

Modular Synthesis of Petri Nets from Regular Languages

Eric Badouel¹, Philippe Darondeau¹, Laure Petrucci²

¹*IRISA, campus de Beaulieu, F-35042 Rennes Cedex*
{Eric.Badouel,Philippe.Darondeau}@irisa.fr

²*LIPN, 99, avenue Jean-Baptiste Clément, F-93430 Villetaneuse*
Laure.Petrucci@lipn.univ-paris13.fr

Abstract

We propose a framework in which the synthesis of Petri nets from products of regular languages may be dealt with in a modular way, without evaluating any global language. For this purpose, we focus on distributed Petri nets, made of subnets residing in different sites of a communication network. The behaviour of each component is specified by a regular language on the union of the alphabets of this component and the components immediately upstream.

1 Introduction

The basic Petri Net Synthesis problem consists in finding whether an automaton or a labelled graph may be realized up to an isomorphism by the reachable state graph of a Petri net with injectively labelled transitions. The problem was first examined for elementary nets, and it was decided using the key concept of *regions* of a graph [12, 13, 4, 10, 5]. The similar problem for P/T-nets was decided later on in [1] using the extended concept of regions defined in [15] and in [3].

Another Petri Net Synthesis problem consists in finding whether a prefix-closed language may be realized by the set of firing sequences of a Petri net with injectively labelled transitions. This problem was solved abstractly in [14] using the concept of regions of a language. For prefix-closed regular languages, the problem was decided later on in [1].

A drawback of the synthesis procedures constructed so far in both contexts is the lack of modularity: the graph or language to be realized is given by a monolithic specification. This limitation restricts the range of the applications of Petri Net Synthesis. In this paper, we will show that the synthesis of nets from regular languages is not incompatible with modular specifications. Under some conditions which we believe reasonable, one can in fact synthesize a bounded P/T-net $\mathcal{N} = (P, T, F, M_0)$ from a finite family of regular languages $L_s \subseteq \widehat{T}_s^*$, where $\widehat{T}_s \subseteq T$ and $T = \cup_s \widehat{T}_s$, without computing their product $\otimes_s L_s$ nor any automaton accepting this product.

If $T = \cup_s \widehat{T}_s$ but no other assumption is made on the alphabets \widehat{T}_s of the languages L_s , we do not know any modular solution to the net synthesis problem. Therefore, we assume that languages L_s bijectively correspond to the sites $s \in S$ of a communication network $G = (S, \rightarrow)$, where $s' \rightarrow s$ represents a channel from s' to s . We assume moreover a partition $T = \uplus_s T_s$ such that for all s , \widehat{T}_s is the union of T_s and all sets $T_{s'}$ such that $s' \rightarrow s$ in G . The idea is that the synthesized P/T-nets \mathcal{N} should be distributed, and a transition $t' \in T_{s'}$ cannot produce tokens consumed by $t \in T_s$ unless there is a channel from s' to s in G . According to this interpretation, \widehat{T}_s is the set of all transitions $t \in T_s$ that can take place at site s plus all remote transitions that can influence the firing of the transitions $t \in T_s$, by sending tokens on channels from s' to s in G .

In this framework, if a specification $\{L_s \mid s \in S\}$ is coherent, that is the languages L_s are the respective projections of some language $L \subseteq T^*$, one can decide in a modular way whether there exists a bounded and distributed net implementing them. Even though coherence can in general not be checked in a modular way, it is fortunately the case when G is minimally connected, which excludes e.g. ring architectures but still covers a lot of practical cases.

The rest of the paper is organized as follows. In Section 2, we recall the concept of Distributed Petri Nets and the way to convert them into communicating finite state machines with bounded channels. In Section 3, we state the Distributed Net Synthesis Problem to be solved, and we show that this problem may be decomposed over sites, thus leading to the Open Net Synthesis Problem. The Open Net Synthesis Problem is addressed in Section 4, using a suitable adaptation of the concept of regions of a language. A brief conclusion completes the paper.

2 Distributed Petri Nets

In this section, we refine the concept of Distributed Petri Nets introduced in [2] and we recall their relationship to communicating finite state machines. To begin with, we recall the definition of Place/Transition nets.

Definition 1. *[P/T-net] A P/T-net is a bi-partite graph $N = (P, T, F)$, where P and T are disjoint sets of vertices, called places and transitions, respectively, and $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a set of directed edges with non-negative integer weights. A marking of N is a map $M : P \rightarrow \mathbb{N}$. The state graph of N is a labelled graph, with markings as vertices, where there is an edge from M to M' with label $t \in T$ (notation: $M[t]M'$) if and only if, for every place $p \in P$, $M(p) \geq F(p, t)$ and $M'(p) = M(p) - F(p, t) + F(t, p)$. The reachability graph of an initialized P/T-net $\mathcal{N} = (P, T, F, M_0)$ with the initial marking M_0 is*

the induced restriction of its state graph on the set of markings that may be reached from M_0 . The net \mathcal{N} is finite if P and T are finite. The net \mathcal{N} is bounded if its reachability graph is finite. The language $L(\mathcal{N})$ of the net \mathcal{N} is the set of words $t_1 t_2 \dots t_l \in T^*$ that label firing sequences $M_0[t_1]M_1[t_2] \dots M_{l-1}[t_l]M_l$. We use $M_0[t_1 \dots t_l]M_l$ as a shorter notation for such a firing sequence.

It follows from the net firing rule that whenever $M[w]M'$ for some word $w \in T^*$, $M'(p) = M(p) + \sum_{t \in T} \psi(w)(t) \times (F(t, p) - F(p, t))$ for every place $p \in P$, where $\psi(w)$ is the firing count vector of w , also called the Parikh image of w .

Definition 2. The Parikh image of a word w of \hat{T}^* , where $\hat{T} = \{t_1, \dots, t_k\}$, is the map $\psi(w) : \hat{T} \rightarrow \mathbb{N}$ such that $\psi(w)(t_h)$ counts the occurrences of t_h in w . By a slight abuse of notation, we sometimes use the alternative map $\psi(w) : \{1, \dots, k\} \rightarrow \mathbb{N}$ such that $\psi(w)(h)$ counts the occurrences of t_h in w .

Distributed Petri Nets are P/T-nets in which the weighted flow relation $F : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ includes predefined constraints $F(p, t) = 0$ or $F(t, p) = 0$. These constraints reflect the architecture of a communication network, and they ensure that the net may be realized with a communicating finite state machine mapped on this architecture.

Definition 3. [Distributed Net Architecture] Given a set of transitions T , a distributed net architecture is defined by a location map $\lambda : T \rightarrow S$ and a communication graph $G = (S, \rightarrow)$ where S is a finite set of sites. For $s \in S$, we note $T_s = \lambda^{-1}\{s\}$ and $\hat{T}_s = T_s \cup \lambda^{-1}\{s' \in S \mid s' \rightarrow s\}$.

T_s is the set of all transitions that can occur at site s . \hat{T}_s is the set of all transitions that can exert direct influence on the firability of transitions in T_s . In the sequel, $\pi_s : T^* \rightarrow T_s^*$ is the unique monoid morphism such that $\pi_s(t) = t$ for $t \in T_s$ and $\pi_s(t) = \epsilon$ (the empty word) otherwise. Similarly, $\hat{\pi}_s : T^* \rightarrow \hat{T}_s^*$ is the unique monoid morphism such that $\hat{\pi}_s(t) = t$ for $t \in \hat{T}_s$ and $\hat{\pi}_s(t) = \epsilon$ otherwise.

Definition 4. [Distributed P/T-net] Given a location map $\lambda : T \rightarrow S$ and a connected communication graph $G = (S, \rightarrow)$, a distributed P/T-net is a P/T-net in which the following requirements are satisfied:

- $(\forall p \in P)(\exists t \in T) F(p, t) \neq 0$
- $(\forall p \in P)(\forall t, t' \in T) F(p, t) \neq 0 \wedge F(p, t') \neq 0 \Rightarrow \lambda(t) = \lambda(t')$
- $(\forall p \in P)(\forall t, t' \in T) F(p, t) \neq 0 \wedge F(t', p) \neq 0 \Rightarrow \lambda(t) = \lambda(t') \vee \lambda(t') \rightarrow \lambda(t)$

In view of the first two requirements in Def. 4, the location map $\lambda : T \rightarrow S$ extends in a unique way to a map $\lambda : T \cup P \rightarrow S$ such that

$F(p, t) \neq 0 \Rightarrow \lambda(p) = \lambda(t)$. So, the places of a distributed net are located in sites. The first two requirements stipulate that a transition t located in site $\lambda(t)$ cannot consume tokens from a place p located in a different site $\lambda(p) \neq \lambda(t)$. The third condition stipulates that a transition t' located in site $\lambda(t')$ can produce and send tokens to a place p only if $\lambda(p) = \lambda(t')$ or there is an edge from $\lambda(t')$ to $\lambda(p)$ in graph G , figuring a channel from $\lambda(t')$ to $\lambda(p)$.

Example 1 *Let us consider a communication system, sketched in figure 1, with four machines PC_1 , PC_2 , PC_3 and PC_4 connected by a network comprising a router. Machines PC_1 and PC_3 send messages, while machines PC_2 and PC_4 wait for incoming messages. Machines PC_1 and PC_3 send messages to machine PC_2 via the router. The router uses a store and forward strategy, i.e. each incoming message is stored and then forwarded to its destination. Copies of all messages from PC_3 to PC_2 are sent directly to PC_4 .*

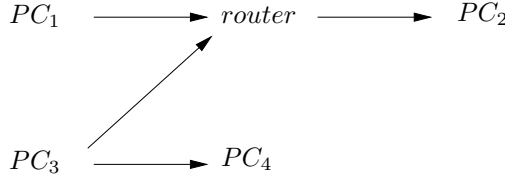


Figure 1: Example of a routing system

Figure 2 shows a distributed Petri net modelling this system. The distribution is such that: $S = \{PC_1, PC_2, PC_3, PC_4, \text{router}\}$ with the communication graph as in figure 1 and:

$$\begin{aligned}
 \lambda(P_{11}) &= \lambda(P_{12}) = \lambda(T_{11}) = \lambda(T_{12}) = PC_1 \\
 \lambda(B_{R2}) &= \lambda(P_{21}) = \lambda(P_{22}) = \lambda(T_{21}) = \lambda(T_{22}) = PC_2 \\
 \lambda(P_{31}) &= \lambda(P_{32}) = \lambda(T_{31}) = \lambda(T_{32}) = PC_3 \\
 \lambda(B_{34}) &= \lambda(P_{41}) = \lambda(P_{42}) = \lambda(T_{41}) = \lambda(T_{42}) = PC_4 \\
 \lambda(R_1) &= \lambda(R_2) = \lambda(P_{R1}) = \lambda(P_{R2}) = \lambda(T_{1R}) = \lambda(T_{3R}) = \lambda(T_{R2}) = \text{router}
 \end{aligned}$$

The sets T_s and \hat{T}_s are thus defined by:

$$\begin{aligned}
 T_{PC_1} &= \{T_{11}, T_{12}\}, \quad \hat{T}_{PC_1} = \{T_{11}, T_{12}\} \\
 T_{PC_2} &= \{T_{21}, T_{22}\}, \quad \hat{T}_{PC_2} = \{T_{21}, T_{22}, T_{1R}, T_{3R}, T_{R2}\} \\
 T_{PC_3} &= \{T_{31}, T_{32}\}, \quad \hat{T}_{PC_3} = \{T_{31}, T_{32}\} \\
 T_{PC_4} &= \{T_{41}, T_{42}\}, \quad \hat{T}_{PC_4} = \{T_{41}, T_{42}, T_{31}, T_{32}\} \\
 T_{\text{router}} &= \{T_{1R}, T_{3R}, T_{R2}\}, \quad \hat{T}_{\text{router}} = \{T_{1R}, T_{3R}, T_{R2}, T_{11}, T_{12}, T_{31}, T_{32}\}
 \end{aligned}$$

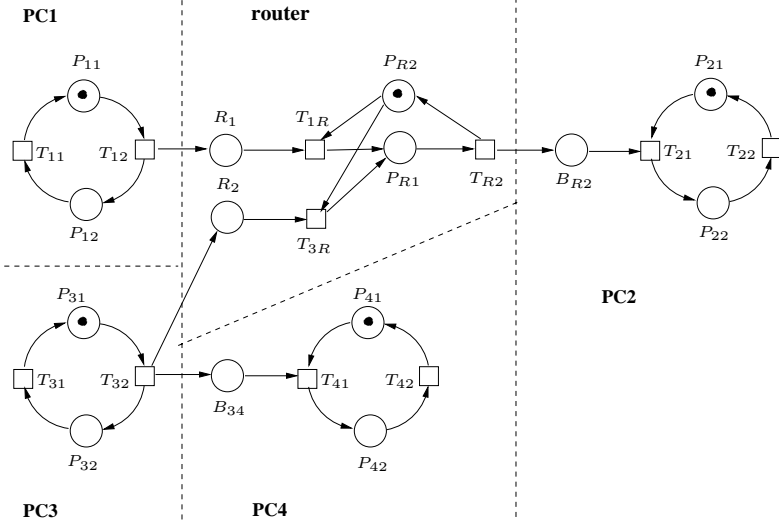


Figure 2: Distributed Petri net modelling the routing system

From now on, $\mathcal{N} = (P, T, F, M_0)$ is a finite initialized P/T-net, bounded and distributed w.r.t. $\lambda : T \cup P \rightarrow S$ and $G = (S, \rightarrow)$. The distributed architecture (λ, G) induces a decomposition of \mathcal{N} into an indexed family of components \mathcal{N}_s for s ranging over S , as follows. Each subnet \mathcal{N}_s is the induced restriction of \mathcal{N} on the subset of places $P_s = P \cap \lambda^{-1}\{s\}$ and on the subset of transitions \hat{T}_s , i.e. $\mathcal{N}_s = (P_s, \hat{T}_s, F_s, M_{0s})$ where M_{0s} is the restriction of M_0 on P_s and F_s is the restriction of F on $(P_s \times \hat{T}_s) \cup (\hat{T}_s \times P_s)$. Thus $\mathcal{N} = \oplus_s \mathcal{N}_s$ according to the definition below.

Definition 5. *The sum of a family of place disjoint nets $\mathcal{N}_s = (P_s, \hat{T}_s, F_s, M_{0s})$, $s \in S$, is the net $\oplus_s \mathcal{N}_s = (P, T, F, M_0)$ defined with $P = \uplus_s P_s$, $T = \cup_s \hat{T}_s$, $F(p, t) = F_s(p, t)$ if $p \in P_s$ and $t \in \hat{T}_s$ for some $s \in S$ else 0, $F(t, p) = F_s(t, p)$ if $p \in P_s$ and $t \in \hat{T}_s$ for some $s \in S$ else 0, and $M_0(p) = M_{0s}(p)$ for $p \in P_s$.*

In order to explain the relationship between the language $L(\mathcal{N})$ of the net and the languages of the net components $L(\mathcal{N}_s)$, we recall the following definition.

Definition 6. *[mixed product of languages, adapted from [11]] Given $L_s \subseteq \hat{T}_s^*$ for s ranging over S , let $\otimes_s L_s \subseteq T^*$ be defined as $\{w \mid \forall s : \hat{\pi}_s(w) \in L_s\}$.*

The following propositions follow immediately from the definition of distributed P/T-nets.

Proposition 7. $L(\mathcal{N}_s) \cdot (\widehat{T}_s \setminus T_s) \subseteq L(\mathcal{N}_s)$.

Proposition 8. $L(\mathcal{N}_s) \supseteq \widehat{\pi}_s(L(\mathcal{N}))$.

Proposition 9. $L(\mathcal{N}) = L(\oplus_s \mathcal{N}_s) = \otimes_s L(\mathcal{N}_s)$.

We state now two other useful properties of the component nets \mathcal{N}_s induced by the decomposition of a bounded and distributed P/T-net.

Proposition 10. *The following two properties hold for all $s \in S$ and $t \in T_s$:*

- i) $(\forall w \in L(\mathcal{N})) \ wt \notin L(\mathcal{N}) \Rightarrow \widehat{\pi}_s(wt) \notin L(\mathcal{N}_s),$
- ii) $(\forall w' \in \widehat{\pi}_s(L(\mathcal{N}))) \ w't \notin \widehat{\pi}_s(L(\mathcal{N})) \Rightarrow w't \notin L(\mathcal{N}_s).$

PROOF. We first prove (i). Let $w \in L(\mathcal{N})$ and $t \in T_s$ be such that $wt \notin L(\mathcal{N})$. In view of the net firing rule, $0 \leq M_0(p) + \sum_{t' \in T} \psi(w)(t') \times (F(t', p) - F(p, t')) < F(p, t)$ for some place $p \in P$, and $F(p, t) > 0$ entails that $\lambda(p) = \lambda(t) = s$. From Def. 4, $F(p, t') = 0$ and $F(t', p) = 0$ for every transition t' outside \widehat{T}_s , while by Def. 5, $F(p, t') = F_s(p, t')$ and $F(t', p) = F_s(t', p)$ for every transition t' in \widehat{T}_s (in particular for t). So, if we set $w' = \widehat{\pi}_s(w)$, then $M_{0s}(p) + \sum_{t' \in \widehat{T}_s} \psi(w')(t') \times (F_s(t', p) - F_s(p, t')) < F_s(p, t)$, showing that $w't = \widehat{\pi}_s(wt) \notin L(\mathcal{N}_s)$.

We now show that (i) entails (ii). Let $w' \in \widehat{\pi}_s(L(\mathcal{N}))$ and $t \in T_s$ such that $w't \notin \widehat{\pi}_s(L(\mathcal{N}))$. Since $w' \in \widehat{\pi}_s(L(\mathcal{N}))$, $w' = \widehat{\pi}_s(w)$ for some $w \in L(\mathcal{N})$. From $w' = \widehat{\pi}_s(w)$ and $w't \notin \widehat{\pi}_s(L(\mathcal{N}))$, necessarily $wt \notin L(\mathcal{N})$ hence by (i), $\widehat{\pi}_s(wt) \notin L(\mathcal{N}_s)$, i.e. $w't \notin L(\mathcal{N}_s)$. ■

Definition 11. *[restricted boundedness] Given $s \in S$ and $L_s \subseteq \widehat{T}_s^*$, the net $\mathcal{N}_s = (P_s, \widehat{T}_s, F_s, M_{0s})$ is said to be bounded in restriction to L_s if there exists some finite bound B such that $(\forall p \in P_s)(\forall w' \in L_s \cap L(\mathcal{N}_s)) \ M_{0s}[w']M' \Rightarrow M'(p) \leq B$.*

Proposition 12. *Let \mathcal{N} be a bounded and distributed P/T-net. Then for all $s \in S$, the component subnet \mathcal{N}_s of \mathcal{N} is bounded in restriction to $\widehat{\pi}_s(L(\mathcal{N}))$.*

PROOF. Let $M_0[w]M$ in \mathcal{N} and $M_{0s}[w']M'$ in \mathcal{N}_s , with $w' = \widehat{\pi}_s(w)$. As \mathcal{N} is a bounded net, for any place $p \in P$, $M(p) \leq B$ for some finite bound B independent of M and p . From Def. 4, for any place $p \in P_s$, $M(p) = M_{0s}(p) + \sum_{t' \in \widehat{T}_s} \psi(w')(t') \times (F_s(t', p) - F_s(p, t')) = M'(p)$, hence $M'(p) \leq B$. ■

In the end of this section, we sketch a simple translation from bounded and distributed P/T-nets to communicating finite state machines with an equivalent behaviour. Let \mathcal{N} be a bounded and distributed P/T-net over (λ, G) . For any place $p \in P$, there exists a

finite bound $B(p)$ such that $M(p) \leq B(p)$ for every reachable marking M of \mathcal{N} . Note that, for any place $p \in P_s$, $B(p)$ may be determined locally from $\hat{\pi}_s(L(\mathcal{N}))$ if this language is known, without computing the global state graph of \mathcal{N} . This will be the case for nets \mathcal{N} synthesized from a specification $\{L_s \mid s \in S\}$ since $\hat{\pi}_s(L(\mathcal{N})) = L_s$ for such nets.

Given a bounded and distributed P/T-net \mathcal{N} , we shall simulate the behaviour of this net with a communicating machine $(\mathcal{A}, \mathcal{C})$ where $\mathcal{A} = \{A_s \mid s \in S\}$ is a set of finite automata and $\mathcal{C} = \{(s, s') \mid s \rightarrow s' \text{ in } G\}$ is a set of channels. The set of messages that may be sent to or received from a channel (s, s') is the set of names of places p' such that $p' \in P_{s'}$ and $F(t, p') > 0$ for some transition $t \in T_s$. For each $s \in S$, the automaton A_s has an extended alphabet $T_s \cup \tau_s$, where T_s is a set of internal actions representing homonymic transitions of \mathcal{N} , and τ_s is a set of communication actions. Each alphabet τ_s comprises two types of actions on channels: an action $!s'(p')$ means sending message p' on channel (s, s') in order to simulate the emission of a token (intended for place p'); an action $?s(p')$ means receiving message p' from channel (s, s') in order to simulate the delayed arrival of a token in place p' . Channels are unordered, i.e. messages sent on channel (s, s') may be received in a different order. The state of a channel may therefore be seen as a vector of counters (one for each p'). The state of the communicating machine is defined as the set of states of the component automata A_s plus the set of states of the channels. An action $!s'(p')$ is always enabled when it is enabled in some automaton A_s such that $(s, s') \in \mathcal{C}$. An action $?s(p')$ is enabled if it is enabled in some automaton A_s such that $(s, s') \in \mathcal{C}$ and this channel contains at least one occurrence of the message p' .

We now define, for each $s \in S$, the set of states Q_s and the transition function δ_s of the automaton A_s . First, we let Q_s be the set of all maps $M' : P_s \rightarrow \mathbb{N}$ such that $(\forall p \in P_s) M'(p) \leq B(p)$. Second, we let $q_{0s} = M_{0s}$ and $A_s = (Q_s, T_s \cup \tau_s, \delta_s, q_{0s})$ where $\delta_s : Q_s \times (T_s \cup \tau_s)^* \rightarrow Q_s$ is the partial transition function defined as follows:

- for each $t \in T_s$ and $q, q' \in Q_s$ such that $q[t]q'$ is a firing step of \mathcal{N}_s (although q is not necessarily reachable from M_{0s}), let $\delta_s(q, t \cdot \text{send}(t)) = q'$ for an arbitrary linearisation $\text{send}(t) \in \tau_s^*$ of the multiset $\{!s'(p') \times F(t, p') \mid s' \neq s \wedge p' \in P_{s'}\}$,
- for each $s' \neq s$ and $p \in P_s$ such that $F(t', p) \neq 0$ for some $t' \in T_{s'}$, provided that $q(p) < B(p)$, let $\delta_s(q, ?s'(p)) = q'$ where $q'(p) = q(p) + 1$ and $q'(p') = q(p')$ for $p' \neq p$.

Note that when a transition produces tokens for one or several dis-

tant places, these tokens are sent on the communication network in one batch while they are received one at a time. In order to restore the symmetry between send actions and receive actions, one may expand the automata A_s to automata with larger sets of states Q'_s and partial transition functions $\delta_s : Q'_s \times (T_s \cup \tau_s) \rightarrow Q'_s$. We claim that the resulting family of finite state machines $\{A_s \mid s \in S\}$ communicating through the channels (s', s) or $s' \rightarrow s$ defined in G , is channel bounded and implements $L(\mathcal{N})$. By adapting the proofs given in [2], one may further show that \mathcal{N} and the considered communicating finite state machines have branching bisimilar behaviours [17] when all communication actions $?s(p)$ and $!s(p)$ are dealt with as silent actions τ .

Remarks: Following the indications given in section 5 of [2], one may define an alternative translation from distributed nets to communicating systems $(\{\mathcal{N}'_s \mid s \in S\}, \mathcal{C})$ where each component \mathcal{N}'_s is a Petri net with possible concurrency.

3 The Distributed Net Synthesis Problem

The problem we want to address may be stated as follows.

Definition 13. *[Net Specification] Given a distributed net architecture (λ, G) with $\lambda : T \rightarrow S$ and $G = (S, \rightarrow)$, a net specification \mathcal{L} is an indexed family of prefix-closed and regular languages $L_s \subseteq \widehat{T}_s^*$ where s ranges over S .*

Problem 1 (Net Synthesis Problem) *Given a distributed net architecture (λ, G) and a specification $\{L_s \mid s \in S\}$, decide whether there exists and construct a bounded and distributed P/T-net \mathcal{N} such that $\widehat{\pi}_s(L(\mathcal{N})) = L_s$ for all $s \in S$.*

We moreover want to solve this problem in a *modular* way, without ever computing the mixed product $\otimes_s L_s$ nor the global language of any net over (λ, G) . In particular, we forbid ourselves to compute from the specification some global net \mathcal{N} and then check whether $\widehat{\pi}_s(L(\mathcal{N})) = L_s$ for all $s \in S$ as required. It is worth noting that a specification $\mathcal{L} = \{L_s \mid s \in S\}$ does generally not define unambiguously any global language L over T , as the following example shows.

Example 2 *Let $S = \{1, 2, 3\}$ with $T_1 = \{a\}$, $T_2 = \{b\}$, $T_3 = \{c\}$, and $G = \{1 \rightarrow 2, 1 \rightarrow 3\}$. Consider the net specification given by $L_1 = \text{pref}\{a\}$, $L_2 = \text{pref}\{ab\}$, and $L_3 = \text{pref}\{ac\}$ where $\text{pref } E$ means the set of all left factors of words in E . Then there exists several languages $L \subseteq T^*$ such that $L_i = \widehat{\pi}_i(L)$ for every i in $\{1, 2, 3\}$, for instance $L = \text{pref}\{abc, acb\}$, $L = \text{pref}\{abc\}$, $L = \text{pref}\{acb\}$, or*

$L = \text{pref}\{ab, ac\}$. In fact, only $L = \text{pref}\{abc, acb\}$ coincides with the language of a solution \mathcal{N} to the distributed net synthesis problem from $\mathcal{L} = \{L_1, L_2, L_3\}$.

It seems however desirable that a specification $\mathcal{L} = \{L_s \mid s \in S\}$ over (λ, G) defines at least one global language L over T , hence the following definition.

Definition 14. [Coherency] Given a distributed net architecture (λ, G) with $\lambda : T \rightarrow S$ and $G = (S, \rightarrow)$, a specification $\{L_s \mid s \in S\}$ is coherent if there exists some language L over T such that $\hat{\pi}_s(L) = L_s$ for all $s \in S$.

Proposition 15. A specification $\mathcal{L} = \{L_s \mid s \in S\}$ is coherent if and only if $(\forall s') L_{s'} = \hat{\pi}_{s'}(\otimes_s L_s)$.

PROOF. Suppose $\hat{\pi}_s(L) = L_s$ for all $s \in S$ then clearly $L \subseteq \otimes_s L_s$ and for all s' , $L_{s'} = \hat{\pi}_{s'}(\otimes_s L_s)$ because $\hat{\pi}_{s'}(L) \subseteq \hat{\pi}_{s'}(\otimes_s L_s) \subseteq L_{s'}$ and $\hat{\pi}_{s'}(L) = L_{s'}$. The converse implication is immediate. ■

Proposition 16. Let $\mathcal{L} = \{L_s \mid s \in S\}$ be a coherent specification and let $\mathcal{N} = \oplus_s \mathcal{N}_s$ be a distributed net. Then $L(\mathcal{N}) = \otimes_s L_s$ if and only if $\hat{\pi}_s(L(\mathcal{N})) = L_s$ for all $s \in S$.

PROOF. Suppose that $\hat{\pi}_s(L(\mathcal{N})) = L_s$ for all $s \in S$. Then $L(\mathcal{N}) \subseteq \otimes_s L_s$. Moreover, for all $s \in S$, $\hat{\pi}_s(L(\mathcal{N})) \subseteq L(\mathcal{N}_s)$ because $L(\mathcal{N}) = \otimes_s L(\mathcal{N}_s)$. Therefore $\otimes_s L_s \subseteq \otimes_s L(\mathcal{N}_s) = L(\mathcal{N})$. Altogether, $L(\mathcal{N}) = \otimes_s L_s$. The converse implication follows from Prop.15. ■

In view of Prop. 16, for coherent specifications, Problem 1 is equivalent to the following.

Problem 2 Given a net architecture (λ, G) and a specification $\{L_s \mid s \in S\}$, decide whether there exists and construct a bounded and distributed P/T-net \mathcal{N} over (λ, G) such that $L(\mathcal{N}) = \otimes_s L_s$.

In view of Def. 14, the coherency of a specification $\mathcal{L} = \{L_s \mid s \in S\}$ is necessary to the existence of solutions to Problem 1, but Problem 2 may have solutions for incoherent specifications. Indeed $L(\mathcal{N}) = \otimes_s L(\mathcal{N}_s)$ for any distributed net \mathcal{N} , but in most cases $L(\mathcal{N}_s)$ is a strict superset of $\hat{\pi}_s L(\mathcal{N})$ and therefore $\{L(\mathcal{N}_s) \mid s \in S\}$ is not a coherent specification.

Coherency is a decidable property since it expresses as $(\forall s \in S) L_s = \hat{\pi}_s(\otimes_s L_s)$ and any mixed product or projection of regular languages is a regular language. However one can generally not check

coherency without computing the mixed product $\otimes_s L_s$. Computing from \mathcal{L} , without a preliminary check of coherency, some net \mathcal{N} candidate as a solution to Problem 1 does not help overcoming the difficulty because, unless coherency is assumed, it is generally not possible to check that $\hat{\pi}_s(L(\mathcal{N})) = L_s$ for all $s \in S$ without computing $L(\mathcal{N})$. The modular synthesis of P/T-nets can therefore not be envisaged without imposing constraints on distributed net architectures, hence the following definition.

Definition 17. [*Minimally Connected*] A communication graph $G = (S, \rightarrow)$ is minimally connected if the underlying undirected multigraph is a graph and this graph is a tree (with arbitrary root vertex).

Example 3 The communication graph of example 1 (figure 1) is minimally connected.

Lemma 18. Let (λ, G) be a net architecture with a minimally connected graph $G = (S, \rightarrow)$. Then $\hat{T}_{s'} \cap \hat{T}_s = T_{s'}$ for every edge $s' \rightarrow s$ of G .

PROOF. By definition, $\hat{T}_{s'}$ is the union of $T_{s'}$ and all subsets $T_{s''}$ such that $s'' \rightarrow s'$. Similarly, \hat{T}_s is the union of T_s and all subsets $T_{s''}$ such that $s'' \rightarrow s$, thus including $T_{s'}$. By minimal connectedness of G , $s'' \rightarrow s'$ entails $s'' \neq s$, and there exists no vertex s'' such that $s'' \rightarrow s$ and $s'' \rightarrow s'$. As subsets $T_{s''}$ are pairwise disjoint, the lemma follows. ■

Lemma 19. Let (λ, G) be a net architecture with a minimally connected graph $G = (S, \rightarrow)$. Then, for any two distinct vertices s' and s , $\hat{T}_{s'} \cap \hat{T}_s = \emptyset$ unless $s' \rightarrow s$ or $s \rightarrow s'$ or $s'' \rightarrow s'$ and $s'' \rightarrow s$ for some (necessarily unique) s'' . Moreover, in the latter case, $\hat{T}_{s'} \cap \hat{T}_s = T_{s''}$.

PROOF. Left to the reader. ■

Proposition 20. Let $\mathcal{L} = \{L_s \mid s \in S\}$ be a specification over (λ, G) where G is minimally connected. Then \mathcal{L} is coherent if and only if $\pi_{s'}(L_{s'}) = \pi_{s'}(L_s)$ for every edge $s' \rightarrow s$ in G .

PROOF. By Prop. 15, $\{L_s \mid s \in S\}$ is coherent if and only if, for any $s' \in S$ and for any $w \in L_{s'}$, there exists an indexed family of words $\{w_s \mid s \in S \wedge w_s \in L_s\}$ such that $w = w_{s'}$ and $w \in \hat{\pi}_{s'}(\otimes_s \{w_s\})$.

By construction of the alphabets T_s and \hat{T}_s , $s \in S$, $T_{s'} = \hat{T}_{s'} \cap \hat{T}_s$ whenever $s' \rightarrow s$, and then $\pi_{s'} \circ \hat{\pi}_{s'} = \pi_{s'} \circ \hat{\pi}_s$. Thus if the specification \mathcal{L} is coherent, $\pi_{s'}(L_{s'}) = \pi_{s'} \circ \hat{\pi}_{s'}(\otimes_s L_s) = \pi_{s'} \circ \hat{\pi}_s(\otimes_s L_s) = \pi_{s'}(L_s)$.

Suppose that $\pi_{s'}(L_{s'}) = \pi_{s'}(L_s)$ for every edge $s' \rightarrow s$ in G . We show that $\{L_s \mid s \in S\}$ is coherent. Let s' be an arbitrary vertex of

G , and let $w_{s'} \in L_{s'}$. For all $n \geq 0$, let S_n be the subset of vertices at a distance at most n from s' , thus $S_0 = \{s'\}$ and $S_m = S_{m+1}$ for some m . We proceed by induction on $n < m$.

Assume that for each $s \in S_n$, w_s has been chosen from L_s such that, for all vertices $s'', s \in S_n$, if $s'' \rightarrow s$ is an edge of G then $\pi_{s''}(w_{s''}) = \pi_{s''}(w_s)$. As G is minimally connected, any vertex $s'' \in S_{n+1} \setminus S_n$ is connected by an edge $s'' \rightarrow s$ or $s \rightarrow s''$ to *exactly one* vertex $s \in S_n$, and there is no edge between two distinct vertices in $S_{n+1} \setminus S_n$. As $(s'' \rightarrow s) \Rightarrow (\pi_{s''}(L_{s''}) = \pi_{s''}(L_s))$ and $(s \rightarrow s'') \Rightarrow (\pi_s(L_s) = \pi_s(L_{s''}))$, one can choose independently for all $s'' \in S_{n+1} \setminus S_n$ some $w_{s''}$ from $L_{s''}$ such that $\pi_{s''}(w_{s''}) = \pi_{s''}(w_s)$ if $s'' \rightarrow s$ or $\pi_s(w_{s''}) = \pi_s(w_s)$ if $s \rightarrow s''$.

Now let $n = m$, thus w_s has been defined for all $s \in S$, and $\pi_{s''}(w_{s''}) = \pi_{s''}(w_s)$ for every edge $s'' \rightarrow s$ in G . At this stage, Lemmas 18 and 19 show that $\otimes_s w_s$ is not the empty set, hence $w \in \hat{\pi}_{s'}(\otimes_s \{w_s\})$. ■

Prop. 20 shows that when the communication graph G is minimally connected, the coherency of specifications over (λ, G) can be checked in a modular way. For the purpose of modular net synthesis, we impose on specifications a slightly stronger requirement of coherency as follows.

Definition 21. [Strong Coherency] A specification $\mathcal{L} = \{L_s \mid s \in S\}$ over (λ, G) is strongly coherent if it is coherent and the following condition is satisfied for every edge $s' \rightarrow s$ in G :

$$(\forall w \in L_s)(\forall t \in T_{s'}) \pi_{s'}(wt) \in \pi_{s'}(L_{s'}) \Rightarrow wt \in L_s.$$

Example 4 We now consider example 1 again. Let PC_1, PC_2, PC_3, PC_4 and router be the subnets shown on figure 2, and let router' be the net obtained by removing places R1 and R2 from router. For any two subnets \mathcal{N}' and \mathcal{N}'' , let $\mathcal{N}' + \mathcal{N}''$ be the net containing the places and transitions of both nets plus the connecting arcs drawn on figure 2. Define $L_{PC_1} = L(PC_1)$, $L_{PC_2} = L(PC_2 + \text{router}')$, $L_{PC_3} = L(PC_3)$, $L_{PC_4} = L(PC_4 + PC_3)$, and $L_{\text{router}} = L(\text{router} + PC_1 + PC_3)$. Then $\mathcal{L} = \{L_s \mid s \in S\}$ is a strongly coherent specification.

For coherent specifications, in view of Prop. 20, the above condition may be rewritten to $(\forall w \in L_s)(\forall t \in T_{s'}) \pi_{s'}(wt) \in \pi_{s'}(L_s) \Rightarrow wt \in L_s$, hence strong coherency may be checked in a modular way.

Proposition 22. Let \mathcal{N} be a distributed P/T-net over (λ, G) . If G is minimally connected, then $\mathcal{L} = \{\hat{\pi}_s(L(\mathcal{N})) \mid s \in S\}$ is strongly coherent.

PROOF. The coherency of \mathcal{L} follows clearly from its definition. It remains to show that \mathcal{L} is strongly coherent. Let $s, s' \in S$ such that

$s' \rightarrow s$ in G . Let $w \in \hat{\pi}_s(L(\mathcal{N}))$ and $t \in T_{s'}$ such that $\pi_{s'}(wt) \in \pi_{s'} \circ \hat{\pi}_{s'}(L(\mathcal{N}))$. We show that $wt \in \hat{\pi}_s(L(\mathcal{N}))$. First, it is possible to choose a word $w' \in \hat{\pi}_{s'}(L(\mathcal{N}))$ such that $\pi_{s'}(w) = \pi_{s'}(w')$ and $w't \in \hat{\pi}_{s'}(L(\mathcal{N}))$. Second, proceeding as in the proof of Prop. 20, it is possible to extend this choice to a full family of words $w_\sigma \in \hat{\pi}_\sigma(L(\mathcal{N}))$, $\sigma \in S$, such that $w_s = w$, $w_{s'} = w'$, and $w_{s''} \in \hat{\pi}_{s''}(\otimes_\sigma w_\sigma)$ for all s'' . By Prop. 8, $w_{s''} \in L(\mathcal{N}_{s''})$ for all s'' . Now consider the second family of words u_σ , $\sigma \in S$, defined with $u_\sigma = w_\sigma t$ if $t \in \hat{T}_\sigma$ and $u_\sigma = w_\sigma$ otherwise. Then, $u_\sigma \in L(\mathcal{N}_\sigma)$ for all $\sigma \in S$, and $u_{s''} \in \hat{\pi}_{s''}(\otimes_\sigma u_\sigma)$ for all s'' . By Prop. 9, $u_\sigma \in \hat{\pi}_\sigma(L(\mathcal{N}))$ for all $\sigma \in S$. In particular, $u_s = wt \in \hat{\pi}_s(L(\mathcal{N}))$. Therefore, \mathcal{L} is strongly coherent. \blacksquare

Imposing minimally connected architectures is restrictive, since this excludes in particular ring architectures. We postpone comments on this point. On the contrary, in view of Prop. 22 and the statement of Problem 1, strong coherency is a necessary condition for a specification to have some distributed net realization. It is therefore not restrictive to assume strong coherency, which can be checked in a modular way for minimally connected architectures. We feel that modular synthesis of nets is not possible without imposing this restriction on architectures. In order to solve Problem 1 for a specification $\mathcal{L} = \{L_s \mid s \in S\}$ over an architecture (λ, G) which is not minimally connected, we can only suggest to choose if possible a non-trivial equivalence relation \equiv on S such that G/\equiv is minimally connected, and to solve Problem 1 for the aggregated specification $\mathcal{L}' = \{L'_x \mid x \in X\}$, where $X = (S/\equiv)$ and $L'_x = \otimes_{s \in X} L_s$.

In the sequel, we consider always strongly coherent specifications over minimally connected architectures. In view of Prop. 16, Problem 1 is then equivalent to the following.

Problem 3 *Given a net architecture (λ, G) in which G is minimally connected, and given a strongly coherent specification $\{L_s \mid s \in S\}$, decide whether there exists and construct a bounded and distributed P/T-net \mathcal{N} over (λ, G) such that $L(\mathcal{N}) = \otimes_s L_s$.*

We will solve Problem 3 without ever computing the mixed product $\otimes_s L_s$ nor the global language of any net over (λ, G) . The keys to the modular synthesis of bounded and distributed P/T-nets are given by the following two propositions.

Proposition 23. *Let $\{L_s \mid s \in S\}$ be a strongly coherent specification, and let $\{L'_s \mid s \in S\}$ be an arbitrary set of prefix-closed languages $L'_s \subseteq \hat{T}_s^*$ such that $L'_s \cdot (\hat{T}_s \setminus T_s) \subseteq L'_s$ for all s . Then $\otimes_s L_s = \otimes_s L'_s$ if and only if, for all $s \in S$: $L_s \subseteq L'_s \subseteq \mathcal{C}((L_s \cdot T_s \setminus L_s) \cdot \hat{T}_s^*)$ where \mathcal{C} means complementation w.r.t. \hat{T}_s^* .*

PROOF.

If part

Suppose that the stated conditions are satisfied. Clearly, $\otimes_s L_s \subseteq \otimes_s L'_s$. In order to show the converse inclusion, we proceed by contradiction, and assume the existence of some minimal word $wt \in (\otimes_s L'_s) \setminus (\otimes_s L_s)$ with $t \in T$. From the minimality assumption, $w \in (\otimes_s L'_s) \cap (\otimes_s L_s)$, and therefore $w \in \otimes_s \{w_s\}$ for some family of words $w_s \in L_s$ such that $w_s = \hat{\pi}_s(w) \in L_s \subseteq L'_s$ for all s . Let $t \in T_{s'}$. Define a second family of words w'_s as follows.

- for $s = s'$ or $s' \rightarrow s$ let $w'_s = w_s t$,
- in any other case let $w'_s = w_s$.

Clearly, $wt \in \otimes_s \{w'_s\}$. As $wt \in (\otimes_s L'_s)$, the word $w'_{s'} = \hat{\pi}_{s'}(wt) = \hat{\pi}_{s'}(w)t$ belongs both to $L'_{s'}$ and to $L_{s'} \cdot T_{s'}$. Therefore, by the second condition of inclusion, $w'_{s'}$ belongs to $L_{s'}$. In order that $wt \notin (\otimes_s L_s)$, it must be the case that $w'_s = w_s t \notin L_s$ for some s such that $s' \rightarrow s$. But this is excluded by the condition of strong coherency, since $w_s \in L_s$, $\pi_{s'}(w_s t) = \pi_{s'}(w'_s) = \pi_{s'}(w'_{s'})$ and $w'_{s'}$ belongs to $L_{s'}$. It follows from this contradiction that $\otimes_s L_s = \otimes_s L'_s$.

Only If part

Suppose that $\otimes_s L_s = \otimes_s L'_s$. By the condition of coherency, $L_s = \hat{\pi}_s(\otimes_s L_s)$, hence $L_s = \hat{\pi}_s(\otimes_s L'_s) \subseteq L'_s$, and the first inclusion holds. Now let $w_{s'} \in L_{s'}$ and $t \in T_{s'}$ such that $w_{s'} t \notin L_{s'}$, and assume for a contradiction that $w_{s'} t \in L'_{s'}$. As the specification $\{L_s \mid s \in S\}$ is coherent, there exists a family of words $w_s \in L_s$, $s \neq s'$, and a word $w \in \otimes_s w_s$ such that $\hat{\pi}_s(w) = w_s$ for all $s \in S$. Define a second family of words w'_s as follows.

- for $s = s'$ or $s' \rightarrow s$ let $w'_s = w_s t$,
- in any other case let $w'_s = w_s$.

Then clearly, $wt \in \otimes_s w'_s$. By assumption, $w'_{s'} = w_{s'} t \in L'_{s'}$. Moreover $s' \rightarrow s$ entails that $w'_s = w_s t \in L'_s$ since $w_s \in L_s \subseteq L'_s$ and $L'_s \cdot T_{s'} \subseteq L'_s$. Therefore, $wt \in \otimes_s L'_s = \otimes_s L_s$. It follows from the coherency of the specification that $\hat{\pi}_{s'}(wt) = w_{s'} t$ belongs to $L_{s'}$, in contradiction with the assumptions. This establishes the second inclusion, hence the proposition. ■

Proposition 24. *Let $\mathcal{L} = \{L_s \mid s \in S\}$ be a strongly coherent specification over (λ, G) where $G = (S, \rightarrow)$ is minimally P -connected. Then $\otimes_s L_s = L(\mathcal{N})$ for some distributed and bounded P/T -net $\mathcal{N} = \oplus_s \mathcal{N}_s$ where $\mathcal{N}_s = (P_s, \hat{T}_s, F_s, M_{0s})$ if and only if the following conditions are satisfied:*

1. $p \in P_s \wedge t \in \hat{T}_s \setminus T_s \Rightarrow F_s(p, t) = 0$,
2. $p \in P_s \Rightarrow F_s(p, t) > 0$ for some $t \in T_s$,

3. $w \in L_s \wedge t \in T_s \wedge wt \in L_s \Rightarrow M_{0s}[w]M'$ with $F_s(p, t) \leq M'(p)$ for $p \in P_s$,
4. $w \in L_s \wedge t \in T_s \wedge wt \notin L_s \Rightarrow M_{0s}[w]M'$ with $F_s(p, t) > M'(p)$ for some p ,
5. the net \mathcal{N}_s is bounded in restriction to L_s .

PROOF.

If part

From conditions 1 and 2, \mathcal{N} is a distributed P/T-net. Therefore, $L(\mathcal{N}) = \otimes_s L(\mathcal{N}_s)$ and $L(\mathcal{N}_s) \cdot (\widehat{T}_s \setminus T_s) \subseteq L(\mathcal{N}_s)$. From conditions 3 and 4, $L_s \subseteq L(\mathcal{N}_s) \subseteq \mathbb{C}((L_s \cdot T_s \setminus L_s) \cdot \widehat{T}_s^*)$. As $\mathcal{L} = \{L_s \mid s \in S\}$ is a strongly coherent specification, $\otimes_s L_s = L(\mathcal{N})$ follows by Prop. 23. It remains to show that \mathcal{N} is bounded. We proceed by contradiction. Let $s \in S$ and $p \in P_s$ and suppose that for any n , there exists some firing sequence $M_0[w]M$ of the net \mathcal{N} with $M(p) > n$. As $\mathcal{N} = \oplus_s \mathcal{N}_s$, there exists some corresponding firing sequence $M_{0s}[\widehat{\pi}_s(w)]M'$ of the net \mathcal{N}_s with $M'(p) = M(p)$. As $w \in L(\mathcal{N}) = \otimes_s L_s$, $\widehat{\pi}_s(w) \in L_s$. Therefore, the net \mathcal{N}_s is not bounded in restriction to L_s , contradicting condition 5.

Only If part

In view of Def. 4 and Propositions 8, 10 and 12, all conditions stated in Prop. 24 are necessary to the existence of a solution to the bounded and distributed net synthesis problem from \mathcal{L} . ■

By Prop. 24, the bounded and distributed net synthesis problem from a strongly coherent specification $\{L_s \mid s \in S\}$ decomposes modularly to $|S|$ independent instances of the following.

Problem 4 (Open Net Synthesis Problem) *Given finite alphabets T and \widehat{T} , with $T \subseteq \widehat{T}$, and a non-empty regular and prefix-closed language $L \subseteq \widehat{T}^*$, decide whether there exists and construct a finite P/T-net $\mathcal{N} = (P, \widehat{T}, F, M_0)$ such that:*

1. $p \in P \wedge t \in \widehat{T} \setminus T \Rightarrow F(p, t) = 0$,
2. $p \in P \Rightarrow F(p, t) > 0$ for some $t \in T$,
3. $w \in L \wedge t \in T \wedge wt \in L \Rightarrow M_0[w]M$ with $F(p, t) \leq M(p)$ for all $p \in P$,
4. $w \in L \wedge t \in T \wedge wt \notin L \Rightarrow M_0[w]M$ with $F(p, t) > M(p)$ for some p ,
5. \mathcal{N} is bounded in restriction to L .

For fixed alphabets \widehat{T} and $T \subseteq \widehat{T}$, a net $\mathcal{N} = (P, \widehat{T}, F, M_0)$ satisfying condition 1 w.r.t. T is called an *open net* in the sequel. Clearly, $L(\mathcal{N}) \cdot (\widehat{T} \setminus T) \subseteq L(\mathcal{N})$ for any such open net.

4 Open Net Synthesis using Regions

In this section, we solve Problem 4 using the concept of regions of a language, originally defined in [14]. The presentation of regions given below draws inspiration from [1] and [8], with minor adaptations reflecting the slightly different statement of the synthesis problem.

Definition 25.*[regions] Let $L \subseteq \hat{T}^*$ be a prefix closed language over $\hat{T} = \{t_1, \dots, t_k\}$. A region of L is a non-negative integer vector*

$$r = \langle r_{init}, r \circ t_1, t_1 \circ r, \dots, r \circ t_k, t_k \circ r \rangle$$

such that for all $t \in \hat{T}$, $wt \in L \Rightarrow r/w \geq r \circ t$ where r/w is defined inductively on the words of \hat{T}^ with $r/\epsilon = r_{init}$ and $r/wt = r/w - r \circ t + t \circ r$. The region r is a bounded region of L if there exists $B \in \mathbb{N}$ such that $r/w \leq B$ for all $w \in L$.*

According to Def. 25, the regions r of L correspond bijectively with the *one-place* nets $\mathcal{N}_r = (\{p_r\}, \hat{T}, F, M_0)$ with language larger than L , viz. $M_0(p_r) = r_{init}$, $F(p_r, t_h) = r \circ t_h$ and $F(t_h, p_r) = t_h \circ r$ for all $t_h \in \hat{T}$. In order to reflect condition 1 in Problem 4, we now introduce open regions as follows.

Definition 26.*[Open Regions] Given two alphabets $T \subseteq \hat{T}$ and a non-empty prefix-closed and regular language $L \subseteq \hat{T}^*$, an open region of L is any bounded region r of L such that $r \circ t_h = 0$ for all letters $t_h \in \hat{T} \setminus T$.*

The suitability of the concept of regions for solving Problem 4 is shown by the following proposition, where \mathbb{C} means the complementation in \hat{T}^* .

Proposition 27. *Problem 4 has a solution \mathcal{N} if and only if for all $w \in L$ and $t \in T$, $wt \notin L \Rightarrow r/w < r \circ t$ for some open region r of L (r disables t after w).*

Let R be any finite and minimal set of open regions of L such that, for any $t \in T$ and for any minimal word $wt \notin L$, some region in R disables t after w . Then $L \subseteq L(\mathcal{N}) \subseteq \mathbb{C}((LT \setminus L) \cdot \hat{T}^)$ for the open net $\mathcal{N} = (P, \hat{T}, F, M_0)$ as follows:*

- P is a set of places p_r in bijective correspondence with the regions $r \in R$,
- $M_0(p_r) = r_{init}$ and for any $t \in \hat{T}$, $F(p_r, t) = r \circ t$ and $F(t, p_r) = t \circ r$. Moreover, \mathcal{N} is bounded in restriction to L , hence it is a solution to Problem 4.

PROOF. Let $\mathcal{N} = (P, \hat{T}, F, M_0)$ be a solution to Problem 4. Let $w \in L$ and $t \in T$ such that $wt \notin L$. In view of the net firing rule, there must exist some place $p \in P$ such that $F(p, t) > M_0(p) +$

$\sum_h \psi(w)(t_h) \times (F(t_h, p) - F(p, t_h))$. Let $r = \langle r_{init}, r^\circ t_1, t_1^\circ r, \dots, r^\circ t_k, t_k^\circ r \rangle$ be the associated vector defined with $r_{init} = M_0(p)$, $r^\circ t_h = F(p, t_h)$ and $t_h^\circ r = F(t_h, p)$ for all $t_h \in \widehat{T}$. Then r is an open region of L and r disables t after w .

Conversely let R be a finite set of open regions of L as described in the proposition, and let \mathcal{N} be the associated net defined *ibidem*. In view of the definition of open regions, the minimality of R , and the correspondence between regions in R and places of \mathcal{N} , conditions 1, 2 and 3 of Problem 4 are clearly satisfied. Condition 4 follows from the hypothesis that for any minimal word $wt \notin L$, some region in R disables t after w . Condition 5 follows from the fact that any open region of L is by definition a bounded region of L . Finally, $L \subseteq L(\mathcal{N})$ and $L(\mathcal{N}) \subseteq \mathbb{C}((LT \setminus L) \cdot \widehat{T}^*)$ are mere restatements of the conditions 3 and 4. ■

The next proposition helps computing the open regions of a regular language.

Proposition 28. *Given an integral vector $r = \langle r_{init}, r^\circ t_1, t_1^\circ r, \dots, r^\circ t_k, t_k^\circ r \rangle$ with non-negative entries, let $r \times w$ be defined for $w \in \widehat{T}^*$ inductively with $r \times \epsilon = 0$ and $r \times wt = r \times w - r^\circ t + t^\circ r$. Then for any prefix-closed regular language L over \widehat{T} , r is an open region of L if and only if the following conditions hold:*

- i) $r \times w = 0$ for every word w such that $vw^* \subseteq L$,
- ii) $r/w \geq r^\circ t$ for every $w \in L$ such that $wt \in L$,
- iii) $r^\circ t_h = 0$ for every letter $t_h \in \widehat{T} \setminus T$.

PROOF. Condition (i) is necessary because $r/vw^n = r/v + n(r \times w)$ for all n . Indeed, if $r \times w < 0$ then it is not possible that $r/vw^n \geq 0$ for all n , as required by the definition of regions, and if $r \times w > 0$ then it is not possible that $r/vw^n < B$ for some bound B for all n , as required by the definition of bounded regions. The remaining two conditions are obviously necessary.

Conversely, suppose that conditions (i,ii,iii) hold. In order to show that r is an open region of L , it suffices to prove that there exists a finite bound $B \in \mathbb{N}$ such that $r/w \leq B$ for all $w \in L$. As L is regular and prefix-closed, $L = L(A)$ for some finite deterministic automaton $A = (Q, \widehat{T}, \delta, q_0)$ where $\delta : Q \times \widehat{T} \rightarrow Q$ is a partial map, all states in Q can be reached from q_0 , and they are all accepting. Define $L' = \{w' \in L \mid w' = uvv' \Rightarrow \delta(q_0, u) \neq \delta(q_0, uv) \vee v = \epsilon\}$. Clearly, L' is a finite set of words. We claim that $B = \max\{r/w' \mid w' \in L'\}$ is an adequate bound. To establish this claim, it suffices to show that for any $w \in L \setminus L'$, $r/w = r/w'$ for some $w' \in L'$. We prove this property by induction on the length of words. Let $w \in L \setminus L'$, hence $w = uvv'$ and $\delta(q_0, u) = \delta(q_0, uv)$ for some $v \neq \epsilon$. As $uv^* \subseteq L$, $r \times v = 0$ by condition (i). Therefore, $r/w = r_{init} + r \times w = r_{init} + r \times uv' = r/uv'$.

Now $uv' \in L$ and it is strictly shorter than w . The validity of the claim follows by induction. \blacksquare

Propositions 27 and 28 provide a basis for deciding Problem 4 and hence for synthesizing open nets from regular languages. The presentation of the decision and synthesis procedure given in the end of the section is cut in two parts. In the first part, we show that the open regions of a regular language may be characterized by a finite linear system. In the second part, we show that deciding Problem 4 amounts to deciding for a *finite* set of minimal words $wt \notin L$ whether some open region disables t after w . The net \mathcal{N} solution to Problem 4 is synthesized from this finite set of disabling regions as indicated in Prop. 27.

4.1 A finite linear characterization of open regions

Let $L \subseteq \hat{T}^*$ be a prefix closed and regular language over $\hat{T} = \{t_1, \dots, t_k\}$. Then $L = L(A)$ for some finite deterministic automaton $A = (Q, \hat{T}, \delta, q_0)$ where $\delta : Q \times \hat{T} \rightarrow Q$ is a partial map, all states in Q can be reached from q_0 , and they are all accepting. We construct from A a finite linear system such that a non-negative integer vector $r = \langle r_{init}, r \circ t_1, t_1 \circ r, \dots, r \circ t_k, t_k \circ r \rangle$ is an open region of L if and only if it is a solution of this system.

The construction of the linear system is based on a partial unfolding of A into a finite automaton $\mathcal{U}A = (Q', \hat{T}, \delta', q'_0)$ with components as follows. Q' is a subset of words of $L(A)$, constructed inductively from the single element $q'_0 = \epsilon$ according to the completion rule stated hereafter. Let $w \in Q'$, then for any $t \in \hat{T}$, $wt \in Q'$ and $\delta'(w, t) = wt$ if and only if $\delta(q_0, wt)$ is defined and differs from $\delta(q_0, v)$ for every prefix v of w . This yields a finite *spanning* tree. A finite number of *chords* are then added by setting $\delta'(vu, t) = v$ whenever $vu, v \in Q'$ and $\delta(q_0, vut) = \delta(q_0, v)$.

Proposition 29. *Let $r = \langle r_{init}, r \circ t_1, t_1 \circ r, \dots, r \circ t_k, t_k \circ r \rangle$ be a non-negative integer vector. Then r is an open region of $L(A)$ if and only if:*

- i) $r \times ut = 0$ for every chord $vu \xrightarrow{t} v$ in $\mathcal{U}A$,
- ii) $r/w \geq r \circ t$ for every edge $w \xrightarrow{t} w'$ in $\mathcal{U}A$,
- iii) $r \circ t_h = 0$ for every letter $t_h \in \hat{T} \setminus T$.

PROOF. In view of Prop. 28, the stated conditions are necessary. Conversely, if (i,ii,iii) hold, then r is an open region of $L(A)$ because $L(A) = L(\mathcal{U}A)$. One may indeed reproduce the reasoning followed in the proof of Prop. 28 to show that r is a bounded region of $L(\mathcal{U}A)$. \blacksquare

Each condition $r \times ut = 0$ in Prop. 29 is equivalent to the linear homogeneous equation:

$$\Sigma_{h=1}^k \vec{X}(h) \times (t_h^\circ r - r^\circ t_h) = 0 \quad (1)$$

where $\vec{X} = \psi(ut)$ is the Parikh image of the cycle ut in \mathcal{UA} . Similarly, each condition $r/w \geq r^\circ t$ in Prop. 29 is equivalent to the linear homogeneous inequality:

$$r_{init} + \Sigma_{h=1}^k \vec{Y}(h) \times (t_h^\circ r - r^\circ t_h) - (r^\circ t) \geq 0 \quad (2)$$

where $\vec{Y} = \psi(w)$ is the Parikh image of the path w in \mathcal{UA} .

Let \mathcal{REG} denote the finite system of linear equations (1) and inequalities (2) in the $2k + 1$ integer variables r_{init} , $t_h^\circ r$ and $r^\circ t_h$ derived from \mathcal{UA} , augmented with $r_{init} \geq 0$, $t_h^\circ r \geq 0$ and $r^\circ t_h \geq 0$ for all h , and $r^\circ t_h = 0$ for every letter $t_h \in \hat{T} \setminus T$.

Proposition 30. *An integer vector r is an open region of $L = L(A)$ if and only if all linear constraints in \mathcal{REG} are satisfied.*

PROOF. This is an immediate consequence of Prop. 29. ■

4.2 The decision and synthesis procedure

From propositions 27 and 30, $L(A) = L(\mathcal{N})$ for some open P/T-net \mathcal{N} if and only if, for each letter $t \in T$ and for each state w of \mathcal{UA} , either $w \xrightarrow{t} w'$ for some w' or $r/w < r^\circ t$ for some open region r . Now $r/w < r^\circ t$ is equivalent to the linear homogeneous inequality:

$$r_{init} + \Sigma_{h=1}^k \vec{Y}(h) \times (t_h^\circ r - r^\circ t_h) - (r^\circ t) < 0 \quad (3)$$

where \vec{Y} is the Parikh image of w . Inequality (3) holds for some open region r of $L(A)$ if and only if the linear system \mathcal{REG} extended with the inequality:

$$r_{init} + \Sigma_{h=1}^k \vec{Y}(h) \times (t_h^\circ r - r^\circ t_h) - (r^\circ t) \leq -1 \quad (4)$$

is feasible in \mathbb{Q}^{2k+1} , which can be decided in polynomial time. Indeed, rational solutions always induce integer solutions. Let R be any minimal set of open regions of $L = L(A)$ large enough for disabling t after w whenever $t \in T$, w is a state of \mathcal{UA} , and $wt \notin L(A)$. Then $L = L(\mathcal{N})$ where \mathcal{N} is constructed as indicated in proposition 27.

5 Conclusion

Let us first summarize the paper. We have refined the model of distributed Petri nets studied in [2], by considering net architectures (λ, G) where $\lambda : T \rightarrow S$ partitions the set of transitions over sites and $G = (S, \rightarrow)$ specifies the possible communications between sites. We have stated conditions under which one can check in a modular way the coherency of a product specification $\{L_s \mid s \in S\}$, where each L_s is a regular language over $\cup\{\lambda^{-1}(s') \mid s' = s \vee s' \rightarrow s\}$. We have shown that the distributed P/T-net synthesis problem may then be decomposed into $|S|$ independent “open” net synthesis problems, which may be solved by a simple adaptation of already known techniques.

We now compare our approach to the synthesis of distributed systems with the earlier approach taken in [6]. In that work, the authors start from global specifications, namely a transition system TS over a global alphabet T . The goal is to produce an implementation by a product of transition systems $\{TS_s \mid s \in S\}$ over some distributed alphabet $\{T_s \mid s \in S\}$ with $T = \cup_s T_s$. The drawback is the impossibility to deal with large specifications, since these are monolithic. We tried in [9] to adopt this framework for the synthesis of P/T-nets from modular specifications. For that purpose, we replaced the global specification TS with a Modular Transition System [16], as follows.

A modular transition system comprises a set of *local modules* TS'_s over *disjoint* alphabets $T'_s = T_s \setminus \cup\{T_{s'} \mid s' \neq s\}$ plus a *synchronizing module* \mathcal{S} . The synchronizing module defines all remaining transitions in $\cup\{T_s \cap T_{s'} \mid s \neq s'\}$ as jumps between vectors of states of the local modules. The synthesis of P/T-nets from modular specifications is unfortunately more symbolic than modular: one can avoid computing a global transition system, but one cannot cut the synthesis problem to smaller independent synthesis problems. This is the reason why we chose here to start from a product specification $\{L_s \mid s \in S\}$, which would rather look as implementations in the spirit of [6].

Our goals differ also significantly from the goals pursued in [6] in that we want to synthesize distributed P/T-nets, and this introduces a lower level of implementation. This led us to consider that the communication network on which the implementation is mapped has some importance, hence the particular form of our product specifications $\{L_s \mid s \in S\}$ where each L_s is a regular language over $\cup\{\lambda^{-1}(s') \mid s' = s \vee s' \rightarrow s\}$.

Bibliography

- [1] Badouel, E., Bernardinello, L., Darondeau, Ph.: Polynomial Algorithms for the Synthesis of Bounded Nets. Proc. CAAP, LNCS **915** (1995) 647-679.

- [2] Badouel, E., Caillaud, B., Darondeau, Ph.: Distributing finite automata through Petri net synthesis. *Formal Aspects of Computing* **13** (2002) 447-470.
- [3] Bernardinello, L., De Michelis, G., Petruni, K., Vigna, S.: On the Synchronic Structure of Transition Systems. In J. Desel, editor, *Structures in Concurrency Theory*, Workshops on Computing, Springer Verlag (1996) 11-31.
- [4] Bernardinello, L.: Synthesis of Net Systems. Proc. ATPN, LNCS **691** (1993) 11-31.
- [5] Cortadella, J., Kishinevsky, M., Lavagno, L., Yakovlev, A.: Deriving Petri Nets from Finite Transition Systems. *IEEE Trans. on Computers* **47**,8 (1998) 859-882.
- [6] Castellani, I., Mukund, M., Thiagarajan, P.S.: Synthesizing Distributed Transition Systems from Global Specifications. Proc. FSTTCS, LNCS **1738** (1999) 219-231.
- [7] Christensen, S., Petrucci, L.: Modular State Space Analysis of Coloured Petri nets. Proc. ATPN, LNCS **935** (1995) 201-217.
- [8] Darondeau, P.: Region Based Synthesis of P/T-nets and its Potential Applications. Proc. ATPN, LNCS **1825** (2000) 16-23.
- [9] Darondeau, P., Petrucci, L.: Modular Automata 2 Distributed Petri Nets 4 Synthesis. INRIA-RR 6192 (2007).
- [10] Desel, J., Reisig, W.: The Synthesis Problem of Petri Nets. *Acta Informatica* **33** (1996) 297-315.
- [11] Duboc, C.: Mixed Product and Asynchronous Automata. *Theoretical Computer Science* **48**, 3 (1986) 183-199.
- [12] Ehrenfeucht, A., Rozenberg, G.: Partial (Set) 2-Structures; *Part I*: Basic Notions and the Representation Problem. *Acta Informatica* **27** (1990) 315-342.
- [13] Ehrenfeucht, A., Rozenberg, G.: Partial (Set) 2-Structures; *Part II*: State Spaces of Concurrent Systems. *Acta Informatica* **27** (1990) 343-368.
- [14] Hoogers, P.W., Kleijn, H.C.M., Thiagarajan, P.S.: A Trace Semantics for Petri Nets. Proc. ICALP, LNCS **623** (1992) 595-604.
- [15] Mukund, M.: Petri Nets and Step Transition Systems. *International Journal of Foundations of Computer Science* **3**,4 (1992) 443-478.
- [16] Petrucci, L.: Modélisation, vérification et applications. Mémoire d'habilitation à diriger des recherches, Université d'Evry (2002).
- [17] van Glabbeek, R.J., Weijland, W.P.: Branching time and abstraction in bisimulation semantics. Proc. IFIP Congress, North-Holland/IFIP (1989) 613-618.