

SYNTACTICAL PROPERTIES OF UNBOUNDED NETS OF PROCESSORS

J. BEAUQUIER ⁽¹⁾, A. CHOQUET ⁽¹⁾, A. PETIT ^(1,2), G. VIDAL-NAQUET ^(1,3)

⁽¹⁾ L.R.I. Bât. 490, Université Paris-Sud, 91405 ORSAY FRANCE.

⁽²⁾ L.I.F.O., Université d'Orléans, 45067 ORLEANS FRANCE.

⁽³⁾ Ecole Supérieure d'Electricité, 91190 GIF SUR YVETTE FRANCE.

Abstract. We present a formal description of the logical links in an unbounded net of processors. This description is provided by a finite transducer. We prove some syntactical properties of the net : we give decision algorithms for its coherence and connectivity.

0. Introduction

In almost every paper on nets of processors, the number of considered processors is bounded. For instance, in systolic or neural architectures (cf. [Quinton...86] and [Kohonen...88] for an introduction), the number of processors, even if arbitrarily large, is nearly always considered to be given. Moreover, the topology of the nets is always very simple and regular : lines, grids,...

In this paper, we consider nets, in which the number of processors, although finite, has no known bound. In fact, that involves the assumption that the net is possibly infinite. And we deal with nets of arbitrary (regular) topology.

In such a net, a processor is physically connected to a finite number of other processors in a "crystalline" way. For example, on a plan : each processor is linked to four others ; in three dimensional space, it is linked to six, as shown in the figure 1.

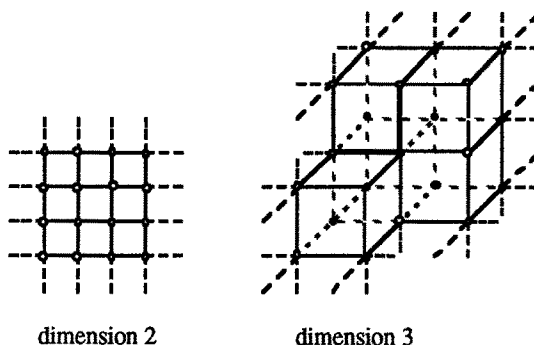


Figure 1.

The aim of this paper is to study some systems in which two processors that are physically connected can be logically connected in different ways. For instance, two processors can be not logically connected, or connected in a synchronous or an asynchronous way. We want to develop a specification allowing to formally prove properties about such systems. The model must take into account the unboundness of the number of processors and, at the opposite, provide a finite specification in order to be operational. The problem is thus to describe a possibly infinite object by a finite mechanism.

The method that we use associates a logical path, i.e. a sequence of logical connections, to each physical path. Such a mechanism is known in the literature as a transducer (cf. [Berstel...79]) and our requirements make that this transducer must be finite. Namely, we associate to a particular point of the net, called the starting point, a finite transducer that maps physical paths starting from him onto logical paths.

For instance, in a two dimensional space, a physical path from a particular point may be described as : north, north, west, north and could be related to a logical path as : (north, synchronous), (north, asynchronous), (west, asynchronous), (north, not connected).

Clearly enough, such a finite transducer does not always specify in an unique way the logical links of a net of processors: since several paths from the starting point can use a same physical link, the logical descriptions of this link can be different according to the considered path. This remark leads to impose some coherence constraints to the specification. We propose some of them and we prove that they can all be decided from a finite transducer.

1. Basic definitions

Throughout the paper, we will assume the classical definitions of formal language theory to be known by the reader. Also the notations are classical (for more details, see e.g. [Berstel...79] or [Hopcroft...69]).

A - Physical and logical links.

We introduce an alphabet of directions. A letter indicates which direction is taken, a word on this alphabet corresponds to a sequence of moves.

Such an alphabet is of size $2q$, $\mathcal{D} = \{dir_1, \dots, dir_{2q}\}$ and for each i , dir_{2i-1} and dir_{2i} will be interpreted later as opposite directions. In other words : if we move in direction dir_{2i-1} and then in direction dir_{2i} , we come back to the start point, and conversely.

Examples.

We will give 2 examples for \mathcal{D} :

▷ In plane programming, each processor is located on a point on a plan.

In this case, $\mathcal{D} = \{N(orth), S(outh), E(ast), W(est)\}$.

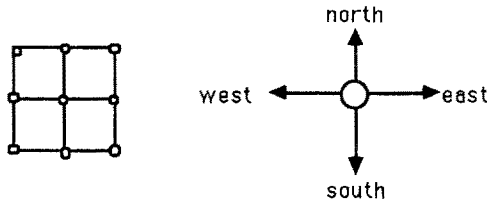


Figure 2.

▷ The processors can be located in the space at the tops of a cube.

In this case, $\mathcal{D} = \{U(p), D(own), F(orward), B(ackward), L(eftrightarrow), R(ight)\}$

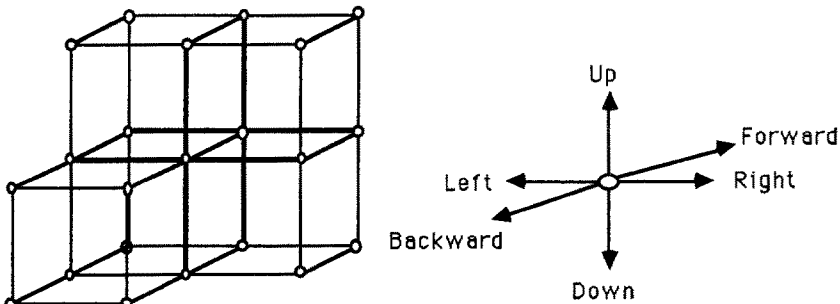


Figure 3.

On the other hand, we have a connexion alphabet \mathcal{N} that describes the nature of the logical links between two processors that are physically connected.

Let us also give two examples for \mathcal{N} .

▷ The basic and simplest example is an alphabet of size 2, that just says if the connexion is open or closed :

$\mathcal{N} = \{C(onnected), D(isconnected)\}$.

▷ We can also make distinctions between the different kinds of connexions. For instance we can have synchronous or asynchronous connexions :

$\mathcal{N} = \{D(isconnected), A(synchronous), S(ynchronous)\}$.

To a physical link we want to associate one (at least) logical nature. But there are an infinity of different physical links so this association cannot be done with a table. As stated in the introduction the mapping between physical and logical links is described by a (finite) strictly alphabetic transducer.

Example.

$\mathcal{D} = \{n, s, e, w\}$, $\mathcal{N} = \{c, d\}$ and the transducer is given by the following scheme.

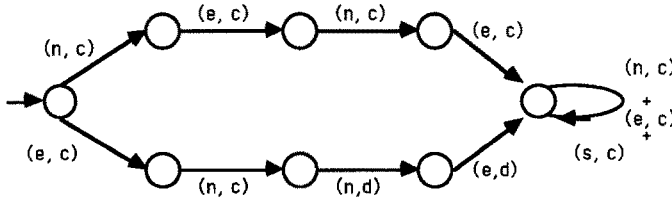


Figure 4.

It is well known that to give a (finite) strictly alphabetic transducer, which transform a word of \mathcal{D}^* in a word of \mathcal{N}^* , is equivalent to give a rational language of $(\mathcal{D} \times \mathcal{N})^*$.

In the example above, the associated rational language is :

$L = \{(n, c)(e, c)(n, c)(e, c) + (e, c)(n, c)(n, d)(e, d)\} \{(n, c) + (e, c) + (s, c)\}^*$.

Throughout this paper, we will quite always consider, instead of the finite strictly alphabetic transducer, the associated rational language. Moreover, this language will be supposed to be prefix-closed.

B - Some morphisms on languages.

In order to study properties of physical and logical paths induced by this language, we have to introduce some morphisms over $(\mathcal{D} \times \mathcal{N})^*$.

a - The first projection on \mathcal{D}^* gives the physical path that is followed.

$$\begin{aligned} p_{\mathcal{D}} : (\mathcal{D} \times \mathcal{N})^* &\longrightarrow \mathcal{D}^*, \\ (dir_1, nat_1) \dots (dir_r, nat_r) &\longrightarrow dir_1 \dots dir_r. \end{aligned}$$

For instance, in the example above, $p_{\mathcal{D}}((e, c)(n, c)(n, d)(e, d)(s, c)) = ennes$.

b - The second projection on \mathcal{N}^* give the nature of the connexions found along the followed path.

$$\begin{aligned} p_{\mathcal{N}} : (\mathcal{D} \times \mathcal{N})^* &\longrightarrow \mathcal{N}^*, \\ (dir_1, nat_1) \dots (dir_r, nat_r) &\longrightarrow nat_1 \dots nat_r. \end{aligned}$$

In the example above, $p_{\mathcal{N}}((e, c)(n, c)(n, d)(e, d)(s, c)) = ccdds$.

c - The Parikh's (or commutative) image [Parikh...66] of this path gives the number of moves in each direction.

$$\begin{aligned} P : (\mathcal{D} \times \mathcal{N})^* &\longrightarrow \mathbb{N}^{2q}, \\ w &\longrightarrow (lp_{\mathcal{D}}(w)_{dir_1}, \dots, lp_{\mathcal{D}}(w)_{dir_{2q}}). \end{aligned}$$

In the example above, $P((e,c)(n,c)(n,d)(e,d)(s,c)) = (2, 1, 2, 0)$. Two words have the same Parikh's image when their projections on \mathcal{D} are permutations of each other.

d - The morphism μ gives the coordinates of the point in \mathbb{Z}^q reached after a path w . It reduces the Parikh's image by eliminating each occurrence of any direction that is counterbalanced by an occurrence of the opposite direction.

$$\begin{aligned} \mu: (\mathcal{D} \times \mathcal{N})^* &\longrightarrow \mathbb{Z}^q, \\ w &\longrightarrow (|p_{\mathcal{D}}(w)|_{\text{dir}_1} - |p_{\mathcal{D}}(w)|_{\text{dir}_2}, \dots, |p_{\mathcal{D}}(w)|_{\text{dir}_{2q-1}} - |p_{\mathcal{D}}(w)|_{\text{dir}_{2q}}). \end{aligned}$$

In the example above, $\mu((e,c)(n,c)(n,d)(e,d)(s,c)) = (1, 2)$.

Intuitively, two paths have the same image by this morphism iff they lead to the same point. This means that we are now working in the free group, instead of working in the free monoid.

We then introduce an equivalence relation \approx on $(\mathcal{D} \times \mathcal{N})^*$: two words are equivalent if they lead to the same point, from a given point, and this is obviously decidable. Formally :

$$u \approx v \Leftrightarrow \mu(u) = \mu(v).$$

In the example above, $(e,c)(n,c)(n,d)(e,d)(s,c)(n,c)(s,c) \approx (e,c)(n,c)(e,d)$. This relation is clearly an equivalence relation ; moreover, it is a congruence.

We will also use the morphism ϕ , from \mathbb{N}^{2q} on \mathbb{Z}^q , defined by $\mu = \phi \circ P$.

Let L be a language over $(\mathcal{D} \times \mathcal{N})^*$, and (dir, nat) an element of $(\mathcal{D} \times \mathcal{N})$.

$L(\text{dir}, \text{nat})^{-1} = \{f \in (\mathcal{D} \times \mathcal{N})^* / f(\text{dir}, \text{nat}) \in L\}$ is the set of all words of L that can be completed, in L , by (dir, nat) .

C - Semi-linear sets.

Let us recall the definition of a semi-linear set.

Definition 1.

$P \subseteq \mathbb{Z}^q$ is linear iff there are a_0, a_1, \dots, a_k in \mathbb{Z}^q such that

$$P = a_0 + \mathbb{N}.a_1 + \dots + \mathbb{N}.a_k.$$

P is semi-linear iff P is a finite union of linear sets.

The link between context-free (and rational) languages and semi-linear sets is given by the following result.

Proposition 1.[Ginsburg...66]

Let L be a context-free language. Then $P(L)$ and $\mu(L)$ are semi-linear sets that can be effectively constructed.

In the sequel, we will use the following fundamental results on semi-linear sets several times.

Proposition 2.[Ginsburg...66]

Let S and S' be semi-linear sets.

- 1 - The emptiness of S is a decidable problem.
- 2 - $S \cap S'$ is a semi-linear set that can be effectively constructed.

2 - Notions of coherence

With the above morphisms we can define precisely the way the logical links of the net are specified by the language of $(\mathcal{D} \times \mathcal{N})^*$. Let M be a point of the physical net defined by its components α in $\mu((\mathcal{D} \times \mathcal{N})^*)$ and dir a direction in \mathcal{D} . The physical link issued from M and of direction dir can have "nat" as logical value if there is some $w(\text{dir}, \text{nat})$ in L such that $\mu(w) = \alpha$.

A - Definitions.

Clearly enough, any language does not necessarily provide an unique logical nature for a given physical link. Since two distinct physical paths can use a same link, this link does not necessarily receive the same logical interpretation in the images of the two paths given by the language.

Now we want to determine some properties on the specification (i.e. the language) that yield properties of the logical links. The strongest property that we can require is the 0-strong coherence: an unique logical link is associate to a given physical link. It is a rather strong property and languages satisfying it can be hard to obtain. So that, in order to make the specification simpler to express, we introduce less constrained forms of coherence. Languages satisfying them are easier to build, but a finite number of physical links may have several associated logical natures.

Strong coherence specify a net everywhere, excepted in the neighbourhood of the starting point.

Definition 2.

Let k be an integer and L a language over $(\mathcal{D} \times \mathcal{N})^*$. We say that L is *strongly coherent with respect to k* , and denote it by $L \in SC(k)$ if and only if :

$$[\forall u, v \in L, u(\text{dir}, \text{nat}) \in L, v(\text{dir}, \text{nat}') \in L, u \approx v, |u| \geq k, |v| \geq k] \Rightarrow \text{nat} = \text{nat}'.$$

We define $SC = \bigcup_{k=0}^{\infty} SC(k)$.

A less constrained form of coherence can be defined in the following way:

Definition 3.

Let k be an integer and L a language over $(\mathcal{D} \times \mathcal{N})^*$. We say that L is *weakly coherent with respect to k* , and denote it by $L \in wc(k)$ iff :

$$\begin{array}{l} u_0(\text{dir}, \text{nat}_1)u_1(\text{dir}, \text{nat}_2)\dots(\text{dir}, \text{nat}_k)u_k(\text{dir}, \text{nat}) \in L \\ v_0(\text{dir}, \text{nat}'_1)v_1(\text{dir}, \text{nat}'_2)\dots(\text{dir}, \text{nat}'_k)v_k(\text{dir}, \text{nat}') \in L \\ u_0 \approx v_0, (\text{dir}, \text{nat}_i)u_i \approx (\text{dir}, \text{nat}'_i)v_i \approx \varepsilon, \forall i \in 1\dots k. \end{array} \quad \left| \quad \Rightarrow \text{nat} = \text{nat}' \right.$$

(where ε denotes the empty word).

We define $wc = \bigcup_{k=0}^{\infty} wc(k)$.

Remark. Obviously, $SC(0) = wc(0)$.

As direct consequences of the definitions, we have the following hierarchical relations on these sets :

Proposition 3.

$$L \in SC(k) \Rightarrow L \in SC(k+1).$$

$$L \in wc(k) \Rightarrow L \in wc(k+1).$$

Note that the reverse implications are false, as shown by the two following examples.

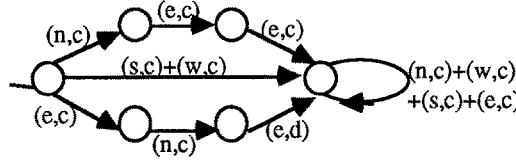


Figure 5.

The language accepted by the above transducer (every state is terminal) is 3-SC, but not 2-SC, since $(n, c)(e, c)(e, c) \in L$, $(e, c)(n, c)(e, d) \in L$, and $(n, c)(e, c) \approx (e, c)(n, c)$.

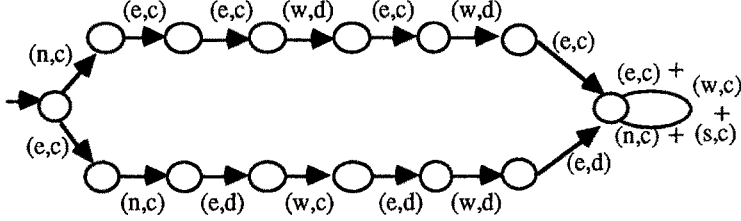


Figure 6.

The language accepted by this transducer (every state is terminal) is 3-wc but not 2-wc.

We also have a relation between the weak and the strong coherence.

Proposition 4.

$$L \in SC(k) \Rightarrow L \in wc(k \div 2).$$

Indeed, $(dir, nat_i)u_i \approx \varepsilon$ implies $|dir, nat_i)u_i| \geq 2$, thus $|u_0(dir, nat_1)u_1 \dots (dir, nat_{k \div 2})u_{k \div 2}| \geq k$, and the same holds for v .

But the reverse implication is false, as shown by the following counter-example :

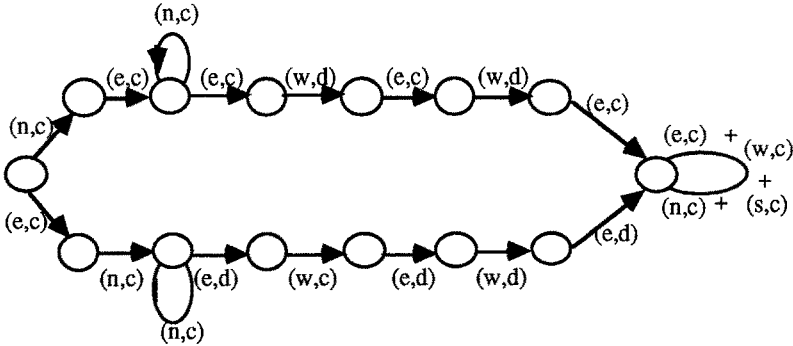


Figure 7.

This language is still 3-wc, but is not k -SC, for any k .

The natural question that arises then is : is it possible to decide, for a given k , whether a language is k weak or strong coherent. Furthermore, can we decide if there is some k for which the language is k weak or strong coherent. Our purpose is now to show that we are able to decide most of these properties.

B. The strong coherence.

We always assume that L is a rational set, but the results of this section can be extended to context free languages.

We are able to decide whether the language L is strongly coherent or not.

Theorem 1.

Let L be a rational language of $(\mathcal{D} \times \mathcal{N})^*$. It is decidable whether L belongs to $SC(0)$ or not.

Proof.

$$L \in SC(0) \Leftrightarrow \left| \begin{array}{l} u(\text{dir}, \text{nat}) \in L \\ v(\text{dir}, \text{nat}') \in L \\ u \approx v \end{array} \right| \Rightarrow \text{nat} = \text{nat}'$$

$$\Leftrightarrow [\text{nat} \neq \text{nat}' \Rightarrow \text{There is no couple } (u, v) \text{ such that } \left| \begin{array}{l} u(\text{dir}, \text{nat}) \in L. \\ v(\text{dir}, \text{nat}') \in L. \\ u \approx v. \end{array} \right|$$

$$\Leftrightarrow [\text{nat} \neq \text{nat}' \Rightarrow \mu(L(\text{dir}, \text{nat})^{-1}) \cap \mu(L(\text{dir}, \text{nat}')^{-1}) = \emptyset].$$

$L(\text{dir}, \text{nat})^{-1}$ is a rational set then $\mu(L(\text{dir}, \text{nat})^{-1})$ is a semi-linear set (Proposition 1). So we have reduce the problem to the emptiness of the intersection of two semi-linear sets. As we recall it, this problem is decidable (Proposition 2). 🍏

It is well known that the emptiness of the intersection of two (semi) linear sets is an NP-complete problem [Karp...72]. One can ask if there exists a solution to our problem with a lower complexity. In fact, our problem is equivalent to the above problem of emptiness. We have the following proposition.

Proposition 5.

The problem of deciding whether a rational language is in $SC(0)$ is NP-complete.

Proof.

Let $A = a_0 + \mathbf{N}.a_1 + \dots + \mathbf{N}.a_k$ and $B = b_0 + \mathbf{N}.b_1 + \dots + \mathbf{N}.b_k$ be two linear sets. In order to simplify the notations, we will suppose that a_i and b_i are in \mathbf{N}^2 . A proof in the general case can be straightforward deduced from the proof in the case $n = 2$.

Thus we define $a_i = (x_i, y_i)$ and $b_j = (z_j, t_j)$ with x_i, y_i, z_j and t_j in \mathbf{N} .

We introduce a direction alphabet $\mathcal{D} = \{n, s, e, w, u, p\}$ and a connexion alphabet $\mathcal{N} = \{\text{nat}, \text{nat}', \text{nat}''\}$.

We will construct a transducer (and thus a language L on $\mathcal{D} \times \mathcal{N}$) such that the 0- strongly coherence of L is equivalent to the emptiness of $A \cap B$.

Let us consider the following transducer :

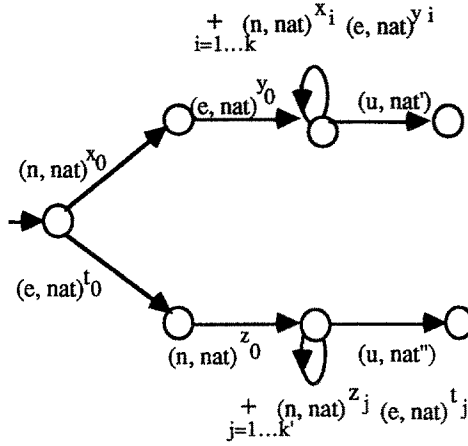


Figure 8.

It is then obvious that the language defined by this transducer is 0-strongly coherent if and only if $\mu [L(u, nat')^{-1}]$ and $\mu [L(u, nat'')^{-1}]$ have an empty intersection. In other words, L is 0-strongly coherent if and only if $A \cap B = \emptyset$ and thus, the Proposition 5 is proved. 🍏

Corollary.

It is decidable whether L belongs to $SC(k)$ for a given integer k , and this is also an NP-complete problem.

Proof.

Indeed, we have $L \in SC(k) \Leftrightarrow [(\mathcal{D} \times \mathcal{N})^k]^{-1}L \in SC(0)$. 🍏

Now suppose that only the language is given. Can we find an integer k such that L belongs to $SC(k)$? That is to say, can we prove that the language belongs to SC ? Again, we can decide the belonging to SC or not.

Theorem 2.

Let L be a rational language of $(\mathcal{D} \times \mathcal{N})^*$. It is decidable whether L belongs to SC or not.

Proof.

$$L \notin SC \Leftrightarrow \forall k \geq 0, \quad \left\{ \begin{array}{l} \exists (nat, nat') \in \mathcal{N}^2, nat \neq nat'. \\ \exists dir \in \mathcal{D}. \\ \exists (u, v) \in [(\mathcal{D} \times \mathcal{N})^*]^2. \end{array} \right. \quad \text{such that}$$

$$[I] \quad \left\{ \begin{array}{l} u(dir, nat) \in L. \\ v(dir, nat') \in L. \\ u \approx v. \\ |u| \geq k, |v| \geq k. \end{array} \right.$$

$$\Leftrightarrow [II] \text{ (since } \mathcal{N} \text{ and } \mathcal{D} \text{ are finite)} \exists (nat, nat') \in \mathcal{N}^2, nat \neq nat', \exists dir \in \mathcal{D} \text{ such that,} \\ \forall k \geq 0, \exists (u, v) \in [(\mathcal{D} \times \mathcal{N})^*]^2 \text{ such that } [I] \text{ holds.}$$

We need to introduce the following sets :

for $(dir, nat, nat') \in (\mathcal{D} \times \mathcal{N}^2)$ with $nat \neq nat'$, we define $E_{dir, nat, nat'} = \mu (L(dir, nat)^{-1}) \cap \mu (L(dir, nat')^{-1})$.

→ 1st case: $\exists (\text{dir}, \text{nat}, \text{nat}') \text{ such that } E_{\text{dir}, \text{nat}, \text{nat}'}$ is infinite.

There exists then an infinite sequence (m_1, \dots, m_r, \dots) in $E_{\text{dir}, \text{nat}, \text{nat}'}$ such that $\|m_i\| \leq \|m_{i+1}\|$ for every i (with, if $m = (\mu_1, \dots, \mu_q)$, $\|m\| = \sum_{i=1 \dots q} |\mu_i|$).

Thus, we can find two sequences (u_i) in $L(\text{dir}, \text{nat})^{-1}$ and (v_i) in $L(\text{dir}, \text{nat}')^{-1}$ such that $\mu(u_i) = \mu(v_i) = m_i$. We have $|u_i| \geq \|m_i\|$, $|v_i| \geq \|m_i\|$, and $u_i \approx v_i$.

Thus, $\forall k \in \mathbb{N}$, $\exists u_k \in L(\text{dir}, \text{nat})^{-1}$, $\exists v_k \in L(\text{dir}, \text{nat}')^{-1}$, $|u_k| \geq k$, $|v_k| \geq k$, $u_k \approx v_k$. It follows that L is not SC.

→ 2nd case: Every $E_{\text{dir}, \text{nat}, \text{nat}'}$ is finite.

For every $E_{\text{dir}, \text{nat}, \text{nat}'}$, we define, for each $m \in E_{\text{dir}, \text{nat}, \text{nat}'}$, $D_1 = P(L(\text{dir}, \text{nat})^{-1}) \cap \phi^{-1}(m)$ and

$D_2 = P(L(\text{dir}, \text{nat}')^{-1}) \cap \phi^{-1}(m)$. These two sets are semi-linear and actually computable. Moreover, the number of $E_{\text{dir}, \text{nat}, \text{nat}'}$ is finite and each of them is a finite set, thus, there is a finite number of sets D_1 and D_2 .

We can then distinguish two sub-cases :

→ 1st subcase : There is some m in some $E_{\text{dir}, \text{nat}, \text{nat}'}$ such that D_1 and D_2 are infinite.

Let (t_k) and (t'_k) be two increasing (for $\| \cdot \|$) sequences respectively of D_1 and D_2 . Then, there exists two sequences (u_k) in $L(\text{dir}, \text{nat})^{-1}$ and (v_k) in $L(\text{dir}, \text{nat}')^{-1}$ such that :

$$\left| \begin{array}{l} t_k = P(u_k). \\ t'_k = P(v_k). \\ |u_k| \text{ and } |v_k| \text{ are increasing sequences.} \\ u_k \approx v_k \approx m. \end{array} \right.$$

And thus, $L \notin \text{SC}$.

→ 2nd sub-case : For each $E_{\text{dir}, \text{nat}, \text{nat}'}$, for each m in $E_{\text{dir}, \text{nat}, \text{nat}'}$, D_1 or D_2 are finite. We will prove by absurd that L is in SC.

In fact, suppose that $L \notin \text{SC}$: Let $(\text{dir}, \text{nat}, \text{nat}')$ be such that [II] holds, and let (u_k) and (v_k) be two sequences such that [I] holds. Since $E_{\text{dir}, \text{nat}, \text{nat}'}$ is finite, there is some m and some subsequences (u'_k) and (v'_k) such that [I] holds, $u'_k \approx v'_k \approx m$, and $|u'_k|$ and $|v'_k|$ are strictly increasing sequences. Then the sequences $(P(u'_k))$ and $(P(v'_k))$ are strictly increasing to, and $P(u'_k) \in D_1$, $P(v'_k) \in D_2$, that are therefore both infinite, which is a contradiction with the hypothesis. \blacksquare

C. The weak coherence.

In this section, we will suppose that the language L is given by a finite deterministic automaton with N states, $\mathcal{A} = (Q, (\mathcal{D} \times \mathcal{N}), \delta, q_0, F)$.

Theorem 3.

Let L be a rational language of $(\mathcal{D} \times \mathcal{N})^*$. It is decidable whether L belongs to $\text{wc}(k)$ or not.

Proof.

Let q_1 and q_2 be two states of the automaton \mathcal{A} , and d a direction of \mathcal{D} . We introduce the set of paths leading from q_1 to q_2 , starting with the direction d .

$L(q_1, q_2, d) = \{(\text{dir}, \text{nat})w \mid \delta(q_1, (\text{dir}, \text{nat})w) = q_2\}$. This set is obviously rational.

Suppose that there exist two words u and v implying that L is not k -weakly coherent.

$u = u_0(\text{dir}, \text{nat}_1)u_1(\text{dir}, \text{nat}_2)\dots u_{k-1}(\text{dir}, \text{nat}_k)u_k(\text{dir}, \text{nat})$.

$v = v_0(\text{dir}, \text{nat}'_1)v_1(\text{dir}, \text{nat}'_2)\dots v_{k-1}(\text{dir}, \text{nat}'_k)v_k(\text{dir}, \text{nat}')$.

with

$$(I) \quad \left| \begin{array}{l} u_0 \approx v_0 \\ (\text{dir}, \text{nat}_i)u_i \approx (\text{dir}, \text{nat}'_i)v_i \approx \varepsilon \\ \text{nat} \neq \text{nat}' \end{array} \right.$$

Then there exist two $(k+1)$ -tuples of states $(q_1, q_2, \dots, q_{k+1})$ and $(q'_1, q'_2, \dots, q'_{k+1})$, such that we have:

$$u = u_0(\text{dir}, \text{nat}_1)u_1(\text{dir}, \text{nat}_2) \dots u_{k-1}(\text{dir}, \text{nat}_k)u_k(\text{dir}, \text{nat}).$$

$$\begin{array}{ccccccc} \uparrow & \uparrow & & \uparrow & & \uparrow & \uparrow \\ q_0 & q_1 & & q_2 & & q_k & q_{k+1} \end{array}$$

$$v = v_0(\text{dir}, \text{nat}'_1)v_1(\text{dir}, \text{nat}'_2) \dots v_{k-1}(\text{dir}, \text{nat}'_k)v_k(\text{dir}, \text{nat}').$$

$$\begin{array}{ccccccc} \uparrow & \uparrow & & \uparrow & & \uparrow & \uparrow \\ q'_0 & q'_1 & & q'_2 & & q'_k & q'_{k+1} \end{array}$$

Moreover, the conditions (I) imply:

$$(II) \quad \left| \begin{array}{l} \mu(\{u / \delta(q_0, u) = q_1\}) \cap \mu(\{u / \delta(q_0, u) = q'_1\}) \neq \emptyset \\ 0 \in \mu(L(q_i, q_{i+1}, \text{dir})) \\ 0 \in \mu(L(q'_i, q'_{i+1}, \text{dir})) \end{array} \right.$$

Conversely, if two $(k+1)$ -tuples of states $(q_1, q_2, \dots, q_{k+1})$ and $(q'_1, q'_2, \dots, q'_{k+1})$ verify that there exist a direction dir and two natures nat and nat' , such that:

$$(III) \quad \left| \begin{array}{l} (II) \\ \delta(q_{k+1}, (\text{dir}, \text{nat})) \text{ is defined} \\ \delta(q'_{k+1}, (\text{dir}, \text{nat}')) \text{ is defined} \end{array} \right.$$

it is obvious that the language L is not k -weakly coherent.

But if elements $(q_1, q_2, \dots, q_{k+1})$, $(q'_1, q'_2, \dots, q'_{k+1})$, dir , nat and nat' are given, the conditions (III) are decidable (Proposition 2). Moreover, there is a finite number of such elements, thus we can decide whether the language L is k -weakly coherent or not. \blacktriangle

In the case where L is a rational language (as it is assumed in this section), the increasing sequence $(\text{wc}(k))$ is stationary:

Proposition 6.

$\forall k \geq N + 1, L \in \text{wc}(k) \Rightarrow L \in \text{wc}(N)$. (where N is the number of states of \mathcal{Q})

Proof.

Let us consider a word u such that :

$$\begin{array}{ccccccc} u = u_0(\text{dir}, \text{nat}_1)u_1(\text{dir}, \text{nat}_2) \dots u_{N-1}(\text{dir}, \text{nat}_N)u_N(\text{dir}, \text{nat}). \\ \uparrow & \uparrow & & \uparrow & & \uparrow & \\ q_1 & q_2 & & q_N & & q_{N+1} \end{array}$$

We run through the states q_1, q_2, \dots, q_{N+1} . We find, since there are only N different states, two integers $i < j$ such that $q_i = q_j$.

Moreover, we have $\mu((\text{dir}, \text{nat}_i)u_i \dots (\text{dir}, \text{nat}_{j-1})u_{j-1}) = \varepsilon$. Therefore, this loop can be iterated, so that we get a word :

$\hat{u} = u_0(\text{dir}, \text{nat}_1) \dots u_{i-1}[(\text{dir}, \text{nat}_{i-1})u_{i-1} \dots u_{j-1}]^s(\text{dir}, \text{nat}_j) \dots u_{N-1}(\text{dir}, \text{nat}_N)u_N(\text{dir}, \text{nat})$ with $N + (s - 1)(j - i) \geq k$.

We proceed in the same way with v .

From the k -weak confluence, we then get $\text{nat} = \text{nat}'$, and thus, L is N -wc. \blacktriangle

Thus, to decide whether a rational language L is weakly coherent, it suffices to test whether L is in $\cup_{i=1 \dots NWC(i)}$, which is decidable from Theorem 3. Thus, we have proved:

Theorem 4.

Let L be a rational language in $(\mathcal{D} \times \mathcal{N})^*$. Then, it is decidable whether L is weakly coherent or not.

3 - Coherent components

In this section we will refine our results. When a transducer is built, it can be possibly incorrect, i.e. that it does not meet any of the previous coherences. Nevertheless, some of these transducers can provide some useful information : they give in fact several (namely a finite number of) specifications. We can then partition the language, each part corresponding to a specification. Let us consider the following example :

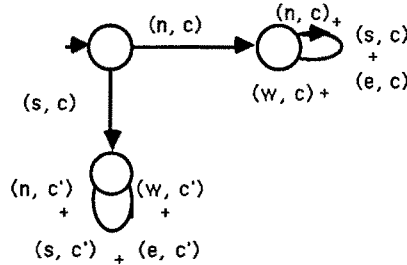


Figure 10.

We have $\mathcal{D} = \{n, s, e, w\}$ and $\mathcal{N} = \{c, c'\}$. This transducer is neither strong nor weak coherent. But, the associated rational language is the union of two coherent languages : $(n, c)(\mathcal{D} \times \{c\})^* \cup (s, c')(\mathcal{D} \times \{c'\})^*$. For the description of such a transducer, we will introduce some notions. In the following, "coherent" will mean "strongly coherent with respect to 0" (Definition 2).

Definition 4.

Let L be a language over $(\mathcal{D} \cup \mathcal{N})^*$, and w be a word in L . The w - *component* induced by w , denoted by $\text{Init}(w, L)$, is the set of the left factors of the words in L that start with w . It is defined by :

$$\text{Init}(w, L) = \text{LF}(w) \cup \{wu / wu \in L\}.$$

A language will then be said to be *partitionable* if it is the union of a finite number of w - components. This means that a finite number of choices (that may be connected to some test instructions in a program corresponding to the transducer) will determine which specification has to be considered among the finite number of specifications described by the transducer.

Definition 5.

A language L over $(\mathcal{D} \cup \mathcal{N})^*$ is *partitionable* iff there exist a finite number of words w_1, \dots, w_p such that :

- 1 - $\text{Init}(w_i, L)$ is coherent for $i = 1 \dots p$.
- 2 - $L = \cup_{i=1 \dots p} \text{Init}(w_i, L)$.

The words w_i are then called *initialization words*, and the sets $\text{Init}(w_i, L)$ *coherent components* of L .

Remark.

$w' <_1 w \Rightarrow \text{Init}(w, L) \subseteq \text{Init}(w', L)$.

It follows that the component $\text{Init}(w, L)$ can be replaced by $\text{Init}(w', L)$ (of course if $\text{Init}(w', L)$ is also coherent) in the decomposition of L .

We want to have the optimal decomposition, i.e with maximal components.

Definition 6.

A coherent component induced by w is said to be a *maximal coherent component* iff :

- 1 - $\text{Init}(w, L)$ is coherent.
- 2 - $w' <_1 w, w' \neq w \Rightarrow \text{Init}(w', L)$ is not coherent.

We want to get a decomposition of a language with the smallest number of components, and such that the initialization words are the shortest ones. For that purpose, we will use the following characterization of partitionable languages :

Property.

L is partitionable iff there exist a finite number of words w_1, \dots, w_q such that

- 1 - $\text{Init}(w_i, L)$ is a maximal coherent component, for $i = 1, \dots, q$.
- 2 - $L = \bigcup_{i=1 \dots q} \text{Init}(w_i, L)$.

The words w_i are called *maximal initialization words*.

Remark.

We can notice that, in the conditions of the property, the set $\{w_1, \dots, w_q\}$ is a prefix set.

One can of course notice that a coherent language is partitionable (because it is equal to $\text{Init}(\epsilon, L)$). But the converse is false, as shown by the transducer of figure 10. The language is partitionable, but is not coherent (and also not k -weak or strong coherent).

Furthermore, there exist languages that are not partitionable. Let us consider the next example for illustrating this :

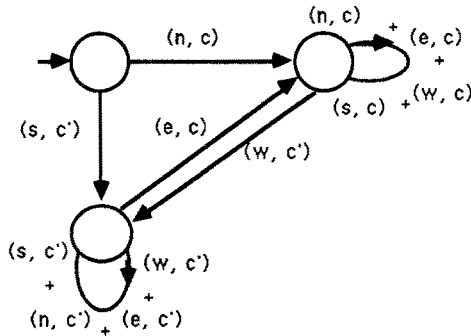


Figure 11.

We then want to know whether a language is partitionable or not. In order to prove that this property is decidable we need the following lemma:

Lemma.

Let $w = w_1 w_2 w_3$ be a word such that :

- 1 - $\text{Init}(w, L)$ is a maximal coherent component.
- 2 - w contains a loop, i.e $\delta(q_0, w_1) = \delta(q_1, w_2) = q_1$.

Then, for $p \in \mathbb{N}^*$, $\text{Init}(w_1 w_2^p w_3, L)$ is a maximal coherent component.

Proof.

Let us first show that $\text{Init}(w_1 w_2^p w_3, L)$ is coherent. Suppose that :

$$\begin{array}{|l} w_1 w_2^p w_3 u(\text{dir}, \text{nat}) \in L \\ w_1 w_2^p w_3 u(\text{dir}, \text{nat}) \in L \end{array}$$

with $\mu(w_1 w_2^p w_3 u(\text{dir}, \text{nat})) = \mu(w_1 w_2^p w_3 u(\text{dir}, \text{nat}))$

Since L is deterministic and w_2 corresponds to a loop, we have :

$$\begin{array}{|l} w_1 w_2 w_3 u(\text{dir}, \text{nat}) \in L \\ w_1 w_2 w_3 v(\text{dir}, \text{nat}') \in L \end{array}$$

with $\mu(w_1 w_2 w_3 u(\text{dir}, \text{nat})) = \mu(w_1 w_2 w_3 v(\text{dir}, \text{nat}'))$.

And this implies that $\text{nat} = \text{nat}'$, according to the fact that $\text{Init}(w, L)$ is a coherent component. Thus, $\text{Init}(w_1 w_2^p w_3, L)$ is coherent.

Moreover, it is a maximal coherent component. Indeed, let us suppose that a left factor w' is also an initialization word. We will show that w admits then also a left factor that is an initialization word :

1 - $w' = w_1 w_2^p w_3'$, with $w_3' <_1 w_3$, $w_3 \neq w_3'$. Let us suppose that $\text{Init}(w', L)$ is coherent.

$\triangleright w_1 w_2^p w_3' u(\text{dir}, \text{nat}) \in L \Rightarrow w_1 w_2 w_3' u(\text{dir}, \text{nat}) \in L$

$\triangleright w_1 w_2^p w_3' v(\text{dir}, \text{nat}') \in L \Rightarrow w_1 w_2 w_3' v(\text{dir}, \text{nat}') \in L$

$\triangleright [\mu(w_1 w_2 w_3 u(\text{dir}, \text{nat})) = \mu(w_1 w_2 w_3 v(\text{dir}, \text{nat}'))]$


\Rightarrow

$$[\mu(w_1 w_2 w_3' u(\text{dir}, \text{nat})) = \mu(w_1 w_2 w_3' v(\text{dir}, \text{nat}'))].$$

As $\text{Init}(w', L)$ is coherent, this implies then that $\text{nat} = \text{nat}'$. And thus, $\text{Init}(w_1 w_2 w_3', L)$ is coherent too. But then, $\text{Init}(w, L)$ is not a maximal coherent component, which is in contradiction with our hypothesis. Thus, $\text{Init}(w_1 w_2^p w_3, L)$ is a maximal coherent component.

2 - $w' = w_1 w_2^q w_2'$, with $q < p$ and $w_2' <_1 w_2$, and $w_2 \neq w_2'$. Applying the same reasoning, we see that the coherence of $\text{Init}(w_1 w_2^q w_2', L)$ implies that $\text{Init}(w_1 w_2', L)$ is also coherent, which is impossible since $\text{Init}(w, L)$ is maximal. Thus, $\text{Init}(w_1 w_2^p w_3, L)$ is a maximal coherent component.

3 - $w' = w_1'$, with $w_1' <_1 w_1$. Then if $\text{Init}(w_1', L)$ is coherent, $\text{Init}(w, L)$ is not maximal, which is impossible according to our hypothesis.

Thus, no left factor of $w_1 w_2^p w_3$ is an initialization word, and $\text{Init}(w_1 w_2^p w_3, L)$ is a maximal coherent component. 

With this lemma we can prove the decidability of the partitionable property:

Theorem 6.

Let L be a rational language of $(\mathcal{A} \cup \mathcal{B})^*$. It is decidable whether L is partitionable or not.

Proof.

We recall that the language L is given by a finite deterministic transducer with N states,

$$\mathcal{A} = (Q, (\mathcal{A} \times \mathcal{B}), \delta, q_0, F).$$

The lemma proves that if a word containing a loop induces a maximal coherent component, there is an infinity of maximal components, and the language is not partitionable. Indeed, suppose that the language is partitionable, and that a word w containing a loop is a maximal initialization word. Let $\alpha_1, \dots, \alpha_p$ be the initialization words of L . Let $k = \max_{i=1, \dots, p} |\alpha_i| + 1$, and q be such that $|w_1 w_2^q w_3| > k$. Since $w_1 w_2^q w_3 \in L$, there is a word α_i such that $w_1 w_2^q w_3 \in \text{Init}(\alpha_i, L)$.

As $|\alpha_i| < |w_1 w_2^q w_3|$, we have $\alpha_i <_1 w_1 w_2^q w_3$, but then $\text{Init}(w_1 w_2^q w_3, L)$ is not a maximal coherent component. According to the precedent lemma, this is impossible.

Thus, if L is partitionable, the maximal initialization words contain no loop. It follows that the potential maximal initialization words are of size less than N , if N is the number of states of the transducer. Then, to decide whether the language L is partitionable or not, we just have :

1 - to test if these words (that are in finite number) are initialization words, starting with the smallest ones, and avoiding the verification for the words that admit a left factor that has still be recognized as an initialization word.

2 - To verify that L is equal to the union of the maximal components that have been found.

Thus, we can decide whether L is partitionable or not. And the theorem is proved. 🍎

Example.

If we consider the example of figure 10, we see that the automaton defines a language that admits 2 maximal coherent components $w_1 = (n, c)$ and $w_2 = (s, c')$.

But the automaton of figure 11 is not partitionable, since it admits an infinity of maximal initialization words.

4. Some properties of such a system

In this section we are interested in detecting some particular properties of the system.

We will first try to detect the possible lack of -utilization of some physical links of the system. A link will be in a idle state if it is never logically connected. Formally:

Definition 7.

A link, defined by its component α in $\mu((\mathcal{D} \times \mathcal{N})^*)$ and a direction dir , is in a idle state iff :

$$\forall nat \in \mathcal{N}, [u(dir, nat) \in L \Rightarrow \mu(u) \neq \alpha]$$

Proposition 7.

It is decidable whether a link is in a idle state.

Proof.

It suffices to test whether $\alpha \notin \bigcup_{nat \in \mathcal{N}} \mu(L(dir, nat)^{-1})$, which is decidable. 🍎

We can also wonder if two points are logically linked together. We introduce three kinds of connectivity. In order to simplify the notations we consider connectivities between the starting point and a given point. Nevertheless these results can be extended in a straightforward way to connectivity between two arbitrarily choosen points of the net.

Definition 8.

- 1 - The point is *weakly connected* to the starting point iff there is a path u in L that leads to this point.
- 2 - The point is *strongly connected* to the starting point iff there is an infinity of paths u in L leading to it.
- 3 - The point is *completely connected* to the starting point iff every path leading to it belongs to L .

Proposition 8.

Let L be a rational language over $(\mathcal{D} \times \mathcal{N})^*$. Each of the 3 kinds of connectivity is decidable.

Proof.

Let us suppose that the considered point is given by its component α in $\mu((\mathcal{D} \times \mathcal{N})^*)$.

- 1 - One just has to test whether α belongs to $\mu(L)$ or not.
- 2 - One just has to test whether $\phi^{-1}(\alpha) \cap P(L)$ is finite or not.
- 3 - One has just to test whether $p_{\mathcal{D}}(\phi^{-1}(\alpha))$ is included in $p_{\mathcal{D}}(L)$ or not. 🍎

Finally, we can also look for paths of a given nature (the nature preserving connectivity).

More formally, let us suppose that a rational set N of logical links is given (a particular case is the case of nat^*). Let us consider a point in the net given by its component α in $\mu((\mathcal{D} \times \mathcal{N})^*)$.

Is there a word u in L such that $\mu(u) = \alpha$ and $p_{\mathcal{N}}(u) \in N$?

Proposition 9.

It is decidable whether a given point is nature preserving connected with the starting point.

Proof.

One just has to verify whether $\alpha \in \mu(L \cap p_{\mathcal{N}}^{-1}(N))$ or not, which is decidable. In the particular case where $N = \text{nat}^*$, it enables to find paths of a given nature, i.e. such that contain only link of a given logical nature. 🍏

5. Perspectives

The title of the paper is intended to stress the fact that the properties we want to describe do not take into account any "meaning" for the nature of the logical links, or the nature of the processors that constitute the network.

In further work, we will investigate properties stemming from the nature of the processors and of the logical links. For example, a processor could be a finite automaton communicating with his neighbours, in the way described in [Beauquier...87]. The different types of logical connections would be then the description of the communicating states. Problems like local deadlock, arise then.

Also mechanisms that can realise these specifications have not been discussed here. Two possible mechanisms can be considered: an automaton that moves on the network, and establishes the different logical links, according to the rules given by the transducer. Another, is the broadcast mechanism, each node tells its neighbour, in wich state it is, and what type of link is established.

6. References

- [Beauquier...87] J. Beauquier, "Systèmes distribués et automates finis", Actes du colloque C3, Angoulême, pp 27-44, 1987
- [Berstel...79] J. Berstel, "Transductions and context free languages", Teubner Studienbucher Informatik, 1979.
- [Ginsburg...66] S. Ginsburg, "The mathematical theory of context free languages" McGraw-Hill, 1966, New York.
- [Karp...72] R.M. Karp, "Reducibility among combinatorial problems", in R.E. Miller and J.W. Thatcher (eds), Complexity of Computer Computations, Penum press, New York, pp. 85-103, 1972.
- [Kohonen...88] T. Kohonen, "An introduction to neural computing", Neural Networks, vol. 1, pp 3-16, 1988.
- [Parikh...66] R.J. Parikh, "On context-free languages", J. Assoc. Comput. Mach. 13, pp 570-581, 1966.
- [Quinton...86] P. Quinton, "An introduction to systolic architectures", Future Parallel Computer, LNCS 272, pp 387-400, 1986.