurs infinitely often in u. Combining this with the already established fact that every symbol in v occurs infinitely often in u, we obtain that $u = wa^\omega$ for some $w \in \Delta^*$, $a \in \Delta$ and $v \in \{a\}^\infty$.

Finally assume that $\mathrm{alph}(w) \setminus \{a\} \neq \varnothing$ and suppose that $v = a^\omega$. then $a^i w a^\omega \neq a^j w a^\omega$ if $i \neq j$, but $a^i w a^\omega \subseteq u \mid\mid\mid v$ for all $i \geq 0$. Thus also in this case $u \mid\mid\mid v$ is infinite, a contradiction. Hence if $v = a^\omega$, then $u = a^\omega$ and $u \mid\mid\mid v = \{a^\omega\}$. If $v \neq a^\omega$, then $v = a^k$ for some $k \geq 1$ and $u = wa^\omega$. In this case $u \mid\mid\mid v = (w \mid\mid\mid a^k)\{a^\omega\}$ by Lemma 6.3.22(1) and thus $u \mid\mid\mid v$ is finite.

(2) (If) Follows directly from Lemma 6.3.22(2).

(Only if) If $u \mid\mid v$ is a finite set, then by Lemma 6.3.3(1) also $u \mid\mid\mid v$ is a finite set and the statement follows from (1). □

As a summary of the results obtained in Corollaries 6.3.11 and 6.3.24 and Lemma 6.3.25 we can now formulate the conditions under which the (fair) shuffles of two words form a finite language.

Theorem 6.3.26. *Let $u, v \in \Delta^\infty$. Then*

(1) $u \mid\mid\mid v$ is finite if and only if either $u, v \in \Delta^*$, or $u = \lambda$, or $v = \lambda$, or there exists an $a \in \Delta$ such that $u, v \in \{a\}^\infty$, or there exists a $w \in \Delta^*$ such that either $u = wa^\omega$ and $v \in \{a\}^*$, or $v = wa^\omega$ and $u \in \{a\}^*$, and

(2) $u \mid\mid v$ is finite if and only if either $u, v \in \Delta^*$, or $u = \lambda$, or $v = \lambda$, or there exists an $a \in \Delta$ such that $u, v \in \{a\}^\infty$, or there exists a $w \in \Delta^*$ such that either $u = wa^\omega$ and $v \in \{a\}^*$, or $v = wa^\omega$ and $u \in \{a\}^*$. □

6.3.3 Commutativity and Associativity

For later use of shuffles in the context of team automata, it is important to know that shuffles are commutative and associative. In Subsection 6.3.2 we showed the commutativity of the (fair) shuffles in Theorems 6.3.8 and 6.3.9 via the alternative definition of (fair) shuffles presented in Lemma 6.3.7. Before we deal with associativity we first present two lemmata that together provide a result (cf. Theorem 6.3.29) that has Theorem 6.3.8(1) as a direct corollary. This result actually is yet another alternative definition for the fair shuffle of two (possibly infinite) words. It sheds light on the particular characteristics of fair shuffles and it plays an important role in the remainder of this section.

First we need some auxiliary definitions. Let Δ be an alphabet. For each $i \in \mathbb{N}$ and $a \in \Delta$ we let $[a, i]$ be a distinct symbol. Let $[\Delta, i] = \{[a, i] \mid a \in \Delta\}$. Thus for all $i, j \in \mathbb{N}$ such that $i \neq j$, $[\Delta, i]$ and $[\Delta, j]$ are disjoint. We moreover assume, for all $i \in \mathbb{N}$, that Δ and $[\Delta, i]$ are disjoint. Let $i \in \mathbb{N}$. We define the homomorphisms $\beta_i : \Delta^* \to [\Delta, i]^*$ and $\overline{\beta}_i : [\Delta, i]^* \to \Delta^*$ by $\beta_i(a) = [a, i]$ and $\overline{\beta}_i([a, i]) = a$, respectively. Note that β_i and $\overline{\beta}_i$ are bijections. Intuitively, β_i is used to uniquely label every symbol in a word before this word is used in a shuffle, after which $\overline{\beta}_i$ can be used to remove this label again.

In addition we define the following homomorphisms. Let $i \in \mathbb{N}$ and let $J \subseteq \mathbb{N}$ be such that $i \notin J$. The homomorphism $\varphi_{i,J} : (\bigcup\{[\Delta, j] \mid j \in \{i\} \cup J\})^* \to \Delta^*$ is defined by $\varphi_{i,J}([a, i]) = a$ and $\varphi_{i,J}([a, j]) = \lambda$, for all $j \in J$, whereas the homomorphism $\psi_J : (\bigcup\{[\Delta, j] \mid j \in J\})^* \to \Delta^*$ is 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 symbol in a word that is labeled by i and to erase every other symbol from that word, whereas ψ_J simply removes all labels in J from every symbol in a word that is labeled by such a label from J.

Lemma 6.3.27. *Let $u, v \in \Delta^\infty$. Then, for all $i, j \in \mathbb{N}$ such that $i \neq j$,*

$$u \mid\mid\mid v \subseteq \psi_{\{i,j\}}(\varphi^{-1}_{i,\{j\}}(u) \cap \varphi^{-1}_{j,\{i\}}(v)).$$

Proof. Without loss of generality we assume that $i = 1$ and $j = 2$. Moreover, we prove only the case that $u \in \Delta^*$ and $v \in \Delta^\infty$. The proofs of the other cases are analogous.

Let $w \in u \mid\mid\mid v$. Hence $w = u_1v_1u_2v_2 \cdots u_nv_n$ with $n \geq 1$, $u_1 \in \Delta^*$, $u_2, u_3, \ldots, u_n, v_1, v_2, \ldots, v_{n-1} \in \Delta^+$, $v_n \in \Delta^\omega$, $u = u_1u_2 \cdots u_n$, and $v = v_1v_2 \cdots v_n$. Now consider $\overline{w} = \beta_1(u_1)\beta_2(v_1)\beta_1(u_2)\beta_2(v_2) \cdots \beta_1(u_n)\beta_2(v_n)$. Recall from the definitions of β_1, β_2, $\varphi_{1,\{2\}}$, and $\varphi_{2,\{1\}}$ that for all $a \in \Delta$, $\varphi_{1,\{2\}}(\beta_1(a)) = a$ and $\varphi_{1,\{2\}}(\beta_2(a)) = \lambda$. Hence it follows immediately that $\varphi_{1,\{2\}}(\overline{w}) = u$. Likewise, $\varphi_{2,\{1\}}(\overline{w}) = v$. Hence $\overline{w} \in \varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v)$. From the definitions of β_1, β_2, and $\psi_{\{1,2\}}$ we recall that for all $a \in \Delta$, $\psi_{\{1,2\}}(\beta_1(a)) = a$ and $\psi_{\{1,2\}}(\beta_2(a)) = a$. This implies that $\psi_{\{1,2\}}(\overline{w}) = w$ and we are done. □

Lemma 6.3.28. *Let $u, v \in \Delta^\infty$. Then, for all $i, j \in \mathbb{N}$ such that $i \neq j$,*

$$\psi_{\{i,j\}}(\varphi^{-1}_{i,\{j\}}(u) \cap \varphi^{-1}_{j,\{i\}}(v)) \subseteq u \mid\mid\mid v.$$

Proof. Without loss of generality we again assume that $i = 1$ and $j = 2$. Furthermore we again proof only the case that $u \in \Delta^*$ and $v \in \Delta^\infty$. The proofs of the other cases are analogous.

Let $w \in \psi_{\{1,2\}}(\varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v))$ and let $\overline{w} \in \varphi^{-1}_{1,\{2\}}(u) \cap \varphi^{-1}_{2,\{1\}}(v)$ be such that $\psi_{\{1,2\}}(\overline{w}) = w$. Since $\varphi_{1,\{2\}}(\overline{w}) = u$ there exist $m \geq 0$, $x_1, x_2, \ldots, x_m \in \Delta^*$, $x_{m+1} \in \Delta^\infty$, and $u_1, u_2, \ldots, u_m \in \Delta^+$ such that $\overline{w} = \beta_2(x_1)\beta_1(u_1)\beta_2(x_2)\beta_1(u_2) \cdots \beta_2(x_m)\beta_1(u_m)\beta_2(x_{m+1})$ and $u = u_1u_2 \cdots u_m$. Observe that the situation that $m = 0$ corresponds to the case that $u = \lambda$. Similarly, $\varphi_{2,\{1\}}(\overline{w}) = v$ and the fact that $v \neq \lambda$ imply that there exist $n \geq 1$, $y_1, y_2, \ldots, y_n \in \Delta^*$, $v_1, v_2, \ldots, v_{n-1} \in \Delta^+$, and $v_n \in \Delta^\omega$ such that $\overline{w} = \beta_1(y_1)\beta_2(v_1)\beta_1(y_2)\beta_2(v_2) \cdots \beta_1(y_n)\beta_2(v_n)$ and $v = v_1v_2 \cdots v_n$. We thus have the situation that $\beta_2(x_1)\beta_1(u_1)\beta_2(x_2)\beta_1(u_2) \cdots \beta_2(x_m)\beta_1(u_m)\beta_2(x_{m+1}) = \beta_1(y_1)\beta_2(v_1)\beta_1(y_2)\beta_2(v_2) \cdots \beta_1(y_n)\beta_2(v_n)$. 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. $x_1 = \lambda$. Now $v \in \Delta^\omega$ implies that $m \neq 0$. Thus we have that $\beta_1(u_1)\beta_2(x_2)\beta_1(u_2)\beta_2(x_3) \cdots \beta_2(x_m)\beta_1(u_m)\beta_2(x_{m+1}) =$

$\beta_1(y_1)\beta_2(v_1)\beta_1(y_2)\beta_2(v_2)\cdots\beta_1(y_n)\beta_2(v_n)$. Again by $[\Delta,1]\cap[\Delta,2]=\varnothing$ and from the fact that $u_i\in\Delta^+$ for all $1\leq i\leq m$, $v_j\in\Delta^+$ for all $1\leq j\leq n-1$, and $v_n\in\Delta^\omega$, we know that $m=n$ and, for all $1\leq i\leq n$, $\beta_1(u_i)=\beta_2(y_i)$ and $\beta_2(v_i)=\beta_2(x_{i+1})$. Consequently $w=\psi_{\{1,2\}}(\overline{w})=u_1v_1u_2v_2\cdots u_nv_n\in u\mid\mid\mid v$.

Next assume that $\beta_1(y_1)=\lambda$, i.e. $y_1=\lambda$. In this case we thus have the situation that $\beta_2(x_1)\beta_1(u_1)\beta_2(x_2)\beta_1(u_2)\cdots\beta_2(x_m)\beta_1(u_m)\beta_2(x_{m+1})=\beta_2(v_1)\beta_1(y_2)\beta_2(v_2)\beta_1(y_3)\cdots\beta_1(y_n)\beta_2(v_n)$. Again by $[\Delta,1]\cap[\Delta,2]=\varnothing$ and from the fact that $u_i\in\Delta^+$ for all $1\leq i\leq m$, $v_j\in\Delta^+$ for all $1\leq j\leq n-1$, and $v_n\in\Delta^\omega$, we know that $n=m+1$, $\beta_1(u_i)=\beta_1(y_{i+1})$ and $\beta_2(v_i)=\beta_2(x_i)$, for all $1\leq i\leq m$, and $\beta_2(v_{m+1})=\beta_2(x_{m+1})$. Consequently $w=\psi_{\{1,2\}}(\overline{w})=v_1u_1v_2u_2\cdots v_mu_mv_{m+1}\in u\mid\mid\mid v$. □

We now combine the two directly preceding lemmata to indeed obtain yet another alternative definition of the fair shuffle of two (possibly infinite) words. Note that since these lemmata use inverse homomorphisms based on the complete two words being shuffled. It therefore serves only as an alternative definition of the *fair* shuffle of these two words.

Theorem 6.3.29. *Let $u,v\in\Delta^\infty$. Then, for all $i,j\in\mathbb{N}$ such that $i\neq j$,*

$$u\mid\mid\mid v=\psi_{\{i,j\}}(\varphi^{-1}_{i,\{j\}}(u)\cap\varphi^{-1}_{j,\{i\}}(v)).$$ □

This theorem now provides a different — rather elegant — proof of Theorem 6.3.8(1) since we know that intersection is commutative and thus $u\mid\mid\mid v=\psi_{\{i,j\}}(\varphi^{-1}_{i,\{j\}}(u)\cap\varphi^{-1}_{j,\{i\}}(v))=\psi_{\{i,j\}}(\varphi^{-1}_{j,\{i\}}(v)\cap\varphi^{-1}_{i,\{j\}}(u))=v\mid\mid\mid u$. The fair shuffle of two words can thus be obtained by applying a combination of (inverse) homomorphisms and intersection to those two words.

Example 6.3.30. (Example 6.3.2 continued) Note that we have $\varphi^{-1}_{1,\{2\}}(u)=\{\beta_2(x_1)\beta_1(a)\beta_2(x_2)\beta_1(b)\beta_2(x_3)\beta_1(c)\beta_2(x_4)\mid x_i\in\Delta^*,\ i\in[3],\ x_4\in\Delta^\infty\}=\{\beta_2(x_1)[a,1]\beta_2(x_2)[b,1]\beta_2(x_3)[c,1]\beta_2(x_4)\mid x_i\in\Delta^*,\ i\in[3],\ x_4\in\Delta^\infty\}$ and $\varphi^{-1}_{2,\{1\}}(v)=\{\beta_1(y_1)\beta_2(c)\beta_1(y_2)\beta_2(d)\beta_1(y_3)\mid y_i\in\Delta^*,\ i\in[2],\ y_3\in\Delta^\infty\}=\{\beta_1(y_1)[c,2]\beta_1(y_2)[d,2]\beta_1(y_3)\mid y_i\in\Delta^*,\ i\in[2],\ y_3\in\Delta^\infty\}$. Thus, e.g., $[a,1][c,2][b,1][d,2][c,1]\in\varphi^{-1}_{1,\{2\}}(u)\cap\varphi^{-1}_{2,\{1\}}(v)$ and hence we now obtain that $\psi_{\{1,2\}}([a,1][c,2][b,1][d,2][c,1])=acbdc\in\psi_{\{1,2\}}(\varphi^{-1}_{1,\{2\}}(u)\cap\varphi^{-1}_{2,\{1\}}(v))$. Finally, note that in Example 6.3.2 we have seen that indeed $acbdc\in u\mid\mid\mid v$. □

This example shows why the construction $\psi_{\{i,j\}}(\varphi^{-1}_{i,\{j\}}(u)\cap\varphi^{-1}_{j,\{i\}}(v))$, with $u,v\in\Delta^\infty$ and $i\neq j\in\mathbb{N}$, in general does not equal $u\mid\mid v$: the inverse homomorphisms are "fair" in the sense that they take only complete words as input.

It remains to prove that (fairly) shuffling is associative. The remainder of this subsection is devoted to this. The setup is as follows. We first use Theorem 6.3.29 to prove the associativity of fairly shuffling (cf. Theorem 6.3.32). Lemma 6.3.4(1) then implies that associativity remains to be proven only in case infinite words are involved. To this aim we subsequently relate the shuffles of possibly infinite words to the shuffles of the finite prefixes of those possibly infinite words (cf. Theorem 6.3.49). We then conclude by using this result to prove associativity (cf. Theorem 6.3.51).

The following lemma streamlines the proof of the result succeeding it, which states that fairly shuffling is associative.

Lemma 6.3.31. *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 different from 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 $i_1 = 1$, $i_2 = 2$, $i_3 = 3$, and $j \neq 1$.

($\subseteq$) Let $z \in \psi_{\{j,1\}}(\varphi^{-1}_{1,\{j\}}(u) \cap \varphi^{-1}_{j,\{1\}}(\psi_{\{2,3\}}(\varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w))))$ and let $\overline{z} \in \varphi^{-1}_{1,\{j\}}(u) \cap \varphi^{-1}_{j,\{1\}}(\psi_{\{2,3\}}(\varphi^{-1}_{2,\{3\}}(v) \cap \varphi^{-1}_{3,\{2\}}(w)))$ be such that $\psi_{\{j,1\}}(\overline{z}) = z$. 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 $\overline{\overline{z}} = a_1\beta_2(\overline{\beta}_j(\overline{b}_1))\beta_3(\overline{\beta}_j(\overline{c}_1))a_2\beta_2(\overline{\beta}_j(\overline{b}_2))\beta_3(\overline{\beta}_j(\overline{c}_2))\cdots$. 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 $\psi_{\{1,2,3\}}(\overline{\overline{z}}) = \psi_{\{j,1\}}(\overline{z}) = z$.

($\supseteq$) 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 let $\overline{z} \in \varphi^{-1}_{1,\{2,3\}}(u) \cap \varphi^{-1}_{2,\{1,3\}}(v) \cap \varphi^{-1}_{3,\{1,2\}}(w)$ 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 $i \geq 1$, $\overline{\beta}_j(\alpha_i) = \psi_{\{2,3\}}(b_i c_i)$ now implies that $\psi_{\{j,1\}}(\overline{u}) = \psi_{\{1,2,3\}}(\overline{z}) = z$. □

Theorem 6.3.32. *Let $u, v, w \in \Delta^\infty$ and let $L_1, L_2, L_3 \subseteq \Delta^\infty$. Then*

(1) $\{u\} \mid\mid\mid (v \mid\mid\mid w) = (u \mid\mid\mid v) \mid\mid\mid \{w\}$ *and*

(2) $L_1 \mid\mid\mid (L_2 \mid\mid\mid L_3) = (L_1 \mid\mid\mid L_2) \mid\mid\mid L_3$.

Proof. (1) From Definition 6.3.1, Theorem 6.3.29, and Lemma 6.3.31 we obtain that $\{u\} \mid\mid\mid (v \mid\mid\mid w) = \{x \mid \exists y \in v \mid\mid\mid w : \ x \in u \mid\mid\mid y\} = \{x \mid \exists y \in \psi_{\{k,\ell\}}(\varphi^{-1}_{k,\{\ell\}}(v) \cap \varphi^{-1}_{\ell,\{k\}}(w)) : \ x \in \psi_{\{i,j\}}(\varphi^{-1}_{i,\{j\}}(u) \cap \varphi^{-1}_{j,\{i\}}(y)),\ i,j,k,\ell \in \mathbb{N},\ i \neq j,\ k \neq \ell\} = \{x \mid x \in \psi_{\{i,j\}}(\varphi^{-1}_{i,\{j\}}(u) \cap \varphi^{-1}_{j,\{i\}}(\psi_{\{k,\ell\}}(\varphi^{-1}_{k,\{\ell\}}(u) \cap \varphi^{-1}_{\ell,\{k\}}(v)))),\ i,j,k,\ell \in \mathbb{N},\ i \neq j,\ k \neq \ell\} = \{x \mid x \in \psi_{\{i,k,\ell\}}(\varphi^{-1}_{i,\{k,\ell\}}(u) \cap \varphi^{-1}_{k,\{i,\ell\}}(v) \cap \varphi^{-1}_{\ell,\{i,k\}}(w)),\ i,k,\ell \in \mathbb{N},\ i \neq k,\ k \neq \ell,\ \ell \neq i\} = \{x \mid x \in \psi_{\{j,\ell\}}(\varphi^{-1}_{j,\{\ell\}}(\psi_{\{i,k\}}(\varphi^{-1}_{i,\{k\}}(u) \cap \varphi^{-1}_{k,\{i\}}(v))) \cap \varphi^{-1}_{\ell,\{j\}}(w),\ i,j,k,\ell \in \mathbb{N},\ i \neq k,\ j \neq \ell\} = \{x \mid \exists z \in \psi_{\{i,k\}}(\varphi^{-1}_{i,\{k\}}(u) \cap \varphi^{-1}_{k,\{i\}}(v)) : \ x \in \psi_{\{j,\ell\}}(\varphi^{-1}_{j,\{\ell\}}(z) \cap \varphi^{-1}_{\ell,\{j\}}(w),\ i,j,k,\ell \in \mathbb{N},\ i \neq k,\ j \neq \ell\} = \{x \mid \exists z \in u \mid\mid\mid v : \ z \in z \mid\mid\mid w\} = (u \mid\mid\mid v) \mid\mid\mid \{w\}$.

(2) By definition and (1) we obtain $L_1 \mid\mid\mid (L_2 \mid\mid\mid L_3) = \{x \in u \mid\mid\mid y \mid u \in L_1,\ y \in L_2 \mid\mid\mid L_3\} = \{x \in \{u\} \mid\mid\mid (v \mid\mid\mid w) \mid u \in L_1,\ v \in L_2,\ w \in L_3\} = \{x \in (u \mid\mid\mid v) \mid\mid\mid \{w\} \mid u \in L_1,\ v \in L_2,\ w \in L_3\} = \{x \in z \mid\mid\mid w \mid z \in L_1 \mid\mid\mid L_2,\ w \in L_3\} = (L_1 \mid\mid\mid L_2) \mid\mid\mid L_3$. □

Due to Lemma 6.3.4(1) this result implies that also in the special case that we deal with finite words (finitary languages) only, shuffling is associative.

Corollary 6.3.33. *Let $u, v, w \in \Delta^*$ and let $L_1, L_2, L_3 \subseteq \Delta^*$. Then*

(1) $\{u\} \mid\mid (v \mid\mid w) = (u \mid\mid v) \mid\mid \{w\}$ *and*

(2) $L_1 \mid\mid (L_2 \mid\mid L_3) = (L_1 \mid\mid L_2) \mid\mid L_3$. □

Hence what remains is the case that infinite words are involved. To this aim we now seek to express the shuffles of possibly infinite words in terms of shuffles of their finite prefixes, which obviously are fair shuffles.

We begin by defining (u, v)-*decompositions* as a way to interleave the finite words u and v by alternating sequences from u and v. The construction of these (u, v)-decompositions resembles a construction used in the proof of Lemma 6.3.7.

Definition 6.3.34. *Let $w \in \Delta^*$. Then*

a decomposition *of* w *is a sequence* $d = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$ *such that* $n \geq 1$, $u_1 \in \Delta^*$, $u_2, u_3, \ldots, u_n, v_1, v_2, \ldots, v_{n-1} \in \Delta^+$, $v_n \in \Delta^*$, *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 also called a* (u, v)-decomposition *of* w. □

Together with Definition 6.3.1(1) this leads to the following result.

Lemma 6.3.35. *Let* $u, v, w \in \Delta^*$. *Then*

there exists a (u, v)*-decomposition of* w *if and only if* $w \in u \mid\mid v$. □

Note that a shuffle $w \in u \mid\mid v$ may have several decompositions.

Example 6.3.36. Let $\Delta = \{a, b, c\}$. Let $u, v \in \Delta^*$ be such that $u = aba$ and $v = babc$. Clearly $w = abababc \in u \mid\mid v$. Note that both $d_1 = (a, ba, ba, bc)$ and $d_2 = (aba, babc)$ are (u, v)-decompositions of w. Hence w does not have a unique decomposition.

Note that also $w' = babcaba \in u \mid\mid v$. It is however easy to see that in this case $(\lambda, babc, aba, \lambda)$ is the unique (u, v)-decomposition of w'. □

If $d = (u_1, v_1, u_2, v_2, \ldots, u_n v_n)$ is a (u, v)-decomposition of a word z, then n intuitively is the number of alternations of sequences from u and v that form $z = u_1 v_1 u_2 v_2 \cdots u_n v_n$.

Definition 6.3.37. *Let* $d = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$, *for some* $n \in \mathbb{N}$, *be a* (u, v)*-decomposition. Then*

n *is the* norm *of* d, *denoted by* $|| d ||$. □

Definition 6.3.38. *Let* $d = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$, *for some* $k \in \mathbb{N}$, *and* $d' = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$, *for some* $n \in \mathbb{N}$, *be two decompositions of two words over an alphabet* Δ. *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, one of the following three cases holds. Either*

- *(a)* $k = n$, $x_k = u_k$, *and* $y_k a = v_k$, *for some* $a \in \Delta$, *or*
- *(b)* $k = n$, $y_k = v_k = \lambda$, *and* $x_k a = u_k$, *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 not difficult 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 6.3.39. *Let $d = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$, for some $k \in \mathbb{N}$, and $d' = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$, for some $n \in \mathbb{N}$, be two decompositions — of two words over an alphabet Δ — such that d precedes d'. Then*

$$x_1x_2\cdots x_k \in \mathit{pref}(u_1u_2\cdots u_n),\ y_1y_2\cdots y_k \in \mathit{pref}(v_1v_2\cdots v_n),\ \text{and}\ x_1y_1x_2y_2\cdots x_ky_k \in \mathit{pref}(u_1v_1u_2v_2\cdots u_nv_n).$$

Proof. If $d = d'$ there is nothing to prove, so let us assume that $d \neq d'$. From Definition 6.3.38 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 $0 \leq j < \ell$. Thus, for all $0 \leq j < \ell - 1$, $s_j \in \mathrm{pref}\,(s_{j+1})$, $t_j \in \mathrm{pref}\,(t_{j+1})$, and $w_j \in \mathrm{pref}\,(w_{j+1})$. Hence $s_0 = x_1x_2\cdots x_k \in \mathrm{pref}\,(s_\ell) = \mathrm{pref}\,(u_1u_2\cdots u_n)$, $t_0 = y_1y_2\cdots y_k \in \mathrm{pref}\,(t_\ell) = \mathrm{pref}\,(v_1v_2\cdots v_n)$, and $w_0 = x_1y_1x_2y_2\cdots x_ky_k \in \mathrm{pref}\,(w_\ell) = \mathrm{pref}\,(u_1v_1u_2v_2\cdots u_nv_n)$. □

A sequence of decompositions — of words w_i into words u_i and words v_i, with $i \geq 0$ — preceding each other, uniquely defines the limit of the words w_i as an element of the shuffle of the limits of the words u_i and the words v_i.

Lemma 6.3.40. *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*

$$u = \lim_{i\to\infty} u_i,\ v = \lim_{i\to\infty} v_i,\ \text{and}\ w = \lim_{i\to\infty} w_i\ \text{exist, and}\ w \in u \mid\mid v.$$

Proof. By Lemma 6.3.39 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 $z_i \in \Delta^*$ such that $u_{i+1} = u_iz_i$. Observe that $u = \lim_{i\to\infty} u_i = u_{N_0} \lim_{i\to\infty} z_1z_2\cdots z_{i-N_0}$. Thus

we 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 6.3.35, 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 6.3.14(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 $z_i \in \Delta^*$ such that $v_{i+1} = v_i z_i$. Observe that $v = \lim_{i\to\infty} v_i = v_{\ell_0} \lim_{i\to\infty} z_1 z_2 \cdots z_{i-v_0}$. 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 6.3.35, 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 6.3.14(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 6.3.38 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 $i \geq 1$. According to Lemma 6.3.39, $u_{j_i} = x_1 x_2 \cdots x_{|| d_{j_i}||-1} s_i \in \mathrm{pref}(u_{j_{i+1}}) = \mathrm{pref}(x_1 x_2 \cdots x_{|| d_{j_{i+1}}||-1} s_{i+1})$, for all $i \geq 1$, and thus $u = \lim_{n\to\infty} x_1 x_2 \cdots x_n$. Analogously we get $v = \lim_{n\to\infty} y_1 y_2 \cdots y_n$, and $w = \lim_{n\to\infty} x_1 y_1 x_2 y_2 \cdots x_n y_n$. 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 $w \in u \mid\mid v$. □

The preceding two lemmata allow us to conclude that whenever the prefixes of an infinite word w are included in the shuffle of the prefixes of two words u and v that do not share a single letter, then w is a shuffle of u and v.

Lemma 6.3.41. *Let $u, v \in \Delta^\infty$ be such that alph$(u) \cap$ alph$(v) = \varnothing$ and let $w \in \Delta^\omega$. Then*

if pref$(w) \subseteq$ pref$(u) \mid\mid$ pref(v), then $w \in u \mid\mid v$.

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 $w[n+1] = w[n]a$ such that $a \in \mathrm{alph}(u)$ 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 $w[n+1] \in u_{n+1} \mid\mid v_{n+1}$. Observe that $\#_a(w[n+1]) = \#_a(w[n]) + 1$. Moreover, for all $b \in \mathrm{alph}(u)$ and for all $c \in \mathrm{alph}(v)$ such that $b \neq a$ and $c \neq a$ we have $\#_b(w[n]) = \#_b(u_n) = \#_b(w[n+1]) = \#_b(u_{n+1})$ and

$\#_c(w[n]) = \#_c(v_n) = \#_c(w[n+1]) = \#_c(v_{n+1})$ because $w[n+1] = w[n]a$ and $\mathrm{alph}(u) \cap \mathrm{alph}(v) = \varnothing$.

Consequently, using the fact that u_{n+1} and u_n are both prefixes of u, and v_{n+1} and v_n are both prefixes of v we conclude that $u_{n+1} = u_n a$ 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 $w[n]$. Then we have $d_n = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k)$, with $k \geq 0$. We define a (u_{n+1}, v_{n+1})-decomposition of $w[n+1]$ as follows.

First let $a \in \mathrm{alph}(u)$. 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 we set $d_{n+1} = (x_1, y_1, x_2, y_2, \ldots, x_k, y_k, a, \lambda)$. 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}.

Next we 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 $w[i]$, $i \geq 0$, such that d_i precedes d_{i+1} for all $i \geq 0$. Hence from Lemmata 6.3.40 it follows that $w = \lim_{n\to\infty} w[n] \in \lim_{n\to\infty} u_n \mid\mid \lim_{n\to\infty} v_n = u \mid\mid v$. □

This result implies that in order to determine whether or not an infinite word is a shuffle of two (possibly infinite) words that do not share a single letter, it suffices to consider only the (finite!) prefixes of those words. Unfortunately, however, condition $\mathrm{alph}(u) \cap \mathrm{alph}(v) = \varnothing$ of Lemma 6.3.41 is necessary to prove that each prefix of w has a unique decomposition into prefixes of u and v. This is illustrated in the following example. We moreover show that there exist an infinite number of prefixes $w[n]$ with a decomposition that does not precede any decomposition of $w[n+1]$.

Example 6.3.42. Let $\Delta = \{a, b\}$. Let $u, v \in \Delta^\omega$ be such that $u = (a^3 b)^\omega$ and $v = b^\omega$. Clearly $\{a^3, a^3 b\} \subseteq \mathrm{pref}(u)$, $\{b^2, b^3\} \subseteq \mathrm{pref}(v)$, and $w = a^3 b^3 \in \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$. We thus note that $d_1 = (a^3, b^3)$ and $d_2 = (a^3 b, b^2)$ are decompositions of w.

Note that also $w' = wa = a^3 b^3 a \in \mathrm{pref}(u) \mid\mid \mathrm{pref}(v)$. The only decompositions of w' based on prefixes of u and v are $d' = (a^3 b, b^2, a, \lambda)$ and $d'' = (a^3, b^2, ba, \lambda)$. It is clear that d_1 does not precede d' nor does it precede d''. Hence w and $w' = wa$ are such that there exists a decomposition d_1 of w that does not precede any decomposition of w'. Note, however, that d' is preceded by d_2.

Let $j \geq 0$ and let $u_j = a^3(ba^3)^j \in \text{pref}(u)$ and $v_j = b^3(b^3)^j \in \text{pref}(v)$. Then clearly both $w_j = (a^3b^4)^j a^3b^3 \in \text{pref}(u) \mid\mid \text{pref}(v)$ and $w'_j = w_j a = (a^3b^4)^j a^3b^3a \in \text{pref}(u) \mid\mid \text{pref}(v)$. Now 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. By the same reasoning as for the case $j = 0$ above it is however easy to see that there does not exist a decomposition of w'_j based on prefixes of u and v that is preceded by d_j. □

In order to generalize Lemma 6.3.41 by dropping the condition $\text{alph}(u) \cap \text{alph}(v) \neq \varnothing$ we need to be able to guarantee the following: if $u, v \in \Delta^\infty$, $w \in \Delta^\omega$, and $\text{pref}(w) \subseteq \text{pref}(u) \mid\mid \text{pref}(v)$, then there exists 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 6.3.43. *(König's Lemma) If G is an infinite finitely-branching rooted tree, then there exists an infinite path through G, starting in the root.* □

The subsequent definition of *limit-closed* languages allows us to first generalize Lemma 6.3.41 to languages and then to infer that the condition $\text{alph}(u) \cap \text{alph}(v) \neq \varnothing$ can — after all — indeed be dropped from Lemma 6.3.41.

Definition 6.3.44. *Let $K \subseteq \Delta^\infty$. Then*

K is limit closed *if for all words $w_1 \leq w_2 \leq \cdots \in pref(K)$, $\lim_{n\to\infty} w_n \in K \cup pref(K)$.* □

Example 6.3.45. All singleton languages $\{u\}$ are limit closed. Also all finitary languages $L = \{\lambda, a, \ldots, a^n \mid n \geq 1\}$ over a unary alphabet are limit closed, whereas a^* is not limit closed due to the fact that $\lim_{n\to\infty} a^n = a^\omega \notin a^* \cup L$. However, $a^* \cup a^\omega$ and a^ω are limit closed. □

Lemma 6.3.46. *Let $K, L \subseteq \Delta^\infty$ be limit closed and let $w \in \Delta^\omega$. Then*

if $pref(w) \subseteq pref(K) \mid\mid pref(L)$, then $w \in K \mid\mid L$.

Proof. Let $\text{pref}(w) \subseteq \text{pref}(K) \mid\mid \text{pref}(L)$.

For $n \geq 0$, let $V_n = \{d \mid d \text{ is a } (u_n, v_n)\text{-decomposition of } w[n],\ u_n \in \text{pref}(K), \text{ and } v_n \in \text{pref}(L)\}$ be the set of all possible decompositions of the prefixes $w[n]$ of w. Note that $V_0 = \{(\lambda, \lambda)\}$ consists of the (λ, λ)-decomposition of $w[0] = \lambda$. 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 $E = \{(d, d') \mid d \text{ directly precedes } d'\}$. 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 6.7.

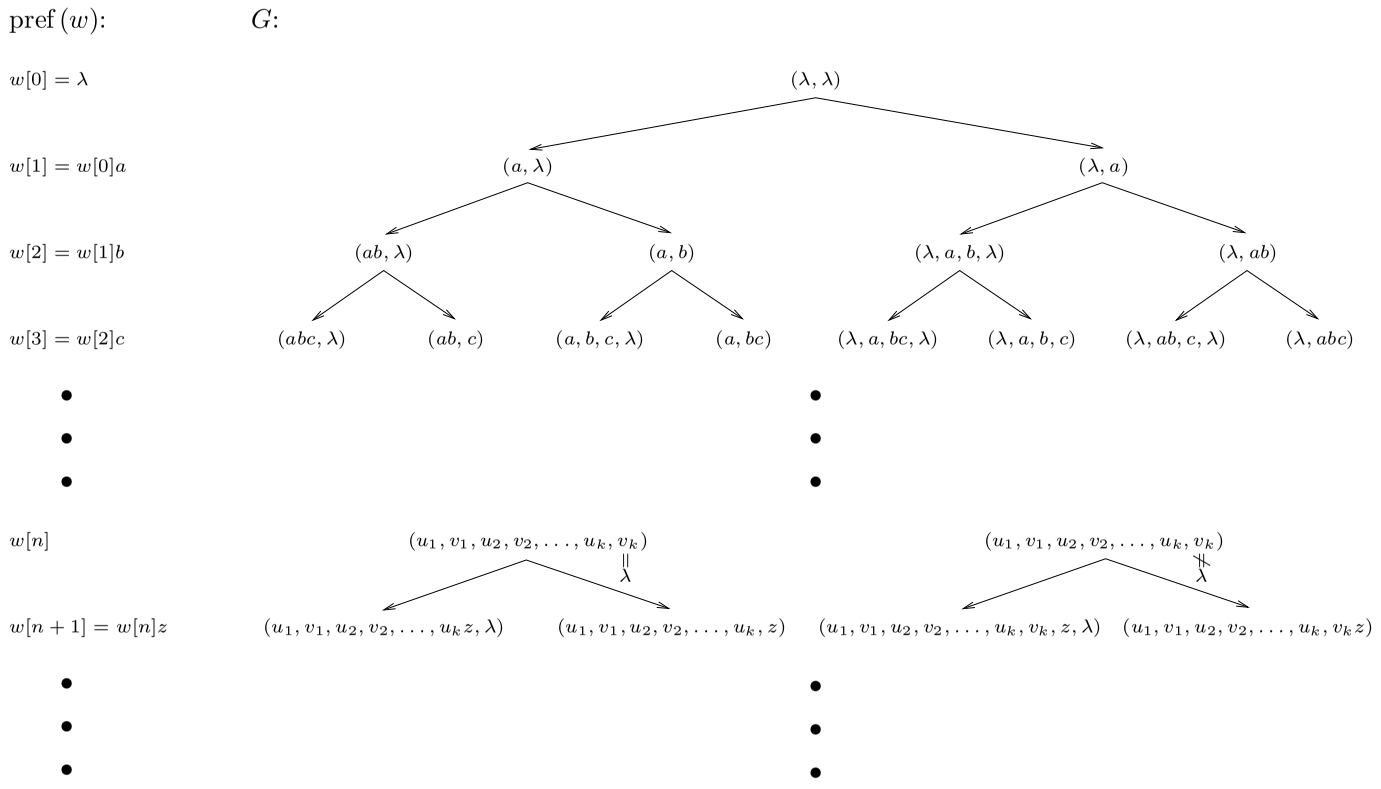

Fig. 6.7. Sketch of tree $G = (\bigcup_{n\geq 0} V_n, E)$.

Except for (λ, λ), every vertex of G has precisely one incoming edge. This can be seen as follows. The fact that $\text{pref}(w) \subseteq \text{pref}(K) \mid\mid \text{pref}(L)$ implies that every vertex has at least one incoming edge, whereas the fact that for every decomposition of a prefix $w[n]$, $n \geq 1$, we can immediately distinguish the unique last symbol of $w[n]$, implies that every vertex has at most one incoming edge. Furthermore, from Definition 6.3.38 it follows that every vertex has at most two outgoing edges, depending on whether the symbol added to $w[n]$, $n \geq 0$, 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 6.3.40 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 K or L are not limit closed, as is shown next.

Example 6.3.47. Let $\Delta = \{a\}$ and let $w = a^\omega \in \Delta^\omega$. Let $K = a^* \subseteq \Delta^\infty$ and let $L = \{\lambda\} \subseteq \Delta^\infty$. Then clearly $\text{pref}(w) = a^* = \text{pref}(K) \mid\mid \text{pref}(L)$, whereas $w = a^\omega \notin a^* = K \mid\mid L$. □

Since all singleton languages are limit closed, we immediately obtain the following result.

Corollary 6.3.48. *Let $u, v \in \Delta^\infty$ and let $w \in \Delta^\omega$. Then*

if $\text{pref}(w) \subseteq \text{pref}(u) \mid\mid \text{pref}(v)$, then $w \in u \mid\mid v$. □

Together with Theorem 6.3.21, this corollary and its preceding lemma imply the following result.

Theorem 6.3.49. *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 $\text{pref}(w) \subseteq \text{pref}(u) \mid\mid \text{pref}(v)$, and

(2) $w \in K \mid\mid L$ if and only if $\text{pref}(w) \subseteq \text{pref}(K) \mid\mid \text{pref}(L)$. □

We have thus been able to express the shuffles of possibly infinite words in terms of the shuffles of finite prefixes of those possibly infinite words. One more result now suffices to prove the associativity of shuffling.

Corollary 6.3.50. *Let $v, w \in \Delta^\infty$. Then*

$v \mid\mid w$ is limit closed.

Proof. Let $y_1 \leq y_2 \leq \cdots \in \operatorname{pref}(v \mid\mid w)$ and let $y = \lim_{n\to\infty} y_n$. Since for all $x \in \operatorname{pref}(y)$, there exists an $i \geq 0$ such that $x \in \operatorname{pref}(y_i) \in \operatorname{pref}(\operatorname{pref}(v \mid\mid w)) = \operatorname{pref}(v \mid\mid w)$, it follows that $\operatorname{pref}(y) \subseteq \operatorname{pref}(v \mid\mid w)$. Consequently, we distinguish two cases.

If $y \in \Delta^*$, then $y \in \operatorname{pref}(v \mid\mid w)$.

If $y \in \Delta^\omega$, then by Theorem 6.3.49(1), $y \in v \mid\mid w$.

Hence $y \in v \mid\mid w \cup \operatorname{pref}(v \mid\mid w)$ and $v \mid\mid w$ is thus limit closed. □

Theorem 6.3.51. *Let $u, v, w \in \Delta^\infty$ and let $L_1, L_2, L_3 \subseteq \Delta^\infty$. Then*

(1) $\{u\} \mid\mid (v \mid\mid w) = (u \mid\mid v) \mid\mid \{w\}$ *and*

(2) $L_1 \mid\mid (L_2 \mid\mid L_3) = (L_1 \mid\mid L_2) \mid\mid L_3$.

Proof. (1) Let $x \in \{u\} \mid\mid (v \mid\mid w)$.

If $x \in \Delta^*$, then Definition 6.3.1 implies that $u, v, w \in \Delta^*$. Consequently, by Corollary 6.3.33(1), $x \in (u \mid\mid v) \mid\mid \{w\}$.

If $x \in \Delta^\omega$, then since we know that $\{u\}$ and $v \mid\mid w$ are limit closed, Theorem 6.3.49(2) implies that $\operatorname{pref}(x) \subseteq \operatorname{pref}(\{u\}) \mid\mid \operatorname{pref}(v \mid\mid w)$. Hence, by Theorem 6.3.21(1), $\operatorname{pref}(x) \subseteq \operatorname{pref}(\{u\}) \mid\mid (\operatorname{pref}(v) \mid\mid \operatorname{pref}(w))$. Then Corollary 6.3.33(2) implies that $\operatorname{pref}(x) \subseteq (\operatorname{pref}(u) \mid\mid \operatorname{pref}(v)) \mid\mid \operatorname{pref}(\{w\})$ and from Theorem 6.3.21(1) we obtain $\operatorname{pref}(x) \subseteq \operatorname{pref}(u \mid\mid v) \mid\mid \operatorname{pref}(\{w\})$. Finally, using the fact that $u \mid\mid v$ and $\{w\}$ are limit closed, Theorem 6.3.49(2) implies that $x \in (u \mid\mid v) \mid\mid \{w\}$.

(2) Analogous to the proof of Theorem 6.3.32(2). □

6.3.4 Conclusion

The associativity of (fairly) shuffling (cf. Theorems 6.3.32 and 6.3.51) directly implies that the order in which we (fairly) shuffle a number of languages is irrelevant, i.e. $L_1 \mid\mid\mid L_2 \mid\mid\mid \cdots \mid\mid\mid L_n$ and $L_1 \mid\mid L_2 \mid\mid \cdots \mid\mid L_n$ unambiguously define the fair shuffle and shuffle, respectively, of the languages L_1, L_2, …, L_n, for an $n \geq 1$. It is thus not necessary to put any brackets in these expressions and we will henceforth refrain from doing so. Using also the commutativity of (fairly) shuffling, we may introduce the following shorthand notations for such *n-ary (fair) shuffles*.

Notation 12. *We denote the fair shuffle $L_1 \mid\mid\mid L_2 \mid\mid\mid \cdots \mid\mid\mid L_n$ and the shuffle $L_1 \mid\mid L_2 \mid\mid \cdots \mid\mid L_n$ of the languages L_1, L_2, …, L_n, for an $n \geq 1$, by $\mid\mid\mid_{i\in[n]} L_i$ and $\mid\mid_{i\in[n]} L_i$, respectively.* □

6.4 Synchronized Shuffles

In this section we generalize the basic shuffle by defining *synchronized shuffles*. Rather than freely interleaving the occurrences of the letters in the words being shuffled, some letters may now be subject to "synchronization". This means that occurrences of those letters in different words are now combined into one occurrence. The resulting word thus has a "backbone" consisting of occurrences of synchronized letters. As a preliminary example, consider the words *wev* and *ave*. If we assume that the letter v needs to be synchronized, then *weave* is a *synchronized shuffle on* v *of* *wev* and *ave*. Its backbone consists of only one element, viz. v. We see that those letters occurring on the left (right) side of v in the original words occur on the left (right) side of v in *weave* as well. Note that *weave* is not an ordinary shuffle of *wev* and *ave*.

As was the case for shuffles, also the idea underlying synchronized shuffles is not new. Instead, it appears in numerous disguises throughout the computer science literature. The oldest reference — once again to the best of our knowledge — to this idea is the *concurrent composition* $P \oplus Q$ *of synchronizing processes* P and Q defined in [Kim76]. Within formal language theory, a slightly adapted version of the idea was introduced in [DeS84] as the '*produit de mixage*' $K \sqcap\!\sqcap L$ of two languages K and L. This operation was renamed *synchronized shuffle* in [LR99]. In the context of process algebra, finally, two further slightly adapted versions of the idea were introduced in [vdS85] as the *weave* $T \underline{w} U$ of two words T and U, and in [Ros97] as the *alphabetized parallel composition* $P \,{}_X\underline{||}_Y\, Q$ of processes P and Q given alphabets X and Y. We will soon see, however, that the synchronized shuffles we define here are more general than any of these operations from the literature. In particular, we define two variants of synchronized shuffles: the *fully synchronized shuffle* and the *relaxed synchronized shuffle*, both obtained by varying the alphabet of letters to be synchronized.

Given two words over two given (possibly different) alphabets, a fully synchronized shuffle requires all letters in the intersection of these two alphabets to be synchronized, while a relaxed synchronized shuffle requires only a specified subset of the letters in this intersection to be synchronized. Both synchronized shuffles are thus defined with respect to two alphabets. We continue our example by again considering the words *wev* and *ave*. Assume that *wev* is a word over the alphabet $\{w, e, v\}$ and that *ave* is a word over the alphabet $\{a, v, e\}$. Then a *fully synchronized shuffle of* *wev* and *ave w.r.t.* $\{w, e, v\}$ and $\{a, v, e\}$ does not exist due to the fact that e and v cannot form one backbone respecting both the order ev from *wev* and the order ve from *ave*. However, a *relaxed synchronized shuffle on* $\{e\}$ *of* *wev* and *ave w.r.t.* $\{w, e, v\}$ and $\{a, v, e\}$ does exist and contains, e.g., *wavev*.

We begin by formally defining the most general synchronized shuffle, in terms of which we consequently define the two variants just discussed — complete with more elaborate examples. Along the way we will compare our synchronized shuffles to the ones from the literature. Subsequently we present a few of their basic properties. Since synchronized shuffles are defined on the basis of the ordinary shuffle, many observations from the previous section continue to hold (with trivial adaptions). We will not draw all such implications, but rather provide a series of connections between the various types of (synchronized) shuffles. Finally, we prove that all three types of synchronized shuffles satisfy notions of commutativity and associativity.

6.4.1 Definitions

We start by defining *synchronized shuffles* as a generalization of the shuffles of the previous section. Given an alphabet Γ and two words u and v, in a synchronized shuffle u and v synchronize on letters from Γ, while all occurrences of other letters are shuffled.

Definition 6.4.1. *Let $u, v \in \Delta^\infty$ and let Γ be an alphabet. Then*

a word $w \in \Delta^\infty$ is a synchronized shuffle *(*S-shuffle *for short) on Γ of u and v if one of the following two cases holds. Either*

(1) $w \in (u_1 \mid\mid v_1)x_1(u_2 \mid\mid v_2)x_2 \cdots x_{n-1}(u_n \mid\mid v_n)$, where for some $n \geq 1$, $u_1, u_2, \ldots, u_{n-1}, v_1, v_2, \ldots, v_{n-1} \in (\Delta \setminus \Gamma)^*$, $u_n, v_n \in (\Delta \setminus \Gamma)^\infty$, and $x_1, x_2, \ldots, x_{n-1} \in \Gamma$ are such that $u = u_1x_1u_2x_2 \cdots x_{n-1}u_n$ and $v = v_1x_1v_2x_2 \cdots x_{n-1}v_n$, or

(2) $w \in (u_1 \mid\mid v_1)x_1(u_2 \mid\mid v_2)x_2 \cdots$, where $u_1, u_2, \ldots, v_1, v_2, \cdots \in (\Delta \setminus \Gamma)^*$, and $x_1, x_2, \cdots \in \Gamma$ are such that $u = u_1x_1u_2x_2 \cdots$ and $v = v_1x_1v_2x_2 \cdots$.

This S-shuffle w on Γ is called fair *if in case (1) $(u_n \mid\mid v_n)$ is fair or if case (2) holds.* □

The sequence $\mathrm{pres}_\Gamma(w)$ is called the *backbone* of w. Note that in case (1) the S-shuffle w has a finite backbone $x_1x_2 \cdots x_{n-1}$, while in case (2) it has an infinite backbone $x_1x_2 \cdots$.

For $u, v \in \Delta^\infty$ the language consisting of all *(fair) S-shuffles* on Γ of u and v is denoted by $u \mid\mid^\Gamma v$ ($u \mid\mid\mid^\Gamma v$) and is defined as $u \mid\mid^\Gamma v = \{w \in \Delta^\infty \mid w \text{ is an S-shuffle on } \Gamma \text{ of } u \text{ and } v\}$ and $u \mid\mid\mid^\Gamma v = \{w \in \Delta^\infty \mid w \text{ is a fair S-shuffle on } \Gamma \text{ of } u \text{ and } v\}$, respectively.

For $L_1, L_2 \subseteq \Delta^\infty$ the *(fair) S-shuffle* on Γ of L_1 and L_2 is denoted by $L_1 \mid\mid^\Gamma L_2$ ($L_1 \mid\mid\mid^\Gamma L_2$) and is defined as the language consisting of all (fair)

S-shuffles on Γ of a word from L_1 and a word from L_2. Thus $L_1 \,||^{\Gamma}\, L_2 = \{w \in u \,||^{\Gamma}\, v \mid u \in L_1,\ v \in L_2\} = \bigcup_{u \in L_1,\, v \in L_2} (u \,||^{\Gamma}\, v)$ and $L_1 \,|||^{\Gamma}\, L_2 = \bigcup_{u \in L_1,\, v \in L_2} (u \,|||^{\Gamma}\, v)$, respectively.

Example 6.4.2. (Example 6.3.2 continued) Recall that $u = abc$ and $v = cd$. Now $u \,||^{\{c\}}\, v = u \,|||^{\{c\}}\, v = \{abcd\}$, whereas $u \,||^{\{b,c\}}\, v = u \,|||^{\{b,c\}}\, v = \varnothing$.

Recall that $w_1 = a^\omega$. Now $w_1 \,||^{\{a\}}\, a = w_1 \,|||^{\{a\}}\, a = \varnothing$ and $w_1 \,||^{\{a\}}\, w_1 = w_1 \,|||^{\{a\}}\, w_1 = \{a^\omega\}$.

Finally, recall that $\Delta = \{a, b, c, d\}$. Let $w_{12} = (ab)^\omega \in \Delta^\omega$ and let $w_{21} = (ba)^\omega \in \Delta^\omega$. Then we have $w_{12} \,||^{\{a\}}\, w_{21} = w_{12} \,|||^{\{a\}}\, w_{21} = \{(bab)^\omega\}$, whereas $w_{12} \,||^{\{a,b\}}\, w_{21} = w_{12} \,|||^{\{a,b\}}\, w_{21} = \varnothing$. □

From Definition 6.4.1 we furthermore obtain that the fair S-shuffle on an alphabet Γ of languages is included in the S-shuffle on Γ of these languages.

We now show that S-shuffles are indeed a generalization of both the concurrent composition as defined in [Kim76] and the 'produit de mixage' as defined in [DeS84] (and later renamed synchronized shuffle in [LR99]). If we syntactically restrict an S-shuffle on an alphabet Γ of languages $L_1, L_2 \subseteq \Delta^*$ to the case that $\Gamma \subseteq \Delta$, then we obtain exactly the concurrent composition operation defined in [Kim76]. If, on the other hand, we define the *alphabet* $\mathrm{alph}(L)$ of a language L as $\mathrm{alph}(L) = \bigcup_{w \in L} \mathrm{alph}(w)$ and allow infinite words in L_1 and L_2, then $L_1 \,|||^{\mathrm{alph}(L_1) \cap \mathrm{alph}(L_2)}\, L_2$ is exactly the 'produit de mixage' of L_1 and L_2 as defined in [DeS84] (which in [LR99] is restricted to finitary languages and renamed synchonized shuffle).

We proceed by defining the *fully synchronized shuffle* as a special case of the synchronized shuffle. Given a word u over Δ_1 and a word v over Δ_2, in a fully synchronized shuffle u and v synchronize on letters from $\Delta_1 \cap \Delta_2$, while all occurrences of other letters are again shuffled. Limited to finite words, the fully synchronized shuffle is exactly the weave operation defined in [vdS85] in the context of process algebra. By allowing infinite words, the fully synchronized shuffle is thus more general than the weave operation.

Definition 6.4.3. *Let $u \in \Delta_1^\infty$ and let $v \in \Delta_2^\infty$. Then*

> *a word $w \in (\Delta_1 \cup \Delta_2)^\infty$ is a* fully synchronized shuffle *(*fS-shuffle *for short) of u and v w.r.t. Δ_1 and Δ_2 if w is an S-shuffle on $\Delta_1 \cap \Delta_2$ of u and v.*

This fS-shuffle of u and v w.r.t. Δ_1 and Δ_2 is called fair *if w is a fair S-shuffle on $\Delta_1 \cap \Delta_2$ of u and v.* □

For $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$ the language consisting of all *(fair) fS-shuffles* of u and v w.r.t. Δ_1 and Δ_2 is denoted by $u \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; v$ ($u \;{}_{\Delta_1}\underline{|||}_{\Delta_2}\; v$) and is defined

as $u \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid w \text{ is an fS-shuffle of } u \text{ and } v \text{ w.r.t. } \Delta_1 \text{ and } \Delta_2\}$ and $u \;{}_{\Delta_1}\underline{|||}_{\Delta_2}\; v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid w \text{ is a fair fS-shuffle of } u \text{ and } v \text{ w.r.t. } \Delta_1 \text{ and } \Delta_2\}$, respectively.

For $L_1 \subseteq \Delta_1^\infty$ and $L_2 \subseteq \Delta^\infty$ the *(fair) fS-shuffle* of L_1 and L_2 w.r.t. Δ_1 and Δ_2 is denoted by $L_1 \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; L_2$ ($L_1 \;{}_{\Delta_1}\underline{|||}_{\Delta_2}\; L_2$) and is defined as the language consisting of all (fair) fS-shuffles of a word from L_1 and a word from L_2 w.r.t. Δ_1 and Δ_2. Thus $L_1 \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; L_2 = \{w \in u \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; v \mid u \in L_1,\, v \in L_2\} = \bigcup_{u \in L_1,\, v \in L_2}(u \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; v)$ and $L_1 \;{}_{\Delta_1}\underline{|||}_{\Delta_2}\; L_2 = \bigcup_{u \in L_1,\, v \in L_2}(u \;{}_{\Delta_1}\underline{|||}_{\Delta_2}\; v)$, respectively.

Example 6.4.4. (Example 6.4.2 continued) Now $u \;{}_{\Delta}\underline{||}_{\Delta}\; v = u \;{}_{\Delta}\underline{|||}_{\Delta}\; v = \varnothing$. Next let $\Delta_1 = \{a, b, c\}$ and let $\Delta_2 = \{c, d\}$. Consequently, let $u = abc \in \Delta_1^*$ and let $v = cd \in \Delta_2^*$. Then $u \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; v = u \;{}_{\Delta_1}\underline{|||}_{\Delta_2}\; v = \{abcd\} = u \;|||^{\{c\}}\; v = u \;||^{\{c\}}\; v$.

We moreover have $w_1 \;{}_{\Delta}\underline{||}_{\Delta}\; a = w_1 \;{}_{\Delta}\underline{|||}_{\Delta}\; a = \varnothing$, with $a \in \Delta^*$, and $w_1 \;{}_{\Delta}\underline{||}_{\Delta}\; w_1 = w_1 \;{}_{\Delta}\underline{|||}_{\Delta}\; w_1 = \{a^\omega\} = w_1 \;|||^{\{a\}}\; w_1 = w_1 \;||^{\{a\}}\; w_1$. Recall that $w_2 = b^\omega \in \Delta^\infty$ and hence $w_1 \;{}_{\Delta}\underline{||}_{\Delta}\; w_2 = w_1 \;{}_{\Delta}\underline{|||}_{\Delta}\; w_2 = \varnothing$. Next let $\Delta_a = \{a\}$ and let $\Delta_b = \{b\}$. Consequently, let $w_1 = a^\omega \in \Delta_a^\infty$ and let $w_2 = b^\omega \in \Delta_b^\infty$. Then $w_1 \;{}_{\Delta_a}\underline{||}_{\Delta_b}\; w_2 = w_1 \;||\; w_2$ and $w_1 \;{}_{\Delta_a}\underline{|||}_{\Delta_b}\; w_2 = w_1 \;|||\; w_2$.

Finally, $w_{12} \;{}_{\Delta}\underline{||}_{\Delta}\; w_{21} = w_{12} \;{}_{\Delta}\underline{|||}_{\Delta}\; w_{21} = \varnothing$. □

Finally we define also the *relaxed synchronized shuffle* as a special case of the synchronized shuffle. Given an alphabet Γ, a word u over Δ_1, and a word v over Δ_2, in a relaxed synchronized shuffle u and v synchronize on letters from $\Gamma \cap \Delta_1 \cap \Delta_2$, while all occurrences of other letters are once again shuffled.

Definition 6.4.5. *Let $u \in \Delta_1^\infty$, let $v \in \Delta_2^\infty$, and let Γ be an alphabet. Then*

> *a word $w \in (\Delta_1 \cup \Delta_2)^\infty$ is a* relaxed synchronized shuffle *(*rS-shuffle *for short) on Γ of u and v w.r.t. Δ_1 and Δ_2 if w is an S-shuffle on $\Gamma \cap \Delta_1 \cap \Delta_2$ of u and v.*

This rS-shuffle on Γ of u and v w.r.t. Δ_1 and Δ_2 is called fair *if w is a fair S-shuffle on $\Gamma \cap \Delta_1 \cap \Delta_2$ of u and v.* □

For $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$ the language consisting of all *(fair) rS-shuffles* on Γ of u and v w.r.t. Δ_1 and Δ_2 is denoted by $u \;{}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2}\; v$ ($u \;{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2}\; v$) and is defined as $u \;{}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2}\; v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid w \text{ is an rS-shuffle on } \Gamma \text{ of } u \text{ and } v \text{ w.r.t. } \Delta_1 \text{ and } \Delta_2\}$ and $u \;{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2}\; v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid w \text{ is a fair rS-shuffle on } \Gamma \text{ of } u \text{ and } v \text{ w.r.t. } \Delta_1 \text{ and } \Delta_2\}$, respectively.

For $L_1 \subseteq \Delta_1^\infty$ and $L_2 \subseteq \Delta^\infty$ the *(fair) rS-shuffle* on Γ of L_1 and L_2 w.r.t. Δ_1 and Δ_2 is denoted by $L_1 \;{}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2}\; L_2$ ($L_1 \;{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2}\; L_2$) and is defined as the language consisting of all (fair) rS-shuffles on Γ of a word

from L_1 and a word from L_2 w.r.t. Δ_1 and Δ_2. Thus $L_1 \; {}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2} \; L_2 = \{w \in u \; {}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2} \; v \mid u \in L_1, \; v \in L_2\} = \bigcup_{u \in L_1, \, v \in L_2} (u \; {}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2} \; v)$ and $L_1 \; {}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2} \; L_2 = \bigcup_{u \in L_1, \, v \in L_2} (u \; {}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2} \; v)$, respectively.

Example 6.4.6. (Example 6.4.4 continued) Now $u \; {}_{\Delta}\underline{||}^{\{c\}}_{\Delta} \; v = u \; {}_{\Delta}\underline{|||}^{\{c\}}_{\Delta} \; v = \{abcd\}$, whereas $u \; {}_{\Delta}\underline{||}^{\{b,c\}}_{\Delta} v = u \; {}_{\Delta}\underline{|||}^{\{b,c\}}_{\Delta} v = \varnothing$. Furthermore, $u \; {}_{\Delta_1}\underline{||}^{\{c\}}_{\Delta_2} \; v = u \; {}_{\Delta_1}\underline{|||}^{\{c\}}_{\Delta_2} \; v = u \; {}_{\Delta_1}\underline{|||}^{\{b,c\}}_{\Delta_2} \; v = u \; {}_{\Delta_1}\underline{||}^{\{b,c\}}_{\Delta_2} \; v = \{abcd\} = u \; {}_{\Delta_1}\underline{||}_{\Delta_2} \; v = u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v = u \; |||^{\{c\}} \; v = u \; ||^{\{c\}} \; v$.

We moreover have $w_1 \; {}_{\Delta}\underline{||}^{\{a\}}_{\Delta} \; a = w_1 \; {}_{\Delta}\underline{|||}^{\{a\}}_{\Delta} \; a = \varnothing$, with $a \in \Delta^*$, and $w_1 \; {}_{\Delta}\underline{||}^{\{a\}}_{\Delta} \; w_1 = w_1 \; {}_{\Delta}\underline{|||}^{\{a\}}_{\Delta} \; w_1 = \{a^\omega\} = w_1 \; {}_{\Delta}\underline{||}_{\Delta} \; w_1 = w_1 \; {}_{\Delta}\underline{|||}_{\Delta} \; w_1 = w_1 \; |||^{\{a\}} \; w_1 = w_1 \; ||^{\{a\}} \; w_1$. We also have $w_1 \; {}_{\Delta}\underline{||}^{\{a\}}_{\Delta} \; w_2 = w_1 \; {}_{\Delta}\underline{|||}^{\{a\}}_{\Delta} \; w_2 = \varnothing$, $w_1 \; {}_{\Delta_a}\underline{||}^{\{a\}}_{\Delta_b} \; w_2 = w_1 \; || \; w_2$, and $w_1 \; {}_{\Delta_a}\underline{|||}^{\{a\}}_{\Delta_b} \; w_2 = w_1 \; ||| \; w_2$.

Finally, here $w_{12} \; {}_{\Delta}\underline{||}^{\{a\}}_{\Delta} \; w_{21} = w_{12} \; {}_{\Delta}\underline{|||}^{\{a\}}_{\Delta} \; w_{21} = \{(bab)^\omega\}$, whereas $w_{12} \; {}_{\Delta}\underline{||}^{\{a,b\}}_{\Delta} w_{21} = w_{12} \; {}_{\Delta}\underline{|||}^{\{a,b\}}_{\Delta} w_{21} = \varnothing$. □

We now take a closer look at the three synchronized shuffles just introduced. We immediately note that the rS-shuffle can be considered to lie inbetween the S-shuffle and the fS-shuffle. In fact, the following results follow directly from Definitions 6.4.1, 6.4.3, and 6.4.5.

Lemma 6.4.7. *Let $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$. Let $K \subseteq \Delta_1^\infty$ and $L \subseteq \Delta_2^\infty$. Let Γ be an alphabet. Then*

(1) if $\Gamma \subseteq \Delta_1 \cap \Delta_2$, then $u \; {}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2} \; v = u \; |||^{\Gamma} \; v$, $u \; {}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2} \; v = u \; ||^{\Gamma} \; v$, $K \; {}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2} \; L = K \; |||^{\Gamma} \; L$, and $K \; {}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2} \; L = K \; ||^{\Gamma} \; L$, and

(2) if $\Gamma \supseteq \Delta_1 \cap \Delta_2$, then $u \; {}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2} \; v = u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v$, $u \; {}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2} \; v = u \; {}_{\Delta_1}\underline{||}_{\Delta_2} \; v$, $K \; {}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2} \; L = K \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; L$, and $K \; {}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2} \; L = K \; {}_{\Delta_1}\underline{||}_{\Delta_2} \; L$. □

We continue by pointing out that for arbitrary alphabets Δ_1, Δ_2, and Γ, both $u \; {}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2} \; v$ and $u \; {}_{\Delta_1}\underline{||}_{\Delta_2} \; v$ are undefined if either $u \notin \Delta_1^\infty$ or $v \notin \Delta_2^\infty$.

Finally, we show how this section's synchronized shuffles are related to the shuffle of the previous section. From Definition 6.4.1 we immediately obtain that the S-shuffle is indeed a generalization of the shuffle.

Lemma 6.4.8. *Let $u, v \in \Delta^\infty$ and let $K, L \subseteq \Delta^\infty$. Then*

(1) $u \; |||^{\varnothing} \; v = u \; ||| \; v$ and $u \; ||^{\varnothing} \; v = u \; || \; v$, and

(2) $K \; |||^{\varnothing} \; L = K \; ||| \; L$ and $K \; ||^{\varnothing} \; L = K \; || \; L$. □

Together with Example 6.3.2, this lemma implies that the inclusions of the fair S-shuffle on an alphabet Γ of languages in the S-shuffle on Γ of these languages may be proper. Furthermore, an S-shuffle on an alphabet Γ of languages is always fair in case both languages are finitary.

Moreover, the rS-shuffle degenerates to the shuffle if there are no letters to synchronize on.

Lemma 6.4.9. *Let $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$. Let $K \subseteq \Delta_1^\infty$ and $L \subseteq \Delta_2^\infty$. Then*

(1) $u \; {}_{\Delta_1}\underline{|||}^{\varnothing}_{\Delta_2} \; v = u \;|||^{\varnothing}\; v = u \;|||\; v$ *and* $u \; {}_{\Delta_1}\underline{||}^{\varnothing}_{\Delta_2} \; v = u \;||^{\varnothing}\; v = u \;||\; v$, *and*

(2) $K \; {}_{\Delta_1}\underline{|||}^{\varnothing}_{\Delta_2} \; L = K \;|||^{\varnothing}\; L = K \;|||\; L$ *and* $K \; {}_{\Delta_1}\underline{||}^{\varnothing}_{\Delta_2} \; L = K \;||^{\varnothing}\; L = K \;||\; L$. □

Similarly, the fS-shuffle is a generalization of the shuffle in case of disjoint alphabets.

Lemma 6.4.10. *Let $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$. Let $K \subseteq \Delta_1^\infty$ and $L \subseteq \Delta_2^\infty$. Let $\Delta_1 \cap \Delta_2 = \varnothing$. Then*

(1) $u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v = u \;|||^{\varnothing}\; v = u \;|||\; v$ *and* $u \; {}_{\Delta_1}\underline{||}_{\Delta_2} \; v = u \;||^{\varnothing}\; v = u \;||\; v$, *and*

(2) $K \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; L = K \;|||^{\varnothing}\; L = K \;|||\; L$ *and* $K \; {}_{\Delta_1}\underline{||}_{\Delta_2} \; L = K \;||^{\varnothing}\; L = K \;||\; L$. □

6.4.2 Basic Observations

We have seen that a (fair) shuffle of two words always exists. From Example 6.4.2 we however conclude that a (fair) S-shuffle of two nonempty words need not exist. In fact, we have the following result.

Lemma 6.4.11. *Let $u, v \in \Delta^\infty$ and let Γ be an alphabet. Then*

(1) for all $w \in u \;||^\Gamma\; v$, $pres_\Gamma(w) = pres_\Gamma(u) = pres_\Gamma(v)$, and

(2) $u \;||^\Gamma\; v = \varnothing$ if and only if $pres_\Gamma(u) \neq pres_\Gamma(v)$.

Proof. (1) This follows immediately from Definition 6.4.1.

(2) (If) Let $u \;||^\Gamma\; v \neq \varnothing$. Then (1) implies that $\mathrm{pres}_\Gamma(u) = \mathrm{pres}_\Gamma(v)$.

(Only if) Let $\mathrm{pres}_\Gamma(u) = \mathrm{pres}_\Gamma(v) = w$. According to Definition 6.4.1 we thus need to distinguish two cases.

If there exists an $n \geq 0$ such that $w = x_1 x_2 \cdots x_n$, with $x_i \in \Gamma$ for all $i \in [n]$, then it must be the case that $u = u_1 x_1 u_2 x_2 \cdots x_n u_{n+1}$ and $v = v_1 x_1 v_2 x_2 \cdots x_n v_{n+1}$, with $u_i, v_i \in (\Delta \backslash \Gamma)^*$ for all $i \in [n]$ and $u_{n+1}, v_{n+1} \in$

$(\Delta \setminus \Gamma)^\infty$. Hence $u \,||^\Gamma\, v = (u_1 \,||\, v_1)x_1(u_2 \,||\, v_2)x_2 \cdots x_n(u_{n+1} \,||\, v_{n+1}) \neq \varnothing$ because for all $i \in [n+1]$, $u_i \,||\, v_i \neq \varnothing$.

If $w = x_1x_2\cdots$, with $x_i \in \Gamma$ for all $i \geq 1$, then it must be the case that $u = u_1x_1u_2x_2\cdots$ and $v = v_1x_1v_2x_2\cdots$, with $u_i, v_i \in (\Delta \setminus \Gamma)^*$ for all $i \geq 1$. Hence $u \,||^\Gamma\, v = (u_1 \,||\, v_1)x_1(u_2 \,||\, v_2)x_2 \cdots \neq \varnothing$ because for all $i \geq 1$, $u_i \,||\, v_i \neq \varnothing$. □

We have also seen that the only (fair) shuffle of an arbitrary word and the empty word is the given word itself. Due to the requirement of a matching backbone, we immediately conclude that this in general does not hold when any of the (fair) synchronized shuffles is considered.

In Lemma 6.3.10, finally, we have seen that the length of every word in the shuffle of two finite words equals the sum of the lengths of those two words. Any synchronized shuffle of two finite words, however, may be a word of length less than the sum of the lengths of those two words. This is due to the fact that each letter from the synchronization alphabet must occur in both words being shuffled, while it occurs only once in the backbone of each synchronized shuffle of those words.

In the remainder of this subsection we seek to express the S-shuffles of possibly infinite words in terms of the S-shuffles of their finite prefixes. We begin by considering the case in which two words that are S-shuffled share a finite backbone (cf. Definition 6.4.1(1)). In such words u and v we can thus distinguish initial prefixes u_1 and v_1 ending with the last letter of the finite backbone, and suffixes u_2 and v_2 containing no more letters from the alphabet of the backbone. It is clear that elements of the S-shuffle of u and v then consist of a prefix that is part of the S-shuffle of u_1 and v_1 and a suffix that is part of the shuffle of u_2 and v_2. This leads to the following result.

Lemma 6.4.12. *Let Γ be an alphabet, let $u_1, v_1 \in ((\Delta \setminus \Gamma)^*\Gamma)^*$, and let $u_2, v_2 \in (\Delta \setminus \Gamma)^\infty$. Then*

(1) $(u_1 \,|||^\Gamma\, v_1)(u_2 \,|||\, v_2) = u_1u_2 \,|||^\Gamma\, v_1v_2$ *and*

(2) $(u_1 \,||^\Gamma\, v_1)(u_2 \,||\, v_2) = u_1u_2 \,||^\Gamma\, v_1v_2$. □

Note that this lemma resembles Lemma 6.3.14. The main difference between the two lemmata is the fact that the statements of Lemma 6.4.12 consist of equalities rather than inclusions from left to right only. The reason lies in the fact that the application of Lemma 6.4.12 is limited to prefixes which end at a predetermined position, viz. at the end of the backbone (which thus dictates the structure of all S-shuffles).

Lemma 6.4.12 consequently allows us to conclude that whenever the prefixes of an infinite word w are included in the S-shuffle of the prefixes of two

words u and v sharing a finite backbone, then w is an element of the S-shuffle of u and v. In fact we prove a more general statement, immediately for prefixes of limited-closed languages (cf. Corollary 6.3.48 and Theorem 6.3.49).

Lemma 6.4.13. *Let $K, L \subseteq \Delta^\infty$ be limit closed, let Γ be an alphabet, and let $w = w_1 w_2$ be such that $w_1 \in ((\Delta \setminus \Gamma)^*\Gamma)^*$ and $w_2 \in (\Delta \setminus \Gamma)^\omega$. Then*

if $\mathit{pref}(w) \subseteq \mathit{pref}(K) \mid\mid^\Gamma \mathit{pref}(L)$, then $w \in K \mid\mid^\Gamma L$.

Proof. Let $\mathrm{pref}(w) \subseteq \mathrm{pref}(K) \mid\mid^\Gamma \mathrm{pref}(L)$. Then there exist an $n \geq 1$, $u_i \in \mathrm{pref}(K)$ and $v_i \in \mathrm{pref}(L)$ such that $w_1 \in u_i \mid\mid^\Gamma v_i$, for all $i \in [n]$. Note that according to Definition 6.4.1, all $u_i, v_i \in ((\Delta \setminus \Gamma)^*\Gamma)^*$. For all $i \in [n]$, let $K_{u_i} = \{u \in (\Delta \setminus \Gamma)^* \mid u_i u \in K\}$ and let $L_{v_i} = \{v \in (\Delta \setminus \Gamma)^* \mid v_i v \in L\}$.

Let $z \in \mathrm{pref}(w_2)$ and consider the word $w_1 z$. Thus $w_1 z \in \mathrm{pref}(K) \mid\mid^\Gamma \mathrm{pref}(L)$ because $w_1 z \in \mathrm{pref}(w)$. Hence there exist $u \in \mathrm{pref}(K)$ and $v \in \mathrm{pref}(L)$ such that $w_1 z \in u \mid\mid^\Gamma v$. Again by Definition 6.4.1 we know that $u = u'u''$ and $v = v'v''$, with $u', v' \in ((\Delta \setminus \Gamma)^*\Gamma)^*$ and $u'', v'' \in (\Delta \setminus \Gamma)^*$, and $w_1 \in u' \mid\mid^\Gamma v'$. Hence there exists an $i \in [n]$ such that $u' = u_i$ and $v' = v_i$. This implies that $w_1 z \in u_i \mathrm{pref}(K_{u_i}) \mid\mid^\Gamma v_i \mathrm{pref}(K_{v_i})$. Consequently, by Lemma 6.4.12(2), $\mathrm{pref}(w_2) \subseteq \bigcup_{i \in [n]} (\mathrm{pref}(K_{u_i}) \mid\mid^\Gamma \mathrm{pref}(L_{v_i})) = \bigcup_{i \in [n]} (\mathrm{pref}(K_{u_i}) \mid\mid \mathrm{pref}(L_{v_i}))$ (the equality follows because $\mathrm{pref}(K_{u_i})$ and $\mathrm{pref}(L_{v_i})$, with $i \in [n]$, do not contain letters from Γ).

Since the number of pairs u_i and v_i, with $i \in [n]$, is finite, it must be the case that there exists a $j \in [n]$ such that for each $z \in \mathrm{pref}(w_2)$ there exists a prefix z' of w_2 such that $z < z'$ and for which $z' \in \mathrm{pref}(K_{u_j}) \mid\mid \mathrm{pref}(L_{v_j})$ and thus $z \in \mathrm{pref}(K_{u_j}) \mid\mid \mathrm{pref}(L_{v_j})$. Hence $\mathrm{pref}(w_2) \subseteq \mathrm{pref}(K_{u_j}) \mid\mid \mathrm{pref}(L_{v_j})$. Since K and L are limit closed, so are K_{u_j} and L_{v_j}. Lemma 6.3.46 then implies that $w_2 \in K_{u_j} \mid\mid L_{v_j}$. Hence with Lemma 6.4.12(2) we obtain $w = w_1 w_2 \in (u_j \mid\mid^\Gamma v_j)(K_{u_j} \mid\mid L_{v_j}) = u_j K_{u_j} \mid\mid^\Gamma v_j L_{v_j} \subseteq K \mid\mid^\Gamma L$. □

A similar statement can be proven for infinite words.

Lemma 6.4.14. *Let $K, L \subseteq \Delta^\infty$ be limit closed, let Γ be an alphabet, and let $w \in ((\Delta \setminus \Gamma)^*\Gamma)^\omega$. Then*

if $\mathit{pref}(w) \subseteq \mathit{pref}(K) \mid\mid^\Gamma \mathit{pref}(L)$, then $w \in K \mid\mid^\Gamma L$.

Proof. Let $\mathrm{pref}(w) \subseteq \mathrm{pref}(K) \mid\mid^\Gamma \mathrm{pref}(L)$. Let $w_1, w_2, \ldots \in (\Delta \setminus \Gamma)^*$ and $x_1, x_2, \ldots \in \Gamma$ be such that $w = w_1 x_1 w_2 x_2 \cdots$.

Since $\mathrm{pref}(w) \subseteq \mathrm{pref}(K) \mid\mid^\Gamma \mathrm{pref}(L)$ we know that for all $n \geq 1$ there exists a sequence $\rho = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$, with $u_i, v_i \in (\Delta \setminus \Gamma)^*$ for all $i \in [n]$, and such that $u_1 x_1 u_2 x_2 \cdots u_n x_n \in \mathrm{pref}(K)$, $v_1 x_1 v_2 x_2 \cdots v_n x_n \in \mathrm{pref}(L)$, and $w_i \in (u_i \mid\mid v_i)$ for all $i \in [n]$. That is, $w_1 x_1 w_2 x_2 \cdots w_n x_n \in$

$(u_1||v_1)x_1(u_2||v_2)x_2\cdots(u_n||v_n)x_n = u_1x_1u_2x_2\cdots u_nx_n||^{\Gamma}v_1x_1v_2x_2\cdots v_nx_n$. We will refer to $w_1x_1w_2x_2\cdots w_nx_n$ as $w(n)$ and to ρ as a $(K\ ||^{\Gamma}\ L)$-deco of $w(n)$.

We say that a $(K\,||^{\Gamma}\,L)$-deco $\rho = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$ of $w(n)$ directly precedes a $(K\,||^{\Gamma}\,L)$-deco ρ' of $w(n+1)$ if $\rho' = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n, u_{n+1}, v_{n+1})$. We furthermore add a trivial ρ_λ which by definition directly precedes every $(K\ ||^{\Gamma}\ L)$-deco of $w(1)$.

For $n \geq 1$, let $V_n = \{\rho \mid \rho \text{ is a } (K\ ||^{\Gamma}\ L)\text{-deco of } w(n)\}$ be the set containing every possible $(K\ ||^{\Gamma}\ L)$-deco of $w(n)$. Let $V_0 = \{\rho_\lambda\}$. Note that each V_n is finite, for $n \geq 0$, and that $V_n \cap V_{n'} = \varnothing$, for all $n > n' \geq 0$. Furthermore, let $E = \{(\rho, \rho') \mid \rho \text{ directly precedes } \rho'\}$. 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. In fact, G is an infinite finitely-branching rooted tree with root ρ_λ. This can be seen as follows. Except for ρ_λ, every vertex $\rho = (u_1, v_1, u_2, v_2, \ldots, u_{k+1}, v_{k+1})$ has exactly one incoming edge, viz. from ρ_λ if $k = 0$ and from $(u_1, v_1, u_2, v_2, \ldots, u_k, v_k)$ if $k \geq 1$. Note that this $(u_1, v_1, u_2, v_2, \ldots, u_k, v_k)$ is indeed a $(K\ ||^{\Gamma}\ L)$-deco of $w(k)$. Since each w_i, with $i \in [n]$, is a finite word, every vertex moreover has a finite number of outgoing edges. Finally, the graph is infinite since it has at least one distinct vertex for every prefix $w(n)$ of w.

We can thus use König's Lemma to conclude that there exists an infinite path π through G, starting in the root ρ_λ. Let $\pi = (\rho_\lambda, \rho_1, \rho_2, \ldots)$, with $\rho_n = (u_1, v_1, u_2, v_2, \ldots, u_n, v_n)$ for all $n \geq 1$. Then by definition $u_1x_1u_2x_2\cdots u_nx_n \in \mathrm{pref}(K)$ and $v_1x_1v_2x_2\cdots v_nx_n \in \mathrm{pref}(L)$. Since K and L are limit closed this implies that $u = \lim_{n\to\infty} u_1x_1u_2x_2\cdots u_nx_n \in K$ and $v = \lim_{n\to\infty} v_1x_1v_2x_2\cdots v_nx_n \in L$. By the definition of the $(K\ ||^{\Gamma}\ L)$-deco of $w(n)$ we thus obtain that $w = w_1x_1w_2x_2\cdots \in (u_1\ |||\ v_1)x_1(u_2\ |||\ v_2)x_2\cdots = u\ ||^{\Gamma}\ v$. Hence $w \in K\ ||^{\Gamma}\ L$. □

The preceding two lemmata allow us to express — as we did for the shuffle in Theorem 6.3.49(2) — the S-shuffle of possibly infinite words in terms of the S-shuffle of finite prefixes of those possibly infinite words.

Theorem 6.4.15. *Let $K, L \subseteq \Delta^\infty$ be limit closed, let $w \in \Delta^\omega$, and let Γ be an alphabet. Then*

$$w \in K\ ||^{\Gamma}\ L \text{ if and only if } \mathit{pref}(w) \subseteq \mathit{pref}(K)\ ||^{\Gamma}\ \mathit{pref}(L).$$

Proof. (If) Let $\mathrm{pref}(w) \subseteq \mathrm{pref}(K)\ ||^{\Gamma}\ \mathrm{pref}(L)$. Then by Definition 6.4.1 and Lemmata 6.4.13 and 6.4.14 it follows that $w \in K\ ||^{\Gamma}\ L$.

(Only if) Let $w \in K\ ||^{\Gamma}\ L$. Then according to Definition 6.4.1 one of the following two cases holds.
Either $w = (u_1\ ||\ v_1)x_1(u_2\ ||\ v_2)x_2\cdots x_{n-1}(u_n\ ||\ v_n)$ for some $n \geq 1$, $u_i, v_i \in$

$(\Delta \setminus \Gamma)^*$ for all $i \in [n-1]$, $u_n, v_n \in (\Delta \setminus \Gamma)^\infty$, and $x_i \in \Gamma$ for all $i \in [n-1]$, and such that $u = u_1x_1u_2x_2 \cdots x_nu_{n+1}$ and $v = v_1x_1v_2x_2 \cdots x_nv_{n+1}$.
Or else $w = (u_1 \mid\mid v_1)x_1(u_2 \mid\mid v_2)x_2 \cdots$ for some $n \geq 1$, $u_i, v_i \in (\Delta \setminus \Gamma)^*$ for all $i \geq 1$, and $x_i \in \Gamma$ for all $i \geq 1$, and such that $u = u_1x_1u_2x_2 \cdots x_nu_{n+1}$ and $v = v_1x_1v_2x_2 \cdots x_nv_{n+1}$.

Consequently we consider a prefix $y \in \mathrm{pref}(w)$. Then in both cases $y = (u_1 \mid\mid v_1)x_1(u_2 \mid\mid v_2)x_2 \cdots x_{k-1}x$ for some $k \geq 1$ and $x \in \mathrm{pref}(u_k \mid\mid v_k)$. Immediately from Definition 6.4.1 and Theorem 6.3.21(1) it then follows that $y \in u_1x_1u_2x_2 \cdots u_{k-1}x_{k-1}\mathrm{pref}(u_k) \mid\mid^\Gamma v_1x_1v_2x_2 \cdots v_{k-1}x_{k-1}\mathrm{pref}(v_k) \subseteq \mathrm{pref}(u) \mid\mid^\Gamma \mathrm{pref}(v)$. Hence $y \in \mathrm{pref}(K) \mid\mid^\Gamma \mathrm{pref}(L)$. □

Since all singleton languages are limit closed, we immediately obtain the following result.

Theorem 6.4.16. *Let $u, v \in \Delta^\infty$, let $w \in \Delta^\omega$, and let Γ be an alphabet. Then*

$w \in u \mid\mid^\Gamma v$ *if and only if* $\mathrm{pref}(w) \subseteq \mathrm{pref}(u) \mid\mid^\Gamma \mathrm{pref}(v)$. □

6.4.3 Commutativity and Associativity

In order to use the (fair) synchronized shuffles in the context of team automata, it is important to establish certain commutativity and associativity properties.

The (fair) S-shuffle is defined on the basis of the (fair) shuffle, which is commutative. Hence the commutativity of the (fair) S-shuffle is a direct consequence of the commutativity of the (fair) shuffle, as stated in Theorem 6.3.8.

Theorem 6.4.17. *Let $u, v \in \Delta^\infty$ and let Γ be an alphabet. Then*

(1) $u \mid\mid\mid^\Gamma v = v \mid\mid\mid^\Gamma u$ *and* $u \mid\mid^\Gamma v = v \mid\mid^\Gamma u$, *and*

(2) $L_1 \mid\mid\mid^\Gamma L_2 = L_2 \mid\mid\mid^\Gamma L_1$ *and* $L_1 \mid\mid^\Gamma L_2 = L_2 \mid\mid^\Gamma L_1$. □

Recall that both rS-shuffles and fS-shuffles are defined in terms of S-shuffles. Consequently, also these synchronized shuffles may be considered commutative in the following sense.

Corollary 6.4.18. *Let $u, v \in \Delta^\infty$, let $L_1, L_2 \subseteq \Delta^\infty$, and let Γ be an alphabet. Then*

(1) $u \;{}_{\Delta_1}\underline{|||}^\Gamma_{\Delta_2}\; v = v \;{}_{\Delta_2}\underline{|||}^\Gamma_{\Delta_1}\; u$ *and* $u \;{}_{\Delta_1}\underline{||}^\Gamma_{\Delta_2}\; v = v \;{}_{\Delta_2}\underline{||}^\Gamma_{\Delta_1}\; u$, *and*

(2) $L_1 \;{}_{\Delta_1}\underline{|||}^\Gamma_{\Delta_2}\; L_2 = L_2 \;{}_{\Delta_2}\underline{|||}^\Gamma_{\Delta_1}\; L_1$ *and* $L_1 \;{}_{\Delta_1}\underline{||}^\Gamma_{\Delta_2}\; L_2 = L_2 \;{}_{\Delta_2}\underline{||}^\Gamma_{\Delta_1}\; L_1$. □

Corollary 6.4.19. *Let $u, v \in \Delta^\infty$ and let $L_1, L_2 \subseteq \Delta^\infty$. Then*

(1) $u \ {}_{\Delta_1}\underline{|||}_{\Delta_2} \ v = v \ {}_{\Delta_2}\underline{|||}_{\Delta_1} \ u$ *and* $u \ {}_{\Delta_1}\underline{||}_{\Delta_2} \ v = v \ {}_{\Delta_2}\underline{||}_{\Delta_1} \ u$, *and*

(2) $L_1 \ {}_{\Delta_1}\underline{|||}_{\Delta_2} \ L_2 = L_2 \ {}_{\Delta_2}\underline{|||}_{\Delta_1} \ L_1$ *and* $L_1 \ {}_{\Delta_1}\underline{||}_{\Delta_2} \ L_2 = L_2 \ {}_{\Delta_2}\underline{||}_{\Delta_1} \ L_1$. □

It remains to prove that also in case of synchronized shuffles a notion of associativity can be upheld. In case of S-shuffles, associativity is easily understood. S-shuffling is associative because $\{u\} \ ||^\Gamma \ (v \ ||^\Gamma \ w)$ equals $(u \ ||^\Gamma \ v) \ ||^\Gamma \ \{w\}$, for words u, v, and w, and an alphabet Γ, and likewise for the fair case. To prove this statement we use the associativity of (fair) shuffling.

Theorem 6.4.20. *Let $u, v, w \in \Delta^\infty$ and let Γ be an alphabet. Then*

(1) $\{u\} \ |||^\Gamma \ (v \ |||^\Gamma \ w) = (u \ |||^\Gamma \ v) \ |||^\Gamma \ \{w\}$ *and*

(2) $\{u\} \ ||^\Gamma \ (v \ ||^\Gamma \ w) = (u \ ||^\Gamma \ v) \ ||^\Gamma \ \{w\}$.

Proof. (1) Let $x \in \{u\} \ |||^\Gamma \ (v \ |||^\Gamma \ w)$. Then by Lemma 6.4.11, $\text{pres}_\Gamma(x) = \text{pres}_\Gamma(u) = \text{pres}_\Gamma(v) = \text{pres}_\Gamma(w)$. Now let $y = \text{pres}_\Gamma(x)$. We distinguish two cases.

First consider that $y \in \Gamma^*$. Then there exists an $n \geq 0$ such that $y = y_1 y_2 \cdots y_n$ with $y_i \in \Gamma$, for all $i \in [n]$. Consequently there exist $x_1, x_2, \ldots, x_n, u_1, u_2, \ldots, u_n, v_1, v_2, \ldots, v_n, w_1, w_2, \ldots, w_n \in \Gamma^*$ and x_{n+1}, $u_{n+1}, v_{n+1}, w_{n+1} \in \Gamma^\infty$ such that $x = x_1 y_1 x_2 y_2 \cdots x_n y_n x_{n+1}$, $u = u_1 y_1 u_2 y_2 \cdots u_n y_n u_{n+1}$, $v = v_1 y_1 v_2 y_2 \cdots v_n y_n v_{n+1}$, and $w = w_1 y_1 w_2 y_2 \cdots w_n y_n w_{n+1}$. By Definition 6.4.1, $x_i \in \{u_i\} \ || \ (v_i \ || \ w_i)$, for all $i \in [n]$, and $x_{n+1} \in \{u_{n+1}\} \ ||| \ (v_{n+1} \ ||| \ w_{n+1})$. Now by Theorem 6.3.51(1), $\{u_i\} \ || \ (v_i \ || \ w_i) = (u_i \ || \ v_i) \ || \ \{w_i\}$, for all $i \in [n]$, and according to Theorem 6.3.32(1), $\{u_{n+1}\} \ ||| \ (v_{n+1} \ ||| \ w_{n+1}) = (u_{n+1} \ ||| \ v_{n+1}) \ ||| \ \{w_{n+1}\}$. This implies, again by Definition 6.4.1, that $x \in (u \ |||^\Gamma \ v) \ |||^\Gamma \ \{w\}$.

Secondly, the case that $y \in \Gamma^\infty$ is analogous.

(2) Analogous. □

Theorem 6.4.21. *Let $L_1, L_2, L_3 \subseteq \Delta^\infty$ and let Γ be an alphabet. Then*

(1) $L_1 \ |||^\Gamma \ (L_2 \ |||^\Gamma \ L_3) = (L_1 \ |||^\Gamma \ L_2) \ |||^\Gamma \ L_3$ *and*

(2) $L_1 \ ||^\Gamma \ (L_2 \ ||^\Gamma \ L_3) = (L_1 \ ||^\Gamma \ L_2) \ ||^\Gamma \ L_3$.

Proof. Analogous to the proof of Theorem 6.3.32(2). □

The statements of the preceding two theorems do not hold when the synchronization alphabet Γ may vary. Given $w_1, w_2, w_3 \in \Delta^*$ and two distinct alphabets Γ and Γ', e.g., $(w_1 \ ||^\Gamma \ w_2) \ ||^{\Gamma'} \ w_3$ in general does not equal $w_1 \ ||^\Gamma \ (w_2 \ ||^{\Gamma'} \ w_3)$. This is shown in the following example.

Example 6.4.22. Let $L_1 = \{a\}$, let $L_2 = \{a, b\}$, and let $L_3 = \{ab\}$. Then $(L_1 \,||^{\{a\}}\, L_2) \,||^{\{b\}}\, L_3 = \{a\} \,||^{\{b\}}\, \{ab\} = \varnothing$, whereas $L_1 \,||^{\{a\}}\, (L_2 \,||^{\{b\}}\, L_3) = \{a\} \,||^{\{a\}}\, \{ab\} = \{ab\}$. □

It is worthwhile to notice here that the synchronized shuffle as studied in [DeS84] and [LR99] is not associative, as is noted in [LR99] and shown in the following example. Recall that the 'produit de mixage' or synchonized shuffle of $L_1, L_2 \subseteq \Delta^\infty$ is defined as $L_1 \,|||^{\mathrm{alph}(L_1)\cap\mathrm{alph}(L_2)}\, L_2$, where $\mathrm{alph}(L)$ — for an alphabet L — is defined as $\mathrm{alph}(L) = \bigcup_{w\in L} \mathrm{alph}(w)$.

Example 6.4.23. (Example 6.4.22 continued) Now $L_1 \,||^{\mathrm{alph}(L_1)\cap\mathrm{alph}(L_2)}\, L_2 = \{a\} \,||^{\{a\}}\, \{a, b\} = \{a\}$ and thus $\{a\} \,||^{\mathrm{alph}(\{a\})\cap\mathrm{alph}(L_3)}\, L_3 = \{a\} \,||^{\{a\}}\, \{ab\} = \{ab\}$, while on the other hand $L_2 \,||^{\mathrm{alph}(L_2)\cap\mathrm{alph}(L_3)}\, L_3 = \{a, b\} \,||^{\{a,b\}}\, \{ab\} = \varnothing$ and thus $L_1 \,||^{\mathrm{alph}(L_1)\cap\mathrm{alph}(\{ab\})}\, \varnothing = \{a\} \,||^{\{a\}}\, \varnothing = \varnothing$. □

In [vdS85] it is noted that the weave operation studied there is on purpose not defined as the synchronized shuffle operation of [DeS84] and [LR99] because in that case it would no longer have been associative.

Contrary to the case of the S-shuffle, the synchronization alphabet of an fS-shuffle or an rS-shuffle depends on the alphabets involved. Hence it is not immediately clear how associativity should be formalized. A natural approach would be to consider fS-shuffling associative if $\{u\} \,{}_{\Delta_1}\!\underline{||}_{\Delta_2\cup\Delta_3}\, (v \,{}_{\Delta_2}\!\underline{||}_{\Delta_3}\, w)$ equals $(u \,{}_{\Delta_1}\!\underline{||}_{\Delta_2}\, v) \,{}_{\Delta_1\cup\Delta_2}\!\underline{||}_{\Delta_3}\, \{w\}$ for all words $u \in \Delta_1^\infty$, $v \in \Delta_2^\infty$, and $w \in \Delta_3^\infty$, and similarly rS-shuffling and the fair cases.

We now present an example to illustrate this idea.

Example 6.4.24. (Example 6.4.4 continued) Recall that we have set $\Delta_1 = \{a, b, c\}$, $\Delta_2 = \{c, d\}$, $u = abc \in \Delta_1^*$, and $v = cd \in \Delta_2^*$. Now we let $\Delta_3 = \{b, c, e\}$ and we let $w = bce \in \Delta_3^*$. Then it follows immediately that $\{u\} \,{}_{\Delta_1}\!\underline{||}_{\Delta_2\cup\Delta_3}\, (v \,{}_{\Delta_2}\!\underline{||}_{\Delta_3}\, w) = \{abc\} \,{}_{\{a,b,c\}}\!\underline{||}_{\{b,c,d,e\}}\, (cd \,{}_{\{c,d\}}\!\underline{||}_{\{b,c,e\}}\, bce) = \{abc\} \,{}_{\{a,b,c\}}\!\underline{||}_{\{b,c,d,e\}}\, \{bcde, bced\} = \{abcde, abced\} = abcd \,{}_{\{a,b,c,d\}}\!\underline{||}_{\{b,c,e\}}\, bce = (abc \,{}_{\{a,b,c\}}\!\underline{||}_{\{c,d\}}\, cd) \,{}_{\{a,b,c,d\}}\!\underline{||}_{\{b,c,e\}}\, \{bce\} = (u \,{}_{\Delta_1}\!\underline{||}_{\Delta_2}\, v) \,{}_{\Delta_1\cup\Delta_2}\!\underline{||}_{\Delta_3}\, \{w\}$.

Next we let $\Gamma = \{b, c\}$. Consequently, it follows immediately that $\{u\} \,{}_{\Delta_1}\!\underline{||}^{\Gamma}_{\Delta_2\cup\Delta_3}\, (v \,{}_{\Delta_2}\!\underline{||}^{\Gamma}_{\Delta_3}\, w) = \{abc\} \,{}_{\{a,b,c\}}\!\underline{||}^{\{b,c\}}_{\{b,c,d,e\}}\, (cd \,{}_{\{c,d\}}\!\underline{||}^{\{b,c\}}_{\{b,c,e\}}\, bce) = \{abc\} \,{}_{\{a,b,c\}}\!\underline{||}^{\{b,c\}}_{\{b,c,d,e\}}\, \{bcde, bced\} = \{abcde, abced\} = abcd \,{}_{\{a,b,c,d\}}\!\underline{||}^{\{b,c\}}_{\{b,c,e\}}\, bce = (abc \,{}_{\{a,b,c\}}\!\underline{||}^{\{b,c\}}_{\{c,d\}}\, cd) \,{}_{\{a,b,c,d\}}\!\underline{||}^{\{b,c\}}_{\{b,c,e\}}\, \{bce\} = (u \,{}_{\Delta_1}\!\underline{||}^{\Gamma}_{\Delta_2}\, v) \,{}_{\Delta_1\cup\Delta_2}\!\underline{||}^{\Gamma}_{\Delta_3}\, \{w\}$. □

This example dealt with finite words and hence fair fS-shuffles and fair rS-shuffles. Before turning to the general case we now prove that indeed fair fS-shuffling and fair rS-shuffling are associative in the sense just discussed. The following characterization of the fair shuffles of two words over disjoint alphabets in terms of preserving homomorphisms turns out to be very useful.

We give here a full direct proof, but the statement can also be proven by modification of Theorem 6.3.29 and its proof (using $\text{pres}_{\Delta_1}^{-1}$ and $\text{pres}_{\Delta_2}^{-1}$ instead of the inverse homomorphisms $\varphi_{i,\{j\}}^{-1}$ and $\varphi_{j,\{i\}}^{-1}$).

Lemma 6.4.25. *Let $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$ be such that $\Delta_1 \cap \Delta_2 = \varnothing$. Then*

$$u \,|||\, v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid pres_{\Delta_1}(w) = u,\ pres_{\Delta_2}(w) = v\}.$$

Proof. ($\subseteq$) Let $w \in u \,|||\, v$. Since $\Delta_1 \cap \Delta_2 = \varnothing$ it follows immediately by Lemma 6.3.7(1) that $\text{pres}_{\Delta_1}(w) = u$ and $\text{pres}_{\Delta_2}(w) = v$.

($\supseteq$) Let $w \in (\Delta_1 \cup \Delta_2)^\infty$ be such that $\text{pres}_{\Delta_1}(w) = u$ and $\text{pres}_{\Delta_2}(w) = v$. We distinguish three cases.

First consider that $u \in \Delta_1^*$. Since $\text{pres}_{\Delta_1}(w) = u$ there exist an $n \geq 0$ and $a_1, a_2, \ldots, a_n \in \Delta_1$ such that $u = a_1 a_2 \cdots a_n$ and $w = \alpha_0 a_1 \alpha_1 a_2 \cdots a_n \alpha_n$, where $\alpha_0, \alpha_1, \ldots, \alpha_{n-1} \in \Delta_2^*$ and $\alpha_n \in \Delta_2^\infty$. Since $\text{pres}_{\Delta_2}(w) = v$ and $\Delta_1 \cap \Delta_2 = \varnothing$, we have $v = \alpha_0 \alpha_1 \cdots \alpha_n$. Now let $\alpha_n = \lim_{m\to\infty} \gamma_1 \gamma_2 \cdots \gamma_m$ with $\gamma_i \in \Delta_2^*$, for all $i \geq 1$. Hence $w = \alpha_0 a_1 \alpha_1 a_2 \cdots \alpha_{n-1} a_n \gamma_1 \lambda \gamma_2 \lambda \cdots$ with $u = a_1 a_2 \cdots a_n$ and $v = \alpha_1 \alpha_2 \cdots \alpha_{n-1} \gamma_1 \gamma_2 \cdots$ and thus, again by Lemma 6.3.7(1), $w \in u \,|||\, v$.

The case that $v \in \Delta_2^*$ is analogous.

Finally, consider that $u \in \Delta_1^\omega$ and $v \in \Delta_2^\omega$. Hence $w \in (\Delta_1 \cup \Delta_2)^\omega$. Let $w = c_1 c_2 \cdots = \lim_{n\to\infty} c_1 c_2 \cdots c_n$ with $c_i \in \Delta_1 \cup \Delta_2$, for all $i \geq 1$. By the definition of homomorphisms on infinite words, $\text{pres}_{\Delta_1}(w) = \lim_{n\to\infty} \text{pres}_{\Delta_1}(c_1 c_2 \cdots c_n) = u$ and $\text{pres}_{\Delta_2}(w) = \lim_{n\to\infty} \text{pres}_{\Delta_2}(c_1 c_2 \cdots c_n) = v$. Now denote $\text{pres}_{\Delta_1}(c_1 c_2 \cdots c_n)$ by u_n and $\text{pres}_{\Delta_2}(c_1 c_2 \cdots c_n)$ by v_n. From the first two cases it then follows that for all $n \geq 1$, $c_1 c_2 \cdots c_n \in u_n \,|||\, v_n$. Hence $\text{pref}(w) \subseteq \text{pref}(u) \,|||\, \text{pref}(v)$, which implies that $w \in u \,||\, v$ by Corollary 6.3.48. Since $\Delta_1 \cap \Delta_2 = \varnothing$ and u and v are both infinite words, w satisfies subcase (c) of case (4) of Definition 6.3.1 and thus $w \in u \,|||\, v$. □

This result implies that also the fair S-shuffles and the fair fS-shuffles can be described in terms of preserving homomorphisms, provided that there is no confusion about the non-synchronizing symbols.

Theorem 6.4.26. *Let Γ be an alphabet and let $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$ be such that $(\Delta_1 \setminus \Gamma) \cap (\Delta_2 \setminus \Gamma) = \varnothing$. Then*

$$u \,|||^\Gamma\, v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid pres_\Gamma(w) = pres_\Gamma(u) = pres_\Gamma(v),\ pres_{\Delta_1}(w) = u,\ pres_{\Delta_2}(w) = v\}.$$

Proof. ($\subseteq$) Let $w \in u \,|||^\Gamma\, v$. As by Lemma 6.4.11(1), $\text{pres}_\Gamma(w) = \text{pres}_\Gamma(u) = \text{pres}_\Gamma(v)$, we only have to prove that $\text{pres}_{\Delta_1}(w) = u$ and $\text{pres}_{\Delta_2}(w) = v$. According to Definition 6.4.1 we can distinguish two cases.

First consider that $w = w_1y_1w_2y_2\cdots y_nw_{n+1}$, where for some $n \geq 1$, $w_1, w_2, \ldots, w_n \in ((\Delta_1 \cup \Delta_2)\setminus\Gamma)^*$, $w_{n+1} \in ((\Delta_1 \cup \Delta_2)\setminus\Gamma)^\infty$, $y_1, y_2, \ldots, y_n \in \Gamma$, $u = u_1y_1u_2y_2\cdots y_nu_{n+1}$, with $u_1, u_2, \ldots, u_n \in (\Delta_1 \setminus \Gamma)^*$ and $u_{n+1} \in (\Delta_1 \setminus \Gamma)^\infty$, and $v = v_1y_1v_2y_2\cdots y_nv_{n+1}$, with $v_1, v_2, \ldots, v_n \in (\Delta_2 \setminus \Gamma)^*$ and $v_{n+1} \in (\Delta_2 \setminus \Gamma)^\infty$, are such that for all $i \in [n+1]$, $w_i \in u_i \mid\mid\mid v_i$. Since $(\Delta_1 \setminus \Gamma) \cap (\Delta_2 \setminus \Gamma) = \varnothing$, it follows from Lemma 6.4.25 that for all $i \in [n+1]$, $\mathrm{pres}_{\Delta_1}(w_i) = u_i$ and $\mathrm{pres}_{\Delta_2}(w_i) = v_i$. Hence $\mathrm{pres}_{\Delta_1}(w) = \mathrm{pres}_{\Delta_1}(w_1)\,\mathrm{pres}_{\Delta_1}(y_1)\,\mathrm{pres}_{\Delta_1}(w_2)\,\mathrm{pres}_{\Delta_1}(y_2)\,\cdots\,\mathrm{pres}_{\Delta_1}(y_n)\,\mathrm{pres}_{\Delta_1}(w_{n+1}) = u_1y_1u_2y_2\cdots y_nu_{n+1} = u$ and, analogously, $\mathrm{pres}_{\Delta_2}(w) = v_1y_1v_2y_2\cdots y_nv_{n+1} = v$.

Secondly, consider that $w = w_1y_1w_2y_2\cdots$, where $w_1, w_2, \ldots \in ((\Delta_1 \cup \Delta_2) \setminus \Gamma)^*$, $y_1, y_2, \ldots \in \Gamma$, $u = u_1y_1u_2y_2\cdots$, with $u_1, u_2, \ldots \in (\Delta_1 \setminus \Gamma)^*$, and $v = v_1y_1v_2y_2\cdots$, with $v_1, v_2, \ldots \in (\Delta_2 \setminus \Gamma)^*$, are such that for all $i \geq 1$, $w_i \in u_i \mid\mid\mid v_i$. Since $(\Delta_1 \setminus \Gamma) \cap (\Delta_2 \setminus \Gamma) = \varnothing$, Lemma 6.4.25 implies that for all $i \geq 1$, $\mathrm{pres}_{\Delta_1}(w_i) = u_i$ and $\mathrm{pres}_{\Delta_2}(w_i) = v_i$. Hence, by the definition of homomorphisms on infinite words, $\mathrm{pres}_{\Delta_1}(w) = u_1y_1u_2y_2\cdots = u$ and $\mathrm{pres}_{\Delta_2}(w) = v_1y_1v_2y_2\cdots = v$.

($\supseteq$) Let $w \in (\Delta_1 \cup \Delta_2)^\infty$ be such that $\mathrm{pres}_\Gamma(w) = \mathrm{pres}_\Gamma(u) = \mathrm{pres}_\Gamma(v)$, $\mathrm{pres}_{\Delta_1}(w) = u$, and $\mathrm{pres}_{\Delta_2}(w) = v$. Observe that $(\Delta_1 \setminus \Gamma) \cap (\Delta_2 \setminus \Gamma) = \varnothing$ implies that $\Delta_1 \cap \Delta_2 \subseteq \Gamma$. Hence, by Lemma 6.4.7(2), $u \;{}_{\Delta_1}\underline{\mid\mid\mid}^{\Gamma}_{\Delta_2}\; v = u \;{}_{\Delta_1}\underline{\mid\mid\mid}_{\Delta_2}\; v$. Moreover, since $\mathrm{pres}_\Gamma(u) = \mathrm{pres}_\Gamma(v)$, we have $w \in u \;{}_{\Delta_1}\underline{\mid\mid\mid}^{\Gamma}_{\Delta_2}\; v$ if and only if $w \in u \mid\mid\mid^\Gamma v$. Thus it suffices to prove that $w \in u \;{}_{\Delta_1}\underline{\mid\mid\mid}_{\Delta_2}\; v$. We distinguish two cases.

First consider that $\mathrm{pres}_{\Delta_1\cap\Delta_2}(w) \in (\Delta_1 \cup \Delta_2)^*$. Then there exists an $n \geq 1$ such that $w = w_1y_1w_2y_2\cdots y_nw_{n+1}$, where for all $i \in [n]$, $w_i \in ((\Delta_1\setminus\Delta_2)\cup(\Delta_2\setminus\Delta_1))^*$ and $y_i \in \Delta_1\cap\Delta_2$, and $w_{n+1} \in ((\Delta_1\setminus\Delta_2)\cup(\Delta_2\setminus\Delta_1))^\infty$. Moreover, $\mathrm{pres}_{\Delta_1}(w) = \mathrm{pres}_{\Delta_1}(w_1)y_1\mathrm{pres}_{\Delta_1}(w_2)y_2\cdots y_n\mathrm{pres}_{\Delta_1}(w_{n+1}) = u$ and $\mathrm{pres}_{\Delta_2}(w) = \mathrm{pres}_{\Delta_2}(w_1)y_1\mathrm{pres}_{\Delta_2}(w_2)y_2\cdots y_n\mathrm{pres}_{\Delta_2}(w_{n+1}) = v$. Hence $u = u_1y_1u_2y_2\cdots y_nu_{n+1}$, with $u_i = \mathrm{pres}_{\Delta_1\setminus\Delta_2}(w_i)$, for all $i \in [n+1]$, and $v = v_1y_1v_2y_2\cdots y_nv_{n+1}$, with $v_i = \mathrm{pres}_{\Delta_2\setminus\Delta_1}(w_i)$, for all $i \in [n+1]$. Since for all $i \in [n+1]$, $u_i \in (\Delta_1 \setminus \Delta_2)^\infty$, $v_i \in (\Delta_2 \setminus \Delta_1)^\infty$, and $(\Delta_1 \setminus \Delta_2) \cap (\Delta_2 \setminus \Delta_1) = \varnothing$, Lemma 6.4.25 implies that for all $i \in [n]$, $w_i \in u_i \mid\mid\mid v_i = u_i \mid\mid v_i$, and $w_{n+1} \in u_{n+1} \mid\mid\mid v_{n+1} \subseteq u_{n+1} \mid\mid v_{n+1}$. Definition 6.4.1(1) now implies that $w \in u \mid\mid\mid^{\Delta_1\cap\Delta_2} v$, which by Definition 6.4.3 means that $w \in u \;{}_{\Delta_1}\underline{\mid\mid\mid}_{\Delta_2}\; v$.

Next consider that $\mathrm{pres}_{\Delta_1\cap\Delta_2}(w) \in (\Delta_1 \cup \Delta_2)^\omega$. Then $w = w_1y_1w_2y_2\cdots$, where for all $i \geq 1$, $w_i \in ((\Delta_1 \setminus \Delta_2) \cup (\Delta_2 \setminus \Delta_1))^*$ and $y_i \in \Delta_1 \cap \Delta_2$. Moreover, $\mathrm{pres}_{\Delta_1}(w) = \mathrm{pres}_{\Delta_1}(w_1)y_1\mathrm{pres}_{\Delta_1}(w_2)y_2\cdots = u$ and $\mathrm{pres}_{\Delta_2}(w) = \mathrm{pres}_{\Delta_2}(w_1)y_1\mathrm{pres}_{\Delta_2}(w_2)y_2\cdots = v$. Hence $u = u_1y_1u_2y_2\cdots$, with $u_i = \mathrm{pres}_{\Delta_1\setminus\Delta_2}(w_i)$, for all $i \geq 1$, and $v = v_1y_1v_2y_2\cdots$, with $v_i = \mathrm{pres}_{\Delta_2\setminus\Delta_1}(w_i)$, for all $i \geq 1$. Since for all $i \geq 1$, $u_i \in (\Delta_1 \setminus \Delta_2)^*$, $v_i \in (\Delta_2 \setminus \Delta_1)^*$, and $(\Delta_1 \setminus \Delta_2) \cap (\Delta_2 \setminus \Delta_1) = \varnothing$, Lemma 6.4.25 implies that for all $i \geq 1$,

$w_i \in u_i \mid\mid\mid v_i = u_i \mid\mid v_i$. Definition 6.4.1(2) now implies that $w \in u \mid\mid\mid^{\Delta_1 \cap \Delta_2} v$, which by Definition 6.4.3 means that $w \in u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v$. □

Finally, we obtain from this result a characterization of fair fS-shuffling.

Corollary 6.4.27. *Let $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$. Then*

$$u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid pres_{\Delta_1}(w) = u, \; pres_{\Delta_2}(w) = v\}.$$

Proof. By Definition 6.4.3, $u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v = u \mid\mid\mid^{\Delta_1 \cap \Delta_2} v$ and $(\Delta_1 \setminus (\Delta_1 \cap \Delta_2)) \cap (\Delta_2 \setminus (\Delta_1 \cap \Delta_2)) = \varnothing$. Moreover, if $\mathrm{pres}_{\Delta_1}(w) = u$ and $\mathrm{pres}_{\Delta_2}(w) = v$, then $\mathrm{pres}_{\Delta_1 \cap \Delta_2}(w) = \mathrm{pres}_{\Delta_1}(\mathrm{pres}_{\Delta_2}(\mathrm{pres}_{\Delta_2}(w))) = \mathrm{pres}_{\Delta_1}(\mathrm{pres}_{\Delta_2}(v)) = \mathrm{pres}_{\Delta_1 \cap \Delta_2}(v)$. Similarly, $\mathrm{pres}_{\Delta_1 \cap \Delta_2}(w) = \mathrm{pres}_{\Delta_1 \cap \Delta_2}(u)$. Hence, by Theorem 6.4.26, $u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v = u \mid\mid\mid^{\Delta_1 \cap \Delta_2} v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \mathrm{pres}_{\Delta_1 \cap \Delta_2}(w) = \mathrm{pres}_{\Delta_1 \cap \Delta_2}(u) = \mathrm{pres}_{\Delta_1 \cap \Delta_2}(v), \; \mathrm{pres}_{\Delta_1}(w) = u, \; \mathrm{pres}_{\Delta_2}(w) = v\} = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \mathrm{pres}_{\Delta_1}(w) = u, \; \mathrm{pres}_{\Delta_2}(w) = v\}$. □

Now we can prove the associativity of fair fS-shuffling.

Theorem 6.4.28. *Let $u \in \Delta_1^\infty$, let $v \in \Delta_2^\infty$, and let $w \in \Delta_3^\infty$. Then*

$$\{u\} \; {}_{\Delta_1}\underline{|||}_{\Delta_2 \cup \Delta_3} \; (v \; {}_{\Delta_2}\underline{|||}_{\Delta_3} \; w) = (u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v) \; {}_{\Delta_1 \cup \Delta_2}\underline{|||}_{\Delta_3} \; \{w\}.$$

Proof. ($\subseteq$) Let $x \in \{u\} \; {}_{\Delta_1}\underline{|||}_{\Delta_2 \cup \Delta_3} \; (v \; {}_{\Delta_2}\underline{|||}_{\Delta_3} \; w)$ and let $y \in v \; {}_{\Delta_2}\underline{|||}_{\Delta_3} \; w$ be such that $x \in {}_{\Delta_1}\underline{|||}_{\Delta_2 \cup \Delta_3} \; y$. Hence, according to Corollary 6.4.27, $\mathrm{pres}_{\Delta_1}(w) = u$ and $\mathrm{pres}_{\Delta_2 \cup \Delta_3}(w) = y$. Moreover, $\mathrm{pres}_{\Delta_2}(y) = v$ and $\mathrm{pres}_{\Delta_3}(y) = w$. Now let $z = \mathrm{pres}_{\Delta_1 \cup \Delta_2}(x)$. By repeatedly applying Corollary 6.4.27 and by using the properties of preserving homomorphisms, we obtain that $\mathrm{pres}_{\Delta_3}(x) = \mathrm{pres}_{\Delta_3}(\mathrm{pres}_{\Delta_2 \cup \Delta_3}(x)) = \mathrm{pres}_{\Delta_3}(y) = w$, and thus $x \in z \; {}_{\Delta_1 \cup \Delta_2}\underline{|||}_{\Delta_3} \; w$. Furthermore, $\mathrm{pres}_{\Delta_1}(z) = \mathrm{pres}_{\Delta_1}(\mathrm{pres}_{\Delta_1 \cup \Delta_2}(x)) = \mathrm{pres}_{\Delta_1}(x) = u$ and $\mathrm{pres}_{\Delta_2}(z) = \mathrm{pres}_{\Delta_2}(\mathrm{pres}_{\Delta_1 \cup \Delta_2}(x)) = \mathrm{pres}_{\Delta_2}(x) = \mathrm{pres}_{\Delta_2}(\mathrm{pres}_{\Delta_2 \cup \Delta_3}(x)) = \mathrm{pres}_{\Delta_2}(y) = v$. Hence $z \in u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v$ and thus we have proven that $x \in (u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v) \; {}_{\Delta_1 \cup \Delta_2}\underline{|||}_{\Delta_3} \; \{w\}$.

($\supseteq$) By Corollary 6.4.19 and ($\subseteq$) we immediately obtain that $(u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v) \; {}_{\Delta_1 \cup \Delta_2}\underline{|||}_{\Delta_3} \; \{w\} = \{w\} \; {}_{\Delta_3}\underline{|||}_{\Delta_1 \cup \Delta_2} \; (u \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; v) = \{w\} \; {}_{\Delta_3}\underline{|||}_{\Delta_1 \cup \Delta_2} \; (v \; {}_{\Delta_2}\underline{|||}_{\Delta_1} \; u) \subseteq (w \; {}_{\Delta_3}\underline{|||}_{\Delta_2} \; v) \; {}_{\Delta_2 \cup \Delta_3}\underline{|||}_{\Delta_1} \; \{u\} = (v \; {}_{\Delta_2}\underline{|||}_{\Delta_3} \; w) \; {}_{\Delta_2 \cup \Delta_3}\underline{|||}_{\Delta_1} \; \{u\} = \{u\} \; {}_{\Delta_1}\underline{|||}_{\Delta_2 \cup \Delta_3} \; (v \; {}_{\Delta_2}\underline{|||}_{\Delta_3} \; w)$. □

This result can be lifted to languages.

Theorem 6.4.29. *Let $L_1 \subseteq \Delta_1^\infty$, let $L_2 \subseteq \Delta_2^\infty$, and let $L_3 \subseteq \Delta_3^\infty$. Then*

$$L_1 \; {}_{\Delta_1}\underline{|||}_{\Delta_2 \cup \Delta_3} \; (L_2 \; {}_{\Delta_2}\underline{|||}_{\Delta_3} \; L_3) = (L_1 \; {}_{\Delta_1}\underline{|||}_{\Delta_2} \; L_2) \; {}_{\Delta_1 \cup \Delta_2}\underline{|||}_{\Delta_3} \; L_3.$$

Proof. Analogous to the proof of Theorem 6.3.32(2). □

The fact that the rS-shuffle is defined in terms of the S-shuffle, together with the associativity of fair fS-shuffling, now allows us to prove that also fair rS-shuffling is associative.

Theorem 6.4.30. *Let $u \in \Delta_1^\infty$, let $v \in \Delta_2^\infty$, let $w \in \Delta_3^\infty$, and let Γ be an alphabet. Then*

$$\{u\} \ {}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2\cup\Delta_3} \ (v \ {}_{\Delta_2}\underline{|||}^{\Gamma}_{\Delta_3} \ w) = (u \ {}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2} \ v) \ {}_{\Delta_1\cup\Delta_2}\underline{|||}^{\Gamma}_{\Delta_3} \ \{w\}.$$

Proof. Since fair rS-shuffles are defined in terms of fair S-shuffles, it is clear that it suffices to prove that $\{u\} \ |||^{\Gamma\cap(\Delta_1\cap(\Delta_2\cup\Delta_3))} \ (v \ |||^{\Gamma\cap\Delta_2\cap\Delta_3} \ w) = (u \ |||^{\Gamma\cap\Delta_1\cap\Delta_2} \ v) \ |||^{\Gamma\cap(\Delta_1\cup\Delta_2)\cap\Delta_3} \ \{w\}$.

Let $\Delta^\ell = \{a_\ell \mid a \in \Delta\}$, for all $\ell \in \{[123],[12],[13],[23],[1],[2],[3]\}$. Consequently, we consider the homomorphism $\varphi : (\Delta_1 \cup \Delta_2 \cup \Delta_3)^\infty \to (\Delta^{[123]} \cup \Delta^{[12]} \cup \Delta^{[13]} \cup \Delta^{[23]} \cup \Delta^{[1]} \cup \Delta^{[2]} \cup \Delta^{[3]})^\infty$, which we use to label each letter from u, v, and w in a specific way: those letters that appear in Γ and in at least two of the alphabets Δ_1, Δ_2, or Δ_3, are labeled by subscripts indicating all of the alphabets from Δ_1, Δ_2, or Δ_3 that they appear in, while all other letters are labeled by subscripts indicating the unique alphabet from Δ_1, Δ_2, or Δ_3 that they appear in. Formally, φ is defined as follows.

$$\varphi(a) = \begin{cases} a_{[123]} & \text{if } a \in \Gamma \cap \Delta_1 \cap \Delta_2 \cap \Delta_3, \\ a_{[12]} & \text{if } a \in (\Gamma \cap \Delta_1 \cap \Delta_2) \setminus \Delta_3, \\ a_{[13]} & \text{if } a \in (\Gamma \cap \Delta_1 \cap \Delta_3) \setminus \Delta_2, \\ a_{[23]} & \text{if } a \in (\Gamma \cap \Delta_2 \cap \Delta_3) \setminus \Delta_1, \\ a_{[1]} & \text{if } a \in (\Delta_1 \setminus \Gamma) \cup ((\Gamma \cap \Delta_1) \setminus (\Delta_2 \cup \Delta_3)), \\ a_{[2]} & \text{if } a \in (\Delta_2 \setminus \Gamma) \cup ((\Gamma \cap \Delta_2) \setminus (\Delta_1 \cup \Delta_3)), \text{ and} \\ a_{[3]} & \text{if } a \in (\Delta_3 \setminus \Gamma) \cup ((\Gamma \cap \Delta_3) \setminus (\Delta_1 \cup \Delta_2)). \end{cases}$$

Now let $\hat{\Delta}_1 = \Delta^{[123]} \cup \Delta^{[12]} \cup \Delta^{[13]} \cup \Delta^{[1]}$, let $\hat{\Delta}_2 = \Delta^{[123]} \cup \Delta^{[12]} \cup \Delta^{[23]} \cup \Delta^{[2]}$, and let $\hat{\Delta}_3 = \Delta^{[123]} \cup \Delta^{[13]} \cup \Delta^{[23]} \cup \Delta^{[3]}$. Hence $\varphi(u) \in \hat{\Delta}_1^\infty$, $\varphi(v) \in \hat{\Delta}_2^\infty$, and $\varphi(w) \in \hat{\Delta}_3^\infty$. From the way we have labeled the alphabets we obtain that $a \in \Gamma \cap (\Delta_1 \cap (\Delta_2 \cup \Delta_3))$ if and only if $a \in (\Gamma \cap \Delta_1 \cap \Delta_2 \cap \Delta_3) \cup ((\Gamma \cap \Delta_1 \cap \Delta_2) \setminus \Delta_3) \cup ((\Gamma \cap \Delta_1 \cap \Delta_3) \setminus \Delta_2)$ if and only if $\varphi(a) \in \Delta^{[123]} \cup \Delta^{[12]} \cup \Delta^{[13]}$ if and only if $\varphi(a) \in \hat{\Delta}_1 \cap (\hat{\Delta}_2 \cup \hat{\Delta}_3)$ and similarly for the other (potential) synchronization symbols. Since φ is injective, it thus follows that $\{u\} \ |||^{\Gamma\cap(\Delta_1\cap(\Delta_2\cup\Delta_3))} \ (v \ |||^{\Gamma\cap\Delta_2\cap\Delta_3} \ w) = \varphi^{-1}(\varphi(u) \ |||^{\hat{\Delta}_1\cap(\hat{\Delta}_2\cup\hat{\Delta}_3)} \ (\varphi(v) \ |||^{\hat{\Delta}_2\cap\hat{\Delta}_3} \ \varphi(w)))$, which by the associativity of Theorem 6.4.28 is equal to $\varphi^{-1}((\varphi(u) \ |||^{\hat{\Delta}_1\cap\hat{\Delta}_2} \ \varphi(v)) \ |||^{(\hat{\Delta}_1\cup\hat{\Delta}_2)\cap\hat{\Delta}_3} \ \varphi(w))$ and this, once again by the labeling of the alphabets, equals $(u \ |||^{\Gamma\cap\Delta_1\cap\Delta_2} \ v) \ |||^{\Gamma\cap(\Delta_1\cup\Delta_2)\cap\Delta_3} \ \{w\}$. □

The associativity of fair rS-shuffling can also be proven for languages.

Theorem 6.4.31. *Let $L_1 \subseteq \Delta_1^\infty$, let $L_2 \subseteq \Delta_2^\infty$, let $L_3 \subseteq \Delta_3^\infty$, and let Γ be an alphabet. Then*

$$L_1 \;{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2\cup\Delta_3}\; (L_2 \;{}_{\Delta_2}\underline{|||}^{\Gamma}_{\Delta_3}\; L_3) = (L_1 \;{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2}\; L_2) \;{}_{\Delta_1\cup\Delta_2}\underline{|||}^{\Gamma}_{\Delta_3}\; L_3.$$

Proof. Analogous to the proof of Theorem 6.3.32(2). □

As was the case for the associativity of S-shuffling, also the statements of the preceding two theorems do not hold when Γ may vary. Given $w_i \in \Delta_i^*$, with $i \in [3]$, and two distinct alphabets Γ and Γ', e.g., in general $(w_1 \;{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2}\; w_2) \;{}_{\Delta_1\cup\Delta_2}\underline{|||}^{\Gamma'}_{\Delta_3}\; w_3$ does not equal $w_1 \;{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2\cup\Delta_3}\; (w_2 \;{}_{\Delta_2}\underline{|||}^{\Gamma'}_{\Delta_3}\; w_3)$. This is shown in the following example.

Example 6.4.32. Let $\Delta_1 = \{a\}$, let $\Delta_2 = \{b\}$, and let $\Delta_3 = \{a, b\}$. Then clearly $(a \;{}_{\Delta_1}\underline{|||}^{\{a\}}_{\Delta_2}\; b) \;{}_{\Delta_1\cup\Delta_2}\underline{|||}^{\{b\}}_{\Delta_3}\; \{ab\} = (a \;{}_{\{a\}}\underline{|||}^{\{a\}}_{\{b\}}\; b) \;{}_{\{a,b\}}\underline{|||}^{\{b\}}_{\{a,b\}}\; \{ab\} = \{ab, ba\} \;{}_{\{a,b\}}\underline{||}^{\{b\}}_{\{a,b\}}\; \{ab\} = \{aab, aba\}$, while $\{a\} \;{}_{\Delta_1}\underline{|||}^{\{a\}}_{\Delta_2\cup\Delta_3}\; (b \;{}_{\Delta_2}\underline{|||}^{\{b\}}_{\Delta_3}\; ab) = \{a\} \;{}_{\{a\}}\underline{|||}^{\{a\}}_{\{a,b\}}\; (b \;{}_{\{b\}}\underline{|||}^{\{b\}}_{\{a,b\}}\; ab) = a \;{}_{\{a\}}\underline{|||}^{\{a\}}_{\{a,b\}}\; ab = \{ab\}$. □

In case of "unfair" fS-shuffling (and thus also in case of "unfair" rS-shuffling) the associativity at the level of words which we have established for the fair case (and for S-shuffling) does not hold. As the following example shows, unfair fS-shuffling with an infinite word may lead to the abortion of its finite partner and thus destroy the associativity.

Example 6.4.33. Let $\Delta_1 = \{a, b\}$, let $\Delta_2 = \{b\}$, and let $\Delta_3 = \{c\}$. Then clearly $a \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; b = \varnothing$ and thus $(a \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; b) \;{}_{\Delta_1\cup\Delta_2}\underline{||}_{\Delta_3}\; \{c^\omega\} = \varnothing$. However, $\{a\} \;{}_{\Delta_1}\underline{||}_{\Delta_2\cup\Delta_3}\; (b \;{}_{\Delta_2}\underline{||}_{\Delta_3}\; c^\omega) = \{a\} \;{}_{\{a,b\}}\underline{||}_{\{b,c\}}\; (b \;{}_{\{b\}}\underline{||}_{\{c\}}\; c^\omega) = \{a\} \;{}_{\{a,b\}}\underline{||}_{\{b,c\}}\; (\{c^n bc^\omega \mid n \geq 0\} \cup \{c^\omega\}) = \{c^n ac^\omega \mid n \geq 0\} \cup \{c^\omega\}$. □

At first sight, adding λ to represent possible abortion appears to be a solution. For this example, adding λ indeed solves the problem, as is shown in the following example.

Example 6.4.34. (Example 6.4.33 continued) We show how adding λ to a, b, and c^ω may solve the problem, viz. $(\{\lambda, a\} \;{}_{\Delta_1}\underline{||}_{\Delta_2}\; \{\lambda, b\}) \;{}_{\Delta_1\cup\Delta_2}\underline{||}_{\Delta_3}\; \{\lambda, c^\omega\} = (\{\lambda, a\} \;{}_{\{a,b\}}\underline{||}_{\{b\}}\; \{\lambda, b\}) \;{}_{\{a,b\}}\underline{||}_{\{c\}}\; \{\lambda, c^\omega\} = \{\lambda, a\} \;{}_{\{a,b\}}\underline{||}_{\{c\}}\; \{\lambda, c^\omega\} = \{\lambda, a, c^\omega\} \cup \{c^n ac^\omega \mid n \geq 0\} = \{\lambda, a\} \;{}_{\{a,b\}}\underline{||}_{\{b,c\}}\; (\{\lambda, b, c^\omega\} \cup \{c^n bc^\omega \mid n \geq 0\}) = \{\lambda, a\} \;{}_{\{a,b\}}\underline{||}_{\{b,c\}}\; (\{\lambda, b\} \;{}_{\{b\}}\underline{||}_{\{c\}}\; \{\lambda, c^\omega\}) = \{\lambda, a\} \;{}_{\Delta_1}\underline{||}_{\Delta_2\cup\Delta_3}\; (\{\lambda, b\} \;{}_{\Delta_2}\underline{||}_{\Delta_3}\; \{\lambda, c^\omega\})$. □

In this example the aborted word consists of one symbol only and thus has λ as its only proper prefix, whereas in general infinite words when unfairly shuffled may still tolerate arbitrary prefixes, in which case adding λ is not a solution. This is shown in the following example.

Example 6.4.35. Note $(\{\lambda, a^\omega\} \underline{\|}_{\{a\}} \{\lambda, b^2\}) \underline{\|}_{\{a,b\}} \{\lambda, b\} = (\{\lambda, b^2, a^\omega\} \cup \{a^m b a^n b a^\omega, a^n b a^\omega \mid m, n \geq 0\}) \underline{\|}_{\{a,b\}} \{\lambda, b\} = \{\lambda, a^\omega\} \cup \{a^n b a^\omega \mid n \geq 0\}$. However, $\{\lambda, a^\omega\} \underline{\|}_{\{a\}} (\{\lambda, b^2\} \underline{\|}_{\{b\}} \{\lambda, b\}) = \{\lambda, a^\omega\} \underline{\|}_{\{a\}} \{\lambda\} = \{\lambda, a^\omega\}$. □

Hence we propose to add not just λ, but all prefixes of the words involved. In the following example we show that this solves the problems encountered in the previous examples.

Example 6.4.36. (Examples 6.4.33 and 6.4.35 continued) The problem we met in Example 6.4.33 is indeed solved in this way, viz. $(\{\lambda, a\} \underline{\|}_{\{a,b\}} \{\lambda, b\}) \underline{\|}_{\{a,b\}} (\{c^n \mid n \geq 0\} \cup \{c^\omega\}) = (\{\lambda, a\} \underline{\|}_{\{a,b\}} \{c\}) (\{c^n \mid n \geq 0\} \cup \{c^\omega\}) = \{c^n a c^\omega, c^m a c^n, c^n \mid m, n \geq 0\} \cup \{c^\omega\} = \{\lambda, a\} \underline{\|}_{\{a,b\}} \{b,c\}) (\{c^n b c^\omega, c^m b c^n, c^n \mid m, n \geq 0\} \cup \{c^\omega\}) = \{\lambda, a\} \underline{\|}_{\{a,b\}} \{b,c\}) (\{\lambda, b\} \underline{\|}_{\{b\}} \{c\}) (\{c^n \mid n \geq 0\} \cup \{c^\omega\}))$.

Moreover, also the problem we met in Example 6.4.35 is indeed solved in this way, viz. $((\{a^n \mid n \geq 0\} \cup \{a^\omega\}) \underline{\|}_{\{a\}} \{\lambda, b, b^2\}) \underline{\|}_{\{a,b\}} \{\lambda, b\} = (\{a^m b a^n b a^\omega, a^m b a^n b a^p, a^n b a^\omega, a^m b a^n, a^n \mid m, n, p \geq 0\} \cup \{a^\omega\}) \underline{\|}_{\{a,b\}} \underline{\|}_{\{b\}} \{\lambda, b\} = \{a^n b a^\omega, a^m b a^n, a^n \mid n \geq 0\} \cup \{a^\omega\} = (\{a^n \mid n \geq 0\} \cup \{a^\omega\}) \underline{\|}_{\{a\}} \underline{\|}_{\{b\}} \{\lambda, b\} = (\{a^n \mid n \geq 0\} \cup \{a^\omega\}) \underline{\|}_{\{a\}} \underline{\|}_{\{b\}} (\{\lambda, b, b^2\} \underline{\|}_{\{b\}} \underline{\|}_{\{b\}} \{\lambda, b\})$. □

This provides us with enough motivation to set out and prove associativity of (general, unfair) fS-shuffling and rS-shuffling at the level of prefix-closed languages. It is relevant to recall at this point that the behavior of a team automaton and that of its constituting component automata are prefix closed. Hence we can still apply this higher-level notion of associativity to behavior of team automata.

First we express (general) S-shuffles in terms of fair S-shuffles and prefixes (cf. Lemma 6.3.4).

Lemma 6.4.37. *Let Γ be an alphabet. Then*

(1) if $u \in \Delta_1^*$ and $v \in \Delta_2^*$, then $u \,||^\Gamma\, v = u \,|||^\Gamma\, v$,

(2) if $u \in \Delta_1^*$ and $v \in \Delta_2^\omega$, then $u \,||^\Gamma v = \bigcup_{u' \in pref(u),\, pres_\Gamma(u') = pres_\Gamma(u)} (u' \,|||^\Gamma v)$, and

(3) if $u \in \Delta_1^\omega$ and $v \in \Delta_2^\omega$, then $u \,||^\Gamma v = \bigcup_{u' \in pref(u),\, pres_\Gamma(u') = pres_\Gamma(u)} (u' \,|||^\Gamma v) \cup \bigcup_{v' \in pref(v),\ pres_\Gamma(v') = pres_\Gamma(v)} (u \,|||^\Gamma\, v') \cup u \,|||^\Gamma\, v$.

Proof. (1) Trivial.

(2) Let $u \in \Delta_1^*$ and let $v \in \Delta_2^\omega$.

($\subseteq$) Let $w \in u \,||^\Gamma\, v$. By Definition 6.4.1, there exists an $n \geq 1$ such that $w \in (u_1 \,||\, v_1)x_1(u_2 \,||\, v_2)x_2 \cdots x_{n-1}(u_n \,||\, v_n)$, where $u_1, u_2, \ldots, u_n \in (\Delta_1 \setminus \Gamma)^*$, $v_1, v_2, \ldots, v_{n-1} \in (\Delta_2 \setminus \Gamma)^*$, $v_n \in (\Delta_2 \setminus \Gamma)^\omega$, $u = u_1x_1u_2x_2 \cdots x_{n-1}u_n$, and $v = v_1x_1v_2x_2 \cdots x_{n-1}v_n$. Then, according to Lemma 6.3.4(2), $u_n \,||\, v_n = \bigcup_{u' \in \mathrm{pref}(u_n)}(u' \,|||\, v)$ and hence we obtain $w \in (u_1 \,||\, v_1)x_1(u_2 \,||\, v_2)x_2 \cdots x_{n-1}(u_n \,||\, v_n) = \bigcup_{u' \in \mathrm{pref}(u_n)}((u_1 \,||\, v_1)x_1(u_2 \,||\, v_2)x_2 \cdots x_{n-1}(u' \,||\, v_n)) = \bigcup_{u' \in \mathrm{pref}(u_n)}(u_1x_1u_2x_2 \cdots u_{n-1}x_{n-1}u' \,|||^\Gamma\, v_1x_1v_2x_2 \cdots v_{n-1}x_{n-1}v_n) = \bigcup_{\bar{u} \in \mathrm{pref}(u),\ \mathrm{pres}_\Gamma(\bar{u}) = \mathrm{pres}_\Gamma(u)}(\bar{u} \,|||^\Gamma\, v)$.

($\supseteq$) This follows immediately from Definitions 6.3.1 and 6.4.1.

(3) Analogous to (2) but now using Lemma 6.3.4(3). □

As a consequence we obtain a characterization of fS-shuffling in terms of prefixes and preserving homomorphisms.

Corollary 6.4.38. *(1) If $u \in \Delta_1^*$ and $v \in \Delta_2^*$, then $u \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, v = \{w \in (\Delta_1 \cup \Delta_2)^* \mid pres_{\Delta_1}(w) = u,\ pres_{\Delta_2}(w) = v\}$,*

(2) if $u \in \Delta_1^*$ and $v \in \Delta_2^\omega$, then $u \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \exists\, u' \in pref(u) :\ pres_{\Delta_2}(u') = pres_{\Delta_2}(u),\ pres_{\Delta_1}(w) = u',\ pres_{\Delta_2}(w) = v\}$, and

(3) if $u \in \Delta_1^\omega$ and $v \in \Delta_2^\omega$, then $u \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, v = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid pres_{\Delta_1}(w) = u,\ pres_{\Delta_2}(w) = v\} \cup \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \exists\, u' \in pref(u) :\ pres_{\Delta_2}(u') = pres_{\Delta_2}(u),\ pres_{\Delta_1}(w) = u',\ pres_{\Delta_2}(w) = v\} \cup \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \exists\, v' \in pref(v) :\ pres_{\Delta_1}(v') = pres_{\Delta_1}(v),\ pres_{\Delta_1}(w) = u,\ pres_{\Delta_2}(w) = v'\}$.

Proof. By Definition 6.4.3, $u \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, v = u \,||^{\Delta_1 \cap \Delta_2}\, v$ and $u \,{}_{\Delta_1}\underline{|||}_{\Delta_2}\, v = u \,|||^{\Delta_1 \cap \Delta_2}\, v$ whenever $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$. Moreover, for $x \in \Delta_1^\infty$, $\mathrm{pres}_{\Delta_1 \cap \Delta_2}(x) = \mathrm{pres}_{\Delta_2}(x)$ and, for $x \in \Delta_2^\infty$, $\mathrm{pres}_{\Delta_1 \cap \Delta_2}(x) = \mathrm{pres}_{\Delta_1}(x)$. The statements now follow by combining Corollary 6.4.27 and Lemma 6.4.37, with $\Gamma = \Delta_1 \cap \Delta_2$. □

With this corollary we can now prove a result similar to Corollary 6.4.27, which was used to prove the associativity of fair fS-shuffling. In this case, however, we (have to) deal with words together with their prefixes.

Lemma 6.4.39. *If $u \in \Delta_1^\infty$ and $v \in \Delta_2^\infty$, then $(\{u\} \cup pref(u)) \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, (\{v\} \cup pref(v)) = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid pres_{\Delta_1}(w) \in \{u\} \cup pref(u),\ pres_{\Delta_2}(w) \in \{v\} \cup pref(v)\}$.*

Proof. Let $u \in \Delta_1^\infty$ and let $v \in \Delta_2^\infty$. We distinguish three cases.

(1) If $u \in \Delta_1^*$ and $v \in \Delta_2^*$, then by Corollary 6.4.38(1), $(\{u\} \cup \mathrm{pref}(u)) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; (\{v\} \cup \mathrm{pref}(v)) = \{w \in (\Delta_1 \cup \Delta_2)^* \mid \mathrm{pres}_{\Delta_1}(w) \in \{u\} \cup \mathrm{pref}(u), \mathrm{pres}_{\Delta_2}(w) \in \{v\} \cup \mathrm{pref}(v)\}$.

(2) If $u \in \Delta_1^*$ and $v \in \Delta_2^\omega$, then the fact that $u \in \mathrm{pref}(u)$ implies that $(\{u\} \cup \mathrm{pref}(u)) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; (\{v\} \cup \mathrm{pref}(v)) = \mathrm{pref}(u) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; (\{v\} \cup \mathrm{pref}(v)) = (\mathrm{pref}(u) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; \mathrm{pref}(v)) \cup (\mathrm{pref}(u) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; \{v\})$. By Corollary 6.4.38(2), $\mathrm{pref}(u) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; \{v\} = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \exists\, u' \in \mathrm{pref}(u),\, u'' \in \mathrm{pref}(u') : \mathrm{pres}_{\Delta_2}(u') = \mathrm{pres}_{\Delta_2}(u''),\ \mathrm{pres}_{\Delta_1}(w) = u'',\ \mathrm{pres}_{\Delta_2}(w) = v\} = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \exists\, u'' \in \mathrm{pref}(u) :\ \mathrm{pres}_{\Delta_1}(w) = u'',\ \mathrm{pres}_{\Delta_2}(w) = v\}$. Combining this with (1), we obtain $\mathrm{pref}(u) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; (\{v\} \cup \mathrm{pref}(v)) = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \mathrm{pres}_{\Delta_1}(w) \in \mathrm{pref}(u),\ \mathrm{pres}_{\Delta_2}(w) \in \{v\} \cup \mathrm{pref}(v)\}$.

(3) If $u \in \Delta_1^\omega$ and $v \in \Delta_2^\omega$, then $(\{u\} \cup \mathrm{pref}(u)) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; (\{v\} \cup \mathrm{pref}(v)) = L_1 \cup L_2 \cup L_3$, with L_1, L_2, and L_3 as follows.
$L_1 = \mathrm{pref}(u) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; (\{v\} \cup \mathrm{pref}(v)) = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \mathrm{pres}_{\Delta_1}(w) \in \mathrm{pref}(u),\ \mathrm{pres}_{\Delta_2}(w) \in \{v\} \cup \mathrm{pref}(v)\}$ by (2).
$L_2 = (\{u\} \cup \mathrm{pref}(u)) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; \mathrm{pref}(v) = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \mathrm{pres}_{\Delta_1}(w) \in \{u\} \cup \mathrm{pref}(u),\ \mathrm{pres}_{\Delta_2}(w) \in \mathrm{pref}(v)\}$ by (2) and the commutativity of fS-shuffling.
$L_3 = u \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; v = \{w \in (\Delta_1 \cup \Delta_2)^\omega \mid \mathrm{pres}_{\Delta_1}(w) = u,\ \mathrm{pres}_{\Delta_2}(w) = v\} \cup L_1' \cup L_2'$, with $L_1' \subseteq L_1$ and $L_2' \subseteq L_2$, by Corollary 6.4.38(3).

Consequently, $(\{u\} \cup \mathrm{pref}(u)) \;{}_{\Delta_1}\underline{\|}_{\Delta_2}\; (\{v\} \cup \mathrm{pref}(v)) = \{w \in (\Delta_1 \cup \Delta_2)^\infty \mid \mathrm{pres}_{\Delta_1}(w) \in \{u\} \cup \mathrm{pref}(u),\ \mathrm{pres}_{\Delta_2}(w) \in \{v\} \cup \mathrm{pref}(v)\}$. □

We have thus found a characterization of fS-shuffling that is insensitive to the order of application.

Theorem 6.4.40. *Let $u_i \in \Delta_i^\infty$, for all $i \in [3]$. Then*

$$(\{u_1\} \cup \mathit{pref}(u_1)) \;{}_{\Delta_1}\underline{\|}_{\Delta_2 \cup \Delta_3}\; ((\{u_2\} \cup \mathit{pref}(u_2)) \;{}_{\Delta_2}\underline{\|}_{\Delta_3}\; (\{u_3\} \cup \mathit{pref}(u_3))) = \{w \in (\Delta_1 \cup \Delta_2 \cup \Delta_3)^\infty \mid \forall\, i \in [3] :\ \mathit{pres}_{\Delta_i}(w) \in \{u_i\} \cup \mathit{pref}(u_i)\}.$$

Proof. ($\subseteq$) Let $w \in (\{u_1\} \cup \mathrm{pref}(u_1)) \;{}_{\Delta_1}\underline{\|}_{\Delta_2 \cup \Delta_3}\; ((\{u_2\} \cup \mathrm{pref}(u_2)) \;{}_{\Delta_2}\underline{\|}_{\Delta_3}\; (\{u_3\} \cup \mathrm{pref}(u_3)))$. By Lemma 6.4.39, $\mathrm{pres}_{\Delta_1}(w) \in \{u_1\} \cup \mathrm{pref}(u_1)$ and there exists a $y \in (\{u_2\} \cup \mathrm{pref}(u_2)) \;{}_{\Delta_2}\underline{\|}_{\Delta_3}\; (\{u_3\} \cup \mathrm{pref}(u_3))$ such that $\mathrm{pres}_{\Delta_2 \cup \Delta_3}(w) = y$. Consequently, $\mathrm{pres}_{\Delta_2}(w) = \mathrm{pres}_{\Delta_2}(\mathrm{pres}_{\Delta_2 \cup \Delta_3}(w)) = \mathrm{pres}_{\Delta_2}(y)$, which by Lemma 6.4.39 is included in $\{u_2\} \cup \mathrm{pref}(u_2)$, and $\mathrm{pres}_{\Delta_3}(w) = \mathrm{pres}_{\Delta_3}(\mathrm{pres}_{\Delta_2 \cup \Delta_3}(w)) = \mathrm{pres}_{\Delta_3}(y)$, which by Lemma 6.4.39 is included in $\{u_3\} \cup \mathrm{pref}(u_3)$.

($\supseteq$) Let $w \in (\Delta_1 \cup \Delta_2 \cup \Delta_3)^\infty$ be such that $\mathrm{pres}_{\Delta_i}(w) \in \{u_i\} \cup \mathrm{pref}(u_i)$, for all $i \in [3]$. Now let $z = \mathrm{pres}_{\Delta_2 \cup \Delta_3}(w)$. Hence $\mathrm{pres}_{\Delta_2}(z) = \mathrm{pres}_{\Delta_2}(w)$ and $\mathrm{pres}_{\Delta_3}(z) = \mathrm{pres}_{\Delta_3}(w)$. By Corollary 6.4.27, $z \in \mathrm{pres}_{\Delta_2}(w) \;{}_{\Delta_2}\underline{|||}_{\Delta_3}\; \mathrm{pres}_{\Delta_3}(w)$

and $w \in \text{pres}_{\Delta_1}(w) \ {}_{\Delta_1}|||_{\Delta_2\cup\Delta_3} \ z \subseteq \text{pres}_{\Delta_1}(w) \ {}_{\Delta_1}|||_{\Delta_2\cup\Delta_3} \ (\text{pres}_{\Delta_2}(w) \ {}_{\Delta_2}|||_{\Delta_3} \ \text{pres}_{\Delta_3}(w)) \subseteq (\{u_1\}\cup\text{pref}(u_1)) \ {}_{\Delta_1}|||_{\Delta_2\cup\Delta_3} \ ((\{u_2\}\cup\text{pref}(u_2)) \ {}_{\Delta_2}|||_{\Delta_3} \ (\{u_3\}\cup \text{pref}(u_3))) \subseteq (\{u_1\} \cup \text{pref}(u_1)) \ {}_{\Delta_1}||_{\Delta_2\cup\Delta_3} \ ((\{u_2\} \cup \text{pref}(u_2)) \ {}_{\Delta_2}||_{\Delta_3} \ (\{u_3\} \cup \text{pref}(u_3)))$. □

It is worth noticing that the proof of this theorem shows how unfair fS-shuffling can be translated into fair fS-shuffling by including prefixes. The associativity of fS-shuffling of prefix-closed languages now follows immediately.

Theorem 6.4.41. *Let $u \in \Delta_1^\infty$, $v \in \Delta_2^\infty$, and $w \in \Delta_3^\infty$. Then*

$$(\{u\} \cup \mathit{pref}(u)) \ {}_{\Delta_1}||_{\Delta_2\cup\Delta_3} \ ((\{v\} \cup \mathit{pref}(v)) \ {}_{\Delta_2}||_{\Delta_3} \ (\{w\} \cup \mathit{pref}(w))) = ((\{u\} \cup \mathit{pref}(u)) \ {}_{\Delta_1}||_{\Delta_2} \ (\{v\} \cup \mathit{pref}(v))) \ {}_{\Delta_1\cup\Delta_2}||_{\Delta_3} \ (\{w\} \cup \mathit{pref}(w)).$$

Proof. This follows directly from Theorem 6.4.40 and the commutativity of fS-shuffling. □

Theorem 6.4.42. *Let $L_i \subseteq \Delta_i^\infty$, for all $i \in [3]$, be prefix closed. Then*

$$L_1 \ {}_{\Delta_1}||_{\Delta_2\cup\Delta_3} \ (L_2 \ {}_{\Delta_2}||_{\Delta_3} \ L_3) = (L_1 \ {}_{\Delta_1}||_{\Delta_2} \ L_2) \ {}_{\Delta_1\cup\Delta_2}||_{\Delta_3} \ L_3.$$

Proof. ($\subseteq$) Let $w \in L_1 \ {}_{\Delta_1}||_{\Delta_2\cup\Delta_3} \ (L_2 \ {}_{\Delta_2}||_{\Delta_3} \ L_3)$. Then by definition there exist words $u_1 \in L_1$, $u_2 \in L_2$, and $u_3 \in L_3$ such that $w \in (\{u_1\} \cup \text{pref}(u_1))_{\Delta_1}||_{\Delta_2\cup\Delta_3}((\{u_2\}\cup\text{pref}(u_2))_{\Delta_2}||_{\Delta_3}(\{u_3\}\cup\text{pref}(u_3)))$. Consequently, by Theorem 6.4.41, $w \in ((\{u_1\}\cup\text{pref}(u_1)) \ {}_{\Delta_1}||_{\Delta_2} \ (\{u_2\}\cup\text{pref}(u_2))) \ {}_{\Delta_1\cup\Delta_2}||_{\Delta_3} \ (\{u_3\}\cup\text{pref}(u_3)) \subseteq (L_1 \ {}_{\Delta_1}||_{\Delta_2} \ L_2) \ {}_{\Delta_1\cup\Delta_2}||_{\Delta_3} \ L_3$ by the fact that L_1, L_2, and L_3 are prefix closed.

($\supseteq$) This follows from (1) and the commutativity of fS-shuffling. □

As before in the case of fair rS-shuffling, the fact that the rS-shuffle is defined in terms of the S-shuffle, together with the associativity of fS-shuffling, allows us to conclude that also rS-shuffling of prefix-closed languages is associative.

Theorem 6.4.43. *Let Γ be an alphabet and let $u_i \in \Delta_i^\infty$, for all $i \in [3]$. Then*

$$(\{u_1\} \cup \mathit{pref}(u_1))_{\Delta_1}||^{\Gamma}_{\Delta_2\cup\Delta_3}((\{u_2\} \cup \mathit{pref}(u_2))_{\Delta_2}||^{\Gamma}_{\Delta_3}(\{u_3\} \cup \mathit{pref}(u_3))) = (((\{u_1\} \cup \mathit{pref}(u_1)) \ {}_{\Delta_1}||^{\Gamma}_{\Delta_2} \ (\{u_2\} \cup \mathit{pref}(u_2))) \ {}_{\Delta_1\cup\Delta_2}||^{\Gamma}_{\Delta_3} \ (\{u_3\} \cup \mathit{pref}(u_3)).$$

Proof. Similar to the proof of Theorem 6.4.30 by renaming the symbols, but now using Theorem 6.4.41. □

Theorem 6.4.44. *Let Γ be an alphabet and let $L_i \subseteq \Delta_i^\infty$, for all $i \in [3]$, be prefix closed. Then*

$$L_1 \ {}_{\Delta_1}||^{\Gamma}_{\Delta_2\cup\Delta_3} \ (L_2 \ {}_{\Delta_2}||^{\Gamma}_{\Delta_3} \ L_3) = (L_1 \ {}_{\Delta_1}||^{\Gamma}_{\Delta_2} \ L_2) \ {}_{\Delta_1\cup\Delta_2}||^{\Gamma}_{\Delta_3} \ L_3.$$

Proof. Analogous to the proof of Theorem 6.4.42. □

6.4.4 Conclusion

The commutativity and associativity of the S-shuffle (cf. Theorems 6.4.17 and 6.4.21) directly imply that the order in which we (fair) S-shuffle — on an alphabet Γ — a number of languages, is irrelevant, i.e. $L_1 \,|||^{\Gamma}\, L_2 \,|||^{\Gamma}\, \cdots \,|||^{\Gamma}\, L_n$ and $L_1 \,||^{\Gamma}\, L_2 \,||^{\Gamma}\, \cdots \,||^{\Gamma}\, L_n$ unambiguously define the fair S-shuffle and the S-shuffle, respectively, on Γ of languages L_1, L_2, ..., L_n, for an $n \geq 1$. We introduce some shorthand notations for such *n-ary (fair) S-shuffles*.

Notation 13. *We denote the fair S-shuffle* $L_1 \,|||^{\Gamma}\, L_2 \,|||^{\Gamma}\, \cdots \,|||^{\Gamma}\, L_n$ *and the S-shuffle* $L_1 \,||^{\Gamma}\, L_2 \,||^{\Gamma}\, \cdots \,||^{\Gamma}\, L_n$*, for an* $n \geq 1$*, by* $|||^{\Gamma}_{i\in[n]}\, L_i$ *and* $||^{\Gamma}_{i\in[n]}\, L_i$*, respectively.* □

Note that contrary to the (fair) shuffle and the (fair) S-shuffle, it is currently impossible to write either the (fair) fS-shuffle or the (fair) rS-shuffle — on an alphabet Γ — of languages L_1, L_2, ..., L_n, for an $n \geq 3$, without brackets since the order in which they are applied determines the synchronization symbols. We now present an example to illustrate this.

Example 6.4.45. Let $L_1 \subseteq \Delta_1^*$, $L_2 \subseteq \Delta_2^*$, $L_3 \subseteq \Delta_3^*$, and $L_4 \subseteq \Delta_4^*$. Then by Theorem 6.4.29, $((L_1 \,{}_{\Delta_1}\underline{|||}_{\Delta_2}\, L_2) \,{}_{\Delta_1\cup\Delta_2}\underline{|||}_{\Delta_3}\, L_3) \,{}_{\Delta_1\cup\Delta_2\cup\Delta_3}\underline{|||}_{\Delta_4}\, L_4 = (L_1 \,{}_{\Delta_1}\underline{|||}_{\Delta_2} L_2) \,{}_{\Delta_1\cup\Delta_2}\underline{|||}_{\Delta_3\cup\Delta_4} (L_3 \,{}_{\Delta_3}\underline{|||}_{\Delta_4} L_4) = L_1 \,{}_{\Delta_1}\underline{|||}_{\Delta_2\cup\Delta_3\cup\Delta_4} (L_2 \,{}_{\Delta_2}\underline{|||}_{\Delta_3\cup\Delta_4} (L_3 \,{}_{\Delta_3}\underline{|||}_{\Delta_4}\, L_4))$.

Now we let Γ be an alphabet. Then $((L_1 \,{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2}\, L_2) \,{}_{\Delta_1\cup\Delta_2}\underline{|||}^{\Gamma}_{\Delta_3}\, L_3) \,{}_{\Delta_1\cup\Delta_2\cup\Delta_3}\underline{|||}^{\Gamma}_{\Delta_4}\, L_4 = (L_1 \,{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2}\, L_2) \,{}_{\Delta_1\cup\Delta_2}\underline{|||}^{\Gamma}_{\Delta_3\cup\Delta_4}\, (L_3 \,{}_{\Delta_3}\underline{|||}^{\Gamma}_{\Delta_4}\, L_4) = L_1 \,{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2\cup\Delta_3\cup\Delta_4}\, (L_2 \,{}_{\Delta_2}\underline{|||}^{\Gamma}_{\Delta_3\cup\Delta_4}\, (L_3 \,{}_{\Delta_3}\underline{|||}^{\Gamma}_{\Delta_4}\, L_4))$ by Theorem 6.4.31. □

There are various ways of writing the n-ary (fair) fS-shuffles and (fair) rS-shuffles, for an $n \geq 3$, which by Theorems 6.4.29, 6.4.31, 6.4.42, and 6.4.44, are equivalent — provided that in the unfair case the languages are prefix closed. We choose the left-associative variants as standard representants of these classes.

Notation 14. *Let* $n \geq 1$.

The fair fS-shuffle of languages L_1, L_2, ..., L_n, *with respect to* Δ_1, Δ_2, ..., Δ_n, *is* $(\cdots(L_1 \,{}_{\Delta_1}\underline{|||}_{\Delta_2}\, L_2) \,{}_{\Delta_1\cup\Delta_2}\underline{|||}_{\Delta_3}\, \cdots) \,{}_{\cup_{i\in[n-1]}\Delta_i}\underline{|||}_{\Delta_n}\, L_n$ *and the fS-shuffle of* L_1, L_2, ..., L_n, *with respect to* Δ_1, Δ_2, ..., Δ_n, *is* $(\cdots(L_1 \,{}_{\Delta_1}\underline{||}_{\Delta_2}\, L_2) \,{}_{\Delta_1\cup\Delta_2}\underline{||}_{\Delta_3}\, \cdots) \,{}_{\cup_{i\in[n-1]}\Delta_i}\underline{||}_{\Delta_n}\, L_n$.

The fair rS-shuffle on an alphabet Γ *of* L_1, L_2, ..., L_n, *with respect to* Δ_1, Δ_2, ..., Δ_n, *is* $(\cdots(L_1 \,{}_{\Delta_1}\underline{|||}^{\Gamma}_{\Delta_2}\, L_2) \,{}_{\Delta_1\cup\Delta_2}\underline{|||}^{\Gamma}_{\Delta_3}\, \cdots) \,{}_{\cup_{i\in[n-1]}\Delta_i}\underline{|||}^{\Gamma}_{\Delta_n}\, L_n$ *and the rS-shuffle on* Γ *of* L_1, L_2, ..., L_n, *with respect to* Δ_1, Δ_2, ..., Δ_n, *is* $(\cdots(L_1 \,{}_{\Delta_1}\underline{||}^{\Gamma}_{\Delta_2}\, L_2) \,{}_{\Delta_1\cup\Delta_2}\underline{||}^{\Gamma}_{\Delta_3}\, \cdots) \,{}_{\cup_{i\in[n-1]}\Delta_i}\underline{||}^{\Gamma}_{\Delta_n}\, L_n$. □

We now introduce some shorthand notations for these *n-ary (fair) fS-shuffles* and *n-ary (fair) rS-shuffles*.

Notation 15. *Let $n \geq 1$.*

We denote the fair fS-shuffle and the fS-shuffle of languages L_1, L_2, ..., L_n, with respect to Δ_1, Δ_2, ..., Δ_n, by $\underline{|||}_{\{\Delta_i | i \in [n]\}} L_i$ and $\underline{||}_{\{\Delta_i | i \in [n]\}} L_i$, respectively.

We denote the fair rS-shuffle and the rS-shuffle on an alphabet Γ of L_1, L_2, ..., L_n, with respect to Δ_1, Δ_2, ..., Δ_n, by $\underline{|||}^{\Gamma}_{\{\Delta_i | i \in [n]\}} L_i$ and $\underline{||}^{\Gamma}_{\{\Delta_i | i \in [n]\}} L_i$, respectively. □

For the next section it is convenient to reformulate some results on the associativity of (fair) fS-shuffling using the new notations.

Theorem 6.4.46. *(1) If $w_i \in \Delta_i^\infty$, for all $i \in [n]$, then $\underline{|||}_{\{\Delta_i | i \in [n]\}} \{w_i\} = \{w \in (\bigcup_{i \in [n]} \Delta_i)^\infty \mid \forall\, i \in [n] :\ pres_{\Delta_i}(w) = w_i\}$, and*

(2) if $L_i \subseteq \Delta_i^\infty$, for all $i \in [n]$, are prefix closed, then $\underline{||}_{\{\Delta_i | i \in [n]\}} L_i = \{w \in (\bigcup_{i \in [n]} \Delta_i)^\infty \mid \forall\, i \in [n] :\ pres_{\Delta_i}(w) \in L_i\}$.

Proof. (1) This follows from the repeated application of Corollary 6.4.27 and the observation that for all $i, j \in [n]$ and $x \in \Delta_i^\infty$, $\mathrm{pres}_{\Delta_i}(\mathrm{pres}_{\Delta_i \cup \Delta_j}(x)) = \mathrm{pres}_{\Delta_i}(x)$.

(2) This follows from Theorem 6.4.40 and its proof. □

6.5 Team Automata Satisfying Compositionality

In this section we combine the relations between the behavior of team automata and that of their constituting component automata — as developed in Section 6.2 — and the (synchronized) shuffles from Sections 6.3 and 6.4.

In our general setup team automata may have an infinite set of component automata. In the context of compositionality, however, it is more realistic to consider team automata composed over a finite set of component automata.

Notation 16. *For the remainder of this chapter we assume that our fixed composable system $\mathcal{S}$ is finite, viz. $\mathcal{I}$ is a finite subset of $\mathbb{N}$.* □

Each *ai* synchronization in a team automaton requires the participation of all its constituting component automata sharing the action being synchronized. This is reflected in the following result, which shows that the behavior of the *maximal-ai* team automaton, in which no *ai* synchronizations are excluded,

can be described as the fS-shuffle of the behavior of its constituting component automata. Corresponding versions of this result have been formulated for other automata-based specification models with composition based on the *ai* principle (see, e.g., [Tut87] and [Jon87]).

Theorem 6.5.1. *Let $\mathcal{T}$ be the $\mathcal{R}^{ai}$-team automaton over $\mathcal{S}$. Then*

$$\mathbf{B}_{\mathcal{T}}^{\Sigma,\infty} = \underline{\|}_{\{\Sigma_i \mid i \in \mathcal{I}\}} \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}.$$

Proof. ($\subseteq$) This follows immediately from Theorem 6.2.9, the prefix closure of the behavior of component automata, and Theorem 6.4.46(2).

($\supseteq$) Let $w \in \underline{\|}_{\{\Sigma_i \mid i \in \mathcal{I}\}} \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$. Note that each $\mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$ is prefix closed. Hence, according to Theorem 6.4.46(2), $\mathrm{pres}_{\Sigma_i}(w) \in \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$, for all $i \in \mathcal{I}$. Consequently, by definition there exist $\alpha_i \in \mathbf{C}_{\mathcal{C}_i}^{\infty}$ such that $\mathrm{pres}_{\Sigma_i}(\alpha_i) = \mathrm{pres}_{\Sigma_i}(w)$, for all $i \in \mathcal{I}$. Hence $\prod_{i\in\mathcal{I}} \alpha_i \in \prod_{i\in\mathcal{I}} \mathbf{C}_{\mathcal{C}_i}^{\infty}$. Since $w \in \Sigma^{\infty}$ is such that $\mathrm{pres}_{\Sigma_i}(w) = \mathrm{pres}_{\Sigma_i}(\alpha_i)$, for all $i \in \mathcal{I}$, Corollary 6.2.15 implies that there exists a $\beta \in \mathbf{C}_{\mathcal{T}}^{\infty}$ such that $\mathrm{pres}_{\Sigma}(\beta) = w$. Hence $w \in \mathbf{B}_{\mathcal{T}}^{\Sigma,\infty}$. □

Example 6.5.2. (Example 6.2.12 continued) Recall the $\mathcal{R}^{ai}$-team automaton $\mathcal{T}^{ai}$ over $\{\mathcal{C}_1, \mathcal{C}_2\}$, depicted in Figure 6.4(b).

Indeed we see that we get $\mathbf{B}_{\mathcal{T}^{ai}}^{\Sigma,\infty} = \{\lambda, a\} = (\{b^n \mid n \geq 0\} \cup \{b^n a \mid n \geq 0\} \cup \{b^\omega\})\ {}_{\{a,b\}}\underline{\|}_{\{a,b\}}\ (\{\lambda\} \cup \{ab^n \mid n \geq 0\} \cup \{ab^\omega\}) = \mathbf{B}_{\mathcal{C}_1}^{\Sigma_1,\infty}\ {}_{\Sigma_1}\underline{\|}_{\Sigma_2}\ \mathbf{B}_{\mathcal{C}_2}^{\Sigma_2,\infty} = (\underline{\|}_{\Sigma_1}\ \mathbf{B}_{\mathcal{C}_1}^{\Sigma_1,\infty})\ {}_{\Sigma_1}\underline{\|}_{\Sigma_2}\ \mathbf{B}_{\mathcal{C}_2}^{\Sigma_2,\infty} = \underline{\|}_{\{\Sigma_i \mid i \in [2]\}}\ \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$.

Now recall the team automaton $\mathcal{T}$ over $\{\mathcal{C}_1, \mathcal{C}_2\}$, depicted in Figure 6.3(a). Note that while $ba \notin \{\lambda, a\} = \underline{\|}_{\{\Sigma_i \mid i \in [2]\}}\ \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$, clearly $ba \in \mathbf{B}_{\mathcal{T}}^{\Sigma,\infty}$ □

Each *free* synchronization in a team automaton is such that only one of its component automata participates — under the assumption that a loop on the action being synchronized is always executed. Hence, if we require $\mathcal{S}$ to be loop limited, then the behavior of the *maximal-free* team automaton over $\mathcal{S}$ equals the shuffle of the behavior of the component automata from $\mathcal{S}$. Actually we prove a more general result, viz. that the behavior of a specific heterogeneous team automaton that is composed according to a mixture of *maximal-free* and *maximal-ai* synchronizations equals the rS-shuffle of the behavior of its constituting component automata.

Theorem 6.5.3. *Let $\bar{\Gamma} = \Sigma \setminus \Gamma$ and let $\mathcal{T}$ be the $\{\mathcal{R}_a^{ai} \mid a \in \Sigma \cap \Gamma\} \cup \{\mathcal{R}_a^{free} \mid a \in \bar{\Gamma}\}$-team automaton over $\mathcal{S}$. Then*

if $\mathcal{S}$ is $\bar{\Gamma}$-loop limited, then $\mathbf{B}_{\mathcal{T}}^{\Sigma,\infty} = \underline{\|}^{\Gamma}_{\{\Sigma_i \mid i \in \mathcal{I}\}}\ \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$.

Proof. Let $\mathcal{T}'$ be the team automaton that is obtained from $\mathcal{T}$ by attaching a label to each action from $\bar{\Gamma}$ depending on the component automaton executing that action, i.e. $\mathcal{T}' = (Q, \Sigma', \delta', I)$ with $\Sigma' = \{[a,i] \mid a \in \bar{\Gamma} \cap \Sigma_i,\ i \in \mathcal{I}\} \cup (\Sigma \cap \Gamma)$ and $\delta' = \{(q,[a,i],q') \mid a \in \bar{\Gamma},\ (q,a,q') \in \delta,\ \mathrm{proj}_i^{[2]}(q,q') \in \delta_{i,a},\ i \in \mathcal{I}\} \cup (\delta \cap (Q \times \Gamma \times Q))$. Since all actions from $\bar{\Gamma}$ are *free* in $\mathcal{T}$, the behavior of $\mathcal{T}$ is an encoding of the behavior of $\mathcal{T}'$. Let $\psi : (\Sigma')^* \to \Sigma^*$ be the homomorphism defined by $\psi([a,i]) = a$ and $\psi(a) = a$. Then clearly $\mathbf{B}_{\mathcal{T}}^{\Sigma,\infty} = \psi(\mathbf{B}_{\mathcal{T}'}^{\Sigma',\infty})$.

For all $i \in \mathcal{I}$, let $\mathcal{C}_i'$ be the component automaton that is obtained from $\mathcal{C}_i$ by labeling each of its actions from $\bar{\Gamma}$ with i, i.e. $\mathcal{C}_i' = (Q_i, \Sigma_i', \delta_i', I_i)$ with $\Sigma_i' = \{[a,i] \mid a \in \bar{\Gamma} \cap \Sigma_i\} \cup (\Gamma \cap \Sigma_i)$ and $\delta_i' = \{(q,[a,i],q') \mid a \in \bar{\Gamma},\ (q,a,q') \in \delta_i\} \cup (\delta_i \cap (Q_i \times \Gamma \times Q_i))$. Obviously, $\mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty} = \psi(\mathbf{B}_{\mathcal{C}_i'}^{\Sigma_i',\infty})$, for all $i \in \mathcal{I}$. Let $\mathcal{S}' = \{\mathcal{C}_i' \mid i \in \mathcal{I}\}$. Since $\mathcal{S}$ is $\bar{\Gamma}$-loop limited it thus follows that $(\delta')_{[a,i]} = \mathcal{R}_{[a,i]}^{free}(\mathcal{S}')$, for all $a \in \bar{\Gamma}$ and for all $i \in \mathcal{I}$. Hence $\mathcal{T}'$ is the $\{\mathcal{R}_a^{ai} \mid a \in (\Sigma \cap \Gamma)\} \cup \{\mathcal{R}_a^{free} \mid a \in \Sigma' \setminus \Gamma\}$-team automaton over $\mathcal{S}'$. Moreover, since the component automata from $\mathcal{S}'$ can share actions from $\Sigma \cap \Gamma$ but not from $\Sigma' \setminus \Gamma$, it follows that for all $K \subseteq \mathcal{I}$, $\bigcap_{k \in K} \Sigma_k' = \bigcap_{k \in K} \Sigma_k \cap \Gamma$. Hence Theorem 4.5.5 implies that $\mathcal{T}'$ is the *maximal-ai* team automaton over $\mathcal{S}'$ as well. Subsequently, Theorem 6.5.1 and Lemma 6.4.7(2) imply that $\mathbf{B}_{\mathcal{T}}^{\Sigma,\infty} = \psi(\mathbf{B}_{\mathcal{T}'}^{\Sigma',\infty}) = \psi(\underline{||}_{\{\Sigma_i' \mid i \in \mathcal{I}\}} \mathbf{B}_{\mathcal{C}_i'}^{\Sigma_i',\infty}) = \psi(\underline{||}_{\{\Sigma_i' \mid i \in \mathcal{I}\}}^{\Gamma} \mathbf{B}_{\mathcal{C}_i'}^{\Sigma_i',\infty})$, which equals $\underline{||}_{\{\psi(\Sigma_i') \mid i \in \mathcal{I}\}}^{\Gamma} \psi(\mathbf{B}_{\mathcal{C}_i'}^{\Sigma_i',\infty}) = \underline{||}_{\{\Sigma_i \mid i \in \mathcal{I}\}}^{\Gamma} \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$ because $\psi(\Sigma' \setminus \Gamma) \cap \Gamma = \varnothing$. □

Example 6.5.4. (Example 6.2.1 continued) The $\mathcal{R}_a^{free} \cup \mathcal{R}_b^{ai}$-team automaton over $\{\mathcal{C}_1, \mathcal{C}_2\}$ is defined as $\mathcal{T}^{fa} = (\{(q_1,q_2),(q_1,q_2'),(q_1',q_2),(q_1',q_2')\}, \{a,b\}, \delta^{fa}, \{(q_1,q_2)\})$, where $\delta^{fa} = \{((q_1,q_2),a,(q_1,q_2')),((q_1,q_2),a,(q_1',q_2)),((q_1,q_2'),b,(q_1,q_2')),((q_1,q_2'),a,(q_1',q_2')),((q_1',q_2),a,(q_1',q_2'))\}$ and it is depicted in Figure 6.6(b).

Clearly $\{\mathcal{C}_1, \mathcal{C}_2\}$ is $\{a\}$-loop limited and indeed we see that $\mathbf{B}_{\mathcal{T}^{fa}}^{\Sigma,\infty} = \underline{||}_{\{\Sigma_i \mid i \in [2]\}}^{\{b\}} \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$. □

The behavior of the *maximal-free* team automaton over a loop limited composable system thus equals the shuffle of the behavior of its constituting component automata.

Theorem 6.5.5. *Let $\mathcal{T}$ be the $\mathcal{R}^{free}$-team automaton over $\mathcal{S}$. Then*

if $\mathcal{S}$ is loop limited, then $\mathbf{B}_{\mathcal{T}}^{\Sigma,\infty} = ||_{i \in \mathcal{I}} \mathbf{B}_{\mathcal{C}_i}^{\Sigma_i,\infty}$.

Proof. This follows immediately from Theorem 6.5.3 with $\Sigma \cap \Gamma = \varnothing$. □

Example 6.5.6. (Examples 6.2.22 and 6.5.4 continued) Recall the $\mathcal{R}^{free}$-team automaton $\mathcal{T}^{free}$ over $\{\mathcal{C}_1, \mathcal{C}_2\}$, depicted in Figure 6.6(a). Recall also that

$\{\mathcal{C}_1, \mathcal{C}_2\}$ is $\{a\}$-loop limited. Indeed $\mathbf{B}^{\{a\},\infty}_{\mathcal{T}^{free}} = \{\lambda, a, aa\} = \{\lambda, a\} \mid\mid \{\lambda, a\} = \mathbf{B}^{\{a\},\infty}_{\mathcal{C}_1} \mid\mid \mathbf{B}^{\{a\},\infty}_{\mathcal{C}_2} = (\mid\mid_{i\in[1]} \mathbf{B}^{\{a\},\infty}_{\mathcal{C}_1}) \mid\mid \mathbf{B}^{\{a\},\infty}_{\mathcal{C}_2} = \mid\mid_{i\in[2]} \mathbf{B}^{\{a\},\infty}_{\mathcal{C}_i}$.

Since $\{\mathcal{C}_1, \mathcal{C}_2\}$ however is not loop limited, it is no surprise that $ab \notin \mathbf{B}^{\Sigma,\infty}_{\mathcal{T}^{free}}$, whereas $ab \in \mid\mid_{i\in[2]} \mathbf{B}^{\Sigma_i,\infty}_{\mathcal{C}_i}$. □

Summarizing, we have thus been able to describe the behavior of three types of team automata in terms of the behavior of their constituting component automata (cf. Theorems 6.5.1, 6.5.3, and 6.5.5). However, we needed the condition of loop limitedness to avoid ambiguity with respect to the participation of component automata in case of loops. The reason is — once again — the maximal interpretation adopted in Section 4.2. In the next chapter we will show how to circumvent this problem by switching to vectors of actions.

The results of this section provide a semantic equivalent of the syntactic hierarchical results presented in Sections 4.3 and 5.2. Recall from those sections that every iterated team automaton over $\mathcal{S}$ can be considered as a team automaton directly composed over $\mathcal{S}$. Hence, if we construct only *maximal-ai* team automata, then the fact that fS-shuffling is associative for prefix-closed languages implies that the behavior of each such (iterated) *maximal-ai* team automaton equals the fS-shuffle of the behavior of its constituting component automata from $\mathcal{S}$. Such (iterated) *maximal-ai* team automata thus satisfy compositionality. A similar reasoning can be applied in case we consider (iterated) *maximal-free* team automata or (iterated) $\{\mathcal{R}^{ai}_a \mid a \in \Sigma \cap \Gamma\} \cup \{\mathcal{R}^{free}_a \mid a \in \Sigma \setminus \Gamma\}$-team automata over $\mathcal{S}$, where Γ is an alphabet. These satisfy compositonality in the sense that their behavior equals the shuffle or rS-shuffle, respectively, of the behavior of their constituting component automata from $\mathcal{S}$. We now illustrate this exposition by an example. Note that the fact that the distinction of input, output, and internal actions is irrelevant here allows us to deal with synchronized automata rather than team automata in this example.

Example 6.5.7. (Example 4.3.1 continued) Assume that all synchronized automata composed in Example 4.3.1 are *maximal-ai* synchronized automata.

Theorem 6.5.1 then implies that $\mathbf{B}^{\Sigma,\infty}_{\mathcal{T}_{1-7}} = \underline{\mid\mid}_{\{\Sigma_i \mid i\in[7]\}} \mathbf{B}^{\Sigma_i,\infty}_{\mathcal{A}_i}$.

Consequently, together with the commutativity of fS-shuffling (cf. Corollary 6.4.19) and the associativity of fS-shuffling for prefix closed languages (cf. Theorems 6.4.29 and 6.4.42) Theorem 6.5.1 furthermore implies that $\mathbf{B}^{\Gamma'',\infty}_{\mathcal{T}''} = \mathbf{B}^{\Gamma',\infty}_{\mathcal{T}'} \underline{\mid\mid}_{\cup_{i\in[6]}\Sigma_i} \mathbf{B}^{\Sigma_7,\infty}_{\mathcal{A}_7} = (\mathbf{B}^{\Gamma_1,\infty}_{\mathcal{T}_{\{2,4,6\}}} \underline{\mid\mid}_{\cup_{i\in\{2,4,6\}}\Sigma_i \cup \cup_{i\in\{1,3,5\}}\Sigma_i} \mathbf{B}^{\Gamma_2,\infty}_{\mathcal{T}_{\{1,3,5\}}}) \underline{\mid\mid}_{(\cup_{i\in\{2,4,6\}}\Sigma_i)\cup(\cup_{i\in\{1,3,5\}}\Sigma_i)} \mathbf{B}^{\Sigma_7,\infty}_{\mathcal{A}_7} = ((\underline{\mid\mid}_{\{\Sigma_i\mid i\in\{2,4,6\}\}} \mathbf{B}^{\Sigma_i,\infty}_{\mathcal{A}_i}) \underline{\mid\mid}_{\cup_{i\in\{2,4,6\}}\Sigma_i \cup \cup_{i\in\{1,3,5\}}\Sigma_i} (\underline{\mid\mid}_{\{\Sigma_i\mid i\in\{1,3,5\}\}} \mathbf{B}^{\Sigma_i,\infty}_{\mathcal{A}_i})) \underline{\mid\mid}_{(\cup_{i\in\{2,4,6\}}\Sigma_i)\cup(\cup_{i\in\{1,3,5\}}\Sigma_i)} \mathbf{B}^{\Sigma_7,\infty}_{\mathcal{A}_7} = (\underline{\mid\mid}_{\{\Sigma_i\mid i\in[6]\}} \mathbf{B}^{\Sigma_i,\infty}_{\mathcal{A}_i}) \underline{\mid\mid}_{\cup_{i\in[6]}\Sigma_i} \mathbf{B}^{\Sigma_7,\infty}_{\mathcal{A}_7} = \underline{\mid\mid}_{\{\Sigma_i\mid i\in[7]\}} \mathbf{B}^{\Sigma_i,\infty}_{\mathcal{A}_i} = \mathbf{B}^{\Sigma,\infty}_{\mathcal{T}_{1-7}}$. □

We close this chapter with an observation on *si* synchronizations. From a behavioral point of view, *si* synchronizations are very different from both *ai* and *free* synchronizations. While an *ai* synchronization of an action requires the participation of every component automaton with that action, a *free* synchronization of an action requires the participation of only and exactly one component automaton with that action. Whether an action of a component automaton is required to participate in an *si* synchronization of that action, however, cannot be decided without information on its current local state. A shuffle that would describe the behavior of a *maximal-si* team automaton in terms of the behavior of its constituting component automata should thus be a type of synchronized shuffle that — depending on local states of the component automata — is able to decide which actions of the component automata must be interleaved and which must be synchronized. This, however, seems impossible due to the simple fact that the behavior of component automata is stripped from all state information.