Fundamenta Informaticae 99 (2010) 1–14
DOI 10.3233/FI-2010-254
IOS Press

Minimal Regions of ENL-Transition Systems

Maciej Koutny* **and Marta Pietkiewicz-Koutny**
School of Computing Science
Newcastle University
Newcastle upon Tyne NE1 7RU, U.K.
{*maciej.koutny,marta.koutny*}*@ncl.ac.uk*

Abstract. One of the possible ways of constructing concurrent systems is their automated synthesis from behavioural specifications. In this paper, we look at a particular instance of this approach which aims at constructing GALS (globally asynchronous locally synchronous) systems from specifications given in terms of transition systems with arcs labelled by steps of executed actions. GALS systems are represented by Elementary Net Systems with Localities (ENL-systems), each locality defining a set of co-located actions. The synthesis procedure is based on the regions of transition systems and we provide a number of criteria aimed at generating a minimal set of regions (conditions) of an ENL-system generating a given transition system.

Keywords: theory of concurrency, Petri nets, localities, analysis and synthesis, step sequence semantics, conflict, theory of regions, transition systems.

1. Introduction

A number of computational systems exhibit behaviour adhering to the 'globally asynchronous locally (maximally) synchronous' paradigm. Examples can be found in hardware design, where a VLSI chip may contain multiple clocks responsible for synchronising different subsets of gates [6], and in biologically inspired membrane systems representing cells within which biochemical reactions happen in synchronised pulses [14]. To capture such systems in a formal manner, [8] introduced *Place/Transition-nets with localities* (PTL-nets), where each locality identifies a distinct set of events which must be executed synchronously, i.e., in a maximally concurrent manner (akin to *local maximal concurrency*).

*Address for correspondence: School of Computing Science, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K

An attractive way of constructing complex computing systems is their automated synthesis from a range of behavioural specifications, e.g., given in terms of suitable transition systems. In such a case, the synthesis procedure is often based on the regions of a transition system [2, 3, 4, 7, 12, 13, 15].

In the paper [9], we adapted the proposal of [8] to the case of Elementary Net Systems (EN-systems) — a fundamental class of safe Petri nets [13] — leading to EN-systems with localities (and context arcs in [10]). We aimed there at finding a characterisation of all transition systems generated by such nets, and in so doing provide a solution to the corresponding synthesis problem from transition systems to Petri nets. The papers [9, 10] suitably adapted the classical theory of regions [3] to cope with local maximal concurrency and this work was later generalised to other classes of Petri nets in [5]. In this paper, we consider again EN-system with localities (ENL-systems) introduced and investigated in [9], but this time we focus on the algorithmic efficiency of the synthesis procedure (note that the problem is NP-complete which can be shown following the argument made in [2]).

> To explain the basic idea behind ENL-systems, let us consider the net in Figure 1 modelling two co-located consumers and one producer residing in a remote location. In the initial state, the net can execute the singleton step $\{c_4\}$. Another enabled step is $\{p_1\}$ which removes the token from b_1 and puts two tokens, into b and b_2. In this new state, there are three enabled steps, viz. $\{p_2\}$, $\{c_1, c_4\}$ and $\{p_2, c_1, c_4\}$. The last one, $\{p_2, c_1, c_4\}$, corresponds to what is usually called *maximal concurrency* as no more activities can be added to it without violating the constrains imposed by the available resources (represented by tokens). However, the previously enabled step $\{c_4\}$ which is still *resource (or token) enabled* is disallowed by the control mechanism of ENL-systems. It rejects a resource enabled step like $\{c_4\}$ since we can add to it c_1 co-located with c_4 obtaining a step which is resource enabled. In other words, the control mechanism employed by ENL-systems (and PTL-nets) is that of *local maximal concurrency* as indeed postulated by the GALS systems execution rule.

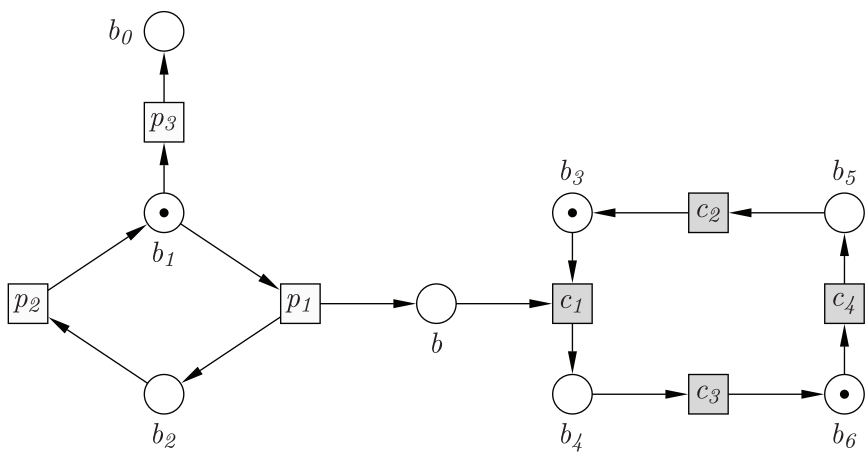

Figure 1. A one-producer/two-consumers system (shading of boxes indicates the co-location of events they represent).

The synthesis procedure of [9] assumed that one is using the full set of regions of a transition system to synthesise an ENL-system. In this paper, we discuss results contributing to an improvement of the synthesis procedure by considering *minimal regions*. In essence, one is interested to construct as small as possible ENL-system generating a given behaviour represented by a transition system. We investigate different ways of achieving this, in particular, by re-evaluating the concept of minimal regions introduced in the context of ENI-system synthesis [15].

2. Preliminaries

In what follows, E is a fixed finite non-empty set of *events*. A *co-location relation* is any equivalence relation $\simeq$ on the set of events. For an event e and a non-empty set of events u (a *step*), we will denote $e \simeq u$ whenever there is at least one event $f \in u$ satisfying $e \simeq f$.

A *step transition system* [1] is a triple $\mathfrak{ts} \stackrel{\text{df}}{=} (S, T, s_{in})$ where S is a non-empty finite set of *states*,

$$T \subseteq S \times (2^E \setminus \{\varnothing\}) \times S$$

is a finite set of *transitions*, and $s_{in} \in S$ is the *initial state*. We will write $s \xrightarrow{u} s'$ (or simply $s \xrightarrow{u}$) whenever (s, u, s') is a transition. Moreover, for every state s:

- $allSteps_s$ is the set of all steps labelling arcs outgoing from s.
- $minSteps_s$ is the set of all minimal steps (w.r.t. set inclusion) in $allSteps_s$.
- E_s is the union of all the steps labelling arcs outgoing from s.
- $\simeq_s$ is the restriction of a co-location relation $\simeq$ to $E_s \times E_s$.

To ease the presentation, we will assume that each event occurs in at least one of the steps labelling the transitions of $\mathfrak{ts}$.

2.1. ENL-systems

A *net* is a tuple $\mathfrak{net} \stackrel{\text{df}}{=} (B, E, F)$ such that B is a finite set of *conditions* disjoint from E, and

$$F \subseteq (B \times E) \cup (E \times B)$$

is the *flow relation*. The meaning and graphical representation of conditions, events and the flow relation is the same as in the standard net theory. For every event e, its *pre-conditions* and *post-conditions* are given respectively by

$$^\bullet e \stackrel{\text{df}}{=} \{b \mid (b, e) \in F\} \text{ and } e^\bullet \stackrel{\text{df}}{=} \{b \mid (e, b) \in F\}$$

(both sets are assumed non-empty and disjoint). The dot-notation extends in the usual way to sets of events. Two events are *in conflict* (or conflicting) if they share a pre-condition or share a post-condition.

An *elementary net system with localities* (ENL-system) is a tuple

$$\mathfrak{enl} \stackrel{\text{df}}{=} (B, E, F, \simeq, c_{in})$$

such that (B, E, F) is the underlying net, $\simeq$ is a co-location relation, and $c_{in} \subseteq B$ is the *initial case* (in general, any subset of B is a *case*). In diagrams, boxes representing co-located events are shaded in the same way (see Figure 1).

The semantics of $\mathfrak{enl}$ is based on steps of simultaneously executed events. We first define *potential steps* as non-empty sets of non-conflicting events. A potential step u is then *resource enabled* at a case c if ${}^\bullet u \subseteq c$ and $u^\bullet \cap c = \varnothing$, and *control enabled* if, in addition, there is no event $e \notin u$ such that $e \simeq u$ and the step $u \cup \{e\}$ is resource enabled at c. If both conditions are satisfied, there is a *transition* from c to the case $c' = (c \setminus {}^\bullet u) \cup u^\bullet$, and we denote this by $c[u\rangle c'$ (or simply $c[u\rangle$). The *step transition system* of $\mathfrak{enl}$ is then given by:

$$\mathfrak{ts}_{\mathfrak{enl}} \stackrel{\text{df}}{=} \Big(C_{\mathfrak{enl}} \, , \, \{(c,u,c') \in 2^B \times 2^E \times 2^B \mid c \in C_{\mathfrak{enl}} \wedge c[u\rangle c'\} \, , \, c_{in} \Big) \, ,$$

where $C_{\mathfrak{enl}}$ — the set of *reachable* cases — is the least set of cases containing c_{in} and closed w.r.t. the transition relation. To ease the presentation, we will assume that $\mathfrak{enl}$ does not have *dead events*, i.e., each event occurs in at least one of the steps labelling the arcs of the transition system $\mathfrak{ts}_{\mathfrak{enl}}$. It is easy to see that the following hold:

Fact 2.1. If a step u is resource enabled at c then there is a step w which is control enabled at c such that $u \subseteq w$ and $e \simeq u$, for every $e \in w \setminus u$.

Fact 2.2. Two events, e and f, resource enabled at a case c are in conflict *iff* there is no step resource enabled at c to which they both belong.

Note that we say that an event e is resource enabled at a case if the singleton step $\{e\}$ is.

2.2. ENL-transition systems

To link the nodes (global states) of a step transition system $\mathfrak{ts}$ with the conditions (local states) of the hypothetical ENL-system corresponding to it, we use the notion of a *region* defined as a triple

$$\mathfrak{r} \stackrel{\text{df}}{=} (in, r, out) \in 2^E \times 2^S \times 2^E$$

such that, for every transition $s \xrightarrow{u} s'$, the following hold:

R1 If $s \in r$ and $s' \notin r$ then $|u \cap in| = 0$ and $|u \cap out| = 1$.

R2 If $s \notin r$ and $s' \in r$ then $|u \cap in| = 1$ and $|u \cap out| = 0$.

R3 If $u \cap out \neq \varnothing$ then $s \in r$ and $s' \notin r$.

R4 If $u \cap in \neq \varnothing$ then $s \notin r$ and $s' \in r$.

Intuitively, the events in in 'enter' the set of states r, and those in out 'exit' r.

There are exactly two *trivial* regions satisfying $r = \varnothing$ or $r = S$, viz. $(\varnothing, \varnothing, \varnothing)$ and $(\varnothing, S, \varnothing)$. Moreover, (in, r, out) is a region iff so is its *complement* $(out, S \setminus r, in)$. In general, a region cannot be identified only by its set of states; in other words, in and out (also denoted $in_{\mathfrak{r}}$ and $out_{\mathfrak{r}}$) may not be recoverable from r.

The set of all non-trivial regions will be denoted by $\mathfrak{R}_{\mathfrak{ts}}$ and, for every state s, $\mathfrak{R}_s$ is the set of all non-trivial regions (in, r, out) containing s, i.e., $s \in r$. The sets of *pre-regions*, ${}^\circ e$, and *post-regions*, e°, of an event e comprise all the non-trivial regions (in, r, out) respectively satisfying $e \in out$ and $e \in in$. This extends in the usual way to sets of events.

To characterise transition systems of ENL-systems, we need two more notions. The set of *potential steps* comprises all non-empty sets u of events such that

$$^\circ e \cap {}^\circ f = e^\circ \cap f^\circ = \varnothing \ ,$$

for each pair of distinct events $e, f \in u$. A potential step u is then *region enabled at state* s if $^\circ u \subseteq \mathfrak{R}_s$ and $u^\circ \cap \mathfrak{R}_s = \varnothing$.

A step transition system $\mathfrak{ts} = (S, T, s_{in})$ is an *ENL-transition system* w.r.t. a co-location relation $\doteqdot$ if the following hold:

A1 Each state is reachable from the initial state.

A2 For every event e, both $^\circ e$ and e° are non-empty.

A3 For all distinct states s and s', $\mathfrak{R}_s \neq \mathfrak{R}_{s'}$.

A4 For every state s and step u, we have that $s \xrightarrow{u}$ *iff* u is region enabled at s and there is no event $e \notin u$ such that $e \doteqdot u$ and the step $u \cup \{e\}$ is region enabled at s.

One can show (see [9]) that the transition system of an ENL-system with the co-location relation $\doteqdot$ is an ENL-transition system w.r.t. $\doteqdot$. Moreover, one can see that the following hold:

Fact 2.3. If a step u is region enabled at a state s then there is a step w such that

$$s \xrightarrow{w} \quad \text{and} \quad u \subseteq w \quad \text{and} \quad e \doteqdot u \ ,$$

for every $e \in w \setminus u$.

Fact 2.4. If $s \xrightarrow{u} s'$ is one of the valid transitions, then the step u is region enabled at s and

$$\mathfrak{R}_s \setminus \mathfrak{R}_{s'} = {}^\circ u \quad \text{and} \quad \mathfrak{R}_{s'} \setminus \mathfrak{R}_s = u^\circ \quad \text{and} \quad {}^\circ u = \biguplus_{e \in u} {}^\circ e \quad \text{and} \quad u^\circ = \biguplus_{e \in u} e^\circ \ .$$

2.3. Synthesising ENL-systems

ENL-systems generate ENL-transition systems. The reverse also is true, and the translation from ENL-transition systems to the corresponding ENL-systems is based on regions.

Let $\mathfrak{ts} = (S, T, s_{in})$ be an ENL-transition system w.r.t. a (given) co-location relation $\doteqdot$. Then the net system *associated* with $\mathfrak{ts}$ is defined as:

$$\mathfrak{enl}^{\doteqdot}_{\mathfrak{ts}} \stackrel{\text{df}}{=} (\mathfrak{R}_{\mathfrak{ts}}, E, F_{\mathfrak{ts}}, \doteqdot, \mathfrak{R}_{s_{in}})$$

where

$$F_{\mathfrak{ts}} \stackrel{\text{df}}{=} \{(\mathfrak{r}, e) \in \mathfrak{R}_{\mathfrak{ts}} \times E \mid \mathfrak{r} \in {}^\circ e\} \cup \{(e, \mathfrak{r}) \in E \times \mathfrak{R}_{\mathfrak{ts}} \mid \mathfrak{r} \in e^\circ\} \ .$$

It turns out that such a construction always produces an ENL-system which, crucially, generates a transition system which is isomorphic to $\mathfrak{ts}$.

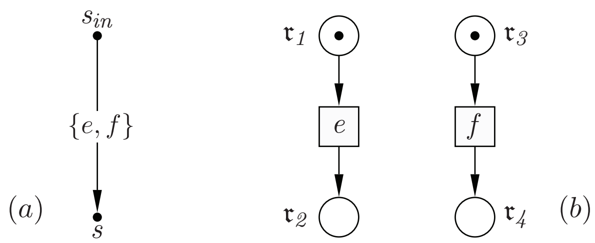

Figure 2. An ENL-transition system with co-located events e and f (a), and the ENL-system resulting from the synthesis (b).

Theorem 2.1. ([9])
Let $\mathfrak{ts}$ be an ENL-transition system w.r.t. a co-location relation $\simeq$. Then $\mathfrak{enl}_{\mathfrak{ts}}^{\simeq}$ is an ENL-system and its step transition system is isomorphic to $\mathfrak{ts}$. Moreover, the isomorphism ψ between $\mathfrak{ts}$ and the step transition system of $\mathfrak{enl}_{\mathfrak{ts}}^{\simeq}$ is given by $\psi(s) \stackrel{\text{df}}{=} \mathfrak{R}_s$, for every state s of $\mathfrak{ts}$.

The above construction is illustrated in Figure 2. Note that the non-trivial regions in this case are:

$$\begin{aligned}\mathfrak{r}_1 &= (\varnothing, \{s_{in}\}, \{e\}) \\ \mathfrak{r}_2 &= (\{e\}, \{s\}, \varnothing\}) \\ \mathfrak{r}_3 &= (\varnothing, \{s_{in}\}, \{f\}) \\ \mathfrak{r}_4 &= (\{f\}, \{s\}, \varnothing)\ .\end{aligned}$$

As mentioned above, the axioms **A1-A4** and the synthesis algorithm are formulated w.r.t. a co-location relation which is assumed to be known. However, one could argue that the distribution of events into different subsystems should be part of realistic synthesis procedure. Given that the number of co-location relations is finite for a given finite set of events, one might, of course, enumerate them all and check the axioms **A1-A4** for each and every one. This, however, would be both wasteful (as many potential relations are clearly inappropriate) and impractical (since the total number of co-location relations for n different events is the n-th number in the fast-growing sequence of *Bell numbers*). Though the following result is straightforward, it is potentially very useful.

Proposition 2.1. Let $\mathfrak{ts}$ be a step transition system. Moreover, let $\simeq$ and $\simeq'$ be two *state consistent* co-location relations, i.e., $\simeq_s$ and $\simeq'_s$ coincide for every state s. Then $\mathfrak{ts}$ is an ENL-transition system w.r.t. $\simeq$ *iff* it is an ENL-transition system w.r.t. $\simeq'$.

Basically, it says that, when checking the axioms **A1-A4**, what really matters are the restrictions of the co-location relations to sets of events locally enabled at each of the states of the transition system. Hence it suffices to check the axioms w.r.t. just one relation for any equivalence class of state consistent co-location relations. In some cases, however, the synthesis problem can be reduced to checking the axioms **A1-A4** for just *one* co-location relation. What is perhaps surprising, the class of such systems has *practical* motivation (see [11]).

An ENL-system has *localised conflicts* (or is ENL/LC-system) if no conflicting non-co-located events are resource enabled at some reachable case.

Proposition 2.2. ([11])
Let $\mathfrak{enl}$ and $\mathfrak{enl}'$ be two ENL-systems with the co-location relations $\simeq$ and $\simeq'$, respectively. If they generate the same transition system $\mathfrak{ts}$ and $\simeq_s = \simeq'_s$, for every state s of $\mathfrak{ts}$, then $\mathfrak{enl}$ has localised conflicts *iff* $\mathfrak{enl}'$ has localised conflicts.

Proposition 2.3. ([11])
Let $\mathfrak{ts}$ be the transition system of an ENL-transition system $\mathfrak{enl}$ with the co-location relation $\simeq$. Then $\mathfrak{enl}$ is an ENL/LC-system *iff* there is no state s of $\mathfrak{ts}$ and two distinct events $e \not\simeq_s f$ in E_s which do not belong to at least one step in $allSteps_s$.

A key property of ENL/LC-systems is captured by the next result.

Theorem 2.2. ([11])
Let $\mathfrak{ts}$ be the transition system of an ENL/LC-system, and s be one of its states. Then two distinct events in E_s are co-located *iff* either there is no step in $allSteps_s$ to which the two events belong, or there is a step in $minSteps_s$ to which the two events belong.

Corollary 2.1. ([11])
Let $\mathfrak{ts}$ be the transition system of an ENL/LC-system with the co-location relation $\simeq$. Then, for every state s of $\mathfrak{ts}$ we have $\simeq_s = \simeq^{\mathfrak{ts},s}$, where:

$$\simeq^{\mathfrak{ts},s} \stackrel{\mathrm{df}}{=} \bigcup_{u \in minSteps_s} u \times u \ \cup \ \Big((E_s \times E_s) \setminus \bigcup_{u \in allSteps_s} u \times u \Big) . \tag{1}$$

For the class of ENL/LC-systems, we are interested in addressing the following problem:

Synthesis Problem 1. Given a step transition system $\mathfrak{ts}$ find as efficient as possible a way of checking whether it is isomorphic to the transition system of an ENL/LC-system, and if so construct such a system.

We can approach this problem in stages. First, for every state s of $\mathfrak{ts}$, we construct $\simeq^{\mathfrak{ts},s}$ as in Corollary 2.1, and then form the co-location relation:

$$\simeq^{\mathfrak{ts}}_{min} \stackrel{\mathrm{df}}{=} \Big(\bigcup_{s} \simeq^{\mathfrak{ts},s} \Big)^{*} .$$

Next, we check whether $\simeq^{\mathfrak{ts},s}$ is equal to $\simeq^{\mathfrak{ts}}_{min} |_{E_s \times E_s}$, for every state s. If this is not the case, we know that Synthesis Problem 1 fails. Otherwise, in view of Corollary 2.1, $\simeq^{\mathfrak{ts}}_{min}$ is the finest (w.r.t. the number of equivalence classes) possible co-location relation for $\mathfrak{ts}$ although we still do not know whether it provides a positive answer to the synthesis problem. To establish this, we proceed to check whether the axioms **A1-A4** are satisfied for the co-location relation $\simeq^{\mathfrak{ts}}_{min}$. If so, $\mathfrak{ts}$ is an ENL-transition system, and we can use the procedure from Section 2.3 to obtain the synthesised ENL-system $\mathfrak{enl}^{\simeq^{\mathfrak{ts}}_{min}}_{\mathfrak{ts}}$.

The above outlines a procedure which takes advantage of the structural (and local) properties of the original step transition system. If it succeeds, we obtain an ENL/LC-system which solves the synthesis problem. Moreover, one can easily characterise all other ENL/LC-systems with this property using Propositions 2.1 and 2.2.

3. Constructing regions of ENL-systems

The aim of this paper is to investigate ways in which the regions of an ENL-transition system are constructed and when some of these are redundant from the point of view of generating a net with the same behaviour. Note that such an investigation is completely independent of the considerations concerning co-location relations presented in the previous section.

We first develop a general characterisation of the set $\mathfrak{R}^r$ comprising all the non-trivial regions based on a given set of states r. We need the following auxiliary notations:

- U^r_{In} are the steps labelling transitions incoming to r, i.e., $u \in U^r_{In}$ iff $s \xrightarrow{u} s'$, for some $s \notin r$ and $s' \in r$.
- U^r_{Out} are the steps labelling transitions outgoing from r, i.e., $u \in U^r_{Out}$ iff $s \xrightarrow{u} s'$, for some $s \in r$ and $s' \notin r$.
- E^r_{In} are the events occurring only in steps labelling transitions incoming to r, i.e., $e \in E^r_{In}$ iff $s \xrightarrow{u} s'$ and $e \in u$ implies $s \notin r$ and $s' \in r$.
- E^r_{Out} are the events occurring only in steps labelling transitions outgoing from r, i.e., $e \in E^r_{Out}$ iff $s \xrightarrow{u} s'$ and $e \in u$ implies $s \in r$ and $s' \notin r$.
- IN^r is the set of all $in \subseteq E^r_{In}$ such that $|in \cap u| = 1$, for every $u \in U^r_{In}$.
- OUT^r is the set of all $out \subseteq E^r_{Out}$ such that $|out \cap u| = 1$, for every $u \in U^r_{Out}$.

It is straightforward to observe that looking for events which enter a set of states r can be done completely separately from looking for events which exit the same set of states. Moreover, any combination of entering and exiting events gives rise to a valid region based on r.

Proposition 3.1. $(in, r, out) \in \mathfrak{R}^r$ *iff* $in \in IN^r$ and $out \in OUT^r$.

Proof:
Follows from **R1-R4** and the definitions above. □

Corollary 3.1. If (in, r, out) and (in', r, out') are two regions of an ENL-transition system, then so is (in, r, out').

Suppose now that we are given the task of finding all the regions in $\mathfrak{R}^r$. The main algorithmic task will be to construct the sets belonging to IN^r (and OUT^r as well). This can be done by a reduction to a version of the vertex covering problem, formulated in the following way.

> Construct an undirected graph with the vertices $U^r_{In} \cup E^r_{In}$ and arcs joining $u \in U^r_{In}$ and $e \in E^r_{In}$ whenever $e \in u$. Then $in \subseteq E^r_{In}$ belongs to IN^r *iff* each vertex in U^r_{In} is adjacent to exactly one vertex belonging to in.

3.1. Sound reductions

Suppose now that we have generated an ENL-system $\mathfrak{enl}$ which is a sub-ENL-system of $\mathfrak{enl}_{\mathfrak{ts}}^{\frown}$ defined in Section 2.3 (in other words, $\mathfrak{enl}$ has been obtained from $\mathfrak{enl}_{\mathfrak{ts}}^{\frown}$ by deleting some conditions together with the adjacent arcs) and that the transition system of $\mathfrak{enl}$ is isomorphic to $\mathfrak{ts}$. We would like to reduce $\mathfrak{enl}$ further by throwing away a condition (region) $\mathfrak{r}$ without violating the property of being an ENL-system with step transition system isomorphic to $\mathfrak{ts}$. Let us denote the resulting ENL-system $\mathfrak{enl}_{\mathfrak{r}}$. Then the following are simple yet useful definition and result.

We call $\mathfrak{r}$ a *sound reduction* if, for every case c reachable in $\mathfrak{enl}$ and every step u, if u is resource enabled at $c \setminus \{\mathfrak{r}\}$ in $\mathfrak{enl}_{\mathfrak{r}}$ then u is resource enabled at c in $\mathfrak{enl}$. In other words $\mathfrak{r}$ is a sound reduction if its deletion does not 'unblock' any events in reachable cases. Hence it is redundant from the behavioural point of view.

Proposition 3.2. If $\mathfrak{r}$ is a sound reduction, then the transition systems of $\mathfrak{enl}$ and $\mathfrak{enl}_{\mathfrak{r}}$ are isomorphic.

Proof:
Consider a reachable case c of $\mathfrak{enl}$. Then the set of resource enabled steps at c in $\mathfrak{enl}$ is exactly the same as the set of resource enabled steps at $c \setminus \{\mathfrak{r}\}$ in $\mathfrak{enl}_{\mathfrak{r}}$. This follows from the definition of sound reduction, and the fact that deleting conditions preserves the resource enabledness of steps. Hence, since the co-location relations of $\mathfrak{enl}$ and $\mathfrak{enl}_{\mathfrak{r}}$ are the same, we have that, for every reachable case c of $\mathfrak{enl}$, the set of control enabled steps at c in $\mathfrak{enl}$ is exactly the same as the set of control enabled steps at $c \setminus \{\mathfrak{r}\}$ in $\mathfrak{enl}_{\mathfrak{r}}$. □

The definition of a sound reduction is *dynamic* and so it may be difficult to apply in practice. Therefore, we will now search for *static* characterisations of sound reductions. In each case, we will be aiming at showing that $\mathfrak{r}$ is redundant if other condition(s) are present. The first static sound reduction is based on pairs of complement regions. In practical terms, it allows one to reduce the number of necessary conditions by half.

Proposition 3.3. If $\mathfrak{r} = (in, r, out)$ and $\overline{\mathfrak{r}} = (out, S \setminus r, in)$ are two conditions in $\mathfrak{enl}$ and deleting $\overline{\mathfrak{r}}$ leads to an ENL-system, then $\overline{\mathfrak{r}}$ is a sound reduction.

Proof:
Suppose that $\mathfrak{enl}_{\overline{\mathfrak{r}}}$ is an ENL-system, but $\overline{\mathfrak{r}}$ is not a sound reduction in $\mathfrak{enl}$. Then there exists a case c in $\mathfrak{enl}$ and a step u resource enabled at $c \setminus \{\overline{\mathfrak{r}}\}$ in $\mathfrak{enl}_{\overline{\mathfrak{r}}}$ and not resource enabled at c in $\mathfrak{enl}$. That means $\overline{\mathfrak{r}}$ is not marked at c and is a pre-condition of some event in u, or $\overline{\mathfrak{r}}$ is marked and is a post-condition of some event in u. Let us consider the first case (similar reasoning can be used in the second case). From the construction of $\mathfrak{enl}_{\mathfrak{ts}}^{\frown}$ and Theorem 2.1 we know that regions become conditions in the synthesised net, and if a condition (equivalent to some region) is marked in some case then the condition equivalent to its complement region is unmarked. We know as well that a complement of a region that is a pre-condition of some event e is a post-condition of this event, i.e.,

$$\overline{\mathfrak{r}} \in {}^{\circ}e \iff e \in out_{\overline{\mathfrak{r}}} \iff e \in in_{\mathfrak{r}} \iff \mathfrak{r} \in e^{\circ} \ .$$

From the above it follows that if $\overline{\mathfrak{r}}$ is not marked at c and is a pre-condition of some event in u then $\mathfrak{r}$ is marked at c and is a post-condition of some event in u. But then $\mathfrak{r}$ would make u resource disabled at $c \setminus \{\overline{\mathfrak{r}}\}$ in $\mathfrak{enl}_{\overline{\mathfrak{r}}}$, producing a contradiction. □

Note that assuming that deleting $\bar{\mathfrak{r}}$ leads to an ENL-system simply means that $\bar{\mathfrak{r}}$ is not a unique pre-condition or post-condition of any event.

3.2. Minimal regions

The next sound reduction we investigate concerns a situation when two (compatible) regions based on disjoint sets of states can be combined into a single (larger) region which is based on the union of their sets of states. It turns out that in such a case the larger region is redundant provided that the two smaller regions are present. To define compatible regions, we need auxiliary notations.

Given an event e and a set of states r, we denote

$$e \in {}^{\blacktriangledown}r \qquad e \in {}^{\triangledown}r \qquad e \in r^{\blacktriangledown} \qquad e \in r^{\triangledown}$$

if, for every transition $s \xrightarrow{u} s'$ with $e \in u$, we have respectively,

$$s' \in r \qquad s' \notin r \qquad s \in r \qquad s \notin r \ .$$

Then two regions of an ENL-transition system, $\mathfrak{r} = (in, r, out)$ and $\mathfrak{r}' = (in', r', out')$, are *compatible* if

$$out \subseteq {}^{\blacktriangledown}r' \cup {}^{\triangledown}r' \quad in \subseteq r'^{\blacktriangledown} \cup r'^{\triangledown} \quad out' \subseteq {}^{\blacktriangledown}r \cup {}^{\triangledown}r \quad in' \subseteq r^{\blacktriangledown} \cup r^{\triangledown}$$

and, moreover, $r \cap r' = \varnothing$. Intuitively, compatibility means that events entering (or exiting) the states of one of the two regions label transitions which either all exit or all never exit (resp. either all enter or all never enter) the states of the other region. As a result, there is never a confusion about the entry/exit status of an event when we consider the union of the sets of states on which the two regions are based.

Proposition 3.4. If $\mathfrak{r} = (in, r, out)$ and $\mathfrak{r}' = (in', r', out')$ are two non-trivial compatible regions of an ENL-transition system $\mathfrak{ts}$, then the following is a (possibly trivial) region of $\mathfrak{ts}$:

$$\mathfrak{r} \oplus \mathfrak{r}' \stackrel{\mathrm{df}}{=} (in \cup in' \setminus F, r \cup r', out \cup out' \setminus F) \ ,$$

where $F \stackrel{\mathrm{df}}{=} {}^{\blacktriangledown}r \cap r'^{\blacktriangledown} \cup {}^{\blacktriangledown}r' \cap r^{\blacktriangledown}$.

Proof:
Suppose $s \xrightarrow{u} s'$ is a transition in $\mathfrak{ts}$. We only show **R1** and **R3** for $\mathfrak{r} \oplus \mathfrak{r}'$ as **R2** and **R4** are symmetric. To prove **R1**, let us assume that $s \in r \cup r'$ and $s' \notin r \cup r'$. We need to show that:

$$|u \cap (in \cup in' \setminus F)| = 0 \quad (*) \qquad \text{and} \qquad |u \cap (out \cup out' \setminus F)| = 1 \quad (**).$$

We have $s' \notin r$ and $s' \notin r'$. By $r \cap r' = \varnothing$, we can assume without loss of generality that $s \in r$ and $s \notin r'$. Because $\mathfrak{r}$ and $\mathfrak{r}'$ are regions we have (from the definition of a region) that

$$|u \cap in| = 0 \ \text{ and } \ |u \cap out| = 1 \ \ \textbf{R1} \ \text{ and } \ u \cap out' = \varnothing \ \ \textbf{R3} \ \text{ and } \ u \cap in' = \varnothing \ \ \textbf{R4} \ .$$

Hence, by $|u \cap in| = 0$ and $u \cap in' = \varnothing$, we have that (*) is satisfied. To prove (**) we can observe that from

$$|u \cap out| = 1 \ \text{ and } \ u \cap out' = \varnothing$$

we have $|u \cap (out \cup out')| = 1$, and we know that there exists $e \in u \cap out$. From $out \subseteq {}^{\blacktriangledown}r' \cup {}^{\triangledown}r'$ and from the fact that we have a transition $s \xrightarrow{u} s'$ in $\mathfrak{ts}$ with $s' \notin r$, it follows that all transitions with steps containing e must not end in r'. Moreover, they must start in r (follows from $e \in u \cap out$ and **R3** for $\mathfrak{r}$). Hence, $e \notin F$ and, consequently, $|u \cap (out \cup out' \setminus F)| = 1$.

To prove **R3**, let us assume that $u \cap (out \cup out' \setminus F) \neq \varnothing$. We need to show that $s \in r \cup r'$ and $s' \notin r \cup r'$. There exists $e \in u$ such that $e \notin F$ and $e \in out \cup out'$. We can assume, without loss of generality, that $e \in u \cap out$ and $e \notin F$. From the definition of region for $\mathfrak{r}$ we have that $s \in r$ and $s' \notin r$. Hence $s \in r \cup r'$. We still need to show that $s' \notin r'$. Suppose $s' \in r'$. Then we have $s \xrightarrow{u} s'$ where $s \in r$, $s' \in r'$ and $e \in u$. But $e \in out \subseteq {}^{\blacktriangledown}r' \cup {}^{\triangledown}r'$, and so $e \in {}^{\blacktriangledown}r'$. Moreover, from $e \in u \cap out$ and **R3** for $\mathfrak{r}$ we know that all transitions labelled with u must start in r. As $e \in u$, we have $e \in r^{\blacktriangledown}$. Hence $e \in F$, a contradiction. As a result, $s' \notin r'$ and, consequently, $s' \notin r \cup r'$. □

Note that in the above proposition we cannot define the set F as

$$(out \cap in') \cup (in \cap out') \ .$$

Consider the example in Figure 2 and the regions $\mathfrak{r}_1$ and $\mathfrak{r}_4$. We can see that with such a definition of F, we would have $F = \varnothing$ and so

$$\mathfrak{r}_1 \oplus \mathfrak{r}_4 = (\{f\}, \{s_{in}, s\}, \{e\})$$

which is not a region. On the other hand, with the definition of F as in the formulation of Proposition 3.4, we have $F = \{e, f\}$ and so

$$\mathfrak{r}_1 \oplus \mathfrak{r}_4 = (\varnothing, \{s_{in}, s\}, \varnothing) = (\varnothing, S, \varnothing) \ ,$$

which is a valid (trivial) region.

The composition of regions by means of $\oplus$ gives rise in a natural way to a pre-order $\prec$ on the set of regions, i.e., $\mathfrak{r} \prec \mathfrak{r}'$ whenever there is $\mathfrak{r}''$ such that $\mathfrak{r} \oplus \mathfrak{r}'' = \mathfrak{r}'$. And, similarly as in the treatment of regions of ordinary EN-systems, one can aim at constructing a solution to the synthesis problem based on *minimal* regions w.r.t. the pre-order $\prec$. That such an approach is sound follows from the next result.

Proposition 3.5. If $\mathfrak{r} = (in, r, out)$, $\mathfrak{r}' = (in', r', out')$ and $\mathfrak{r} \oplus \mathfrak{r}'$ are three conditions in $\mathfrak{enl}$, then $\mathfrak{r} \oplus \mathfrak{r}'$ is a sound reduction.

Proof:
Suppose that there is a case c such that u is resource enabled at $c \setminus \{\mathfrak{r} \oplus \mathfrak{r}'\}$ in $\mathfrak{enl}_{\mathfrak{r} \oplus \mathfrak{r}'}$ and u is not resource enabled at c in $\mathfrak{enl}$. That means $\mathfrak{r} \oplus \mathfrak{r}'$ is not marked at c and is a pre-condition of some event in u, or $\mathfrak{r} \oplus \mathfrak{r}'$ is marked at c and is a post-condition of some event in u. Let us consider the first case (the second one can be dealt with similarly). As $\mathfrak{enl}$ is a sub-ENL-system of $\mathfrak{enl}_{\mathfrak{ts}}^{\frown}$ obtained from the procedure described in Section 2.3, we know that if $\mathfrak{r} \oplus \mathfrak{r}'$ is not marked at c then neither $\mathfrak{r}$ nor $\mathfrak{r}'$ are marked at c. Moreover, if $\mathfrak{r} \oplus \mathfrak{r}'$ is a pre-condition of some event e in u ($\mathfrak{r} \oplus \mathfrak{r}' \in {}^{\circ}e$) then

$$e \in out_{\mathfrak{r} \oplus \mathfrak{r}'} = out \cup out' \setminus ({}^{\blacktriangledown}r \cap r'^{\blacktriangledown} \cup {}^{\blacktriangledown}r' \cap r^{\blacktriangledown}) \ .$$

That means $\mathfrak{r} \in {}^{\circ}e$ or $\mathfrak{r}' \in {}^{\circ}e$, and $e \in u$. So, one of the conditions $\mathfrak{r}$ or $\mathfrak{r}'$ is a pre-condition of u and they are not marked at $c \setminus \{\mathfrak{r} \oplus \mathfrak{r}'\}$ in $\mathfrak{enl}_{\mathfrak{r} \oplus \mathfrak{r}'}$. Consequently u is not resource enabled at that case in $\mathfrak{enl}_{\mathfrak{r} \oplus \mathfrak{r}'}$, producing a contradiction. □

Figure 3 shows an example of synthesis using only minimal regions. Note that the non-trivial regions in this case are:

$$\begin{aligned}
\mathfrak{r}_1 &= (\varnothing, \{s_{in}\}, \{e, f\}) \\
\mathfrak{r}_2 &= (\{e\}, \{s_1\}, \varnothing) \\
\mathfrak{r}_3 &= (\{f\}, \{s_2\}, \varnothing) \\
\mathfrak{r}_4 &= (\varnothing, \{s_{in}, s_2\}, \{e\}) \\
\mathfrak{r}_5 &= (\varnothing, \{s_{in}, s_1\}, \{f\}) \\
\mathfrak{r}_6 &= (\{e, f\}, \{s_1, s_2\}, \varnothing) \ .
\end{aligned}$$

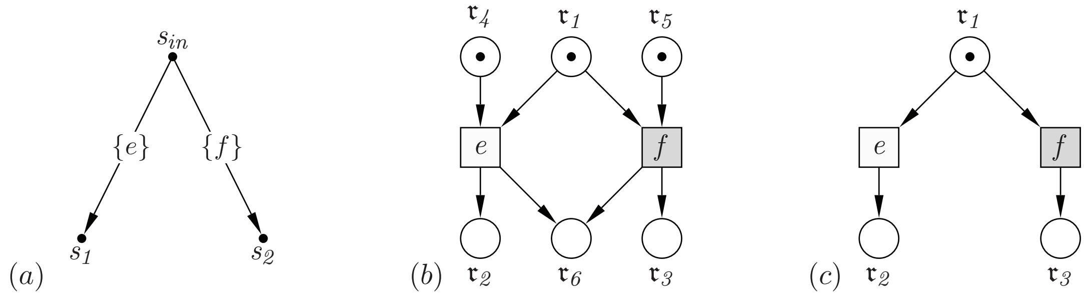

Figure 3. An ENL-transition system with non co-located events e and f (a), the ENL-system resulting from the synthesis (b), and the reduced ENL-system solution for (a) that uses only regions minimal w.r.t. $\prec$.

3.3. Companion regions

The two previous kinds of sound reductions concerned regions based on disjoint sets of states of the original transition system. In fact, they generalised similar notions previously developed for the synthesis of EN-systems. However, the same cannot be said about the third kind of sound reduction as it refers to regions based on *the same* set of states. The key idea here is that if we have a set of regions based on the same set of states — or a set of *companion regions* — then it does not really matter which region is entered or exited by a given event.

Consider the ENL-transition system $\mathfrak{ts}$ in Figure 4(a) and the ENL-system $\mathfrak{enl}$ in Figure 4(b) generating it. The latter is a sub-ENL-system of $\mathfrak{enl}_{\mathfrak{ts}}^{E\times E}$ obtained by deleting conditions complementary to those retained in $\mathfrak{enl}$ (see Proposition 3.3). Consider now the four conditions in the middle row of Figure 4(b). Clearly, they are nothing but four companion regions,

$$\begin{aligned}
\mathfrak{r}_1 &= (\{e\}, \{s\}, \{g\}) \\
\mathfrak{r}_2 &= (\{f\}, \{s\}, \{g\}) \\
\mathfrak{r}_3 &= (\{e\}, \{s\}, \{h\}) \\
\mathfrak{r}_4 &= (\{f\}, \{s\}, \{h\}) \ ,
\end{aligned}$$

belonging to $\mathfrak{R}^{\{s\}}$. It is not difficult to see that not all four are actually needed to construct an ENL-system generating $\mathfrak{ts}$. In fact, we can retain just $\mathfrak{r}_1$ and $\mathfrak{r}_4$, or just $\mathfrak{r}_2$ and $\mathfrak{r}_3$, shown in Figure 4(c,d), without causing any harm.

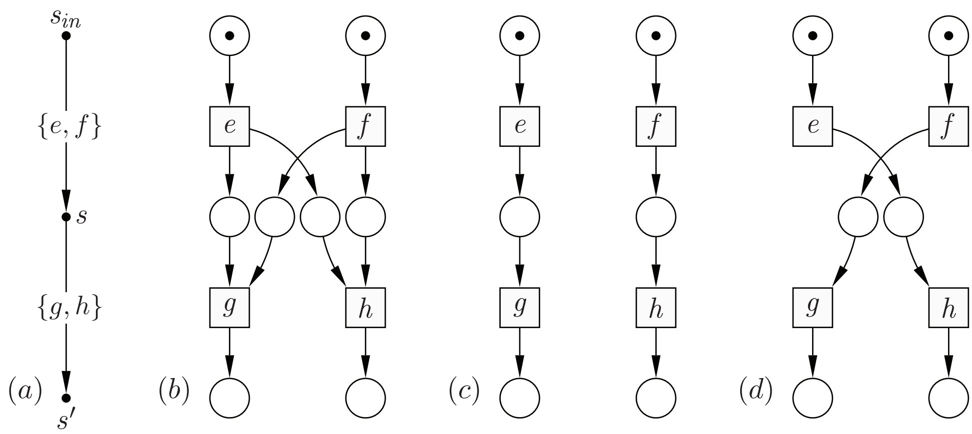

Figure 4. An ENL-transition system (a), and three ENL-systems generating it (b, c, d).

The above observation lies behind our third sound reduction which essentially states that a region is redundant if for every event which enters or exits it, there is another companion region with the same property.

Proposition 3.6. Let $\mathfrak{r} = (in, r, out)$ be a condition in $\mathfrak{enl}$ such that

- $in \subseteq \bigcup\{in' \mid (in', r, out')$ is condition in $\mathfrak{enl}$ different from $\mathfrak{r}\}$.
- $out \subseteq \bigcup\{out' \mid (in', r, out')$ is condition in $\mathfrak{enl}$ different from $\mathfrak{r}\}$.

Then $\mathfrak{r}$ is a sound reduction.

Proof:
Similar to the proofs of Propositions 3.3 and 3.5. □

4. Concluding remarks

In this paper, we discussed how one could synthesise GALS systems represented by Petri nets from their behavioural specifications given in terms of step transition systems. In particular, we investigated how this problem might be solved without considering all potential regions generated by a step transition system. In doing so, we proposed three sound reductions aimed at avoiding redundancy in terms of constructed conditions during the synthesis procedure.

In our future work we plan to consider more relaxed versions of Synthesis Problem 1. For example, one can assume that $\mathfrak{ts}$ gives an upper bound on the desirable behaviour of the synthesised net, and the goal is to retain as much as possible of the behaviour specified by $\mathfrak{ts}$ in the constructed ENL/LC-system.

Acknowledgement. This research was partially supported by the RAE&EPSRC DAVAC project, and NSFC Grants 60910004 and 2010CB328102.

References

[1] Arnold, A.: Finite Transition Systems. Prentice Hall International (1994)

[2] Badouel, E., Bernardinello, L., Darondeau, Ph.: The Synthesis Problem for Elementary Net Systems is NP-complete. Theoretical Computer Science **186** (1997) 107–134

[3] Badouel, E., Darondeau, Ph.: Theory of Regions. In: Reisig, W., Rozenberg, G. (eds.): Lectures on Petri Nets I: Basic Models, Advances in Petri Nets. Lecture Notes in Computer Science **1491**. Springer-Verlag, Berlin Heidelberg New York (1998) 529–586

[4] Bernardinello, L.: Synthesis of Net Systems. In: Marsan, M.A. (ed.): Application and Theory of Petri Nets 1993. Lecture Notes in Computer Science **691**. Springer-Verlag, Berlin Heidelberg New York (1993) 89–105

[5] Darondeau, P., Koutny, M., Pietkiewicz-Koutny, M., Yakovlev, A.: Synthesis of Nets with Step Firing Policies. In: van Hee, K.M., Valk, R. (eds.): PETRI NETS 2008. Lecture Notes in Computer Science **5062**. Springer-Verlag, Berlin Heidelberg New York (2008) 112–131

[6] Dasgupta, S., Potop-Butucaru, D., Caillaud, B., Yakovlev, A.: Moving from Weakly Endochronous Systems to Delay-Insensitive Circuits. Electronic Notes in Theoretical Computer Science **146** (2006) 81–103

[7] Desel, J., Reisig, W.: The Synthesis Problem of Petri Nets. Acta Informatica **33** (1996) 297–315

[8] Kleijn, H.C.M., Koutny, M., Rozenberg, G.: Towards a Petri Net Semantics for Membrane Systems. In: Freund, R., Paun, G., Rozenberg, G., Salomaa, A. (eds.): WMC 2005. Lecture Notes in Computer Science **3850**. Springer-Verlag, Berlin Heidelberg New York (2006) 292–309

[9] Koutny, M., Pietkiewicz-Koutny, M.: Transition Systems of Elementary Net Systems with Localities. In: Baier, C., Hermanns, H. (eds.): CONCUR 2006. Lecture Notes in Computer Science **4137**. Springer-Verlag, Berlin Heidelberg New York (2006) 173–187

[10] Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Elementary Net Systems with Context Arcs and Localities. Fundamenta Informaticae **88** (2008) 307–328

[11] Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Petri Nets with Localities. Scientific Annals of Computer Science **19** (2009) 1–23

[12] Mukund, M.: Petri Nets and Step Transition Systems. International Journal of Foundations of Computer Science **3** (1992) 443–478

[13] Nielsen, M., Rozenberg, G., Thiagarajan, P.S.: Elementary Transition Systems. Theoretical Computer Science **96** (1992) 3–33

[14] Păun, G.: Membrane Computing, An Introduction. Springer-Verlag, Berlin Heidelberg New York (2002)

[15] Pietkiewicz-Koutny, M.: The Synthesis Problem for Elementary Net Systems with Inhibitor Arcs. Fundamenta Informaticae **40** (1999) 251–283