Regular Sets of Pomsets with Autoconcurrency

Jean Fanchon[1] and Rémi Morin[2]

[1] LAAS-CNRS
7, avenue du colonel Roche, F-31077 Toulouse, France
[2] LIF
39 rue F. Joliot-Curie, F-13453 Marseille cedex 13, France

Abstract. Partially ordered multisets (or pomsets) constitute one of the most basic models of concurrency. We introduce and compare several notions of regularity for pomset languages by means of contexts and residues of different kinds. We establish some interesting closure properties that allow us to relate this approach to SP-recognizability in the particular case of series-parallel pomsets. Finally we introduce the framework of compatible languages which generalizes several classical formalisms (including message sequence charts and firing pomsets of Petri nets). In this way, we identify regular sets of pomsets as recognizable subsets in the monoid of multiset sequences.

Partially ordered multisets or partial words constitute one of the most basic models of concurrency [14]. Process algebras like CCS and TCSP, and system models like Petri nets, have been given pomset semantics for many years, and several pomset algebras have been designed. Nevertheless the study of pomset languages from the point of view of recognizability or regularity still offers many interesting problems to investigate.

A key difficulty one encounters in the case of pomsets as opposed to words is the lack of a finite set of operators generating all of them [7]. As an alternative approach, different accepting finite devices for pomset languages have been defined such as, e.g., graph acceptors applied to pomsets [15] and asynchronous automata applied to restricted classes of pomsets without autoconcurrency [4]. In these works the recognizable languages have until now no characterization in an algebraic framework.

On the other hand, some restricted frameworks have been defined and investigated with an algebraic approach of recognizable sets of pomsets. In particular, the trace monoids [3], the concurrency monoids of stably concurrent automata [5] and more recently the monoid of basic message sequence charts [12] have been studied in details.

To our knowledge, until now little work has been done on languages of pomsets with autoconcurrency. The main exception is the algebra of series-parallel pomsets, for which Kleene-like results [11] and Büchi-like results [9] were established recently. In [6], a definition of recognizability by vertex substitution has been studied. It extends to the class of all pomsets the algebraic definition of [11] for SP-pomsets, and fulfills several closure properties.

L. Brim et al. (Eds.): CONCUR 2002, LNCS 2421, pp. 402–417, 2002.
© Springer-Verlag Berlin Heidelberg 2002

Investigating different interpretations of what a pomset language expresses, we introduce in this paper three candidate extensions to pomset languages of the definition of context known for word languages. The corresponding context equivalences $\asymp^{\mathcal{L}}$, $\asymp_{\mathrm{w}}^{\mathcal{L}}$, and $\asymp_{\mathrm{s}}^{\mathcal{L}}$ are shown to give rise to three distinct notions of regularity: More precisely we show

$$\asymp^{\mathcal{L}} \text{ is of finite index} \implies \asymp_{\mathrm{w}}^{\mathcal{L}} \text{ is of finite index} \implies \asymp_{\mathrm{s}}^{\mathcal{L}} \text{ is of finite index}$$

for a given pomset language $\mathcal{L}$ — but none of the converses holds in general. By means of several closure properties, we show also that these notions of regularity are weaker than the algebraic SP-recognizability [11] when we restrict to series-parallel pomsets.

In order to relate our three notions of regularity with the algebraic definition of recognizability in monoids, we introduce the class of *compatible languages*. The latter are characterized by their corresponding step sequences. They include Mazurkiewicz traces [3], local trace languages [10], subsets of message sequence charts [1, 13], pomset languages of stably concurrent automata [5], CCI sets of P-traces [2], and firing pomset languages of Place/Transition nets [8, 16] — but not series-parallel pomsets. Differently from the more general case of weak languages, we show that regularity of a compatible set of pomsets means exactly recognizability of the corresponding step extensions in the free monoid of multiset sequences. Thus, this wide framework of compatible pomset languages proves to have some interesting properties, although it generalizes many classical formalisms of concurrency theory. For this reason, we believe that these compatible languages deserve to be further investigated. In particular, realization of regular languages by labelled Petri nets and correspondence with MSO definability would lead to nice generalizations of results known in particular frameworks.

1 Basics

In this section, we present our main notations and definitions concerning pomsets. We recall the simple relationship between weak languages and basic languages of pomsets. We introduce the notion of *compatible languages* and show that this concept generalizes several classical frameworks of concurrency theory.

Preliminaries. In this paper, we consider a finite alphabet Σ. For any words $u, v \in \Sigma^\star$, we write $u \leqslant v$ if u is a prefix of v, i.e. there is $z \in \Sigma^\star$ such that $u.z = v$. The set of multisets over Σ is denoted by $\mathcal{M}(\Sigma)$ and $\mathcal{M}(\Sigma)^\star$ denotes the set of finite sequences of multisets over Σ, also called *step sequences*.

A *pomset* over Σ is a triple $t = (E, \preccurlyeq, \xi)$ where $(E, \preccurlyeq)$ is a finite partial order and ξ is a mapping from E to Σ. A pomset is *without autoconcurrency* if $\xi(x) = \xi(y)$ implies $x \preccurlyeq y$ or $y \preccurlyeq x$ for all $x, y \in E$. A pomset can be seen as an abstraction of an execution of a concurrent system. In this view, the elements e of E are *events* and their label $\xi(e)$ describes the basic action of the system that is performed when the event e occurs. Furthermore, the order describes

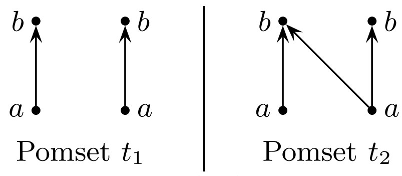


Fig. 1. Two pomsets with the same step extensions

the causal dependence between the events. In particular, a pomset is without autoconcurrency if two occurrences of an action cannot run simultaneously. We denote by $\mathbb{P}(\Sigma)$ the class of all pomsets over Σ. As usual, we shall identify isomorphic pomsets as a single object and consider only classes of pomsets closed under isomorphisms.

A *prefix* of a pomset $t = (E, \preccurlyeq, \xi)$ is the restriction $t' = (E', \preccurlyeq_{|E'}, \xi_{|E'})$ of t to some downward-closed subset E' of E. An *order extension* of a pomset $t = (E, \preccurlyeq, \xi)$ is a pomset $t' = (E, \preccurlyeq', \xi)$ such that $\preccurlyeq \subseteq \preccurlyeq'$. We denote by $\mathrm{OE}(t)$ the set of order extensions of t. A *linear extension* of t is an order extension that is linearly ordered. Now, words can naturally be considered as linear pomsets. Therefore, the set $\mathrm{LE}(t)$ of linear extensions of a pomset t over Σ may be identified to a subset of $\Sigma^\star$: $\mathrm{LE}(t) = \mathrm{OE}(t) \cap \Sigma^\star$.

More generally, any sequence of multisets $w \in \mathcal{M}(\Sigma)^\star$ written $w = p_1...p_n$ can be identified with a pomset over the set of events $p_1 \uplus ... \uplus p_n$ such that each event of p_i is below each event of p_j for $1 \leqslant i < j \leqslant n$. For any pomset t, the set of *step extensions* $\mathrm{SE}(t)$ consists of the multiset sequences that are order extensions of t: $\mathrm{SE}(t) = \mathrm{OE}(t) \cap \mathcal{M}(\Sigma)^\star$. As the next example shows, two distinct pomsets may share the same set of step extensions.

Example 1.1. Consider the two pomsets t_1 and t_2 of Fig. 1. One easily checks that $\mathrm{SE}(t_1) = \mathrm{SE}(t_2)$.

Note that this phenomenon is typical of pomsets with autoconcurrency: For any two pomsets t_1 and t_2 *without* autoconcurrency, if $\mathrm{LE}(t_1) = \mathrm{LE}(t_2)$ then t_1 and t_2 are isomorphic. This holds also if $\mathrm{SE}(t_1) = \mathrm{SE}(t_2)$ since $\mathrm{LE}(t) = \mathrm{SE}(t) \cap \Sigma^\star$.

Thus, pomsets with autoconcurrency are in general more expressive than their step extensions. We want here to study particular classes of pomsets (called *compatible sets*) that are faithfully described by the corresponding step extensions. As we shall observe below (Prop. 1.9), this includes the languages of firing pomsets of Place/Transition nets.

Weak and Basic Sets of Pomsets. The map OE from pomsets to sets of pomsets can be extended to languages of pomsets by putting $\mathrm{OE}(\mathcal{L}) = \bigcup_{t \in \mathcal{L}} \mathrm{OE}(t)$. Obviously, $\mathcal{L} \subseteq \mathrm{OE}(\mathcal{L}) = \mathrm{OE}(\mathrm{OE}(\mathcal{L}))$ for any language of pomsets $\mathcal{L}$. Now, a language is *weak* if it contains all its order extensions.

Definition 1.2. *A language of pomsets $\mathcal{L}$ is* weak *if $\mathcal{L} = \mathrm{OE}(\mathcal{L})$.*

Typical examples of weak languages are the sets of firing pomsets of Petri nets. Consider for instance the example of a Producer-Consumer system illustrated by the Petri net of Fig. 2. In this net, p represents a production of one

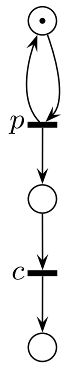


Fig. 2. Petri net $\mathcal{N}$

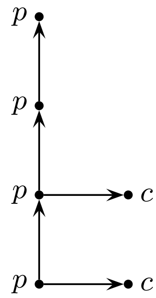


Fig. 3. Firing pomset t with autoconcurrency

item and c a consumption. Some concurrency may occur between p and c or even between distinct occurrences of c when the middle place contains several tokens. Usually, concurrent behaviors are described by partial orders. For instance here, a concurrent execution of the system is depicted in Fig. 3.

More generally, we let a *Petri net* be a quadruple $\mathcal{N} = (S, T, W, \mathrm{M}_{in})$ where

- S is a set of places and T is a set of transitions such that $S \cap T = \emptyset$;
- W is a map from $(S \times T) \cup (T \times S)$ to $\mathbb{N}$, called *weight function*;
- M_{in} is a map from S to $\mathbb{N}$, called *initial marking*.

We let $\mathrm{Mar}_{\mathcal{N}}$ denotes the set of all markings of $\mathcal{N}$ that is to say functions $\mathrm{M} : S \to \mathbb{N}$; a multiset p of transitions is *enabled* at $\mathrm{M} \in \mathrm{Mar}_{\mathcal{N}}$ if $\forall s \in S$, $\mathrm{M}(s) \geqslant \sum_{t \in T} p(t) \cdot W(s,t)$; in this case, we note $\mathrm{M}\,[p\rangle\,\mathrm{M}'$ where $\mathrm{M}'(s) = \mathrm{M}(s) + \sum_{t \in T} p(t) \cdot (W(t,s) - W(s,t))$ and say that the transitions of p may be *fired* concurrently and lead to the marking M'. A *multiset firing sequence* consists of a sequence of markings M_0,..., M_n and a sequence of multisets of transitions p_1,..., p_n such that $\mathrm{M}_0 = \mathrm{M}_{in}$ and $\forall k \in [1, n]$, $\mathrm{M}_{k-1}\,[p_k\rangle\,\mathrm{M}_k$. A *firing pomset* [8, 16] of $\mathcal{N}$ is a pomset $t = (E, \preccurlyeq, \xi)$ such that for all prefixes $t' = (E', \preccurlyeq', \xi')$ of t, and for all linear extensions $a_1...a_n \in \mathrm{LE}(t')$, the sequence $\{a_1\}...\{a_n\}.\xi(\min_{\preccurlyeq}(E \setminus E'))$ is a multiset firing sequence of $\mathcal{N}$. For instance, the pomset t of Fig. 3 is a firing pomset of the Producer-Consumer. Clearly, the language of firing pomsets of any Petri net is weak.

As far as the description of concurrency is concerned, some redundancy of information may appear in weak languages. In order to focus on restricted but representative parts of weak languages, we look at *basic* sets of pomsets.

DEFINITION 1.3. *A language of pomsets $\mathcal{L}$ is* basic *if $t_1 \in \mathrm{OE}(t_2)$ implies $t_1 = t_2$, for all $t_1, t_2 \in \mathcal{L}$.*

We can check easily that two basic sets that have the same order extensions are equal: $\mathrm{OE}(\mathcal{L}) = \mathrm{OE}(\mathcal{L}')$ implies $\mathcal{L} = \mathcal{L}'$ for all basic sets of pomsets $\mathcal{L}$ and $\mathcal{L}'$. Therefore the map OE from basic sets of pomsets to weak sets of pomsets is one-to-one. Actually, we shall see that this map is also onto.

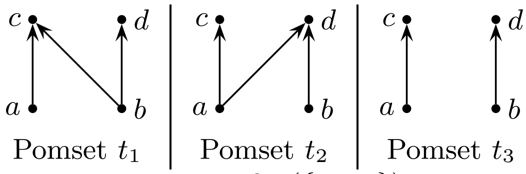


Fig. 4. The weak language $\mathrm{OE}(\{t_1, t_2\})$ is not compatible

Definition 1.4. *Let $\mathcal{L}$ be a language of pomsets. The basis of $\mathcal{L}$ consists of all pomsets $t \in \mathcal{L}$ which are no order extension of some other pomsets of $\mathcal{L}$:*

$$\mathrm{Basis}(\mathcal{L}) = \{t \in \mathcal{L} \mid t' \in \mathcal{L} \wedge t \in \mathrm{OE}(t') \Rightarrow t = t'\}.$$

For any weak language $\mathcal{L}$, $\mathrm{OE}(\mathrm{Basis}(\mathcal{L})) = \mathcal{L}$ and $\mathrm{Basis}(\mathcal{L})$ is a basic set of pomsets. Thus, we obtain *a one-to-one correspondence between weak languages of pomsets and basic languages of pomsets.* However, as already observed in [10], this duality may not preserve some regularity properties such as MSO definability.

Compatible Sets of Pomsets. We focus now on the step extensions of a language $\mathcal{L}$: It consists of step sequences that are order extensions of $\mathcal{L}$. Formally, we put $\mathrm{SE}(\mathcal{L}) = \mathrm{OE}(\mathcal{L}) \cap \mathcal{M}(\Sigma)^\star$. Equivalently, we also have $\mathrm{SE}(\mathcal{L}) = \bigcup_{t \in \mathcal{L}} \mathrm{SE}(t)$. The *step compatible closure* $\overline{\mathcal{L}}$ of a language $\mathcal{L}$ is the set of all pomsets whose step extensions are step extensions of $\mathcal{L}$: $\overline{\mathcal{L}} = \{t \in \mathbb{P}(\Sigma) \mid \mathrm{SE}(t) \subseteq \mathrm{SE}(\mathcal{L})\}$. Obviously $\mathcal{L} \subseteq \overline{\mathcal{L}}$ and $\overline{\mathcal{L}} = \overline{\mathrm{OE}(\mathcal{L})}$ because $\mathrm{SE}(\mathcal{L}) = \mathrm{SE}(\mathrm{OE}(\mathcal{L}))$. Hence $\mathcal{L} \subseteq \mathrm{OE}(\mathcal{L}) \subseteq \overline{\mathcal{L}}$. We say that $\mathcal{L}$ is *compatible* if it is maximal among all languages that share the same step extensions.

Definition 1.5. *A language of pomsets $\mathcal{L}$ is* compatible *if $\mathcal{L} = \overline{\mathcal{L}}$.*

We observe that any compatible set that includes t_1 or t_2 of Fig. 1 contains both of them because $\mathrm{SE}(t_1) = \mathrm{SE}(t_2)$ (Example 1.1). So neither $\{t_1\}$ nor $\{t_2\}$ is a compatible set of pomsets. Actually any compatible language is weak since $\mathcal{L} \subseteq \mathrm{OE}(\mathcal{L}) \subseteq \overline{\mathcal{L}}$. As the next example shows, the converse fails.

Example 1.6. Consider the pomsets t_1, t_2 and t_3 of Fig. 4. Then the weak language $\mathcal{L} = \mathrm{OE}(\{t_1, t_2\})$ is not compatible since $t_3 \in \overline{\mathcal{L}} \setminus \mathcal{L}$.

It turns out that several classical formalisms of concurrency theory are included in the framework of compatible languages. For instance, the next result shows that languages of Mazurkiewicz traces [3], subsets of message sequence charts [1, 13], pomset languages of stably concurrent automata [5], and CCI sets of P-traces [2] are particular cases of basic languages of pomsets whose order extensions are *compatible* languages of pomsets (without autoconcurrency).

Proposition 1.7. *Let $\mathcal{L}$ be a language of pomsets* without autoconcurrency *such that for all t_1, $t_2 \in \mathcal{L}$, if $\mathrm{LE}(t_1) \cap \mathrm{LE}(t_2) \neq \emptyset$ then $t_1 = t_2$. Then $\mathcal{L}$ is basic and $\mathrm{OE}(\mathcal{L})$ is compatible.*

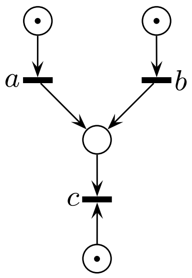


Fig. 5. A Petri net...

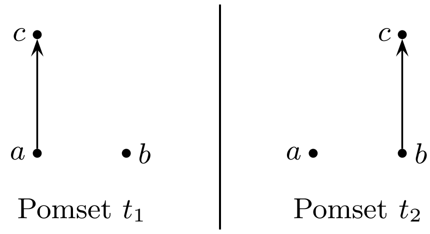


Fig. 6. ... and two of its firing pomsets

Proof. The language $\mathcal{L}$ is clearly basic. We take $t \in \overline{\mathrm{OE}(\mathcal{L})}$. Since $\overline{\mathcal{L}} = \overline{\mathrm{OE}(\mathcal{L})}$, we have $t \in \overline{\mathcal{L}}$. Note also that t is without autoconcurrency. Let $u_1 \in \mathrm{LE}(t)$. Since $t \in \overline{\mathcal{L}}$, there is $t_1 \in \mathcal{L}$ such that $u_1 \in \mathrm{LE}(t_1)$. Assume now that u_2 is another linear extension of t that only differs from u_1 by the order of two adjacent actions, i.e. $u_1 = a_1.a_2...a_n$ and $u_2 = a_1...a_{k-1}.a_{k+1}.a_k.a_{k+2}...a_n$. Then the step sequence $w = \{a_1\}...\{a_{k-1}\}.\{a_k.a_{k+1}\}.\{a_{k+2}\}...\{a_n\}$ is also a step extension of t. Therefore there is $t_2 \in \mathcal{L}$ such that $w \in \mathrm{SE}(t_2)$. Since t_1 and t_2 share u_1 as a linear extension, we get $t_1 = t_2$ hence $u_2 \in \mathrm{LE}(t_1)$. Since all linear extensions of t can be obtained from u_1 by successive permutations of adjacent (independent) actions, we conclude that $\mathrm{LE}(t) \subseteq \mathrm{LE}(t_1)$. Since t and t_1 are without autoconcurrency, t is actually an order extension of t_1, hence $t \in \mathrm{OE}(\mathcal{L})$. □

Remark 1.8. In Prop. 1.7 above, the assumption that $\mathcal{L}$ is without autoconcurrency cannot be removed: The set of order extensions of the basic language which consists only of the pomset t_2 of Fig. 1 is not compatible since $\mathrm{SE}(t_1) = \mathrm{SE}(t_2)$ (Example 1.1).

As claimed above, the framework of compatible languages includes the partial word semantics of Place/Transition nets.

Proposition 1.9. *The set of firing pomsets $\mathcal{L}$ of a Petri net $\mathcal{N}$ is compatible.*

Proof. Let $t = (E, \preccurlyeq, \xi)$ be a pomset of $\overline{\mathcal{L}}$. Let $t' = (E', \preccurlyeq_{|E'}, \xi_{|E'})$ be a prefix of t and $u' \in \mathrm{LE}(t')$ be a linear extension of t'. We put $M = \min_{\preccurlyeq}(E \setminus E')$ and $p = \xi(M)$. We consider a linear extension v' of the restriction of t to the events of $E \setminus (E' \cup M)$. Then $u'.p.v'$ is a step extension of t. Consequently, $u'.p.v' \in \mathrm{SE}(t_1)$ for some $t_1 \in \mathcal{L}$ and $u'.p$ is a step firing sequence of $\mathcal{N}$. Thus $t \in \mathcal{L}$ as well. □

To conclude this section, we consider the Petri net of Fig. 5. The basis $\mathrm{Basis}(\mathcal{L})$ of its firing pomsets contains the pomsets t_1 and t_2 of Fig. 6. Since $abc \in \mathrm{LE}(t_1) \cap \mathrm{LE}(t_2)$, not all compatible languages without autoconcurrency satisfy the condition of Prop. 1.7. Indeed, firing pomset languages of Petri nets are often much more involved than this restricted framework.

2 Regularity via Contexts and Residues

In this section, we investigate and relate several approaches that extend the usual notion of regularity from words to pomsets with autoconcurrency. We introduce two notions of *context* and two corresponding equivalence relations w.r.t. a given language. We compare the indexes of these two approaches (Theorem 2.4) and present a simplified equivalent definition by means of residues (Theorem 2.9). With the help of closure properties (Prop. 2.13), we compare this new approach of regularity to the more algebraic one known in the particular case of series-parallel pomset languages (Prop. 2.14).

Regularity via Contexts. The most basic operation on pomsets is the *strong concatenation*. Given two pomsets $t_1 = (E_1, \preccurlyeq_1, \xi_1)$ and $t_2 = (E_2, \preccurlyeq_2, \xi_2)$ over Σ, the product $t_1 \cdot t_2$ is the pomset that puts each event of t_2 after all events of t_1, i.e. $t_1 \cdot t_2 = (E_1 \uplus E_2, \preccurlyeq_1 \cup \preccurlyeq_2 \cup E_1 \times E_2, \xi_1 \cup \xi_2)$. This operation is the basis of the first notion of context which identifies a language and its order extensions. In other words, this approach adopts a weak interpretation of causality.

Definition 2.1. *Let $\mathcal{L}$ be a language of pomsets and u be a pomset over Σ. The* weak context *of u w.r.t. $\mathcal{L}$ is $\mathcal{L}\backslash_{\mathrm{w}} u = \{(x, y) \in \mathbb{P}(\Sigma)^2 \mid x \cdot u \cdot y \in \mathrm{OE}(\mathcal{L})\}$. We say that u and v are* weak context equivalent *w.r.t. $\mathcal{L}$ if $\mathcal{L}\backslash_{\mathrm{w}} u = \mathcal{L}\backslash_{\mathrm{w}} v$. In that case we write $u \asymp_{\mathrm{w}}^{\mathcal{L}} v$ or simply $u \asymp_{\mathrm{w}} v$ if $\mathcal{L}$ is clear from the context.*

In the literature, many other products were designed in different frameworks, in particular for Mazurkiewicz traces, message sequence charts, or in the more general framework of stably concurrent automata. Following this trend, we introduce a weak product that somehow does not specify the causalities between the two components, giving rise to several possible pomsets. This operation is therefore better defined for languages than for single pomsets. Given two sets of pomsets $\mathcal{L}_1$ and $\mathcal{L}_2$, the *weak concatenation* $\mathcal{L}_1 \diamond \mathcal{L}_2$ consists of all pomsets $t = (E, \preccurlyeq, \xi)$ that can be split into a prefix belonging to $\mathcal{L}_1$ and a remaining part belonging to $\mathcal{L}_2$, i.e. such that $E = E_1 \uplus E_2$ where E_1 is downward-closed w.r.t. $\preccurlyeq$, $(E_1, \preccurlyeq_{|E_1}, \xi_{|E_1})$ belongs to $\mathcal{L}_1$ and $(E_2, \preccurlyeq_{|E_2}, \xi_{|E_2})$ belongs to $\mathcal{L}_2$. We can easily check that this weak concatenation is associative on sets of pomsets. We can now define our second kind of contexts.

Definition 2.2. *Let $\mathcal{L}$ be a language of pomsets and u be a pomset over Σ. The* context *of u w.r.t. $\mathcal{L}$ is $\mathcal{L}\backslash u = \{(x, y) \in \mathbb{P}(\Sigma)^2 \mid (\{x\} \diamond \{u\} \diamond \{y\}) \cap \mathcal{L} \neq \emptyset\}$. Then u and v are* context equivalent *w.r.t. $\mathcal{L}$, and we write $u \asymp^{\mathcal{L}} v$, if $\mathcal{L}\backslash u = \mathcal{L}\backslash v$.*

We want now to compare these two equivalence relations. In the following lemma, we denote by $\mathrm{OE}(\mathcal{L}\backslash v)$ the set of pairs of pomsets (x, y) such that there are some pomsets x' and y' satisfying $x \in \mathrm{OE}(x')$, $y \in \mathrm{OE}(y')$, and $(x', y') \in \mathcal{L}\backslash v$.

Lemma 2.3. *Let $\mathcal{L}$ be a language of pomsets and u be a pomset over Σ. Then*

$$\mathcal{L}\backslash_{\mathrm{w}} u = \bigcup_{v : u \in \mathrm{OE}(v)} \mathrm{OE}(\mathcal{L}\backslash v).$$

Proof. Consider first $(x, y) \in \mathcal{L}\backslash_{\mathrm{w}} u$. Then $x \cdot u \cdot y$ is an order extension of some pomset $t \in \mathcal{L}$. Therefore there are three pomsets x', u', and y' such that

$x \in \mathrm{OE}(x')$, $u \in \mathrm{OE}(u')$, $y \in \mathrm{OE}(y')$, and $t \in \{x'\} \diamond \{u'\} \diamond \{y'\}$. Thus $(x', y') \in \mathcal{L}\backslash u'$ and $(x, y) \in \mathrm{OE}(\mathcal{L}\backslash u')$. Conversely, assume now that $(x, y) \in \mathrm{OE}(\mathcal{L}\backslash v)$ for some pomset v which admits u as order extension. Then there are some pomsets x' and y' satisfying $x \in \mathrm{OE}(x')$, $y \in \mathrm{OE}(y')$, and $(x', y') \in \mathcal{L}\backslash v$. Thus $\{x'\} \diamond \{v\} \diamond \{y'\} \cap \mathcal{L}$ is not empty. Let t belong to this set. Clearly $x \cdot u \cdot y$ is an order extension of t, i.e. $(x, y) \in \mathcal{L}\backslash_{\mathrm{w}} u$. □

As an immediate consequence, we observe that the equivalence relation $\asymp^{\mathcal{L}}$ is more restrictive than $\asymp^{\mathcal{L}}_{\mathrm{w}}$.

Theorem 2.4. *Let $\mathcal{L}$ be a language of pomsets. If $\asymp^{\mathcal{L}}$ is of finite index then $\asymp^{\mathcal{L}}_{\mathrm{w}}$ is of finite index, too.*

Example 2.10 below will show that the converse fails: It may happen that $\asymp^{\mathcal{L}}_{\mathrm{w}}$ is of finite index while $\asymp^{\mathcal{L}}$ has infinitely many equivalence classes. A second basic relationship between these two kinds of contexts is obtained as follows.

Lemma 2.5. *Let $\mathcal{L}$ be a language of pomsets and u be a pomset over Σ. Then $\mathcal{L}\backslash_{\mathrm{w}} u = \mathrm{OE}(\mathcal{L})\backslash u$.*

Proof. Assume first that $(x, y) \in \mathcal{L}\backslash_{\mathrm{w}} u$. Then $t = x \cdot u \cdot y$ is a linear extension of some pomset of $\mathcal{L}$; it also belongs obviously to $\{x\} \diamond \{u\} \diamond \{y\}$ hence $(x, y) \in \mathrm{OE}(\mathcal{L})\backslash u$. Conversely, if $(x, y) \in \mathrm{OE}(\mathcal{L})\backslash u$ then there is some pomset v in $\{x\} \diamond \{u\} \diamond \{y\} \cap \mathrm{OE}(\mathcal{L})$; furthermore $x \cdot u \cdot y$ is an order extension of v: It belongs to $\mathrm{OE}(\mathcal{L})$ and consequently $(x, y) \in \mathcal{L}\backslash_{\mathrm{w}} u$. □

This shows that for any weak language $\mathcal{L}$, $\asymp^{\mathcal{L}} = \asymp^{\mathcal{L}}_{\mathrm{w}}$. In particular, for word languages these two extensions are equivalent and correspond actually to the usual context equivalence for words. In the sequel, we choose the finer approach and focus on the weak concatenation.

Definition 2.6. *A language of pomsets $\mathcal{L}$ is* regular *if $\asymp^{\mathcal{L}}$ is of finite index.*

Consider a weak language $\mathcal{L}$. By Lemma 2.5, $\asymp^{\mathrm{Basis}(\mathcal{L})}_{\mathrm{w}} = \asymp^{\mathcal{L}}$. Now if $\mathrm{Basis}(\mathcal{L})$ is regular then Theorem 2.4 asserts that $\asymp^{\mathrm{Basis}(\mathcal{L})}_{\mathrm{w}}$ is of finite index. In that way, we get the following result.

Corollary 2.7. *For all weak languages $\mathcal{L}$, if $\mathrm{Basis}(\mathcal{L})$ is regular then $\mathcal{L}$ is regular.*

Example 2.11 below will show that the converse fails: We exhibit a non-regular basic language $\mathcal{L}$ such that $\mathrm{OE}(\mathcal{L})$ is regular.

Regularity via Residues. Similarly to properties well-known for words or in monoids, we introduce now two notions of residue that will turn out to lead to an alternative definition for both $\asymp^{\mathcal{L}}$ and $\asymp^{\mathcal{L}}_{\mathrm{w}}$. This simpler approach will help us to present our counter-examples for the converses of Theorem 2.4 and Corollary 2.7 and to establish some closure properties of regularity (Prop. 2.13).

Definition 2.8. *Let $\mathcal{L}$ be a class of pomsets and u be a pomset over Σ.*

1. *The* weak residue *of u w.r.t. $\mathcal{L}$ is $\mathcal{L}/_{\mathrm{w}}\, u = \{y \in \mathbb{P}(\Sigma) \mid u \cdot y \in \mathrm{OE}(\mathcal{L})\}$. We say that u and v are* weak residue equivalent *w.r.t. $\mathcal{L}$ if $\mathcal{L}/_{\mathrm{w}}\, u = \mathcal{L}/_{\mathrm{w}}\, v$. In that case we write $u \dot{\asymp}_{\mathrm{w}} v$.*
2. *The* residue *of u w.r.t. $\mathcal{L}$ is $\mathcal{L}/u = \{y \in \mathbb{P}(\Sigma) \mid (\{u\} \diamond \{y\}) \cap \mathcal{L} \neq \emptyset\}$. We say that u and v are* residue equivalent *w.r.t. $\mathcal{L}$, and we write $u \dot{\asymp} v$, if $\mathcal{L}/u = \mathcal{L}/v$.*

Note that if u is not a prefix of $\mathcal{L}$ then the residue $\mathcal{L}/u$ is empty. So all pomsets that are not prefixes of some pomset of $\mathcal{L}$ are residue equivalent. Clearly, if $\asymp$ (resp. $\asymp_{\mathrm{w}}$) is of finite index then $\dot{\asymp}$ (resp. $\dot{\asymp}_{\mathrm{w}}$) is of finite index, too. We show here that the converse holds.

Theorem 2.9. *Let $\mathcal{L}$ be a language of pomsets.*

1. *$\asymp$ is of finite index if and only if $\dot{\asymp}$ is of finite index.*
2. *$\asymp_{\mathrm{w}}$ is of finite index if and only if $\dot{\asymp}_{\mathrm{w}}$ is of finite index.*

Proof. We assume that $\dot{\asymp}$ is of finite index. For all $u \in \mathbb{P}(\Sigma)$, we define the equivalence relation $\equiv_u$ as follows: $v \equiv_u w$ if for all pomsets $z \in \mathbb{P}(\Sigma)$,

$$\{u\} \diamond \{v\} \diamond \{z\} \cap \mathcal{L} \neq \emptyset \Longleftrightarrow \{u\} \diamond \{w\} \diamond \{z\} \cap \mathcal{L} \neq \emptyset.$$

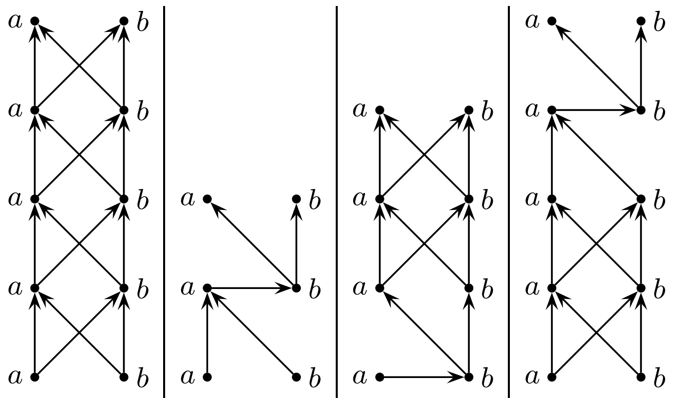
We observe that

$$\{u\} \diamond \{v\} \diamond \{z\} \cap \mathcal{L} \neq \emptyset \Leftrightarrow \exists x \in \{u\} \diamond \{v\} : \{x\} \diamond \{z\} \cap \mathcal{L} \neq \emptyset \Leftrightarrow z \in \cup_{x \in \{u\} \diamond \{v\}} \mathcal{L}/x$$

Thus $v \equiv_u w$ iff $\cup_{x \in \{u\} \diamond \{v\}} \mathcal{L}/x = \cup_{x \in \{u\} \diamond \{w\}} \mathcal{L}/x$. Since there are only a finite number of different contexts $\mathcal{L}/x$, there is a finite number of possible unions of such sets, which shows that $\equiv_u$ has finite index for each $u \in \mathbb{P}(\Sigma)$. Furthermore there is only a finite number of possible different equivalences $\equiv_u$. Now $v \asymp w$ iff for all pomsets $u \in \mathbb{P}(\Sigma)$, $v \equiv_u w$. Therefore $\asymp$ is the intersection of a finite number of equivalences of finite index, and has thus a finite index. The proof for $\asymp_{\mathrm{w}}$ is similar. □

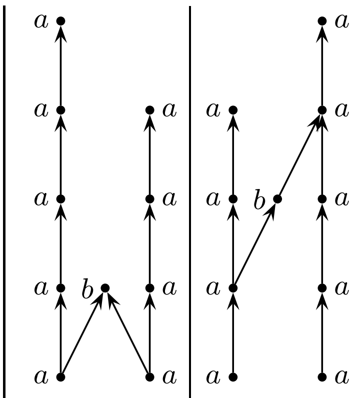
As shown in the sequel, this relationship provides us with simpler means for the study of regularity. It helps us in particular to exhibit some announced counter-examples (Theorem 2.4 and Cor. 2.7).

Example 2.10. We consider the language $\mathcal{L} \subseteq \mathcal{M}(\Sigma)^\star$ which consists of all iterations of the step $\{a, b\}$ such as pomset t_5 of Fig. 7. These pomsets have no autoconcurrency: The events labelled a (resp. b) are linearly ordered. We let $\mathcal{L}'$ be the language consisting of all pomsets t' build from any pomset $t \in \mathcal{L}$ by adding a single causality relation from a 2^n-th a-event to the corresponding 2^n-th b-event. Such pomsets are illustrated by t'_3, t'_4 and t'_5 on Fig. 7. Then $\mathcal{L}$ has only 4 distinct residues: $\emptyset$, $\mathcal{L}/a$, $\mathcal{L}/b$, and $\mathcal{L}/\{a, b\}$. Therefore $\asymp^{\mathcal{L}}$ and $\asymp^{\mathcal{L}}_{\mathrm{w}}$ are of finite index (Th. 2.9 and Th. 2.4). Since $\mathrm{OE}(\mathcal{L}) = \mathrm{OE}(\mathcal{L} \cup \mathcal{L}')$, $\asymp^{\mathcal{L}}_{\mathrm{w}} = \asymp^{\mathcal{L} \cup \mathcal{L}'}_{\mathrm{w}}$ is of finite index, too (Lemma 2.5). However we can easily see that $\mathcal{L} \cup \mathcal{L}'$ shows infinitely many residues, hence $\asymp^{\mathcal{L} \cup \mathcal{L}'}$ is of infinite index.



Pomset t_5 | Pomset t'_3 | Pomset t'_4 | Pomset t'_5

Fig. 7. Pomsets of Example 2.10



Pomset t | Pomset t'

Fig. 8. Pomsets of Ex. 2.11

Example 2.11. We consider the subset $\mathcal{L}_0$ of all pomsets t that consist of two rows of a-events and an additional b-event. The language $\mathcal{L} \subset \mathcal{L}_0$ restricts to the pomsets such that either the b-event covers the first a of each row, or covers the k-th a-event of one row and is covered by the $2k$-th a-event of the other row. Examples of such pomsets are depicted in Fig. 8. We claim that $\mathcal{L}$ is basic and not regular. However, we observe that $\dot{\asymp}^{\mathrm{OE}(\mathcal{L})}$ has index 5: If t is not a prefix of some $t' \in \mathrm{OE}(\mathcal{L})$, then $\mathrm{OE}(\mathcal{L})/t = \emptyset$. Now $\mathrm{OE}(\mathcal{L})/\varepsilon = \mathrm{OE}(\mathcal{L})$, $\mathrm{OE}(\mathcal{L})/a = \mathrm{OE}(\mathcal{L}/a)$, and moreover for all pomsets t that are prefixes of $\mathcal{L}$ different from the empty pomset ε and the singleton pomset a, $\mathrm{OE}(\mathcal{L})/t$ depends only on whether b occurs in t. To be precise, let $\mathcal{L}_1$ be the language of all pomsets consisting of two rows of a. Then $\mathrm{OE}(\mathcal{L})/t$ equals $\mathrm{OE}(\mathcal{L}_1)$ if b occurs in t and $\mathrm{OE}(\mathcal{L}_0)$ otherwise.

We turn now our attention to some operations on pomsets languages that preserve regularity. First, the *parallel composition* of two pomsets $t_1 = (E_1, \preccurlyeq_1, \xi_1)$ and $t_2 = (E_2, \preccurlyeq_2, \xi_2)$ executes t_1 and t_2 independently: $t_1 \| t_2 = (E_1 \uplus E_2, \preccurlyeq_1 \uplus \preccurlyeq_2, \xi_1 \uplus \xi_2)$. This composition is naturally extended to languages as follows: $\mathcal{L}_1 \| \mathcal{L}_2 = \{t_1 \| t_2 \mid t_1 \in \mathcal{L}_1 \wedge t_2 \in \mathcal{L}_2\}$. Similarly, the strong concatenation of $\mathcal{L}_1$ and $\mathcal{L}_2$ is $\mathcal{L}_1 \cdot \mathcal{L}_2 = \{t_1 \cdot t_2 \mid t_1 \in \mathcal{L}_1 \wedge t_2 \in \mathcal{L}_2\}$. Now the *strong iteration* of $\mathcal{L}$ is simply $\mathcal{L}^\star = \bigcup_{k \in \mathbb{N}} \mathcal{L}^k$ where $\mathcal{L}^0$ consists of the empty pomset and $\mathcal{L}^{k+1} = \mathcal{L} \cdot \mathcal{L}^k$.

Lemma 2.12. *Let $\mathcal{L}$, $\mathcal{L}_1$ and $\mathcal{L}_2$ be some pomset languages. Then for all pomsets $u \in \mathbb{P}(\Sigma)$, we have:*

1. $(\mathcal{L}_1 \cup \mathcal{L}_2)/u = \mathcal{L}_1/u \cup \mathcal{L}_2/u$
2. $(\mathcal{L}_1 \| \mathcal{L}_2)/u = \bigcup_{u = u_1 \| u_2} (\mathcal{L}_1/u_1 \| \mathcal{L}_2/u_2)$
3. $(\mathcal{L}_1 \cdot \mathcal{L}_2)/u = (\mathcal{L}_1/u) \cdot \mathcal{L}_2 \cup (\bigcup_{v : u \in \mathcal{L}_1 \cdot v} \mathcal{L}_2/v)$
4. $(\mathcal{L}^\star)/u = \bigcup_{i \in \mathbb{N}} (\bigcup_{v : u \in \mathcal{L}^i \cdot v} (\mathcal{L}/v).\mathcal{L}^\star)$

Proof. We only sketch the third point. Consider a residue $r \in (\mathcal{L}_1 \cdot \mathcal{L}_2)/u$. For some pomset $w = (E, \preccurlyeq, \xi)$, we have $w \in (\mathcal{L}_1 \cdot \mathcal{L}_2) \cap (\{u\} \diamond \{r\})$. Therefore $w = u_1 \cdot u_2 = (E_1 \uplus E_2, \preccurlyeq, \xi)$ where $u_i = (E_i, \preccurlyeq_{|E_i}, \xi_{|E_i})$ belongs to $\mathcal{L}_i$. Similarly, $E = E_u \uplus E_r$ and E_u is a prefix of w. Recall that $x \preccurlyeq y$ for all $x \in E_1$ and all $y \in E_2$ because $w = u_1 \cdot u_2$. Consequently, only two cases may occur:

1. $E_u \subseteq E_1$. Then $E_r = (E_1 \setminus E_u) \uplus E_2$ and $r \in (\mathcal{L}_1/u) \cdot \mathcal{L}_2$.
2. $E_1 \subseteq E_u$. Then $E_2 = (E_u \setminus E_1) \uplus E_r$ and $r \in \mathcal{L}_2/v$ for some pomset v such that $u \in \mathcal{L}_1 \cdot \{v\}$.

Thus $(\mathcal{L}_1 \cdot \mathcal{L}_2)/u \subseteq (\mathcal{L}_1/u) \cdot \mathcal{L}_2 \cup (\bigcup_{v:u \in \mathcal{L}_1 \cdot v} \mathcal{L}_2/v)$. We check now that the opposite inclusion holds. Assume that $r \in (\mathcal{L}_1/u) \cdot \mathcal{L}_2$. Then $r = v \cdot u_2$ with $v \in \mathcal{L}_1/u$ and $u_2 \in \mathcal{L}_2$. Moreover $\{u\} \diamond \{v\} \cap \mathcal{L}_1 \neq \emptyset$ hence $((\{u\} \diamond \{v\}) \cdot \{u_2\}) \cap (\mathcal{L}_1 \cdot \mathcal{L}_2) \neq \emptyset$. Now $(\{u\} \diamond \{v\}) \cdot \{u_2\} \subseteq \{u\} \diamond \{v \cdot u_2\}$. Therefore $\{u\} \diamond \{r\} \cap (\mathcal{L}_1 \cdot \mathcal{L}_2) \neq \emptyset$. Assume now that $r \in \mathcal{L}_2/v$ for some pomset v such that $u \in \mathcal{L}_1 \cdot v$, i.e. $u = u_1 \cdot v$ for some $u_1 \in \mathcal{L}_1$. Since $\{v\} \diamond \{r\} \cap \mathcal{L}_2 \neq \emptyset$, we have $(\{u_1\} \cdot (\{v\} \diamond \{r\})) \cap \mathcal{L}_1 \cdot \mathcal{L}_2 \neq \emptyset$. Now $\{u_1\} \cdot (\{v\} \diamond \{r\}) \subseteq \{u_1 \cdot v\} \diamond \{r\} = \{u\} \diamond \{r\}$. Therefore $\{u\} \diamond \{r\} \cap (\mathcal{L}_1 \cdot \mathcal{L}_2) \neq \emptyset$ in both cases, i.e. $r \in (\mathcal{L}_1 \cdot \mathcal{L}_2)/u$. □

Immediate consequences of these observations, we establish the following closure properties.

Proposition 2.13. *The class of regular languages is closed under union, strong concatenation, strong iteration, and parallel composition.*

Consequently, the class of weak regular languages is also closed under union, strong concatenation, and strong iteration. However, the parallel composition of two non trivial weak languages is never weak. Moreover the class of regular languages is not closed under intersection nor complementation.

Comparison with SP-recognizability. The algebra of series-parallel pomsets on the alphabet Σ, denoted $SP(\Sigma)$, is the subset of $\mathbb{P}(\Sigma)$ generated by the singleton pomsets, identified with Σ, and the strong concatenation and parallel composition on pomsets. Note that the empty pomset does not belong to $SP(\Sigma)$. It is known [7] that $SP(\Sigma)$ is isomorphic to the quotient of the free term algebra $T(\Sigma, \bullet, ||)$ by the following set of equations :

1. Associativity: $t \bullet (t' \bullet t'') =_e (t \bullet t') \bullet t''$ and $t \parallel (t' \parallel t'') =_e (t \parallel t') \parallel t''$
2. Commutativity: $t \parallel t' =_e t' \parallel t$

We denote by $SP_k(\Sigma) \subseteq SP(\Sigma)$ the set of series-parallel pomsets (SP-pomsets for short) of width bounded by k, i.e. such that any antichain has at most k events. In the following, we denote in the same way a term of the initial algebra $T(\Sigma, \bullet, ||)$ and the unique associated pomset in $SP(\Sigma)$. We call SP-algebra any quotient of $SP(\Sigma)$, i.e. any finitely generated model of $T(\Sigma, \bullet, ||)/ =_e$.

Let α be a symbol not in Σ. We define the set of unary SP-contexts $Ctxt(\alpha)$ as $Ctxt(\alpha) = \{C \in SP(\Sigma \cup \{\alpha\}) : |C|_\alpha = 1\}$, i.e. the set of SP-pomsets labelled on $\Sigma \cup \{\alpha\}$ (or SP-terms on $\Sigma \cup \{\alpha\}$) such that a single event (a single leaf of the term) is labelled by α. If $C \in Ctxt(\alpha)$ and $u \in SP(\Sigma)$, we denote by $C(u) \in SP(\Sigma)$ the pomset resulting from the substitution of α by u in C. Let $\mathcal{L} \subseteq SP(\Sigma)$ and $u \in SP(\Sigma)$, the *set of contexts of u in $\mathcal{L}$* is $\mathcal{L} \backslash_\alpha u = \{C \in Ctxt(\alpha) : C(u) \in \mathcal{L}\}$. The *syntactic SP-congruence of $\mathcal{L}$* is the equivalence on $SP(\Sigma)$ defined by $u \sim_{\mathcal{L}} v \iff \mathcal{L} \backslash_\alpha u = \mathcal{L} \backslash_\alpha v$.

1. A subset $\mathcal{L} \subseteq SP(\Sigma)$ is *SP-recognizable* if there is a SP-homomorphism $\sigma : SP(\Sigma) \longrightarrow B$ on a finite SP-algebra B such that $\mathcal{L} = \sigma^{-1}(\sigma(\mathcal{L}))$.
2. Equivalently $\mathcal{L} \subseteq SP(\Sigma)$ is SP-recognizable if the *syntactic SP-congruence of $\mathcal{L}$ is of finite index.*

Note that *SP-recognizability* coincide for languages of linear pomsets $\mathcal{L} \subseteq \Sigma^\star$ with the standard definition of recognizability/regularity for word languages.

Lodaya and Weil showed in [11] that *bounded width SP-pomset languages are SP-recognizable iff they are series-rational*, i.e. constructed from singletons by union, strong concatenation, strong iteration, and parallel composition. Proposition 2.13 asserts that regular languages are closed under these operations. Therefore series-rational languages are regular, and thus bounded width SP-recognizable languages are regular. We see now that this inclusion is strict.

Proposition 2.14. *The class of bounded width SP-recognizable SP-pomset languages is* strictly included *in the class of regular languages of SP-pomsets.*

Proof. To show that the inclusion is strict, we consider a non-recognizable word language $\mathcal{L} \subseteq \Sigma^\star$ and show that for all $k \geqslant 2$ the complement language $N = SP_k(\Sigma) \setminus \mathcal{L}$ is regular but not SP-recognizable.

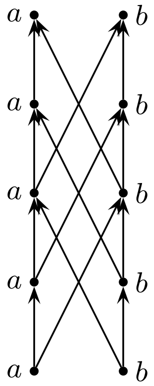
First $N = SP_k(\Sigma) \setminus \mathcal{L}$ is regular. For all pomsets $u \in SP(\Sigma) \setminus SP_k(\Sigma)$, we have $N/u = \emptyset$ because $\forall v \in SP(\Sigma)\ \{u\} \diamond \{v\} \subseteq SP(\Sigma) \setminus SP_k(\Sigma)$. For all pomsets $u \in SP_k(\Sigma)$, we have $N/u = SP_k(\Sigma)$ because for all $v \in SP_k(\Sigma)$ either $u \| v \in N$ or $u \cdot v \in N$.

For all $u \in SP(\Sigma) \setminus \Sigma^\star$, there is no SP-context $C \in Ctxt(\alpha)$ such that $C(u) \in \mathcal{L}$. Thus $N \backslash_\alpha u = Ctxt(\alpha)$: All $u \in SP(\Sigma) \setminus \Sigma^\star$ are SP-equivalent w.r.t. N. If $u \in \Sigma^\star$ then for all $v \in \Sigma^\star$ we have $(\alpha \cdot v)(u) \in N$ iff $v \notin \mathcal{L}/u$ i.e. $\alpha \cdot v \in N \backslash_\alpha u$ iff $v \notin \mathcal{L}/u$. Since $\mathcal{L}$ is not recognisable, there is an infinite number of sets $\Sigma^\star \setminus \mathcal{L}/u$, and consequently an infinite number of sets $N \backslash_\alpha u$. Therefore N is not SP recognisable. □

3 Recognizable Compatible Languages

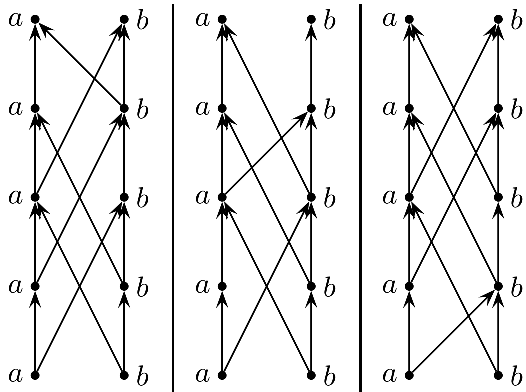
In this section we compare the properties of a pomset language with those of its step extensions. We aim at relating the regularity of a language $\mathcal{L}$ to the usual property of recognizability of its step extensions $\mathrm{SE}(\mathcal{L})$ within the monoid $\mathcal{M}(\Sigma)^\star$. Recall that a subset $\mathcal{L}$ of a monoid $\mathbb{M}$ is called *recognizable* if there exists a finite monoid $\mathbb{M}'$ and a monoid morphism $\eta : \mathbb{M} \rightarrow \mathbb{M}'$ such that $\mathcal{L} = \eta^{-1} \circ \eta(\mathcal{L})$. Equivalently, $\mathcal{L}$ is recognizable if, and only if, the collection of all sets $\mathcal{L}/x = \{y \in \mathbb{M} \mid x \cdot y \in \mathcal{L}\}$ is finite. In particular the set of recognizable subsets of any monoid is closed under union, intersection and complement.

Since $\mathrm{SE}(\mathcal{L})$ is preserved by order extensions, we shall consider weak languages only. We will easily observe that any weak regular language $\mathcal{L}$ describes a recognizable language $\mathrm{SE}(\mathcal{L})$ of $\mathcal{M}(\Sigma)^\star$. However two detailed examples will show that the converse fails. As main result, we prove that regularity corresponds to recognizability for compatible sets of pomsets (Def. 1.5 and Th. 3.5).



Pomset t_5

Fig. 9.



Pomset t' Pomset t'' Pomset t'''

Fig. 10. Pomsets of $\mathcal{L}'$ in Ex. 3.4

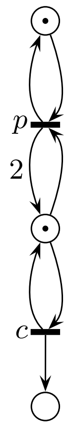


Fig. 11.

Towards an Algebraic Recognizability. Our first stage from regularity to recognizability of step extensions is the restriction of contexts to step sequences.

Definition 3.1. *Let $\mathcal{L}$ be a class of pomsets and u be a pomset over Σ. The* step context *of u w.r.t. $\mathcal{L}$ is $\mathcal{L}\backslash_{\mathrm{s}} u = \{(x, y) \in \mathcal{M}(\Sigma)^{\star 2} \mid x \cdot u \cdot y \in \mathrm{OE}(\mathcal{L})\}$. We say that u and v are* step equivalent*, and we write $u \asymp_{\mathrm{s}} v$, if $\mathcal{L}\backslash_{\mathrm{s}} u = \mathcal{L}\backslash_{\mathrm{s}} v$.*

Clearly if $\asymp_{\mathrm{w}}$ is of finite index then $\asymp_{\mathrm{s}}$ is of finite index, too. As the next example will show, the converse fails even for weak languages. To see this, we use again an equivalent approach by means of residues: Similarly to the previous section, the *step residue* of u w.r.t. $\mathcal{L}$ is $\mathcal{L}/_{\mathrm{s}}\, u = \{y \in \mathcal{M}(\Sigma)^{\star} \mid u \cdot y \in \mathrm{OE}(\mathcal{L})\}$ and two pomsets u and v are called *step residue equivalent*, written $u \dot{\asymp}_{\mathrm{s}} v$, if $\mathcal{L}/_{\mathrm{s}}\, u = \mathcal{L}/_{\mathrm{s}}\, v$. Similarly to Theorem 2.9, we can prove that $\asymp_{\mathrm{s}}$ is of finite index if and only if $\dot{\asymp}_{\mathrm{s}}$ is of finite index.

Example 3.2. We denote by t_n the pomset which consists of n linearly ordered a-events and n linearly ordered b-events such that the k-th event a is below the $(k + 2)$-th event b and symmetrically the k-th event b is below the $(k + 2)$-th event a, as shown on Fig. 9. For $n \geqslant 2$, we denote by t'_n the order extension of t_n with a single additional causality relation, from the $(n - 1)$-th event a to the n-th event b. Similarly we let t''_n be the order extension of t_n with a single additional causality relation, from the $(n - 1)$-th event b to the n-th event a. Now consider the language $\mathcal{L}$ which consists of all t_n for $n \neq 2^k$ together with all t'_{2^k} and all t''_{2^k}. We claim first that $\mathcal{L}$ is not regular because there are infinitely many distinct weak residues $\mathcal{L}/_{\mathrm{w}}\, t_n$. However $\asymp_{\mathrm{s}}^{\mathcal{L}}$ is of finite index. Note also that this gap appears for $\mathrm{OE}(\mathcal{L})$ as well, because $\asymp_{\mathrm{w}}^{\mathcal{L}} = \asymp_{\mathrm{w}}^{\mathrm{OE}(\mathcal{L})}$ and $\asymp_{\mathrm{s}}^{\mathcal{L}} = \asymp_{\mathrm{s}}^{\mathrm{OE}(\mathcal{L})}$.

We can also use these step residues to prove easily that any regular language is recognizable.

Lemma 3.3. *If $\asymp_{\mathrm{s}}$ is of finite index then $\mathrm{SE}(\mathcal{L})$ is recognizable in the monoid $\mathcal{M}(\Sigma)^{\star}$.*

Proof. We observe that $\mathrm{SE}(\mathcal{L})/u = \mathcal{L}/_{\mathrm{s}}\, u \cap \mathcal{M}(\Sigma)^{\star}$ for all $u \in \mathcal{M}(\Sigma)^{\star}$. □

As the next example shows, the converse fails even for weak languages.

EXAMPLE 3.4. We consider again the pomsets t_n of Example 3.2 and Figure 9. We let $\mathcal{L}'$ be the language consisting of all pomsets t' that are order extensions of some t_n with a single additional causality, either from a 2^k-th a-event to the corresponding (2^k+1)-th b-event, or from a 2^k-th b-event to the corresponding (2^k+1)-th a-event. Examples of such pomsets are depicted on Fig. 10. Consider the language $\mathcal{L}$ of all t_n. For all prefixes t of t_n, $-2 \leqslant |t|_a - |t|_b \leqslant 2$ and this number determines $\mathcal{L}/t$. Consequently $\mathcal{L}$ has 6 residues (including $\emptyset$) and $\asymp^{\mathcal{L}}$ and $\asymp_{\mathrm{s}}^{\mathcal{L}}$ are of finite index (Th. 2.9 and Th. 2.4). Therefore $\mathrm{SE}(\mathcal{L})$ is recognizable in the monoid $\mathcal{M}(\Sigma)^{\star}$ (Lemma 3.3). We observe also that $\mathrm{SE}(\mathcal{L}) = \mathrm{SE}(\mathcal{L}')$. However we claim that $\asymp_{\mathrm{s}}^{\mathcal{L}'}$ is of infinite index. Thus $\mathrm{SE}(\mathcal{L}')$ is recognizable in the monoid $\mathcal{M}(\Sigma)^{\star}$ but $\asymp_{\mathrm{s}}^{\mathcal{L}'}$ is of infinite index. Furthermore $\mathcal{L}'' = \mathrm{OE}(\mathcal{L}')$ is a weak language such that $\asymp_{\mathrm{s}}^{\mathcal{L}''} = \asymp_{\mathrm{s}}^{\mathcal{L}'}$ is also of infinite index while $\mathrm{SE}(\mathcal{L}'') = \mathrm{SE}(\mathcal{L}')$ is a recognizable language of $\mathcal{M}(\Sigma)^{\star}$.

Characterization of Regular Compatible Languages. Let $\mathcal{L}$ be a weak language of pomsets. We have observed that the following implications hold:

$\mathcal{L}$ is regular $\Longrightarrow$ $\asymp_{\mathrm{s}}^{\mathcal{L}}$ is of finite index $\Longrightarrow$ $\mathrm{SE}(\mathcal{L})$ is recognizable in $\mathcal{M}(\Sigma)^{\star}$ but none of the converses holds (Examples 3.2 and 3.4). We show now that this gap between regularity and recognizability vanishes for compatible languages.

THEOREM 3.5. *Let $\mathcal{L}$ be a* compatible *language of pomsets. The following are equivalent:*

(i) *$\mathcal{L}$ is regular.*
(ii) *$\asymp_{\mathrm{s}}^{\mathcal{L}}$ is of finite index.*
(iii) *$\mathrm{SE}(\mathcal{L})$ is recognizable in the monoid $\mathcal{M}(\Sigma)^{\star}$.*

Proof. Since $\mathcal{L}$ is weak, $\asymp^{\mathcal{L}} = \asymp_{\mathrm{w}}^{\mathcal{L}}$. By Lemma 3.3, we need just to show that (iii) implies (i). Since $\mathcal{L}$ is compatible, for all pomsets u and r,

$$u \cdot r \in \mathcal{L} \Longleftrightarrow \mathrm{SE}(u \cdot r) \subseteq \mathrm{SE}(\mathcal{L}) \Longleftrightarrow \mathrm{SE}(u) \cdot \mathrm{SE}(r) \subseteq \mathrm{SE}(\mathcal{L})$$

It follows that $\mathcal{L}/u = \overline{\mathcal{L}_u}$ where $\mathcal{L}_u = \bigcap_{v \in \mathrm{SE}(u)} \mathrm{SE}(\mathcal{L})/v$. Consequently, if $\mathrm{SE}(\mathcal{L})$ is recognizable then there are only finitely many residues $\mathcal{L}/u$. □

This result shows that the duality between compatible languages and their corresponding step extensions identifies our notion of regularity with the algebraic recognizability in $\mathcal{M}(\Sigma)^{\star}$. As a consequence, we notice that our notion of regularity corresponds to the one considered for local trace languages [10].

Other Comparisons with Related Works. Recall first that $\mathrm{LE}(\mathcal{L})$ is regular as soon as $\mathcal{L}$ is regular because $\mathrm{LE}(\mathcal{L})/u = (\mathrm{SE}(\mathcal{L})/u) \cap \Sigma^{\star}$ for all words $u \in \Sigma^{\star}$. As the next example shows, the converse fails even for compatible sets of pomsets.

Example 3.6. Consider the compatible set $\mathcal{L}$ of all firing pomsets of the Petri net depicted in Fig. 11. Since each place always contains at least one token, each transition is permanently enabled. Therefore $\mathrm{LE}(\mathcal{L}) = \{p, c\}^\star$ is regular. Now, transition p produces a new token at each occurrence. Therefore the step $c^n = \{c, ..., c\}$ is enabled after a firing sequence s only if s contains at least $n - 1$ occurrences of p. Consequently $\mathrm{SE}(\mathcal{L})$ and $\mathcal{L}$ are not regular.

In practice however, it is often the case that regularity of $\mathrm{LE}(\mathcal{L})$ is equivalent to regularity of $\mathcal{L}$, as the next result shows.

Proposition 3.7. *Let $\mathcal{L}$ be a language of pomsets. We assume that for all pomsets t_1 and t_2 such that t_1 is a prefix of some pomset of $\mathcal{L}$ and t_2 is a postfix of some pomset of $\mathcal{L}$, the following requirement is satisfied:*

$$(\forall u_1 \in \mathrm{LE}(t_1), \forall u_2 \in \mathrm{LE}(t_2), u_1.u_2 \in \mathrm{LE}(\mathcal{L})) \Longrightarrow (\{t_1\} \diamond \{t_2\}) \cap \mathcal{L} \neq \emptyset$$

If $\mathrm{LE}(\mathcal{L})$ is recognizable in the free monoid $\Sigma^\star$ then $\mathcal{L}$ is regular.

This result applies to several frameworks of compatible languages: It shows in particular that for Mazurkiewicz traces and messages sequence charts, our notion of regularity corresponds precisely to the notion of regularity defined in these frameworks. As a consequence, for the particular case of message sequence charts, the notion regularity studied in this paper is stronger than the algebraic recognizability studied in [12].

Conclusion

In this paper we introduced a new general notion of regularity for pomset languages with autoconcurrency. We showed how this approach extends similar notions in classical frameworks of concurrency theory, such as Mazurkiewicz traces and message sequence charts. However a mismatch between regularity and algebraic recognizability appears in some particular cases such as series-parallel pomsets and message sequence charts. We showed that the particular class of compatible languages enjoys several nice properties and includes many formalisms from the literature. We believe that these languages are a good candidate for further generalizations, in particular for a connection between regularity and MSO-definability.

Acknowledgment

We thank Dietrich Kuske for his useful observation that regular pomset languages are not closed under complementation.

References

[1] R. Alur and M. Yannakakis: *Model Checking of Message Sequence Charts.* CONCUR'99, LNCS **1664** (1999) 114–129 403, 406

[2] A. Arnold: *An extension of the notion of traces and asynchronous automata.* Theoretical Informatics and Applications **25** (1991) 355–393 403, 406

[3] V. Diekert and Y. Métivier: *Partial Commutations and Traces.* Handbook of Formal languages, vol. **3** (1997) 457–533 402, 403, 406

[4] M. Droste, P. Gastin, and D. Kuske: *Asynchronous cellular automata for pomsets.* Theoretical Computer Science Vol. **247** (2000) 1–38 402

[5] M. Droste and D. Kuske: *Logical definability of recognizable and aperiodic languages in concurrency monoids.* LNCS **1092** (1996) 233–251 402, 403, 406

[6] J. Fanchon: *A syntactic congruence for the recognizability of pomset languages.* RR 99008 (LAAS, Toulouse, 1999) 402

[7] J. L. Gischer: *The equational theory of pomsets.* Theoretical Comp. Science **61** (1988) 199–224 402, 412

[8] J. Grabowski: *On partial languages.* Fund. Informatica **IV(2)** (1981) 427–498 403, 405

[9] D. Kuske: *Infinite series-parallel posets: logic and languages.* LNCS **1853** (2000) 648–662 402

[10] D. Kuske and R. Morin: *Pomsets for Local Trace Languages — Recognizability, Logic & Petri Nets.* CONCUR 2000, LNCS **1877** (2000) 426–440 403, 406, 415

[11] K. Lodaya and P. Weil: *Series-parallel languages and the bounded-width property.* Theoretical Comp. Science **237** (2000) 347–380 402, 403, 413

[12] R. Morin: *Recognizable Sets of Message Sequence Charts.* STACS 2002, LNCS **2030** (2002) 332–342 402, 416

[13] M. Mukund, K. Narayan Kumar, and M. Sohoni: *Synthesizing distributed finite-state systems from MSCs.* CONCUR 2000, LNCS **1877** (2000) 521–535 403, 406

[14] V. Pratt: *Modelling concurrency with partial orders.* Int. J. of Parallel Programming **15** (1986) 33–71 402

[15] W. Thomas: *Automata Theory on Trees and Partial Orders.* TAPSOFT 97, LNCS **1214** (1998) 20–34 402

[16] W. Vogler: *Modular Construction and Partial Order Semantics of Petri Nets.* LNCS **625** (1992) 252–275 403, 405