Projection of Rational Pomset Languages

Thomas Gazagnaire [1]

Citrix Systems R&D Ltd.
Black Horse House, Castle Park
Cambridge CB3 0BL, United Kingdom

Key words: concurrency - formal languages - partial orders - pomsets - projection - rational languages - finitely-generated languages

1 Introduction

Over the last few decades, partial-order models have become more and more popular for specifying and verifying concurrent and distributed systems. These models are intuitive and very expressive. They have been used to describe concurrent systems communicating either by shared memory [15,3], or by (a)synchronous message-passing [19,14], as well as more general concurrent systems [22,10]. The trade-off from expressiveness is that verification of partial-order models is much more difficult, as model-checking against partial-order properties [1,18] or logical formulas [20] is not decidable in general. However, interesting classes of partial-order models have been defined, combining good expressiveness while keeping verification problems decidable [18,9,8,5]. Moreover, when considering model-checking of sequential models, abstraction techniques [21] are commonly used. Indeed, they decrease the size of the model and thus speed-up model-checking algorithms whose complexities are usually exponential with respect to the model size. *Exact* abstraction techniques do not add behaviors inconsistent with the initial model. These techniques are essential for counterexample refinement based on abstraction [2,11].

In this paper we focus on exact abstraction techniques for partial-order models based on rational pomset expressions (which subsume rational Mazurkiewicz traces languages and HMSCs). We extend the work of Genest, Hélouët and Muscholl [7] which show that it is decidable to know whether the projected language of an HMSC is still the language of an HMSC. However, they use a direct construction based on the existential boundness of HMSC channels. As pomsets allow auto-concurrency, it seems difficult to translate this technique directly to rational pomset expressions. Thus we adopt a rather indirect

Email address: thomas.gazagnaire@citrix.com (Thomas Gazagnaire).

[1] Part of this work was done while the author was at IRISA - ENS Cachan, France.

approach by cleaving this problem into two parts: (i) showing that it is decidable to know whether the projected language of a rational pomset expression is finitely generated (finitely-generated languages of HMSCs have been studied in [13]); and (ii) characterizing recognizable languages of pomsets ([4] studies regular languages of pomsets and [18,17] characterize recognizable languages of Mazurkiewicz trace and MSCs). To glue these two parts together, one can apply the McKnight theorem[16], which states that any recognizable language which is finitely generated is also rational. We focus in this paper on (i) only. First, we introduce the model of pomsets in Section 2. Then, we define the boxed-pomset model in Section 3 and give its basic properties in Section 4. Finally in Section 5, we tackle part (i) using boxed pomsets.

2 Pomsets

First of all, let us recall some well-known definitions related to formal languages. A monoid is a structure $(\mathbb{M}, *, 1)$ where $\mathbb{M}$ is a set of elements and the operator $* : (\mathbb{M} \times \mathbb{M}) \to \mathbb{M}$ is an associative composition law with 1 being its neutral element. As usual, the set of rational expressions of $\mathbb{M}$, denoted by $REX(\mathbb{M})$, is defined by induction as the set of symbols $\emptyset \cup \mathbb{M} \cup \{\alpha\beta\} \cup \{\alpha + \beta\} \cup \{\alpha^\star\}$, where $\alpha, \beta \in REX(\mathbb{M})$. The language $\mathcal{L}_{\mathbb{M}}$ of a rational expression is a function from $REX(\mathbb{M})$ to $\mathbb{M}$ defined by induction as the following: $\mathcal{L}_{\mathbb{M}}(\emptyset) = 1$, for any $m \in \mathbb{M}$, $\mathcal{L}_{\mathbb{M}}(m) = \{m\}$. For any $\alpha, \beta \in REX(\mathbb{M})$, we have $\mathcal{L}_{\mathbb{M}}(\alpha\beta) = \{m * n \mid m \in \mathcal{L}_{\mathbb{M}}(\alpha), n \in \mathcal{L}_{\mathbb{M}}(\beta)\}$ and $\mathcal{L}_{\mathbb{M}}(\alpha + \beta) = \mathcal{L}_{\mathbb{M}}(\alpha) \cup \mathcal{L}_{\mathbb{M}}(\beta)$. Furthermore, we have $\mathcal{L}_{\mathbb{M}}(\alpha^n) = \{m * n \mid m \in \mathcal{L}_{\mathbb{M}}(\alpha), n \in \mathcal{L}_{\mathbb{M}}(\alpha^{n-1})\}$, for any $n > 1$ (and $\alpha^1 = \alpha$). Finally $\mathcal{L}_{\mathbb{M}}(\alpha^\star)$ is defined as $1 \bigcup_{n \geq 1} \mathcal{L}_{\mathbb{M}}(\alpha^n)$.

Furthermore, through the rest of the paper, we fix an infinite set $\mathbb{E}$ of events and a non-empty set Σ of labels. We also fix a mapping $\lambda : \mathbb{E} \to \Sigma$ which assigns to each event a given label. A *labeled partial order* (or simply an *lpo*) over Σ and λ is a structure $(E, \leq)$ where $E \subset \mathbb{E}$ and $\leq \subseteq E \times E$ partially orders E, ie. $\leq$ is reflexive, antisymmetric and transitive. We denote by $(E_p, \leq_p)$ the components of the pomset p and by $\preccurlyeq_p$ the transitive reduction of $\leq_p$. An *isomorphism* of lpo f: $(E_1, \leq_1) \to (E_2, \leq_2)$ is a bijective map which preserves labels and ordering, ie. such that for all $e, e' \in E_1$, $e \leq_1 e' \Leftrightarrow f(e) \leq_2 f(e')$ and $\lambda(e) = \lambda(f(e))$. The isomorphism class of an lpo $(E, \leq)$ is called a *pomset* ([22,10]) and is denoted by $[E, \leq]$. Intuitively a pomset is an lpo in which we pay no attention to the choice of the set E, other than its labels and their respective ordering. We denote by $\mathbb{P}$ the set of all possible pomsets and by $[E_p, \leq_p]$ the components of the pomset p. When $\leq_p = \emptyset$, we simply write $[E_p]$. The pomset model is a natural extension of strings and partial orders, which can be seen as pomsets having all theirs events totally ordered and as pomsets with a one-to-one labelling.

We denote by $(p \setminus E)$ the restriction of the pomset p to events which are not in E, ie. $E_{(p \setminus E)} = (E_p \setminus E)$ and $\leq_{(p \setminus E)} = \leq_p \cap (E_p \setminus E)^2$. The *projection* of a pomset p on an alphabet Γ is a function $\pi_\Gamma : \mathbb{P} \to \mathbb{P}$ which restricts p to its observable labels, i.e. $\pi_\Gamma(p) = (p \setminus \lambda^{-1}(\Sigma \setminus \Gamma))$. From now on, we fix a relation $R \subseteq \Sigma \times \Sigma$ (we do not assume any properties for R, so it might not be reflexive or symmetric). We define the *composition* of the pomsets p_1 and p_2, denoted by $p_1 \circ p_2$, as that augment the union of the two pomsets obtained by adding to the order all pairs $(e, e') \in E_{p_1} \times E_{p_2}$ such that $(\lambda(e), \lambda(e')) \in R$, and then taking the transitive closure of the result. More formally, we define $p_1 \circ p_2$ as $[E_{p_1} \uplus E_{p_2}, (\leq_{p_1} \cup \leq_{p_2} \cup \leq_R)^*]$ where $\leq_R = \{(e, e') \in E_{p_1} \times E_{p_2} \mid (\lambda(e), \lambda(e')) \in R)\}$. This composition extends some well-known composition laws as parallel composition where $R = \emptyset$, strong concatenation where $R = \Sigma \times \Sigma$, Mazurkiewicz trace composition [3] where R is a so-called dependency relation, that is a symmetric and reflexive relation, MSC sequential composition [14,19] where R is the disjoint union of total relations and Causal MSC composition [5] where R is a disjoint union of dependency relations. One can check that $\circ$ is associative, well-defined and admits the empty pomset $1_\mathbb{P}$ as neutral element. Thus, $\mathbb{P}$ is a a monoid and rational pomset expressions $REX(\mathbb{P})$ and their languages $\mathcal{L}_\mathbb{P}$ are well-defined.

Figure 1 gives an example of pomset composition and projection. Each event e is represented by its label $\lambda(e)$ only. Moreover, for convenience, only the transitive reduction of $\leq$ is drawn (also called its Hasse diagram). On this figure, we fix $\Sigma = \{a, b\}$, $R = \{(b, a)\}$ and $\Gamma = \{a\}$. Obviously, this example shows that $\pi_\Gamma(p_1 \circ p_1) \neq \pi_\Gamma(p_1) \circ \pi_\Gamma(p_1)$. We show in the following section that a slight extension of pomsets that we call *boxed pomsets* need to be used to satisfy this equality.

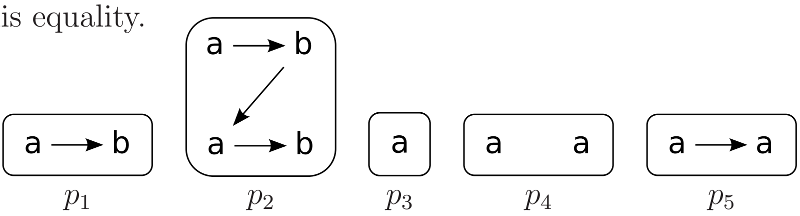

Fig. 1. Composition and projection of pomsets with $\Sigma = \{a, b\}$, $R = \{(b, a)\}$ and $\Gamma = \{a\}$. We have $p_1 \circ p_1 = p_2$, , $\pi_\Gamma(p_1) = p_3$, $p_3 \circ p_3 = p_4$, $\pi_\Gamma(p_2) = p_5$ and $p_4 \neq p_5$.

3 Boxed Pomsets

Figure 1 shows clearly that the projection of pomset composition is, in general, not equals to the composition of pomset projection. The *boxed-pomset* model have been introduced in [6] to correct this draw-back. In this section, we redefine them in a simpler and much concise way as well as giving complete proofs of their properties in Section 4. However, it is possible to show that these models are equivalent.

Let us first introduce some notations. We identify subsets Γ of Σ and pomsets p having exactly $|\Gamma|$ elements such that $\lambda(E_p) = \Gamma$ and $\leq_p = \emptyset$. Furthermore, for any $E, F \in \mathbb{E}$ we define $\langle E, F\rangle$ as $\{(e, f) \in E \times F \mid \lambda(e) = \lambda(f)\}$. For a partial-order relation $\leq$, we write $E \rightsquigarrow_\leq F$ if $\lambda(F) = \lambda(E)$ and $\langle E, F\rangle \subseteq \leq$.

Definition 1 (Boxed Pomsets) *A* boxed pomset $b = [E, \leq]$ *is a pomset whose events can be partitioned into the three sets* E^-, E^i *and* E^+ *such that: (i)* $[E^-] = [E^+] = \lambda(E_b)$*; (ii)* $E^- \rightsquigarrow_{\leq} (E^i \uplus E^+)$*; and (iii)* $(E^- \uplus E^i) \rightsquigarrow_{\leq} E^+$.

The sets E^- and E^+ are called the b's *input* and *output* and the pomset $b|E^i$ the b's *inside*. A boxed pomset can be seen as a simple pomset augmented with explicit extrema events, for, at least, every label appearing in it. This simple pomset corresponds to the boxed-pomset inside and the collection of minimal and maximal events to its input and output. Indeed, (i) and (ii) uniquely define E^- as the collection of minimal events of b (for each label); and (i) and (iii) uniquely define E^+ as the collection of maximal events of b (for each label). Thus, we can detail the components of boxed pomset b as $[E_b^- \uplus E_b^i \uplus E_b^+, \leq_b]$. When necessary, we use E_b^{-+} instead of $E_b^- \uplus E_b^+$. Moreover, (i) implies that $\lambda(E_b^i) \subseteq \lambda(E_b)$, that is E_b^i might not contains all the label appearing in b. The collection of all boxed pomsets is denoted by $\mathbb{B}$. Examples of boxed pomsets are given in Figure 2. They are represented likely as pomsets, but we draw separate rectangles to distinguish clearly their input, inside and output. Input is always on the top, output on the bottom.

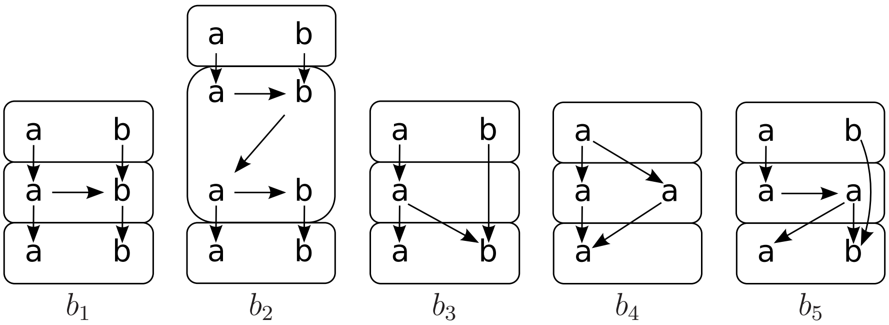

Fig. 2. Composition and projection of boxed pomsets with $\Sigma = \{a, b\}$, $R = \{(b, a)\}$ and $\Gamma = \{a\}$. We have $b_1 \bullet b_1 = b_2$, $\bar{\pi}_\Gamma(b_1) = b_3$, $b_3 \bullet b_3 = b_5$ and $\bar{\pi}_\Gamma(b_2) = b_5$.

We introduce now two natural maps B and U which links pomsets and boxed pomsets. We define first the *boxing operator* $B : \mathbb{P} \to \mathbb{B}$ which builds, from a pomset p, a boxed pomset having p as inside: giving two isomorphic pomsets $q = [E_q]$ and $r = [E_r]$ such that $\lambda(q) = \lambda(r) = \lambda(p)$, $B(p)$ is defined as the boxed pomset $[E_q \uplus E_p \uplus E_r, \leq]$, where $\leq$ is the transitive closure of the union of $\leq_p$, $\langle E_q, E_p\rangle$ and $\langle E_p, E_r\rangle$. Remark that $\leq$ is the minimal partial order containing $\leq_p$ which also ensures that (ii) and (iii) of definition 1 are satisfied. By construction, $B(p)$ is a well-formed boxed pomset admitting p as inside. Conversely, we define the *unboxing operator* $U : \mathbb{B} \to \mathbb{P}$ which extracts the inside of a boxed pomset as $U(b) = (b \setminus E_b^{-+})$. By construction, for any $p \in \mathbb{P}$, $U \cdot B(p) = p$. Figure 1 and Figure 2 illustrate these definitions, where the following equalities hold: $B(p_1) = b_1$, $B(p_4) = b_4$, $U(b_2) = p_2$ and $U(b_3) = p_3$.

As for pomsets, we define the *projection* of a boxed pomset b on an observable alphabet Γ as a function $\bar{\pi}_\Gamma : \mathbb{B} \to \mathbb{B}$ which restricts only the inside of

b to events which are labeled by Γ, ie. with no modifications to its input and output. More formally, $\bar{\pi}_\Gamma(b) = (b \setminus (E_b^i \cap \lambda^{-1}(\Sigma \setminus \Gamma))$. In the same way, we define the *composition* of two boxed pomsets b and c, denoted by $b \bullet c$, as that restricts the boxing of b and c pomset composition by removing events which belong to b and c inputs and outputs. that is, $b \bullet c = (B(b \circ c) \setminus (E_b^{-+} \uplus E_c^{-+}))$. This operation is associative and well-defined, admitting $1_\mathbb{B} = B(1_\mathbb{P}) = 1_\mathbb{P}$ as neutral element. Thus $(\mathbb{B}, \bullet, 1_\mathbb{B})$ is a monoid and rational boxed-pomset expressions and their languages are well-defined. Figure 2 gives an example of boxed-pomset composition and projection. When fixing $\Sigma = \{a, b\}$, $R = \{(b, a)\}$ and $\Gamma = \{a\}$, let us remark that $\bar{\pi}_\Gamma(b_1 \bullet b_1) = \bar{\pi}_\Gamma(b_1) \bullet \bar{\pi}_\Gamma(b_1)$. Lemma 1 states that this property is always satisfied for boxed pomsets - which is actually the reason why we consider them.

As usual, all mappings $f : X \to Y$ defined above are extended naturally to $f : 2^X \to 2^Y$ by applying the operator to each elements of the sets, ie. $f(X) \equiv \{f(x) \mid x \in X\}$. We also extend f to rational expressions, ie. obtaining $f : REX(X) \to REX(Y)$ using the following construction: given a rational expression $\alpha \in REX(X)$, $f(\alpha)$ is obtained by simply replacing each pomset x in α by the boxed pomset $f(p)$. Remark that this operation does not imply that the languages are mapped, as f might not be a morphism. We use $\cdot$ to denote the usual composition of functions, ie. $f \cdot g$ is the function $x \to f(g(x))$.

4 Properties of Rational Boxed-Pomset Expressions

We start this section by characterizing more precisely the behavior of the boxed-pomset projection and of the boxing operator. Indeed, Lemma 1 and Lemma 2 show that these operations are monoid morphisms.

Lemma 1 *For any $\Gamma \subseteq \Sigma$, $\bar{\pi}_\Gamma : \mathbb{B} \to \mathbb{B}$ is a monoid morphism.*
That is, for any $\delta \in REX(\mathbb{B})$, $\mathcal{L}_\mathbb{B} \cdot \bar{\pi}_\Gamma(\delta) = \bar{\pi}_\Gamma \cdot \mathcal{L}_\mathbb{B}(\delta)$ and $\bar{\pi}_\Gamma(1_\mathbb{B}) = 1_\mathbb{B}$.

Proof: Let us fix $b, c \in \mathbb{B}$, $E = \lambda^{-1}(\Sigma \setminus \Gamma)$ and $X = \bar{\pi}_\Gamma(b) \bullet \bar{\pi}_\Gamma(c)$. We have $X = \Big(B\Big((b \setminus E_1) \circ (c \setminus E_2)\Big) \setminus E_3\Big)$ where $E_1 = (E_b^i \cap E)$, $E_2 = (E_c^i \cap E)$ and $E_3 = (E_b^{-+} \uplus E_c^{-+})$. Let us show that $X = \bar{\pi}_\Gamma(b \bullet c)$ as well.
As extremal events of b and c are not deleted in $(b \setminus E_1)$ and $(c \setminus E_2)$, we have $(b \setminus E_1) \circ (c \setminus E_2) = ((b \circ c) \setminus (E_1 \cup E_2))$. Moreover, if a set S does not contain any extremal events of a pomset p, then $B(p \setminus S) = (B(p) \setminus S)$. Thus, $B((b \setminus E_1) \circ (c \setminus E_2)) = (B(b \circ c) \setminus (E_1 \cup E_2))$. Finally, we can write $X = (B(b \circ c) \setminus (E_1 \cup E_2 \cup E_3))$, which is exactly $\bar{\pi}_\Gamma(b \bullet c)$. □

Lemma 2 *The boxing operator $B : \mathbb{P} \to \mathbb{B}$ is a monoid morphism.*
That is, for any $\alpha \in REX(\mathbb{P})$, $\mathcal{L}_\mathbb{B} \cdot B(\alpha) = B \cdot \mathcal{L}_\mathbb{P}(\alpha)$ and $B(1_\mathbb{P}) = 1_\mathbb{B}$.

Proof: Let us fix $p, q \in \mathbb{P}$, $X = B(p) \circ B(q)$, and $E = (E_{B(p)}^{-+} \uplus E_{B(q)}^{-+})$. We have $B(p) \bullet B(q) = (B(X) \setminus E)$. Let us show that $(B(X) \setminus E) = B(p \circ q)$ as well.

Let us consider the pomset $(X \setminus (E^-_{B(p)} \uplus E^+_{B(q)}))$, whose partial-order is the transitive closure of the union of $\leq_p$, $\langle E_p, E^+_{B(p)} \rangle$, $\leq_q$, $\langle E^-_{B(q)}, E_q \rangle$ and $\{(e,f) \in E^+_{B(p)} \times E^-_{B(q)} \mid (\lambda(e), \lambda(f)) \in R\}$. When restricted to events of E_p and E_q, it is exactly $\leq_{p \circ q}$. That is, $(X \setminus E) = p \circ q$. Furthermore, $\leq_{B(X)}$ can be defined as the transitive closure of $\leq_X \cup \langle (E^-_X, E^-_{B(p)} \uplus E^-_{B(q)}) \rangle \cup \langle (E^+_{B(p)} \uplus E^+_{B(q)}), E^+_X \rangle$. Thus, $\leq_{(B(X) \setminus E)}$ is the transitive closure of $\leq_X \cup \langle E^-_X, E_X \rangle \cup \langle E_X, E^+_X \rangle$, which is also $\leq_{B(X \setminus E)}$. Thus, $(B(X) \setminus E) = B(p \circ q)$. □

Next, Proposition 1 underlines the key property of boxed pomsets: a rational boxed-pomset expression can be obtained from any rational pomset expressions such that the language of the former is equivalent (up to an unboxing operation) to the projected language of the latter. Moreover, this boxed-pomset expression is very simple to obtain as it is suffices to apply $\bar{\pi} \cdot B$ to the rational pomset expression.

Proposition 1 *For any* $\Gamma \subseteq \Sigma$, $\pi_\Gamma \cdot \mathcal{L}_{\mathbb{P}} = U \cdot \mathcal{L}_{\mathbb{B}} \cdot \bar{\pi}_\Gamma \cdot B$.

Proof: For any $p \in \mathbb{P}$, $U \cdot B(p) = p$. Thus, for any $\alpha \in REX(\mathbb{P})$ we have $\pi_\Gamma \cdot \mathcal{L}_{\mathbb{P}}(\alpha) = \pi_\Gamma \cdot U \cdot B \cdot \mathcal{L}_{\mathbb{P}}(\alpha)$. Then, one can check that for any $b \in \mathbb{B}$ and $\Gamma \subseteq \Sigma$, $\pi_\Gamma \cdot U(b) = \left(b \setminus \left(E_b^{-+} \uplus (E_b^i \cap \lambda^{-1}(\Sigma \setminus \Gamma))\right)\right)$, which is exactly $U \cdot \bar{\pi}_\Gamma(b)$. This leads to $\pi_\Gamma \cdot \mathcal{L}_{\mathbb{P}}(\alpha) = U \cdot \bar{\pi}_\Gamma \cdot B \cdot \mathcal{L}_{\mathbb{P}}(\alpha)$. Finally, using Lemmas 2 and 1 we obtain that for any $\alpha \in \mathbb{P}$, $\pi_\Gamma \cdot \mathcal{L}_{\mathbb{P}}(\alpha) = U \cdot \mathcal{L}_{\mathbb{B}} \cdot \bar{\pi}_\Gamma \cdot B(\alpha)$. □

As example, let us consider the projection on $\Gamma = \{a\}$ of the language of the rational pomset expression $p_1^\star p_4^\star$, where p_1 and p_4 are the pomsets defined on Figure 1. Applying Proposition 1, we obtain that $\pi_\Gamma \cdot \mathcal{L}_{\mathbb{P}}(p_1^\star p_4^\star) = U \cdot \mathcal{L}_{\mathbb{B}}(b_3^\star b_4^\star)$, where b_3 and b_4 are the boxed pomsets defined on Figure 2. This language contains all the pomsets which are the union of n totally ordered events labeled by a and $2m$ concurrent events labeled by a, for any $m, n \geq 0$. Clearly, we cannot exhibit a finite collection P of pomsets such that any pomsets belonging to $\pi_\Gamma \cdot \mathcal{L}_{\mathbb{P}}(p_1^\star p_4^\star)$ can be decomposed into a product of elements of P, ie. the projected language is not finitely generated.

5 Projection of Rational Pomset Languages

The purpose of this section is to show that one can decide whether the projection of a rational pomset language is generated by a finite number of prime pomsets, ie. by a finite number of pomsets which can not be decomposed into the product of two non-neutral elements. More formally, a pomset p is a *prime* pomset if, for any pomsets q and r such that $p = q \circ r$, then either $q = 1_{\mathbb{P}}$ or $r = 1_{\mathbb{P}}$. Then, the prime set of a pomset p, denoted by $\|p\|$, is defined as the smallest set such that any decomposition of p into $p_1 \circ \ldots \circ p_n$ with $p_1 \ldots p_n$

are prime pomsets, implies that $\{p_1, \ldots, p_n\} \subseteq \|p\|$. As usual, we naturally extend this definition to set of pomsets: the prime set of L is the union of the prime sets of every pomsets that belong to L, ie. $\|L\| = \bigcup_{p \in L} \|p\|$. Finally, a language L is *finitely generated* if its prime set $\|L\|$ is finite.

First of all, computing the prime set of a given pomset can be done efficiently, as shown by Lemma 3. Indeed, this is equivalent to compute the strongly connected components of a graph based on the considered pomset. This technique is an extension of the ones used to compute the prime decomposition of MSCs [12] and of causal MSCs [5].

Lemma 3 *Any pomset $p \in \mathbb{P}$ is prime if, and only if the directed graph $\mathcal{G}_p = (E_p, (\leq_p \cup R_1 \cup R_2))$ is strongly connected, where:*
$R_1 = \{(e, f) \in E_p \times E_p \mid (f \preccurlyeq_p e) \wedge (\lambda(f), \lambda(e)) \notin R\}$; and
$R_2 = \{(e, f) \in E_p \times E_p \mid (e \not\leq_p f) \wedge (f \not\leq_p e) \wedge (\lambda(f), \lambda(e)) \in R\}$.

Proof: First of all, let us show that if p is not prime then $\mathcal{G}_p$ is not strongly connected. Indeed, let us suppose that $p = q \circ r$ with $q, r \neq 1_{\mathbb{P}}$. Then, let us remark that $R_1 \cap (E_r \times E_q) = \emptyset$ and $R_2 \cap (E_r \times E_q) = \emptyset$. This implies that elements of E_q and E_r are not in the same strongly connected component, that is E_p contains at least two strongly connected components.
Second, let us show that if $\mathcal{G}_p$ is not strongly connected then p is not prime. Indeed, let us denote by E_1, a maximal strongly connected component of $\mathcal{G}_p$ which is a strict subset of E_p, without incoming edges and let us fix $E_2 = (E_p \setminus E_1)$ (and thus $E_2 \neq \emptyset$). We then consider P to be the transitive closure of $(\leq_p \cup R_1 \cup R_2)$ and $Q = P \cap (E_1 \times E_2)$. First, as elements of E_1 and E_2 are not in the same strongly connected component, Q is not reflexive. Thus, $Q = (P \setminus P^{-1}) \cap (E_1 \times E_2)$. That is, Q is the transitive closure of the union of $\{(e, f) \in E_1 \times E_2 \mid e \leq_p f \wedge (\lambda(e), \lambda(f)) \in R\}$ and $(R_2 \setminus R_2^{-2}) \cap (E_1 \times E_2)$. The former set is exactly the relation added when we compute the composition of p_1 and p_2, the restrictions of p to E_1 and E_2. That is $p = p_1 \circ p_2$. □

Lemma 3 leads to a direct algorithm to decompose a pomset into its prime set. However, computing the prime set a projected rational pomset language is much more difficult as it might be not finitely generated and thus might have an infinite prime set. The solution we propose consists to use the boxed pomsets introduced in Section 3 and 4 as an intermediate model to finitely represent the projected languages. Proposition 2 shows that this boxed-pomset representation have a nice fix-point property: it is sufficient to bound their $\star$-iterations to deduce properties on their unbounded languages.

Proposition 2 *For any $b \in \mathbb{B}$, $\|U(b^{|\Sigma|+1})\| = \|U(b^{|\Sigma|+2})\|$ iff $\|U(b^\star)\|$ is finite.*

Proof: Let us have a n big enough and let us denote by $b_k = [E_k^{-+} \uplus E_k^i, \leq_k]$ the k-th occurrence of b in $b \circ \ldots \circ b$, where b is composed n times using $\circ$.

Let us also write $U(b^n) = [E, \leq]$ (with $\preccurlyeq$ the transitive reduction of $\leq$) and $\mathcal{G}_{U(b^n)} = (E, \rightarrow)$ where $\rightarrow$ is $(\leq \cup R_1 \cup R_2)$ as defined in Lemma 3.
First of all, $U(b^n)$ is defined as $((b \circ \ldots \circ b) \setminus (\bigcup_{1 \leq k \leq n} E_k^{-+})$, where b is composed n times using $\circ$. When considering the associated graphs, we obtain that $\mathcal{G}_{U(b^n)}$ is the restriction of $\bigcup_{1 \leq k \leq n} \mathcal{G}_{b_k}$ to E, augmented with $\bigcup_{k \neq l}(\rightarrow \cap (E_k^i \times E_l^i))$.
Moreover, one can check that if (e_k, e_l) is in $\rightarrow$, with $e_k \in E_k$ and $e_l \in E_l$, then for any N such that $k + N \geq 1$ and $l + N \geq 1$ we have (e_{k+N}, e_{l+N}) in$\rightarrow$ as well, with e_{k+N},e_{l+N} the corresponding events in E_{k+N} and E_{l+N}. Moreover, if (e_k, e_l) is in $\rightarrow$ then either (i) $e_k \leq e_l$; or (ii) $(e_l \preccurlyeq e_k) \wedge (\lambda(e_l), \lambda(e_k)) \notin R$; or (iii) $(e_l \not\leq e_k) \wedge (e_k \not\leq e_l) \wedge (\lambda(e_l), \lambda(e_k)) \in R$. However, in case $k > l$, then only case (ii) is possible. As $\lambda(E_l^+) = \lambda(E_k^-) \subseteq \Sigma$, $k > l$ implies $k - l \leq |\Sigma|$.
$-(\Rightarrow)$, let us assume that $\|U(b^{|\Sigma|+1})\| = \|U(b^{|\Sigma|+2})\|$. As strongly connected components can appear only if there are some edges $(e_k, e_l) \in (\rightarrow \cap (E_k^i \times E_l^i))$, for $k > l$, then we must have $\|U(b^{|\Sigma|+1+N})\| = \|U(b^{|\Sigma|+N+2})\|$ for any $N > 0$, that is $U(b^\star)$ is finitely generated and its prime set is exactly $\|U(b^{|\Sigma|+1})\|$.
$-(\Leftarrow)$, let us assume that $\|U(b^{|\Sigma|+1})\| \neq \|U(b^{|\Sigma|+2})\|$. Then, there exists k and an edge $(e, f) \in\rightarrow$, where $e \in E_{|\Sigma|+2}$, $f \in E_k$ and $1 < k \leq |\Sigma| + 1$ such that a strongly connected of new kind is created in $\mathcal{G}_{U(b^{|\Sigma|+2})}$. As this edge already exists between the corresponding events in $E_{|\Sigma|+1}$ and E_{k-1} this new component has to the be union of strongly connected components of $\mathcal{G}_{U(b^{|\Sigma|+1})}$. Thus, it will continue to grow at each iteration. That is, $\|U(b^\star)\|$ is infinite. □

Finally, Proposition 2 can be used in conjunction with Proposition 1 to decide whether the projection of a rational pomset language is finitely generated. This leads to Theorem 1 which is the main result of this paper.

Theorem 1 *It is decidable to know whether the projection of a rational pomset language is finitely generated.*

Proof: Let us have $\alpha \in REX(\mathbb{P})$ and $\Gamma \subseteq \Sigma$. We want to decide whether $\pi_\Gamma \cdot \mathcal{L}_\mathbb{P}(\alpha)$ is finitely generated. Proposition 1 indicates that is equivalent to decide whether $U \cdot \mathcal{L}_\mathbb{B} \cdot \bar{\pi}_\Gamma \cdot B(\alpha)$ is finitely generated. Then, Proposition 2 shows that is equivalent equivalent to compare the prime pomsets of finite languages $U \cdot \mathcal{L}_\mathbb{B} \cdot \bar{\pi}_\Gamma \cdot B(\alpha_{|\Sigma|+1})$ and $U \cdot \mathcal{L}_\mathbb{B} \cdot \bar{\pi}_\Gamma \cdot B(\alpha_{|\Sigma|+2})$, where α_k is the rational expression obtained by syntactically replacing every $\star$ by a k. □

–Acknowledgements– The author would like to thank Loïc Hélouët, Blaise Genest and Philippe Darondeau for their helpful comments and suggestions.

References

[1] R. Alur and M. Yannakakis. Model Checking of Message Sequence Charts. In *CONCUR'99*, pages 114–129.

[2] E.M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith. Counterexample-Guided Abstraction Refinement. In *CAV'00*, pages 154–169.

[3] V. Diekert and G. Rozenberg, editors. *Book of Traces.* World Scientific, 1995.

[4] J. Fanchon and R. Morin. Regular Sets of Pomsets with Autoconcurrency. In *CONCUR'02*, pages 402–417.

[5] T. Gazagnaire, B. Genest, L. Hélouët, P.S. Thiagarajan, and S. Yang. Causal Message Sequence Charts. In *CONCUR'07*, pages 166–180.

[6] T. Gazagnaire and L. Hélouët. Event Correlation with Boxed Pomsets. In *FORTE'07*, pages 160–176.

[7] B. Genest, L. Hélouët, and A. Muscholl. High-Level Message Sequence Charts and Projections. In *CONCUR'03*, pages 308–322.

[8] B. Genest, D. Kuske, and A. Muscholl. A Kleene Theorem and Model Checking Algorithms for Existentially Bounded Communicating Automata. *IC*, 204(6), 2006.

[9] B. Genest, A. Muscholl, H. Seidl, and M. Zeitoun. Infinite-State High-Level MSCs: Model-Checking and Realizability. In *ICALP'02*, pages 657–668.

[10] J.L. Gischer. The Equational Theory of Pomsets. *TCS*, 61, 1988.

[11] S.G. Govindaraju and D.L. Dill. Counterexample-Guided Choice of Projections in Approximate Symbolic Model Checking. In *ICCAD'00*, pages 115–119.

[12] L. Hélouët and P. Le Maigat. Decomposition of Message Sequence Charts. In *SAM'00*, pages 47–60.

[13] J.G. Henriksen, M. Mukund, K. Narayan Kumar, and P.S. Thiagarajan. On Message Sequence Graphs and Finitely Generated Regular MSC Languages. In *ICALP'00*, pages 675–686.

[14] ITU-TS. *Recommendation Z.120: Message Sequence Chart (MSC).* 1999.

[15] A. Mazurkiewicz. Concurrent Program Schemes and Their Interpretations. Technical Report DAIMA PB–78, Aarhus University, 1977.

[16] J.D. McKnight. Kleene Quotient Theorem. *Pacific Journ. of Math.*, 1964.

[17] Rémi Morin. Recognizable Sets of Message Sequence Charts. In Helmut Alt and Afonso Ferreira, editors, *STACS*, volume 2285 of *Lecture Notes in Computer Science*, pages 523–534. Springer, 2002.

[18] A. Muscholl and D. Peled. Message Sequence Graphs and Decision Problems on Mazurkiewicz Traces. In *MFCS'99*, pages 81–91.

[19] OMG. Unified Modeling Language Specification, 2003.

[20] D. Peled. Specification and Verification of Message Sequence Charts. In *FORTE'00*, pages 139–154.

[21] A. Pnueli. Keynote Address: Abstraction, Composition, Symmetry, and a Little Deduction: The Remedies to State Explosion. In *CAV'00*, page 1.

[22] V. Pratt. Modeling concurrency with partial orders. *IJPP*, 15(1):33–71, 1986.