A Syntactic Congruence for the Recognizability of Pomset Languages.

Jean Fanchon , LAAS du CRNS, 7 av. du Col. Roche, 31077 Toulouse France.
fanchon@laas.fr.

Abstract

In the present paper, we contribute to the theory of recognizable pomset (or partial word) languages from an algebraic point of view. By substitutions in pomsets labelled with variables and pomset contexts, we define the syntactic equivalence of a pomset language, and use finite index ones to define recognizable languages. We show that the induced recognisable pomset languages have extended closure properties similar to the ones of recognizable tree languages [GS]. They constitute a generalisation of recognisable serie-parallel pomset languages as defined in [LW].

1 Introduction.

Partially ordered multisets or partial words constitute one of the most basic models of concurrency. Process algebras like CCS and TCSP, and system models like Petri Nets, have been given pomset semantics since many years, and different pomset algebras have been defined. Nevertheless the study of pomset languages from the point of view of recognisability or regularity is far from being closed.

The definition of algebraic recognisability for a subset L of a $\sum$-algebra, i.e. L saturates a finite index $\sum$-congruence (or is the inverse image of a subset by a $\sum$-morphism on a finite $\sum$-algebra), encounters in the case of pomsets the lack of a finite set of operators generating all pomsets [Gi]. This could be one of the reasons for which there is up to now no general theory of pomset languages recognisability, extended from those on words or trees languages as it has been done for Mazurkiewics traces. In a complementary approach, different accepting finite devices for pomset languages have been defined, graph acceptors applied to pomsets[Th], asynchronous automata whether cellular or not, applied (up to now) to restricted classes of pomsets [K], automata with concurrency relations applied to concurrency monoids [DK]. In these works the accepted languages have until now no caracterisation in an algebraic framework .

Recently the restricted algebra of serie-parallel pomsets yielded a fairly extended theory and kleene-like results [LW]. We generalise here part of this work, by the use of open pomsets, i.e. pomsets labelled by variables. If $PW(A)$ is the set of pomsets labelled on the alphabet A, $PW(A \cup X)$ is the set of open pomsets with variables in X. By substitution of the variable labelled nodes by pomsets in an open pomset, we define a specific operation whose arity is the number of variables labelling the pomset, and called a pomset defined operation [Gi]. We can then adapt from trees the notion of context, and define the syntactic equivalence of a pomset language, which is a congruence w.r.t. all pomset defined operations . We define recognizable pomset languages as the ones with finite index syntactic congruences, i.e. inducing a finite number of equivalence classes. This definition applies as well to open pomset languages and to closed ones. In case of pomset algebras generated from the labels by a finite number of open pomsets, the definition matches the usual algebraic recognizability theory, and extends the case of the serie-parallel algebra mentionned previously.

We show that many of the closure properties defined on recognizable tree languages are valid in the open pomset framework, like boolean operations and inverse image by a morphism. As for trees, a syntactic monoid congruence w.r.t. a language can be defined and the cases where it has finite index caracterise recognisable languages. The proofs of these properties are very similar to the ones for tree languages, remembering that they cannot be based (yet) on finite acceptors or automata.

Furthermore, we show that recognisable languages are closed for pomset defined operations. Unlike the previous ones, the proof of this property is more involved and requires a carefull insight into the pomsets structures. We conjecture that they are also closed for substitutions in recognisable languages.

This definition of recognisability cannot pretend to be general. The substitution rules considered here are simple when viewed as graph rewriting rules [Co], but they are the most natural extention to pomsets of the substitutions in tree languages. Furthermore they yield also a natural extention of the serie-parallel pomset algebra, and should open into new views of finite accepting devices for pomset languages.

2 Open Pomsets and substitutions.

2.1 Pomsets.

A partially ordered set (poset) is a pair $s = (E, \leq)$ where E is a set and $\leq$ a binary reflexive, antisymmetric and transitive relation on E . In the present work, we consider only **finite partial orders**, where the set E is finite. The complement of $\leq$ in $E \times E$ is denoted $co_{\leq} = \{(x, y), \neg(x \leq y \, OR \, y \leq x)\}$, with infix notation.We omit the subscript when the order relation is implicit. If $E' \subseteq E, e \in E$, we define $e\natural E' \iff \forall e' \in E', e\natural e'$ for $\natural \in \{\leq, \geq, co\}$.

Given an alphabet A, an A-**labelled poset** , A-LP in short, is a triple $s = (E_s, \leq_s, l_s)$ where $(E_s, \leq_s)$ is a finite partial order whose elements are called events, and $l : E_s \longrightarrow A$ a labelling function on events . An **isomorphism of two** A-LP is a bijection between their events sets preserving the order relations and the labellings . The **set of pomsets on** A (partially ordered multisets on A), also called **partial words** , denoted $PW(A)$, is the set of **isomorphism classes of A-labelled posets**. We denote $[E, \leq, l]$ the partial word of which $(E, \leq, l)$ is a representative . The alphabet A and the monoid A^* are included in $PW(A)$ in a natural way: a word $u = a_1 a_2 .... a_n$ in A^* is identified with the partial word $[\{1, 2, ...n\}, \leq, l]$, where the order is the standart one on naturals, and the label maps the i-th integer on the i-th letter in u. The empty pomset is denoted $\varepsilon = [\oslash, \oslash, \oslash]$. The number of occurrences of a letter $a \in A$ in a pomset is $|[E, \leq, l]|_a = Card\{e \in E | l(e) = a\}$, the alphabet of a pomset is $\alpha([E, \leq, l]) = \{a \in A | \exists e \in E, l(e) = a\}$, and the multiset of a pomset $s$is $\mu(s) \in \aleph^A$,$\mu(s)(a) = |s|_a$.

2.1.1 Pomsets with variables.

Let A be an alphabet of "constant" labels, X an alphabet of "variable" labels such that $A \cap X = \oslash$, the set of **open pomsets on** A **with variables in** X is $P(A, X) = PW(A \cup X)$,i.e. the set of pomsets on $A \cup X$.

The **set of variables** of an open pomset is $Var[E, \leq, l] = l(E) \cap X$, the **arity** of an open pomset is $\rho(s) = Card(Var(s))$. The "number of holes" $\sharp(s)$ of a pomset $s = [E, \leq, l]$, is the number of occurrences of variable labels in E, i.e. $\sharp(s) = Card(l^{-1}(X))$. A pomset $s \in PW(A \cup X)$ such that $Var(s) = \{x_1, .., x_n\}$is also denoted $s(x_1, .., x_n)$.

2.2 Morphisms and operations.

Pomset substitutions or morphisms are defined by mappings from the label alphabet $A \cup X$ into the set of pomsets $P(A, X)$. The image of a pomset is obtained by replacing its events by the images of their labels through such a mapping. The figure 1 shows how a morphism substitutes the pomset $f(x)$ to a single event labelled by x. The figure 2 shows the final result of a substitution applied to a pomset.

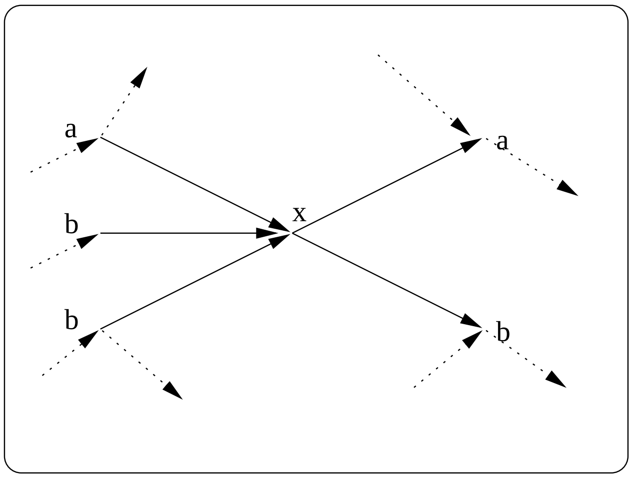
Formally any mapping $\varphi : A \cup X \longrightarrow PW(A \cup X)$, extends to a mapping called morphism $\overline{\varphi} : PW(A \cup X) \longrightarrow PW(A \cup X)$ as follows:

Let be $\varphi(a) = [E_a, \leq_a, l_a], a \in A \cup X$, and $s = [E, \leq, l]$, then $\overline{\varphi}(s) = [E', \leq', l']$ with:

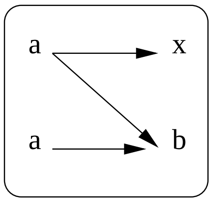
1. $E' = \cup\{\{e\} \times E_a \mid e \in E \, , \, l(e) = a\}$.
2. $\leq' = \{((e_1, e_1'), (e_2, e_2')), e_1 \leq e_2, e_1 \neq e_2\} \cup \{((e, e_1), (e, e_2)), e_1 \leq_{l(e)} e_2\}$
3. $l'(e, e') = l_{l(e)}(e')$

Clearly the definition of $\overline{\varphi}(s)$ is given up to isomorphism of labelled posets, and is independent of the particular representatives of the components s and $\varphi(z), z \in A \cup X$. In many cases a simpler equivalent definition of $\overline{\varphi}(s)$ can be used , for example in the case where $\varphi(l(e)) = l(e)$ for all events $e \in E$ except for a single one (see "contexts" below).

s =



f(x)=



f(a) = a , f(b) = b

f(s) =

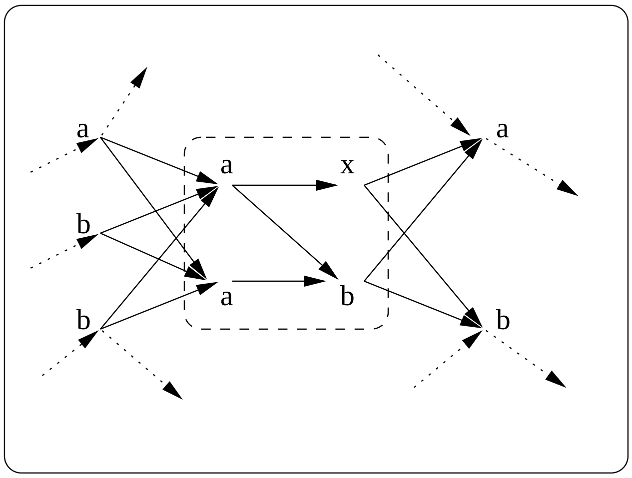


Figure 1.

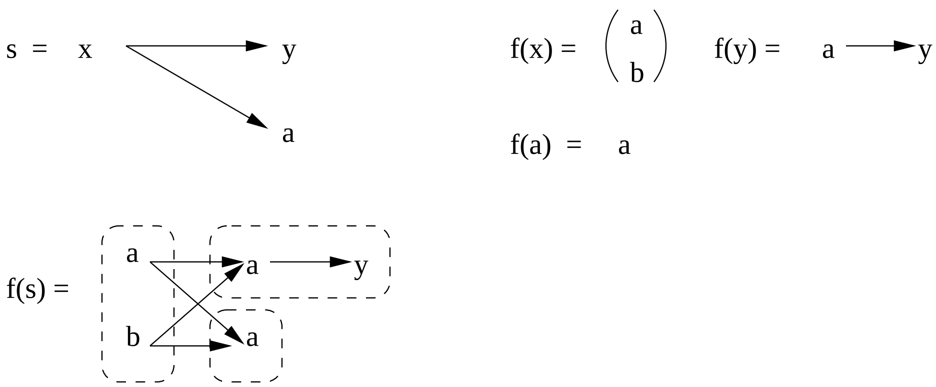


Figure 2.

2.2.1 Pomset definable operations.

Pomset definable operations are based on substitutions of variables, i.e. mappings $\varphi : X \longrightarrow PW(A, X)$. Any open pomset s of arity n with variables $Var(s) = \{x_1, .., x_n\}$ defines a n-ary operation $\overline{s} : P(A, X)^n \longrightarrow P(A, X)$, where the image $\overline{s}(u_1, .., u_n)$ is the image of s by a substitution $\overline{\varphi}$, $\overline{s}(u_1, .., u_n) = \overline{\varphi}(s)$, where $\varphi : X \longrightarrow PW(A, X)$ is such that $\varphi(x_i) = u_i, i = 1..n$. In the following, we adopt the standart tree notation $\overline{s}(u_1, .., u_n) = s[u_1/x_1, .., u_n/x_n]$ abreviated in $s[u_i/x_i]$ when no ambiguity arise.

2.2.2 Sequence and parallel composition.

Sequence and parallel composition are particular cases of pomset defined operations. Let be $Seq = [\{1, 2\}, \{(1, 2)\}, \{(1, x_1), (2, x_2)\}]$ and $Par = [\{1, 2\}, \oslash, \{(1, x_1), (2, x_2)\}]$, then the pomsets $Seq[u/x_1, v/x_2]$and $Par[u/x_1, v/x_2]$ are the standart sequential and parallel compositions of the pomsets u and v. We denote in the following $u \bullet v = Seq[u/x_1, v/x_2]$ and $u \parallel v = Par[u/x_1, v/x_2]$.

In the example of figure 2, we can write $f(s) = s[(a \parallel b)/x, (a \bullet y)/y]$

2.2.3 Languages extentions.

Pomset defined operations can be extended to substitutions of variables by languages of pomsets, i.e. subsets of $P(A, X)$. Let be $L_i \subseteq P(A, X), i = 1..n$, and $s \in P(A, X)$ of arity n, with $Var(s) = \{x_1, .., x_n\}$then $s[L_1/x_1, .., L_n/x_n]$ is the language obtained by all the possible substitutions of occurrences of the variables x_i in s by elements from L_i, with the property that two different occurrences of the same variable x_k may be replaced by different elements of L_k. Formally we have:

1. **If** s **is linear**, i.e. each variable occurs only once in u, i.e. $\forall x \in Var(s), Card(l^{-1}(x)) = 1$, then $s[L_1/x_1, .., L_n/x_n] = \{s[u_1/x_1, .., u_n/x_n], u_i \in L_i, i = 1..n\}$.
2. **Otherwise** let be $m = \sharp(s)$ the number of holes in s, we can find a linear pomset s' with arity m and variables $\{x'_1, .., x'_m\}$, and a map $\phi : [m] \longrightarrow [n]$such that $s = s'[x_{\phi(i)}/x'_i]$. We then define $s[L_1/x_1, .., L_n/x_n] = s'[L_{\phi(1)}/x'_1, .., L_{\phi(m)}/x'_m]$.

If $L \subseteq P(A, X)$, then $L[L_1/x_1, .., L_n/x_n] = \{s[L_1/x_1, .., L_n/x_n], s \in L\}$.

2.3 Contexts.

Let ξ be a new symbol, $\xi \notin A \cup X$. A **context** is a pomset $[E, \leq, l] \in PW(A \cup X \cup \{\xi\})$such that $Card\{e \in E, l(e) = \xi\} = 1$ The set of pomset contexts is denoted $Ct(\xi)$, or Ct for short.

Let be $C, D \in Ct(\xi)$, $u \in P(A, X)$, we denote by $C[u] = C[u/\xi] \in P(A, X)$ the pomset resulting from the substitution of ξ by the pomset u in C. The result of the substitution of ξ in C by a context D is a new context (obvious) denoted $C.D$, i.e $C.D = C[D/\xi] \in Ct$.

Contexts with ξ substitutions have a monoidal structure : substitution in Ct is associative, $\forall C, D, F \in Ct, C.(D.F) = (C.D).F$, and ξ is a neutral element, $\forall C \in Ct, \xi.C = C.\xi = C$. (Obvious, note that ξ in that case represents the pomset with a single event labelled by ξ).

2.3.1 Lemma.

$(Ct, ., \xi)$ is a monoid .

The result of a substitution in a context can be described as follows:

Let be $C \in Ct(\xi)$ with $C = [E_c \cup \{e_0\}, \leq_c, l_c]$ where $e_0 = l_c^{-1}(\xi)$, and $v = [E_v, \leq_v, l_v] \in P(A, X)$, such that $E_c \cap E_v = \oslash$, then the result of the substitution $u = C[v]$ is $u = [E_c \cup E_v, \leq_u, l_u]$, where the event e_0 has been replaced in C by the set of events E_v. The relation$\leq_u$ is the reflexive closure of $\leq_c \cup \leq_v \cup (\Downarrow_c e_0 \times E_v) \cup (E_v \times \Uparrow_c e_0)$ where $\Downarrow_c e_0 = \{e \in E_c,\ e <_c e_0\}$ ($\Uparrow_c e_0$ similarly). The order $\leq_u$ between the events of E_c (resp. E_v) in $C[v]$ is the same as in C (resp. v) and the predecessors in u (resp. successors) of some event of E_v , and belonging to E_c, are the predecessors of e_0 in C (resp. successors). In the partial order $(E_c \cup E_v, \leq_u)$, the subset E_v has specific properties , which caracterise so-called refined subsets:

2.3.2 Refined subsets and subpomsets.

Let be $u = (E, \leq)$ a partial order, a subset $E' \subseteq E$ is a **refined subset** of u iff $\forall e \in E - E'$,$\forall e' \in E'$,

1. $e \; co \; e' \Longrightarrow \forall e'' \in E', e \; co \; e''$
2. $e \leq e' \Longrightarrow \forall e'' \in E', e \leq e''$
3. $e' \leq e \Longrightarrow \forall e'' \in E', e'' \leq e$.

Note that the points 2 and 3 imply the point 1.

The following lemma gives a more precise caracterisation of refined subsets.

2.3.3 Lemma.

E' is a refined subset of $u = (E, \leq)$ (with $\leq' = \leq /E'$) iff $\forall e \in E - E'$, $\forall e' \in E'$:

1. $e \rightarrow e' \Longrightarrow e' \in Min(E', \leq') \bigwedge \forall e'' \in Min(E', \leq'), e \rightarrow e''$.
2. $e' \rightarrow e \Longrightarrow e' \in Max(E', \leq') \bigwedge \forall e'' \in Max(E', \leq'), e'' \rightarrow e$.

Proof.

We prove the “only if” direction for 1: for any $e'' \in Min(E', \leq')$, we have $e \leq e''$ (definition 2.3.2). If $\neg e \rightarrow e''$, then for some $e''' \in E - E'$, $e \leq e''' \leq e''$, thus $e \leq e'' \leq e'$ (definition 2.3.2) which contradicts $e \rightarrow e'$. Now $e' \in Min(E', \leq'$)follows directly. The proof of 2 is similar.

The “if” direction is left to the reader.

Q.E.D.

A pomset $v = [E', \leq', l']$ is a **refined subpomset** of $u = [E, \leq, l]$ iff E' is a refined subset of $(E, \leq)$, $\leq' = \leq /E'$, and $l' = l/E'$.

Note that all the labels of u are refined subpomsets of u as singleton pomsets.

The following lemma is a direct consequence of the previous definitions.

2.3.4 Lemma.

$\forall u, v \in P(A,X)$, $u = C[v]$ for some context C iff v is a refined subpomset of u .

We say that v **occurs in** u **at the context** C iff $u = C[v]$.

Let be $u \in P(A,X)$, we denote by $Ref(u) = \{v \in P(A,X), \exists C \in Ct, u = C[v]\}$ the (obviously) finite **set of refined subpomsets** of u. This definition can easily be extended to refined subpomsets of a context $u \in Ct$, which can be contexts themselves. It is easy to see that $u \in Ct \iff Ref(u) \cap Ct \neq \oslash$, i.e. ξ is a label of u iff the pomset ξ is a refined subpomset of u.

Notations .

$u^{-\xi}L = \{C \in Ct(\xi), C[u] \in L\}$

$C^{-1}(L) = \{u \in P(A,X), C[u] \in L\}$

3 Recognisable languages.

3.1 Syntactic equivalence.

The syntactic equivalence on pomsets is similar to the one for trees, replacing tree contexts by pomset ones.

3.1.1 Definition

Let be $L \subseteq P(A,X)$, the **syntactic equivalence** of L is the relation $\sim_L \subseteq P(A,X) \times P(A,X)$ defined by

$$u \sim_L v \iff \forall C \in Ct(\xi) : (C[u] \in L \iff C[v] \in L)$$

We denote by $[u]_L$ the class of a pomset u.

It is an equivalence relation and due to definitions, we have $u \sim_L v \iff u^{-\xi}L = v^{-\xi}L$.

The syntactic equivalence is a "congruence" for all pomset defined operations:

3.1.2 Lemma (Context congruence) .

$$u \sim_L v \Longrightarrow \forall C \in Ct(\xi), C[u] \sim_L C[v]$$

Proof.

$u \sim_L v \Longleftrightarrow \forall C \in Ct(\xi), (C[u] \in L \Longleftrightarrow C[v] \in L)$

$\Longrightarrow \forall D, F \in Ct(\xi), (D.F[u] \in L \Longleftrightarrow D.F[v] \in L)$

$\Longleftrightarrow \forall F \in Ct(\xi)(F[u] \sim_L F[v])$. **Q.E.D**.

The same holds for n-ary terms,

3.1.3 Lemma (Congruence for pomset defined operations).

$$(\forall i \in [n], u_i \sim_L v_i) \Longrightarrow (\forall s \in P(A, X), s[u_i/x_i] \sim_L s[v_i/x_i])$$

Proof: It is a consequence of the previous property. We prove that for each $i = 1..n$, $s[u_1/x_1, .., u_i/x_i, .., v_n/x_n] \sim_L s[u_1/x_1, .., v_i/x_i, .., v_n/x_n]$, by considering the contexts obtained by substituting ξ consecutively to each occurrence of x_i in $s[u_1/x_1, .., u_{i-1}/x_{i-1}, v_{i+1}/x_{i+1}, .., v_n/x_n]$ **Q.E.D**.

3.2 Syntactic monoid congruence.

As for trees, we can define a syntactic monoid congruence $\approx_L$ on contexts, as follows:

for any $C, C' \in Ct$

$C \approx_L C' \Longleftrightarrow (\forall D \in Ct, \forall u \in P(A, X), D.C[u] \in L \Longleftrightarrow D.C'[u] \in L)$

This definition is equivalent to the following: $C \approx_L C' \Longleftrightarrow \forall u \in P(A, X), C[u] \sim_L C'[u]$.

The relation $\approx_L$is an equivalence relation (obvious) and a congruence w.r.t. context substitution.

3.2.1 Lemma.

$C \approx_L C' \Longrightarrow (\forall D_1, D_2 \in Ct,\ D_1.C.D_2 \approx_L D_1.C'.D_2$

Proof: $\forall D \in Ct, \forall u \in P, (D.C[u] \in L \Longleftrightarrow D.C'[u] \in L)$ implies that

$\forall D, D_1, D_2 \in Ct, \forall u \in P,\ ((D.D_1).C[D_2[u]] \in L \Longleftrightarrow (D.D_1).C'[D_2[u]] \in L)$ **Q.E.D**.

3.3 Recognizable languages.

3.3.1 Definition.

A language $L \subseteq P(A, X)$ is **recognizable** iff its syntactic congruence $\sim_L$ has finite index.

Note that this definition applied to a set of totally ordered pomsets coincides with the usual one on words: if $L \subseteq (A \cup X)^*$, then L is recognisable as a word languages if and only if it is recognisable as a pomset language. One can see that a context $C \in Ct$ such that $C[u] \in (A \cup X)^*$ can be written $C = v \bullet \xi \bullet w$. with $v, w \in (A \cup X)^*$, and thus the set $u^{-\xi} L = \{C \in Ct(\xi), C[u] \in L\}$ is isomorphic to the set $\{(v, w) \in (A \cup X)^*, v \bullet u \bullet w \in L\}$.

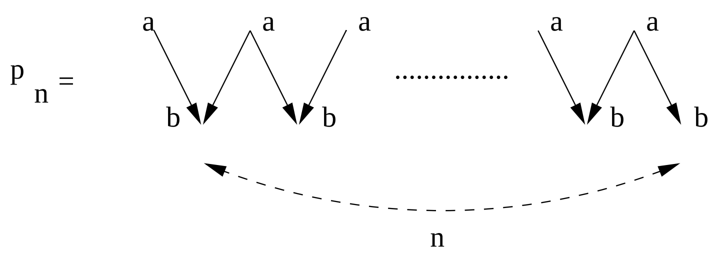
Examples:

The parallel iteration $||_* a \bullet b$ is recognisable while the parallel iteration of $a \parallel b$ is not.

$||_* \; u = \{||_n \; u, n \in \aleph\}$where

$||_0 \; u = \varepsilon$and $||_{n+1} \; u = u \parallel (||_n \; u)$.

The language $L = \{p_n, n \in \aleph\}$described below is recognisable, while not generable by a finite number of pomset operations.



We denote by $Rec(A, X)$ the class of recognizable languages of $P(A, X)$. As for tree languages , languages with a finite index syntactic congruence coincide with those with a finite index syntactic monoid equivalence. We first show the following lemma.

3.3.2 Lemma.

If L is recognizable, the set $\{C^{-1}(L), C \in Ct\}$ is finite and any set $C^{-1}(L)$ is recognizable.

Proof. Any set $C^{-1}(L)$ is saturated for the equivalence $\sim_L$, because we have :

$u \sim_L v \Longleftrightarrow \forall C \in Ct, (u \in C^{-1}(L) \Longleftrightarrow v \in C^{-1}(L))$. Due to the finite number of classes $[u]_L$, there is only a finite number of unions of such classes, and thus of sets $C^{-1}(L)$.

The recognisability of $C^{-1}(L)$:

$u \sim_L v \Longrightarrow \forall C, D \in Ct(\xi) : (C.D[u] \in L \Longleftrightarrow C.D[v] \in L)$

$\Longrightarrow \forall C, D \in Ct(\xi) : (D[u] \in C^{-1}(L) \Longleftrightarrow D[v] \in C^{-1}(L))$

$\Longrightarrow \forall C \in Ct(\xi) : u \sim_{C^{-1}(L)} v$.

Thus the equivalence $\sim_{C^{-1}(L)}$ has finite index if $\sim_L$ has **Q.E.D**.

3.3.3 Proposition.

A language is recognizable iff its syntactic monoid congruence has finite index.

Proof.

Suppose $L \in Rec(A)$, we show that the syntactic monoid congruence has finite index. We have $C_1 \approx_L C_2 \Longleftrightarrow (\forall D \in Ct,\ \forall u \in P,\ D.C_1[u] \in L \Longleftrightarrow D.C_2[u] \in L)$ and this is equivalent to $\forall D \in Ct,\ C_1^{-1}(D^{-1}(L)) = C_2^{-1}(D^{-1}(L))$. We have shown in 3.3.2 that there are only a finite number of sets $D^{-1}(L)$, namely $\{D_i^{-1}(L), i = 1, n\}$ for some $\{D_i\} \subset Ct$. Thus we have $\forall i \in [n], C_1^{-1}(D_i^{-1}(L)) = C_2^{-1}(D_i^{-1}(L))$. Let be for each $i = 1..n$ the equivalence relation on Ct defined by $C_1 \approx_i C_2 \Longleftrightarrow C_1^{-1}(D_i^{-1}(L)) = C_2^{-1}(D_i^{-1}(L))$. Due to the fact that the sets $D_i^{-1}(L), i = 1, n$ are recognisable , $\approx_i$ has finite index. The equivalence $\approx_L$is the intersection of a finite number of equivalences of finite index, and has finite index too.

Suppose $\approx_L$ has finite index, the weaker equivalence $C_1 \approx_L^{\varepsilon} C_2 \Longleftrightarrow (\forall D \in Ct,\ D.C_1[\varepsilon] \in L \Longleftrightarrow D.C_2[\varepsilon] \in L)$ has finite index too. (Recall that $\varepsilon = [\oslash, \oslash, \oslash]$). We have $u \sim_L v$ iff for some $C_1, C_2 \in Ct$ such that $u = C_1[\varepsilon] \bigwedge v = C_2[\varepsilon]$, $C_1 \approx_L^{\varepsilon} C_2$ holds. Then $\sim_L$ has finite index too.

Q.E.D.

4 Closure properties.

4.1 Boolean operators and morphisms.

In this section, we implicitely extend substitutions $\varphi : A \cup X \longrightarrow P(A, X)$ by $\varphi(\xi) = \xi$, and we denote by the same symbol φ their extention $\overline{\varphi} : Ct(\xi) \longrightarrow Ct(\xi)$ as defined in section 2.2.

4.1.1 Theorem.

Recognizable pomset languages are closed under boolean operators, alphabetic morphism and inverse morphism image.

Proof:

1) **Intersection**. Let $L_1, L_2 \in Rec(A,X)$,

$u^{-\xi}(L_1 \cap L_2) = \{C \in Ct, C[u] \in L_1 \cap L_2\} = u^{-\xi}L_1 \cap u^{-\xi}L_2$. As far as they are a finite number of sets $u^{-\xi}L_i, i = 1,2$, they are a finite number of their intersections, and $\sim_{L_1 \cap L_2}$has finite index.

2)**Complement**. $u^{-\xi}\overline{L} = \{C \in Ct, C[u] \notin L\} = \overline{u^{-\xi}L}$. The number of sets $u^{-\xi}\overline{L}$ and $u^{-\xi}L$ are equal , if $\sim_L$ has finite index, so does $\sim_{\overline{L}}$.

3)**Alphabetic morphism** . Let be $\varphi : A \cup X \longrightarrow A \cup X$.

Claim: $C[u] \in \varphi(L) \Longrightarrow \exists D \in Ct,\ v \in P(A,X), C = \varphi(D), u = \varphi(v)$. If $\varphi$is not alphabetic this not true: let be $L = \{a \bullet b\}$, and $\varphi(a) = \varphi(b) = a \bullet b$, with $C = a \bullet \xi$, $u = b \bullet a \bullet b$, then $C[u] \in \varphi(L)$ but $C \neq \varphi(D)$ for any context D.

$u^{-\xi}\varphi(L) = \{C \in Ct(\xi),\ C[u] \in \varphi(L)\} = \cup\{\varphi(D), D[v] \in L\}$ for all $v \in \varphi^{-1}(u)$.

$= \cup\{\varphi(v^{-\xi}L), v \in \varphi^{-1}(u)\}$. The number of sets $v^{-\xi}L$ being finite, the number of their images by φ is finite too, and thus the number of different unions of such sets.

4) **Inverse morphism.** Let be $\varphi : A \cup X \longrightarrow P(A,X)$.

Claim: If $C \in Ct,\ u \in P(A,X)$ then $\varphi(C[u]) = \varphi(C)[\varphi(u)]$(left to the reader: show that 1) $\varphi(u)$ is a refined subpomset of $\varphi(C[u])$, and 2) it occurs at the context $\varphi(C)$).

$u^{-\xi}\varphi^{-1}(L) = \{C \in Ct(\xi), C[u] \in \varphi^{-1}(L)\} = \{C \in Ct(\xi), \varphi(C[u]) \in L\} = \{C \in Ct(\xi), \varphi(C)[\varphi(u)]) \in L\} = \{C \in Ct(\xi), \varphi(C) \in \varphi(u)^{-\xi}L\} = \varphi^{-1}(\varphi(u)^{-\xi}L)$. The number of sets $v^{-\xi}L, v \in P(A,X)$ being finite, the number of their inverse images by φ is finite too.**Q.E.D**.

4.2 Pomset defined operations.

As mentionned in the introductions, the closure properties with respect to pomset defined operations are more involved to prove. The properties are that if $L \in Rec(A,X)$ and $L_i \in Rec(A,X)$,$i = 1..n$, then for any context C, we have $C[L] \in Rec(A,X)$, and for any pomset u, $u[L_1/x_1, .., L_n/x_n] \in Rec(A,X)$. The two proofs follow the same scheme.

4.2.1 Theorem.

Recognizable pomset languages are closed under context substitution.

$L \in Rec(A,X) \Longrightarrow \forall C \in Ct(\xi), C[L] \in Rec(A,X)$

Proof.

We show that there is an equivalence relation $\equiv$ on $P(A,X)$ with finite index, such that

$u \equiv v \Longrightarrow \forall D \in Ct, D[u] \in C[L] \Longleftrightarrow D[v] \in C[L]$. If $u \equiv v$ and $D[u] = C[s]$ for some $s \in L$, then we must have $D[v] = C[s']$for some $s' \in L$.

Let be $D[u] = C[s] = [E, \leq, l]$, with $s \in L$ and $C = [E_C \cup \{e_C\}, \leq_C, l_C]$ where $l_C(e_C) = \xi$ and $D = [E_D \cup \{e_D\}, \leq_D, l_D]$ where $l_D(e') = \xi$. We have (up to isomorphisms) $E = E_D \cup E_u = E_C \cup E_s$ where E_u and E_s are two refined subsets of the same partial order $(E, \leq)$. Then four situations may occur, depending on their relative positions, as depicted by the figure 3 below. For each of these situations we define an equivalence $\equiv_i$, $i = 1,4$ of finite index .

1. $E_u \cap E_s = \oslash$, then E_u is a refined subset of $(E_C \cup \{e_C\}, \leq_C)$such that $e_C \notin E_u$. Let be for each $u \in Ref(C) - Ct$, $[u]_{\equiv_1} = \{u\}$, and for $u \in P(A,X) - Ref(C)$, $[u]_{\equiv_1} = P(A,X) - Ref(C)$. The finiteness of $Ref(C)$ yields a finite number of classes $[u]_{\equiv_1}$. If $u \equiv_1 v$ then $D[v] = C[s]$.
2. $E_u \subset E_s$, in that case we have $s = G[u]$ for some $G \in Ct$, and $D = C.G$. We define $\equiv_2 = \sim_L$.If $u \equiv_2 v$ then $s' = G[v] \in L$, and thus $D[v] = C[s'] \in C[L]$.
3. $E_s \subset E_u$, then $u = G[s]$, $C = D.G$, where G is a refined subpomset of C which is a context. The equivalence $\equiv_3$ defined by $w \equiv_3 v \Longleftrightarrow \forall G \in Ref(C) \cap Ct, (w \in G[L] \Longleftrightarrow v \in G[L])$, has a finite index , $Ref(C)$ being finite. If $u \equiv_3 v$ then for some $s' \in L$, $v = G[s'] \in L$, and thus $D[v] = D.G[s'] = C[s'] \in C[L]$.
4. $E_u \cap E_s \neq \oslash$, $E_u - E_s \neq \oslash$and $E_s - E_u \neq \oslash$. This situation has three subcases A, B and C below, and the equivalence $\equiv_4$ is defined looking at each of them.

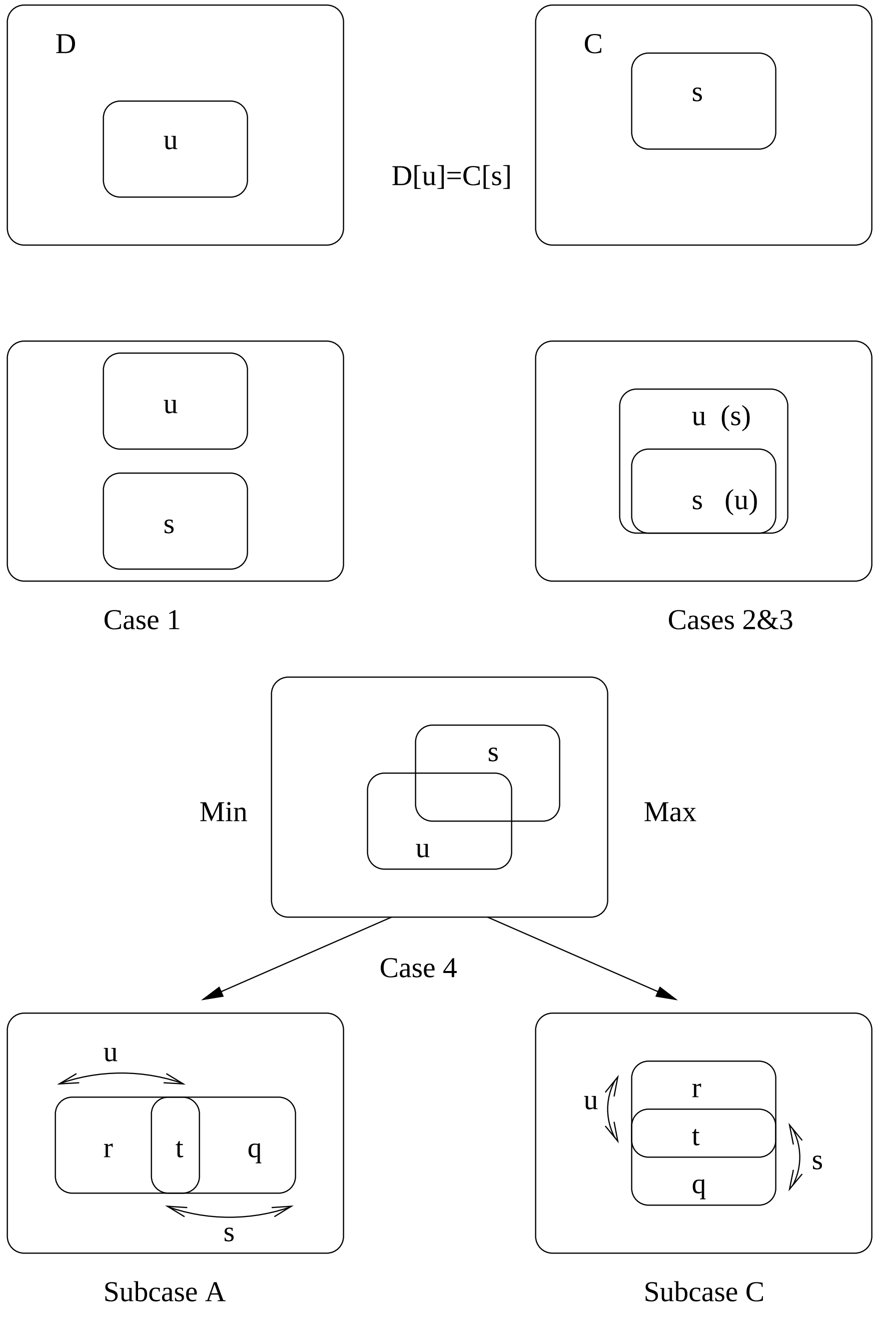

D
u
D[u]=C[s]
C
s
u
s
Case 1
u (s)
s (u)
Cases 2&3
Min
s
u
Max
Case 4
u
r
t
q
s
Subcase A
u
r
t
q
s
Subcase C

Figure 3.

A) $\exists e \in E_u - E_s$, $\exists e' \in E_s, e \leq e'$, and in that case $e \leq E_s$, because E_s is a refined subset. This implies that $\forall e'' \in E_s - E_u$, $E_u \leq e''$ because $e \leq e''$ and E_u is a refined subset and also $\forall e'' \in E_u - E_s$, $e'' \leq E_s$ because $\exists e''' \in E_s - E_u$, $e'' \leq e'''$ and E_s is a refined subset. We have then the set relation $(E_u - E_s) \leq (E_u \cap E_s) \leq (E_s - E_u)$, and any of $(E_u - E_s)$, $(E_u \cap E_s)$, $(E_s - E_u)$ is a refined subset. This can occur only if for some pomsets q, r, t, we have $u = r \bullet t$ and $s = t \bullet q$, and for some context G, $D = G[\xi \bullet q]$, $C = G[r \bullet \xi]$. The number of pomsets r such that $C = G[r \bullet \xi]$ for some context G is finite, and let be $R = \{r \in P(A, X), \exists G \in Ct,\ C = G[r \bullet \xi]\}$, we define $u \equiv_A v \Longleftrightarrow \forall r, t \in P(A, X), (u = r \bullet t \bigwedge r \in R \Longrightarrow v = r \bullet t' \bigwedge t \sim_L t')$. This equivalence has finite index bounded by $2^{**}(Card(R) \times Index(\sim_L))$.

B) $\exists e \in E_u - E_s$, $\exists e' \in E_s, e' \leq e$, this situation is related to the previous one by inversion of u and s, yielding the relation $(E_s - E_u) \leq (E_u \cap E_s) \leq (E_u - E_s)$. This can occur only iff for some pomsets q, r, t and context G, $s = r \bullet t$, $u = t \bullet q$, and $C = G[\xi \bullet q]$, $D = G[r \bullet \xi]$.

Let be $Q = \{q \in P(A, X), \exists G \in Ct,\ C = G[\xi \bullet q]\}$, which is finite, we define

$u \equiv_B v \Longleftrightarrow \forall t, q \in P(A, X), (u = t \bullet q \bigwedge q \in Q \Longrightarrow v = t' \bullet q \bigwedge t \sim_L t')$. This equivalence has finite index bounded by $2^{**}(Card(Q) \times Index(\sim_L))$.

C) $\exists e \in E_u - E_s$, $\exists e' \in E_s$, $e'\ co\ e$, and in that case $e\ co\ E_s$, because E_s is a refined subset.

This implies that $\forall e'' \in E_s - E_u$, $E_u\ co\ e''$ because $e\ co\ e''$ and E_u is a refined subset and also $\forall e'' \in E_u - E_s$, $e'' co E_s$ because $\exists e''' \in E_s - E_u$, $e'' co e'''$ and E_s is a refined subset. We have then the set relations $(E_u - E_s) co (E_u \cap E_s) co (E_s - E_u)$ and $(E_u - E_s)\ co\ (E_s - E_u)$, and the three sets are refined subsets. This can occur only iff for some pomsets q, r, t and context G, $u = r \parallel t$, and $s = t \parallel q$, $D = G[\xi \parallel r]$, $C = G[q \parallel \xi]$. Let be $S = \{q \in P(A, X), \exists G \in Ct,\ C = G[\xi \parallel q]\}$, which is finite, we define

$u \equiv_C v \Longleftrightarrow \forall r, t \in P(A, X), (u = t \parallel q \bigwedge q \in S \Longrightarrow v = t' \parallel q \bigwedge t \sim_L t')$. This equivalence has finite index bounded by $2^{**}(Card(S) \times Index(\sim_L))$.

We define $\equiv_4 = (\equiv_A \cap \equiv_B \cap \equiv_C)$, which has finite index.

The equivalence $\equiv = \cap\{\equiv_i, i = 1..4\}$ has finite index and is such that $u \equiv v \Longrightarrow \forall D \in Ct,\ (D[u] \in C[L] \Longleftrightarrow D[v] \in C[L])$. The proof of this last property is based on the way the equivalence has been constructed, considering that $D[u] \in C[L]$ yields one of the four situations described above.**Q.E.D.**

A pomset u is **linear** iff each variable occurs only once in u, i.e. $\forall x \in Var(u), Card(l^{-1}(x)) = 1$. Recognizable pomset languages are closed under linear pomset operation.

4.2.2 Theorem.

Let be $C \in P(A,X)$, then if C is linear ,

$\{L_i, i = 1..n\} \subset Rec(A,X) \Longrightarrow C[L_i/x_i] \in Rec(A,X)$

Proof. By induction on the arity of $C = [E, \leq, l]$, i.e. $\rho(C) = Card(Var(X))$. The case $\rho(C) = 1$ has been worked out in the previous theorem. We have to prove that $C[L_i/X_i] \in Rec(A,X)$ holds under the hypothesis that $\rho(D) < \rho(C)$ $\Longrightarrow D[L_i/x_i] \in Rec(A,X)$.

The proof follows the line of the previous one.

Suppose that $D[u] = C[s_i/x_i] = [E, \leq, l]$, with $s_i = [E_i, \leq_i, l_i] \in L_i$, $C = [E_C \cup \{e_C\}, \leq_C, l_C]$ where $l_C(e_C) = \xi$ and $D = [E_D \cup \{e_D\}, \leq_D, l_D]$ where $l_D(e') = \xi$, we have (up to isomorphisms) $E = E_D \uplus E_u = E_C \uplus (\cup_i E_i)$ where E_u and $\{E_i, i = 1..n\}$ are refined subsets of s, and where the sets E_i are disjoint. We construct an equivalence relation $\equiv$ following the lines of the previous proof, the four cases and their corresponding equivalences being submitted to the following changes:

1. $E_u \cap (\cup_i E_i) = \oslash$, then u is a refined subpomset of C with no occurrence of variables. The equivalence $\equiv_1$ is similar to the previous one, considering the elements of $Ref(C)$ where no variable appear.
2. $\exists k \in \{1..n\}$, $E_u \subseteq E_k$, this may occur only for a single k. We let $\equiv_2 = \cap\{\sim_{L_i}, i = 1..n\}$, which has finite index.
3. $\exists k \in \{1..n\}, E_k \subseteq E_u$, and $E_u \cap E_i \neq \oslash \Longrightarrow E_i \subseteq E_u$. The equivalence $\equiv_3$ is defined by $w \equiv_3 v \Longleftrightarrow \forall G \in Ref(C), (w \in G[L_i/x_i] \Longleftrightarrow v \in G[L_i/x_i])$, and has a finite index ($Ref(C)$ is finite).
4. $\exists k \in \{1..n\}$, $E_u \cap E_k \neq \oslash$, $E_u - E_k \neq \oslash$and $E_k - E_u \neq \oslash$. Compare to the previous theorem, new situations may occur, depicted in the figure 4.

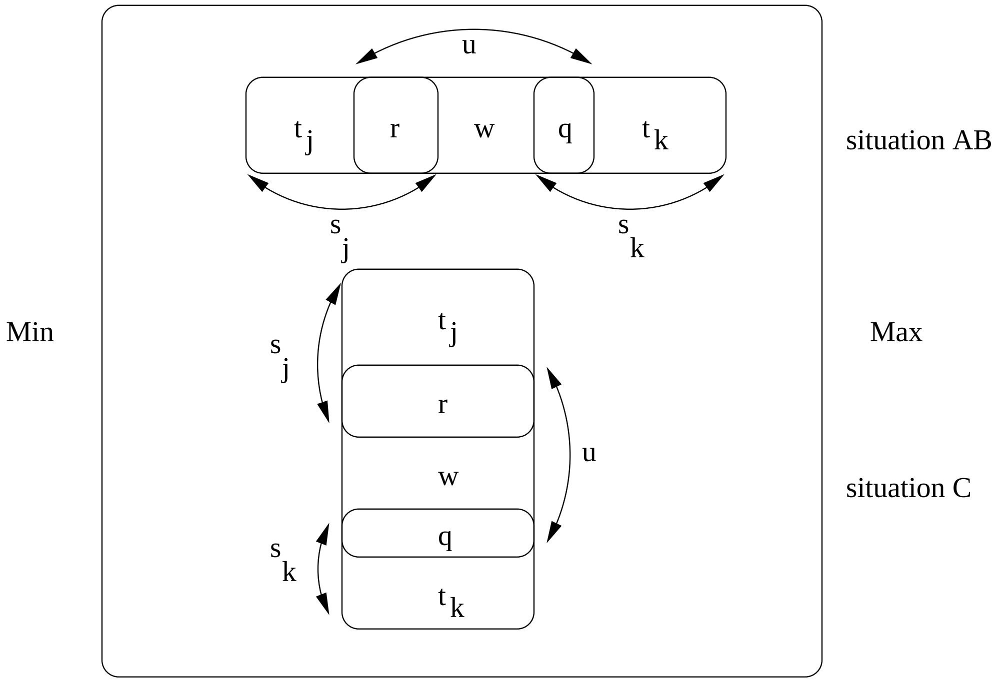


Figure 4.

Due to the disjunction of the sets E_i and the relationships between E_u and E_k as worked out in the previous theorem case 4.A,B,C, there is at most one index $j \neq k$ such that E_j satisfy the same property as E_k, and two situations may occur:

AB)For some pomsets t_j, t_k, r, p, q, w, we have $u = r \bullet p \bullet q$, $s_j = t_j \bullet r$ and $s_k = q \bullet t_k$, with $p = w[s_i/x_i]$ and $j, k \notin Var(w)$ (linearity of C), and for some context G, $D = G[t_j \bullet \xi \bullet t_k]$, $C = G[x_j \bullet w \bullet x_k]$. The number of pomsets w such that $C = G[x_j \bullet w \bullet x_k]$ for some context G and indexes j, k , is finite. Let be $W = \{w \in P(A, X), \exists G \in Ct, \exists j, k \in \{1..n\}\ C = G[x_j \bullet w \bullet x_k]\}$, the arity of any $w \in W$ is stricly lower than the one of C and the induction hypothesis applies, thus $\forall w \in W, w[L_i/x_i] \in Rec(A, X)$ and $w[L_i/x_i]$ has a finite index syntactic congruence denoted $\equiv_w$. Due to the finitness of W, the equivalence $\equiv_W = \cap\{\equiv_w,\ w \in W\}$has finite index. We define

$u \equiv_{AB} v \Longleftrightarrow (u = r \bullet p \bullet q \Longrightarrow v = r' \bullet p' \bullet q' \bigwedge p \equiv_W p' \bigwedge r \equiv_2 r' \bigwedge q \equiv_2 q')$.

Then with the hypothesis above, if $u \equiv_{AB} v$, we have $D[v] = G[t_j \bullet r' \bullet p' \bullet q' \bullet t_k]$, with

$r \sim_{L_j} r' \Longrightarrow t_j \bullet r' = s'_j \in L_j$,

$q \sim_{L_k} q' \Longrightarrow q' \bullet t_k = s'_k \in L_k$

$p \equiv_W p' \Longrightarrow p' = w[s'_i/x_i, i \notin \{j,k\}]$ where $s'_i \in L_i$, for all i such that $x_i \in Var(w)$.

thus $D[v] = G[s'_j \bullet w[s'_i/x_i] \bullet s'_k]$. Due to linearity of C, we have $Var(G) \cap (Var(w) \cup \{j,k\}) = \oslash$. Then if $s'_i = s_i$ for all $x_i \in Var(G)$, we have $D[v] = C[s'_i/x_i] \in C[L_i/x_i]$.

This equivalence has finite index bounded by $2^{**}(Index(\equiv_W) \times Index(\equiv_2) \times Index(\equiv_2))$.

C)In a similar way, using the induction hypothesis, we define a finite index equivalences $\equiv_C$, for the case C depicted in figure 4, using the set $W' = \{w \in P(A,X), \exists G \in Ct, \exists j,k \in \{1..n\}\ C = G[x_j \parallel w \parallel x_k]\}$ instead of W.

We define then the equivalence $\equiv_4 = (\equiv_{AB} \cap \equiv_C)$. Due to its construction, the equivalence $\equiv = \cap\{\equiv_i, i = 1..4\}$ has finite index and satisfies the required property, i.e. $u \equiv v \Longrightarrow \forall D \in Ct, (D[u] \in C[L_i/x_i] \Longleftrightarrow D[v] \in C[L_i/x_i])$.**Q.E.D.**

It follows that the closure property is valid for any pomset defined operation due to the definition 2.2.3 .

4.2.3 Corollary.

Let be $C \in P(A,X)$, then $\{L_i, i = 1..n\} \subset Rec(A,X) \Longrightarrow C[L_i/x_i] \in Rec(A,X)$

5 Conclusion: Two directions for future work.

Besides the generalisation of the closure properties , the caracterisation of recognisable languages by other means like acceptance by specific devices, in particular Petri-Nets, should be possible. Two particular cases of the framework presented here seem to be relevant.

5.1 Finitely generated pomset algebras.

Finitely generated pomset algebras are generated by a finite set of pomsets as follows:

Let be $\Gamma \subseteq P(A,X)$ a finite subset of open pomsets, the pomset algebra $P(\Gamma, A, X)$is the least subset of $P(A,X)$such that

1. $\Gamma \cup A \cup X \subset P(\Gamma, A, X)$

2. $C(x_1,..,x_n), C_1,..,C_n \in P(\Gamma, A, X) \Longrightarrow C[C_i/x_i] \in P(\Gamma, A, X)$

The closed algebra $P(\Gamma, A)$ is the set of closed pomsets of $P(\Gamma, A, X)$, $P(\Gamma, A) = P(\Gamma, A, X) \cap PW(A)$.

Example: The algebra of serie -parallel pomsets is $P(\Gamma, A)$with $\Gamma = \{x_1 \bullet x_2, x_1 \parallel x_2\}$.

The definition of contexts and syntactic congruence $\sim_L$, can be easily adapted from the framework. The set of recognisable languages of $P(\Gamma, A, X)$, is denoted $Rec(\Gamma, A, X)$, and all properties presented in the paper are valid for $Rec(\Gamma, A, X)$. The finite number of patterns in such algebras should yield specific caracterizations of $Rec(\Gamma, A, X)$, hopefully in terms of regular systems of equations.

5.2 Initial context and associated congruence.

Initial contexts and initial syntactic congruence seem to be a possible bridge towards pomset acceptance, in particular asynchronous cellular automata . A context $[E, \leq, l]$is called *initial* iff its ξ-labelled event is minimal in $(E, \leq)$. The **set of initial contexts** is

$ICt(\xi) = \{C = [E, \leq, l] \in Ct(\xi),\ l^{-1}(\xi) \in Min(E, \leq)\}$. It constitutes a submonoid of $(Ct, ., \xi)$, and we can define a corresponding syntactic equivalence for pomset languages. Let be $L \subseteq P(A, X)$, the **initial syntactic equivalence** of L is the relation $\sim_L^i \subseteq P(A, X) \times P(A, X)$ defined by

$$u \sim_L^i v \Longleftrightarrow \forall C \in ICt(\xi) : (C[u] \in L \Longleftrightarrow C[v] \in L)$$

An obvious property is that if $\sim_L$ has finite index, then $\sim_L^i$ has too. We are not able now to conjecture whether the two equivalences define the same class of recognisable languages.

Acknowledgments: Thanks to Marc Boyer , Bernard Berthomieu and Michel Diaz for the inprovements due to their comments and examples.

References

[Co] Courcelle.B. *Graph rewriting: an algebraic and logical approach.* in Handbook of Theoretical Computer Science, vol.B, J.B.van Leeuwen, Elsevier,N.Y.

[DK] Droste.M,Kuske.D. *Automata with concurrency relations, a survey.* Workshop on Algebraic and Syntactic Aspects of Concurrency.P.Gastin and A.Petit eds. LITP 95/48,1995.

[Gi] Gischer. J. *The equational theory of pomsets.* Theoretical Computer Science 61 (1988)199-224.

[GS] Gecseg.F,Steinby.M. *Tree languages.* Handbook of Formal Languages vol.3. Springer-Verlag 1997.

[Ku] Kuske,D. *Asynchronous Cellular and Asynchronous Automata for Pomsets.* Lecture Notes in Computer Science, Vol 1466, CONCUR 98, 517-523, Springer -Verlag 1998.

[LW] Lodaya.K, Weil.P. *Series-parallel pomsets: Algebra, Automata and Languages.* STACS 98, Lecture Notes in Computer Science, Vol. 1373, Springer -Verlag 1998.

[Pr] Pratt.W. *Modeling concurrency with partial orders.* Int.J.Parallel Programming 15 (1987) 33-71.

[Th] Thomas.W. *Automata theory on Trees and Partial Orders.* Lecture Notes in Computer Science, Vol. 1214, TAPSOFT 97,Springer -Verlag 1998.