Minimising the Synthesised ENL-systems

Aishah Ahmed[1], Marta Pietkiewicz-Koutny[1,*]

[1]School of Computing, Newcastle University, Newcastle upon Tyne NE4 5TG, United Kingdom

Abstract

Elementary Net Systems with Localities (ENL-systems) is a class of Petri nets introduced to model GALS (globally asynchronous locally synchronous) systems, where some of the components might be considered as logically or physically close and acting synchronously, while others might be considered as loosely connected or residing at distant locations and communicating with the rest of the system in an asynchronous way. The specification of the behaviour of a GALS system comes very often in the form of a transition system. The automated synthesis, based on regions, is an approach that allows to construct Petri net models from their transition system specifications. In our previous papers we developed algorithms and tool support for the synthesis of ENL-systems from step transition systems, where arcs are labelled by steps (sets) of executed actions. In this paper we focus on the minimisation of the synthesised nets. In particular, we discuss the properties of minimal, companion, and complementary regions, and their role in the process of minimisation of ENL-systems. Furthermore, we propose a strategy to eliminate redundant regions. Our theoretical results are backed by experiments (the algorithms for the minimisation are implemented within the WORKCRAFT framework).

Keywords

theory of concurrency, Petri nets, localities, analysis and synthesis, step sequence semantics, theory of regions, transition systems, WORKCRAFT framework

1. Introduction

A number of computational systems exhibit behaviour that follows 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 [1], and in biologically inspired membrane systems representing cells within which biochemical reactions happen in synchronised pulses [2]. To formalise such systems, [3] introduced *Place/Transition-nets with localities* (PTL-nets), where each locality defines a distinct set of events which must be executed synchronously, i.e., in a maximally concurrent manner (often called *local maximal concurrency*).

An attractive way of constructing complex computing systems is their automated synthesis from behavioural specifications given in terms of suitable transition systems. In such a case, the synthesis procedure is often based on the regions of a transition system, a notion introduced in [4], and later used to solve the synthesis problem for different classes of Petri nets [5, 6, 7, 8, 9, 10]. A comprehensive survey of the synthesis problem and region theory is presented in [11].

ATAED'22: Algorithms & Theories for the Analysis of Event Data, June 20–21, 2022, Bergen, Norway

[*] Corresponding author.

a.ahmad@ncl.ac.uk (A. Ahmed); marta.koutny@ncl.ac.uk (M. Pietkiewicz-Koutny)

The vast majority of results in the area of synthesis of Petri nets use the standard transition systems, where the arcs are labelled with single events/actions, as initial specifications of systems' behaviour. In this paper, however, we follow the approach, used in [12, 13, 14, 15, 10], employing step transition systems instead, where arcs are labelled with sets of executed events/actions.

The nets with localities, as already mentioned, were first introduced in [3] using as a base a class of Place/Transition nets. The idea of actions' localities was later adapted to Elementary Net Systems (EN-systems) in [12], where a solution to the synthesis problem for ENL-systems was presented. Further advances in the area of synthesising nets with localities from step transition systems are the subjects of [13, 14, 15]. The last of them, [15], concentrated on finding the rules for reducing the number of regions that are essential to synthesise ENL-systems.

In our previous papers, [16, 17], we developed algorithms and tool support for the synthesis of ENL-systems from step transition systems. In this paper we continue the work started in [15] and focus on the minimisation of ENL-systems. The nets obtained from the synthesis procedure, called saturated nets, contain many conditions that are redundant from their behaviour point of view. Removing such conditions is important to get more manageable and readable solutions to the synthesis problem. The approaches to remove redundant conditions from nets were investigated in the literature and implemented in several tools [18, 19]. Many synthesis procedures concentrate on returning smaller solutions based on so-called minimal regions [20, 21, 22, 23, 24]. In our approach, minimal regions, as defined for our class of step transition systems, are also important for building smaller solutions to the synthesis problem. Furthermore, in this paper, following [6], we are interested in the role of minimal regions in defining state-machine components of the synthesised and minimised ENL-systems. Our theoretical results are backed by experiments (the algorithms for the minimisation are implemented within the WORKCRAFT framework [25, 26]).

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 $\{p_2\}$ which removes the token from b_1 and puts two tokens, into b_0 and b_2. In this new state, there are three enabled steps, viz. $\{p_1\}$, $\{c_1, c_4\}$ and $\{p_1, c_1, c_4\}$. The last one, $\{p_1, c_1, c_4\}$, 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.

The paper is organised as follows. The next section recalls some basic notions concerning step transition systems as well as ENL-systems and their synthesis. Section 3 recalls, from [15], three reduction rules that can be used to safely eliminate redundant regions/conditions from the synthesised nets. Section 4 defines minimal regions for the class of ENLST-systems and Section 5 discusses the properties of different kinds of regions and their roles in the process of minimisation of ENL-systems. Section 6 presents a strategy to eliminate redundant regions from the synthesised nets. The paper ends with a conclusion that includes some directions for future work. Proofs are omitted due to the page limit.

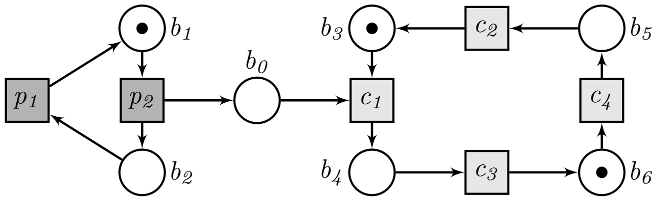

Figure 1: An ENL-system modelling a system comprising one producer and two co-located consumers (the shading of boxes indicates the co-location of events they represent).

2. Preliminaries

In this section, we recall suitably adapted notions and results from [12, 14, 15].

Throughout the paper, E is a fixed finite nonempty set of *events*. A *step* is a nonempty set of events, and a *co-location relation* $\bumpeq$ is any equivalence relation over E. For every event $e \in E$, $[e]_{\bumpeq}$ is the equivalence class of $\bumpeq$ to which e belongs (i.e., the *locality* of e). For an event e and a step U, we denote $e \bumpeq U$ whenever there is at least one event $f \in U$ satisfying $e \bumpeq f$.

Definition 1. *A* step transition system *(or* ST*-system) is a triple* $\mathfrak{ts} = (Q, A, q_0)$*, where* Q *is a nonempty finite set of* states*,* $A \subseteq Q \times (\mathbb{P}(E) \setminus \{\varnothing\}) \times Q$ *is a set of* transitions (arcs)*, and* $q_0 \in Q$ *is the* initial state*.* ♢

In diagrams, ST-systems are represented as labelled directed graphs, and singleton steps annotating transitions are denoted without brackets.

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

The set of all steps labelling transitions outgoing from q will be denoted by $allSteps_q$. For a transition $\mathfrak{t} = (q, U, q') \in A$, we have therefore $U \in allSteps_q$, and respectively call q and q' the *source* and *target* of $\mathfrak{t}$. Furthermore, $\mathfrak{t}$ is *thick* if $|U| \geq 2$.

$\mathfrak{ts}$ is called *thin* if, for every event $e \in E$, there is $(q, \{e\}, q') \in A$.

A sequence of transitions $(q_1, U_1, q_2)(q_2, U_2, q_3) \ldots (q_k, U_k, q_{k+1})$ is a *path* from q_1 to q_{k+1}. A state q is *reachable* if there is a path from q_0 to q.

Two ST-systems, $\mathfrak{ts} = (Q, A, q_0)$ and $\mathfrak{ts}' = (Q', A', q_0')$, are *isomorphic* if there is a bijection $f : Q \to Q'$ such that $f(q_0) = q_0'$ and $A' = \{(f(q), U, f(q')) \mid (q, U, q') \in A\}$. We denote this by $\mathfrak{ts} \cong \mathfrak{ts}'$.

Definition 2. *An* elementary net system with localities w.r.t. a co-location relation $\bumpeq$ *(or* ENL$_{\bumpeq}$*-system) is a tuple* $\mathfrak{enl} = (B, E, F, \bumpeq, c_0)$*, where* B *is a finite set of* conditions *such that* $B \cap E = \varnothing$*,* $F \subseteq (B \times E) \cup (E \times B)$ *is the* flow relation*, and* $c_0 \subseteq B$ *is the* initial case *(in general, any* $c \subseteq B$ *is a* case*).*
We will also say that $\mathfrak{enl}$ *is an elementary net system with localities (or* ENL*-system) if mentioning* $\bumpeq$ *is not important.* ♢

In diagrams, conditions (local states) are represented by circles, events (actions) by boxes, the flow relation by directed arcs, and each case (global state) by tokens (small black dots) placed inside those conditions which belong to this case. Moreover, boxes representing co-located events are shaded in the same way (see Figure 1).

For every event e, its *pre-conditions* and *post-conditions* are given respectively by ${}^\bullet e = \{b \mid (b, e) \in F\}$ and $e^\bullet = \{b \mid (e, b) \in F\}$, and both sets are assumed to be nonempty and disjoint. The dot-notation extends to sets of events in the usual way, e.g., ${}^\bullet U = \bigcup\{{}^\bullet e \mid e \in U\}$. Two distinct events, e and f, are *in conflict* (or conflicting) if they share a pre-condition, or share a post-condition. Furthermore, we assume that there are no isolated conditions in $\mathfrak{enl}$.

The semantics of $\mathfrak{enl}$ is based on steps of simultaneously executed events, and can be understood as *local maximal concurrency*. We first define *potential steps of* $\mathfrak{enl}$ as all nonempty sets of mutually 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 $e \bumpeq U$ and the step $U \cup \{e\}$ is resource enabled at c. We denote these respectively by $U \in \mathit{resenabled}(c)$ and $U \in \mathit{enabled}(c)$. A control enabled step $U \in \mathit{enabled}(c)$ can be *executed* leading from c to the case $c' = (c \setminus {}^\bullet U) \cup U^\bullet$. We denote this by $c[U\rangle c'$.

The set of *reachable* cases of $\mathfrak{enl}$, denoted *reach*$_{\mathfrak{enl}}$, is the least set of cases containing c_0 such that if $c \in$ *reach*$_{\mathfrak{enl}}$ and $c[U\rangle c'$, then $c' \in$ *reach*$_{\mathfrak{enl}}$.

The ST-system *generated* by $\mathfrak{enl}$ is $\mathfrak{ts}_{\mathfrak{enl}} = (\mathit{reach}_{\mathfrak{enl}}, A, c_0)$, where $A = \{(c, U, c') \mid c \in \mathit{reach}_{\mathfrak{enl}} \wedge c[U\rangle c'\}$.

$\mathfrak{enl}$ is a *net realisation* of an ST-system $\mathfrak{ts}$ if $\mathfrak{ts}_{\mathfrak{enl}} \cong \mathfrak{ts}$.

To ease the presentation, we assume that $\mathfrak{enl}$ does not have *dead events*, i.e., for each event e, there are $c \in$ *reach*$_{\mathfrak{enl}}$ and $U \in \mathit{enabled}(c)$ such that $e \in U$.

Definition 3. *Let* $\mathfrak{enl} = (B, E, F, \bumpeq, c_0)$ *be an* ENL$_{\bumpeq}$*-system. We say that* $\mathfrak{enl}$ *is a* state machine ENL$_{\bumpeq}$*-system iff:*

1. $\forall e \in E : \ |{}^\bullet e| = 1 = |e^\bullet|$;

2. $|c_0| = 1$. ♢

Definition 4. *Let* $\mathfrak{enl} = (B, E, F, \bumpeq, c_0)$ *be an* ENL$_{\bumpeq}$*-system. A* subsystem *of* $\mathfrak{enl}$ *is an* ENL$_{\bumpeq}$*-system* $\mathfrak{enl}' = (B', E', F', \bumpeq', c_0')$ *such that the following conditions hold:*

1. $B' \subseteq B$ *and* $E' \subseteq E$;

2. $\forall b \in B' \ \forall e \in E : \ ((b, e) \in F \vee (e, b) \in F) \implies e \in E'$;

3. $F' = F \cap ((B' \times E') \cup (E' \times B'))$; $\bumpeq' = \bumpeq \cap E' \times E'$ *and* $c_0' = c_0 \cap B'$.

A subsystem $\mathfrak{enl}'$ *is* connected *if the graph* $(B' \cup E', F')$ *is connected. Also, point 2 above says that the subsystem* $\mathfrak{enl}'$ *is* generated by the subset of conditions B' *of the* ENL$_{\bumpeq}$*-system* $\mathfrak{enl}$. ♢

Definition 5. *Let* $\mathfrak{enl} = (B, E, F, \bumpeq, c_0)$ *be an* ENL$_{\bumpeq}$*-system. We say that* $\mathfrak{enl}$ *is a* state machine decomposable ENL$_{\bumpeq}$*-system iff there exists a set of connected subsystems of* $\mathfrak{enl}$, $\mathfrak{enl}_i = (B_i, E_i, F_i, \bumpeq_i, c_0^i)$ *($i = 1, \ldots, m$), satisfying the following:*

1. $\forall i \in \{1, \ldots, m\}$: $\mathfrak{enl}_i$ *is a state machine* ENL$_{\bumpeq_i}$*-system;*

2. $B = \bigcup_i B_i,\ E = \bigcup_i E_i$ *and* $F = \bigcup_i F_i$.

The $\mathfrak{enl}_i$ *are called* state machine (or sequential) components *of* $\mathfrak{enl}$. ♢

The general synthesis problem we consider can be formulated thus:

Problem 1. *Given an* ST*-system* $\mathfrak{ts}$ *and a co-location relation* $\simeq$*, find an effective way of checking whether there is an* ENL$_\simeq$*-system which is a net realisation of* $\mathfrak{ts}$*. If the answer is positive construct such an* ENL$_\simeq$*-system.* ♢

The above problem can be approached by considering a link between the nodes (global states) of an ST-system with the conditions (local states) of a hypothetical ENL-system realising it, captured by the notion of regions with explicit input and output events.

Definition 6. *A* region (with explicit input and output events) *of an* ST*-system* $\mathfrak{ts} = (Q, A, q_0)$ *is a triple* $\mathfrak{r} = (\mathit{in}, r, \mathit{out}) \in \mathbb{P}(E) \times \mathbb{P}(Q) \times \mathbb{P}(E)$*, such that* $\mathit{in} = \mathit{out} = \varnothing$ *implies* $r = Q$ *or* $r = \varnothing$ *and, for every transition* (q, U, q') *of* $\mathfrak{ts}$*, the following hold:*

R1 *If* $q \in r$ *and* $q' \notin r$ *then* $|U \cap \mathit{out}| = 1$.

R2 *If* $q \notin r$ *and* $q' \in r$ *then* $|U \cap \mathit{in}| = 1$.

R3 *If* $U \cap \mathit{out} \neq \varnothing$ *then* $q \in r$ *and* $q' \notin r$.

R4 *If* $U \cap \mathit{in} \neq \varnothing$ *then* $q \notin r$ *and* $q' \in r$. ♢

In a region $\mathfrak{r} = (\mathit{in}, r, \mathit{out})$, the set *in* comprises events responsible for entering the set of states r, and *out* comprises events responsible for leaving r. Note that $\bar{\mathfrak{r}} = (out, Q \setminus r, in)$ is also a region (the *complement* of region $\mathfrak{r}$).

In general, a region $\mathfrak{r}$ cannot be identified only by its set of states r. However, if $\mathfrak{ts}$ is *thin*, then its different regions are based on different sets of states.

There are exactly two *trivial* regions satisfying $r = \varnothing$ or $r = Q$, viz. $(\varnothing, \varnothing, \varnothing)$ and $(\varnothing, Q, \varnothing)$.

The set of all non-trivial regions of $\mathfrak{ts}$ will be denoted by $\mathfrak{R}_{\mathfrak{ts}}$ and, for every state q, $\mathfrak{R}_q = \{\mathfrak{r} \in \mathfrak{R}_{\mathfrak{ts}} \mid q \in r\}$ is the set of all non-trivial regions $(\mathit{in}, r, \mathit{out})$ containing q.

The sets of *pre-regions* and *post-regions* of an event e, ${}^\circ e$ and e°, comprise all the non-trivial regions $(\mathit{in}, r, \mathit{out})$ respectively satisfying $e \in \mathit{out}$ and $e \in \mathit{in}$, viz. ${}^\circ e = \{\mathfrak{r} \in \mathfrak{R}_{\mathfrak{ts}} \mid e \in \mathit{out}\}$ and $e^\circ = \{\mathfrak{r} \in \mathfrak{R}_{\mathfrak{ts}} \mid e \in \mathit{in}\}$. This extends in the usual way to sets of events, for example, ${}^\circ U = \bigcup\{{}^\circ e \mid e \in U\}$. Also, we will write $e \in \mathfrak{r}^\circ \iff \mathfrak{r} \in {}^\circ e$ and $e \in {}^\circ\mathfrak{r} \iff \mathfrak{r} \in e^\circ$.

The set of *potential steps* of $\mathfrak{ts}$ comprises all nonempty 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 $q \in Q$ if ${}^\circ U \subseteq \mathfrak{R}_q$ and $U^\circ \cap \mathfrak{R}_q = \varnothing$. We denote this by $U \in \mathit{regenabled}(q)$.

Definition 7. *An* ST*-system* $\mathfrak{ts}$ *is an* ENL step transition system *w.r.t. a co-location relation* $\simeq$ *(or* ENLST$_\simeq$*-system) if the following hold:*

A1 *Each state is reachable.*

A2 *For every event e, both $^\circ e$ and e° are nonempty.*

A3 *For all distinct states q and q', $\mathfrak{R}_q \neq \mathfrak{R}_{q'}$.*

A4 *For every state q and step U, $U \in allSteps_q$* iff *$U \in regenabled(q)$ and there is no event $e \notin U$ such that $e \bumpeq U$ and $U \cup \{e\} \in regenabled(q)$.*

We also say that $\mathfrak{ts}$ is an ENLST*-system (if mentioning $\bumpeq$ is not important).* ♢

One can show that the ST-system generated by an ENL$_{\bumpeq}$-system is an ENLST$_{\bumpeq}$-system. The converse is also true, and a suitable translation is based on the regions of ST-systems.

Definition 8. *Let $\mathfrak{ts}$ be an* ENLST$_{\bumpeq}$*-system. The tuple* associated *with $\mathfrak{ts}$ is defined by $\mathfrak{enl}^{\bumpeq}_{\mathfrak{ts}} = (\mathfrak{R}_{\mathfrak{ts}}, E, F_{\mathfrak{ts}}, \bumpeq, \mathfrak{R}_{q_0})$, where q_0 is the initial state of $\mathfrak{ts}$ and*

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

♢

The above construction always produces an ENL$_{\bumpeq}$-system which generates an ST-system isomorphic to $\mathfrak{ts}$ (see [12]).

Theorem 1. *Let $\mathfrak{ts}$ be an* ENLST$_{\bumpeq}$*-system. Then $\mathfrak{enl}^{\bumpeq}_{\mathfrak{ts}}$ is an* ENL$_{\bumpeq}$*-system such that $\mathfrak{ts} \cong \mathfrak{ts}_{\mathfrak{enl}^{\bumpeq}_{\mathfrak{ts}}}$. Moreover, the unique isomorphism ψ between $\mathfrak{ts}$ and $\mathfrak{ts}_{\mathfrak{enl}^{\bumpeq}_{\mathfrak{ts}}}$ is given by $\psi(q) = \mathfrak{R}_q$, for every state q of $\mathfrak{ts}$.*

3. Optimising solutions to the synthesis problem

In this section we recall some notions and results from [15].

The ENL-system $\mathfrak{enl}^{\bumpeq}_{\mathfrak{ts}}$ obtained from the synthesis of the ENLST-system $\mathfrak{ts}$ may contain many conditions which are *redundant* from the point of view of its behaviour, $i.e.$, deletion of such conditions (and their adjacent arcs) would lead to a net that generates the ST-system, which is still isomorphic to $\mathfrak{ts}$.

Suppose that we have reduced $\mathfrak{enl}^{\bumpeq}_{\mathfrak{ts}}$ in this way obtaining a sub-ENL-system $\mathfrak{enl}$. We would like to reduce $\mathfrak{enl}$ further by deleting a condition/region $\mathfrak{r}$ (and its adjacent arcs) without, as before, violating the property of it being an ENL-system [1] and making sure that the resultant net still generates the ST-system isomorphic to $\mathfrak{ts}$. We denote the ENL-system after such one step reduction: $\mathfrak{enl}_{\mathfrak{r}}$.

In [15], it was proved that complement regions are very often redundant:

Reduction Rule 1. *If $\mathfrak{r} = (in, r, out)$ and $\bar{\mathfrak{r}} = (out, Q \setminus r, in)$ are two conditions in $\mathfrak{enl}$ and deleting $\bar{\mathfrak{r}}$ leads to an* ENL*-system, then the* ST*-systems generated by $\mathfrak{enl}$ and $\mathfrak{enl}_{\bar{\mathfrak{r}}}$ are isomorphic and $\bar{\mathfrak{r}}$ is redundant.* ♢

[1] Every ENL-system $\mathfrak{enl} = (B, E, F, \bumpeq, c_0)$ should satisfy: $\forall e \in E\ ({}^\bullet e \neq \varnothing\ \wedge\ e^\bullet \neq \varnothing\ \wedge\ {}^\bullet e \cap e^\bullet = \varnothing)$.

Another source of redundancy among conditions/regions in the synthesised net are “big" regions that are compositions of smaller regions. To define a composition operator, [15] introduces the concept of *compatible* regions.

Definition 9. *A region (in, r, out) is compatible with another region (in', r', out') iff the following three conditions hold:*

1. $r \cap r' = \varnothing$.
2. *For every $e \in out$ exactly one of the following holds:*
   - *For all the transitions (q, U, q') such that $e \in U$ we have $q' \in r'$.*
   - *For all the transitions (q, U, q') such that $e \in U$ we have $q' \notin r'$.*
3. *For every $e \in in$ exactly one of the following holds:*
   - *For all the transitions (q, U, q') such that $e \in U$ we have $q \in r'$.*
   - *For all the transitions (q, U, q') such that $e \in U$ we have $q \notin r'$.*

If region $\mathfrak{r}$ is compatible with region $\mathfrak{r}'$ and region $\mathfrak{r}'$ is compatible with $\mathfrak{r}$ we say that the two regions are compatible. ♢

In [15], it was proved that the *composition of two compatible regions* (defined below) is also a region. If $\mathfrak{r} = (in, r, out)$ and $\mathfrak{r}' = (in', r', out')$ are two non-trivial compatible regions of an ENLST-system $\mathfrak{ts}$, then the following is a (possibly trivial) region of $\mathfrak{ts}$:

$$\mathfrak{r} \oplus \mathfrak{r}' \overset{\text{df}}{=} (in \cup in' \setminus H, r \cup r', out \cup out' \setminus H),$$

where H is a set of events that belong only to steps labelling transitions hidden/buried in $r \cup r'$ (with its source in r and its target in r' or the other way round). The region $\mathfrak{r} \oplus \mathfrak{r}'$ is called the composition of $\mathfrak{r}$ and $\mathfrak{r}'$. In [15], the following reduction rule was proved:

Reduction Rule 2. *If $\mathfrak{r} = (in, r, out)$, $\mathfrak{r}' = (in', r', out')$ and $\mathfrak{r} \oplus \mathfrak{r}'$ are three conditions/regions in $\mathfrak{enl}$, then the ST-systems generated by $\mathfrak{enl}$ and $\mathfrak{enl}_{\mathfrak{r} \oplus \mathfrak{r}'}$ are isomorphic and $\mathfrak{r} \oplus \mathfrak{r}'$ is redundant.* ♢

The third reduction rule considers regions of $\mathfrak{ts} = (Q, A, q_0)$ based on the same set of states. We will call such regions *companion* regions. For a given set of states r, they will belong to the set denoted by $\mathfrak{R}^r_{\mathfrak{ts}}$.

In [15], it was proved that if the events contained in the set in (out) of a region $\mathfrak{r} = (in, r, out)$ can be found in the in (out) sets of other companion regions then $\mathfrak{r}$ is redundant and can be deleted. Formally:

Reduction Rule 3. *Let $\mathfrak{r} = (in, r, out)$ be a condition/region of $\mathfrak{enl}$ such that:*

$$in \subseteq \bigcup\{in' \mid (in', r, out') \textit{ is condition in } \mathfrak{enl} \textit{ different from } \mathfrak{r}\} \quad (1)$$

$$out \subseteq \bigcup\{out' \mid (in', r, out') \textit{ is condition in } \mathfrak{enl} \textit{ different from } \mathfrak{r}\} \quad (2)$$

Then the ST-systems generated by $\mathfrak{enl}$ and $\mathfrak{enl}_{\mathfrak{r}}$ are isomorphic and $\mathfrak{r}$ is redundant. ♢

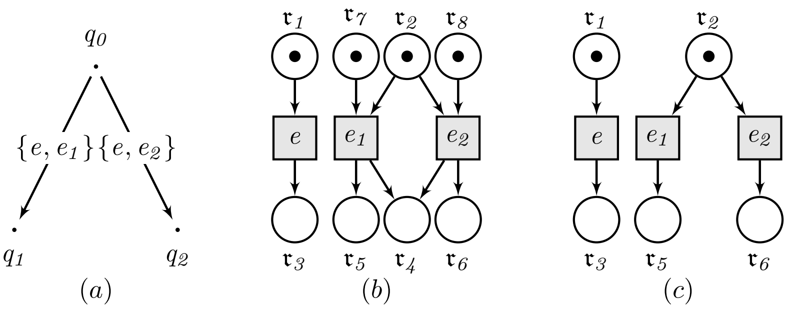
q_0
$\{e, e_1\}\{e, e_2\}$
q_1
q_2
(a)
$\mathfrak{r}_1$ $\mathfrak{r}_7$ $\mathfrak{r}_2$ $\mathfrak{r}_8$
e e_1 e_2
$\mathfrak{r}_3$ $\mathfrak{r}_5$ $\mathfrak{r}_4$ $\mathfrak{r}_6$
(b)
$\mathfrak{r}_1$ $\mathfrak{r}_2$
e e_1 e_2
$\mathfrak{r}_3$ $\mathfrak{r}_5$ $\mathfrak{r}_6$
(c)

Figure 2: An ENLST-system with three co-located events e, e_1 and e_2 (a); the ENL-system resulting from its synthesis (b); and the reduced ENL-system solution for (a) that uses only minimal regions (c).

4. Minimal regions

For many classes of Petri nets, for which the synthesis problem was investigated, a region was defined as a subset of states of a transition system. For such classes of nets and their transition systems a minimal region was defined $w.r.t.$ the set inclusion $\subset$ [6, 7, 23, 27]. Also, composition of regions (as sets) was defined by using the set union operator ($\bigcup$), which is both commutative and associative.

The regions of ENLST-systems are triples of the form: $\mathfrak{r} = (in, r, out)$. The minimal regions in this class of (step) transition systems are defined $w.r.t.$ the strict pre-order $\prec$ on the set of regions, that utilises the idea of regions' composition by means of $\oplus$:

$\mathfrak{r} \prec \mathfrak{r}'$ iff there is a non-trivial region $\mathfrak{r}''$ such that $\mathfrak{r} \oplus \mathfrak{r}'' = \mathfrak{r}'$ [15].

Formally, we have the following definition of a *minimal region*:

Definition 10. *A region* $\mathfrak{r} \in \mathfrak{R}_{\mathfrak{ts}}$ *is* minimal *iff* $\forall\, \widehat{\mathfrak{r}} \in \mathfrak{R}_{\mathfrak{ts}} : \ \widehat{\mathfrak{r}} \not\prec \mathfrak{r}$. ◇

The set of minimal regions of $\mathfrak{ts}$ w.r.t. $\prec$ will be denoted by $\mathfrak{R}^{min}_{\mathfrak{ts}}$.

We observe that if a non-trivial region is *non-minimal* then it can be represented as a composition of two other non-trivial regions. This follows from the definition of the relation $\prec$ and the fact that the composition operator $\oplus$ is commutative, which, in turn, follows immediately from the definition of $\oplus$.

As an example, consider the ENLST-system in Figure 2(a). Its non-trivial regions are:

$$\begin{array}{lllllll}
\mathfrak{r}_1 & = & (\varnothing, \{q_0\}, \{e\}) & \qquad & \mathfrak{r}_3 & = & \bar{\mathfrak{r}}_1 = (\{e\}, \{q_1, q_2\}, \varnothing) \\
\mathfrak{r}_2 & = & (\varnothing, \{q_0\}, \{e_1, e_2\}) & & \mathfrak{r}_4 & = & \bar{\mathfrak{r}}_2 = (\{e_1, e_2\}, \{q_1, q_2\}, \varnothing) \\
\mathfrak{r}_5 & = & (\{e_1\}, \{q_1\}, \varnothing) & & \mathfrak{r}_7 & = & \bar{\mathfrak{r}}_5 = (\varnothing, \{q_0, q_2\}, \{e_1\}) \\
\mathfrak{r}_6 & = & (\{e_2\}, \{q_2\}, \varnothing) & & \mathfrak{r}_8 & = & \bar{\mathfrak{r}}_6 = (\varnothing, \{q_0, q_1\}, \{e_2\})
\end{array}$$

The minimal regions of the ENLST-system in Figure 2(a) are: $\mathfrak{r}_1$, $\mathfrak{r}_2$, $\mathfrak{r}_3$, $\mathfrak{r}_5$ and $\mathfrak{r}_6$. The remaining regions are non-minimal (their set is denoted by $\mathfrak{R}^{\oplus}_{\mathfrak{ts}}$): $\mathfrak{r}_4 = \mathfrak{r}_5 \oplus \mathfrak{r}_6$ $(H = \varnothing)$; $\mathfrak{r}_7 = \mathfrak{r}_2 \oplus \mathfrak{r}_6$ $(H = \{e_2\})$; $\mathfrak{r}_8 = \mathfrak{r}_2 \oplus \mathfrak{r}_5$ $(H = \{e_1\})$.

The reduced ENL-system solution for the ENLST-system in Figure 2(a) that uses only regions minimal $w.r.t.$ $\prec$ is shown in Figure 2(c).

Note that the operator $\oplus$ is not associative as can be shown by using, again, the example of the ENLST-system in Figure 2(a). We can observe that: $(\mathfrak{r}_5 \oplus \mathfrak{r}_6) \oplus \mathfrak{r}_1 \neq \mathfrak{r}_5 \oplus (\mathfrak{r}_6 \oplus \mathfrak{r}_1)$. While $\mathfrak{r}_5$ and $\mathfrak{r}_6$ are compatible and their composition produces $\mathfrak{r}_4$ ($\mathfrak{r}_4 = \mathfrak{r}_5 \oplus \mathfrak{r}_6$), regions $\mathfrak{r}_6$ and $\mathfrak{r}_1$ are not compatible, because $\mathfrak{r}_1$ is not compatible with $\mathfrak{r}_6$, and they cannot be composed.

Also notice that there might be two companion regions (regions based on the same set of states) such that one of them is a minimal region and the second one is a non-minimal region. See, for example, regions $\mathfrak{r}_3 = (\{e\}, \{q_1, q_2\}, \varnothing)$ and $\mathfrak{r}_4 = (\{e_1, e_2\}, \{q_1, q_2\}, \varnothing)$ of the ENLST-system in Figure 2(a), where $\mathfrak{r}_3$ is minimal and $\mathfrak{r}_4$ is non-minimal. So, the minimality of a region cannot be decided by looking at its set of states only.

The following result, about the representation of non-trivial regions, is similar to the results proved for other classes of nets (and their transition systems) that can be found in the literature: Elementary Net Systems [6], pure and bounded Place/Transition Nets [7], Safe Nets [23] or Elementary Net Systems with Inhibitor Arcs (ENI-systems) [27].

Theorem 2. *Every $\mathfrak{r} = (in, r, out) \in \mathfrak{R}_{\mathfrak{ts}}$ can be represented as a composition of minimal regions, where for each pair of different minimal regions in this representation, $\widehat{\mathfrak{r}} = (\widehat{in}, \widehat{r}, \widehat{out})$ and $\widetilde{\mathfrak{r}} = (\widetilde{in}, \widetilde{r}, \widetilde{out})$, $\widehat{r} \cap \widetilde{r} = \varnothing$.*

Theorem 2 and Reduction Rule 2 imply that one can construct a solution to the synthesis problem based on minimal regions $w.r.t.$ the strict pre-order $\prec$. The consequence of the fact that the operator $\oplus$ is not associative, is that we cannot drop the brackets, when we represent a non-trivial region of an ENLST-system as a composition of its minimal regions (for example, $\mathfrak{r} = \mathfrak{r}_1 \oplus (\mathfrak{r}_2 \oplus \ldots (\mathfrak{r}_{n-2} \oplus (\mathfrak{r}_{n-1} \oplus \mathfrak{r}_n)) \ldots)$, where $\mathfrak{r}_i$ ($i = 1, \ldots, n$) are minimal regions in this representation of $\mathfrak{r}$).

5. Properties of regions

In this section we gather facts regarding relationships of complementary, compatible, companion and minimal regions of an ENLST-system $\mathfrak{ts} = (Q, A, q_0)$.

Fact 1. *Any pair of complementary regions of $\mathfrak{ts}$, $\mathfrak{r}$ and $\bar{\mathfrak{r}}$, form a pair of compatible regions and $\mathfrak{r} \oplus \bar{\mathfrak{r}} = (\varnothing, Q, \varnothing)$.* ♢

From Fact 1 it follows that if $\mathfrak{ts}$ has only minimal regions among non-trivial regions, then only the pairs of complementary regions can be composed resulting in a trivial region.

Fact 2. *Let $\mathfrak{r}_1 = (in_1, r_1, out_1)$ and $\mathfrak{r}_2 = (in_2, r_2, out_2)$ be two non-trivial compatible regions of an ENLST-system $\mathfrak{ts}$. Then $in_1 \cap in_2 = \varnothing$ and $out_1 \cap out_2 = \varnothing$.*

Now, we introduce a notion of *strong compatibility* of regions.

Definition 11. *A region (in, r, out) is strongly compatible with another region (in', r', out') iff the following three conditions hold:*

1. $r \cap r' = \varnothing$.

2. *For every $e \in out$ exactly one of the following holds:*
   - $e \in in'$*.*
   - *For all the transitions (q, U, q') such that $e \in U$ we have $q' \notin r'$.*
3. *For every $e \in in$ exactly one of the following holds:*
   - $e \in out'$*.*
   - *For all the transitions (q, U, q') such that $e \in U$ we have $q \notin r'$.*

If region $\mathfrak{r}$ is strongly compatible with region $\mathfrak{r}'$ and region $\mathfrak{r}'$ is strongly compatible with $\mathfrak{r}$ we say that the two regions are strongly compatible. ♢

Fact 3. *Two regions, which are strongly compatible are compatible.*

The composition operator defined for the strongly compatible regions (rather than compatible regions) will be denoted by $\oplus_s$. The strict pre-order relation for the set of regions that utilises operator $\oplus_s$ instead of $\oplus$ will be denoted by $\prec_s$. The set of minimal regions of $\mathfrak{ts}$ w.r.t. $\prec_s$ will be denoted by $\mathfrak{R}_{\mathfrak{ts}}^{min,s}$.

The implication of Fact 3 is that Reduction Rule 2 works with strongly compatible regions (we can replace operator $\oplus$ by $\oplus_s$ in that rule). Also, we can strengthen Fact 1 to:

Fact 4. *Any pair of complementary regions of $\mathfrak{ts}$, $\mathfrak{r}$ and $\bar{\mathfrak{r}}$, form a pair of strongly compatible regions and $\mathfrak{r} \oplus_s \bar{\mathfrak{r}} = (\varnothing, Q, \varnothing)$.* ♢

We will give examples of compatible and strongly compatible pairs of regions using the ENLST-system in Figure 2(a). The pairs of regions $\{\mathfrak{r}_1, \mathfrak{r}_4\}$ and $\{\mathfrak{r}_2, \mathfrak{r}_3\}$ are compatible, but not strongly compatible. However, the pairs $\{\mathfrak{r}_1, \mathfrak{r}_3\}$, $\{\mathfrak{r}_2, \mathfrak{r}_4\}$, $\{\mathfrak{r}_2, \mathfrak{r}_5\}$, $\{\mathfrak{r}_2, \mathfrak{r}_6\}$, $\{\mathfrak{r}_5, \mathfrak{r}_6\}$, $\{\mathfrak{r}_5, \mathfrak{r}_7\}$ and $\{\mathfrak{r}_6, \mathfrak{r}_8\}$ are strongly compatible pairs of regions. Observe also that the pairs $\{\mathfrak{r}_1, \mathfrak{r}_5\}$ and $\{\mathfrak{r}_1, \mathfrak{r}_6\}$ are not compatible, because $\mathfrak{r}_1$ is not compatible with neither $\mathfrak{r}_5$ nor $\mathfrak{r}_6$.

The next proposition shows that unlike $\oplus$ operator, $\oplus_s$ is associative.

Proposition 1. *The operator $\oplus_s$ is associative.*

Proposition 2. *Let $\mathfrak{r}_1 = (in_1, r_1, out_1)$ and $\mathfrak{r}_2 = (in_2, r_2, out_2)$ be compatible regions of an* ST*-system $\mathfrak{ts} = (Q, A, q_0)$, which do not satisfy the conditions to be strongly compatible regions of $\mathfrak{ts}$. Then there exists a companion region of $\mathfrak{r}_1$, $\mathfrak{r}_1' \in \mathfrak{R}_{\mathfrak{ts}}^{r_1}$, and a companion region of $\mathfrak{r}_2$, $\mathfrak{r}_2' \in \mathfrak{R}_{\mathfrak{ts}}^{r_2}$, such that $\mathfrak{r}_1$ and $\mathfrak{r}_2'$ are strongly compatible and $\mathfrak{r}_1'$ and $\mathfrak{r}_2$ are strongly compatible. Furthermore, $\mathfrak{r}_1 \oplus_s \mathfrak{r}_2' = \mathfrak{r}_1' \oplus_s \mathfrak{r}_2 = \mathfrak{r}_1 \oplus \mathfrak{r}_2$.*

To illustrate the result of Proposition 2, we can use again the ENLST-system in Figure 2(a). A pair of its regions, $\mathfrak{r}_4$ and $\mathfrak{r}_1$, are compatible, but not strongly compatible regions. However, there are regions $\mathfrak{r}_3 \in \mathfrak{R}_{\mathfrak{ts}}^{r4}$ and $\mathfrak{r}_2 \in \mathfrak{R}_{\mathfrak{ts}}^{r1}$, such that $\mathfrak{r}_4$ and $\mathfrak{r}_2$ are strongly compatible and $\mathfrak{r}_3$ and $\mathfrak{r}_1$ are strongly compatible. Also, we have $\mathfrak{r}_4 \oplus_s \mathfrak{r}_2 = \mathfrak{r}_3 \oplus_s \mathfrak{r}_1 = \mathfrak{r}_4 \oplus \mathfrak{r}_1 = (\varnothing, Q, \varnothing)$.

Corollary 1. *Let $\mathfrak{ts} = (Q, A, q_0)$ be an* ENLST*-system. Then*

1. $\mathfrak{R}_{\mathfrak{ts}}^{min} = \mathfrak{R}_{\mathfrak{ts}}^{min,s}$.

2. *Every* $\mathfrak{r} \in \mathfrak{R}_{\mathfrak{ts}}$ *can be represented as a composition of minimal regions, where each pair of different minimal regions in this representation,* $\mathfrak{r} = (in, r, out)$ *and* $\mathfrak{r}' = (in', r', out')$, *is a pair of strongly compatible regions.*

The next result, about special families of non-trivial regions of $\mathfrak{ts}$, is inspired by a result proved for the class of Elementary Net Systems in [6]. We have adapted this result here to the context of ENL-systems by changing one of the original conditions that a family of regions should satisfy, but the implied result is the same: a family of regions that satisfy the conditions of Theorem 3, treated as a set of conditions of the synthesised net, would generate a state machine component of this net. Points 2 and 3 of the consequent of Theorem 3 guarantee the satisfaction of Definition 3(1) and the point 1 of the consequent of Theorem 3 guarantees the satisfaction of Definition 3(2).

Theorem 3. *Let* $\mathfrak{R} = \{\mathfrak{r}_1, \mathfrak{r}_2, \ldots, \mathfrak{r}_n\}$ *be a family of non-trivial regions of* $\mathfrak{ts} = (Q, A, q_0)$, *where* $\mathfrak{r}_i = (in_i, r_i, out_i)$, $i \in \{1, \ldots, n\}$, *satisfy the following:*

1. *Every two different regions* $\mathfrak{r}_i, \mathfrak{r}_j \in \mathfrak{R}$ *are strongly compatible regions.*
2. $\forall\, \widehat{\mathfrak{r}} = (\widehat{in}, \widehat{r}, \widehat{out}) \in \mathfrak{R}_{\mathfrak{ts}} :\ \widehat{\mathfrak{r}} \notin \mathfrak{R} \implies (\exists \mathfrak{r}_i \in \mathfrak{R} :\ \widehat{r} \cap r_i \neq \varnothing).$

Then:

1. $\bigcup r_i = Q$;
2. $\forall e \in E :\ |{}^\circ e \cap \mathfrak{R}| \leq 1$ *and* $|e^\circ \cap \mathfrak{R}| \leq 1$;
3. $\forall e \in E :\ e \in {}^\circ\mathfrak{r}_i \iff \exists j :\ e \in \mathfrak{r}_j{}^\circ$.

The saturated ENL-system that is a solution to Problem 1 for a given ENLST-system $\mathfrak{ts}$, $\mathfrak{enl} = \mathfrak{enl}^{\widehat{\ }}_{\mathfrak{ts}}$, and is based on all non-trivial regions, is state machine decomposable (see Definition 5), as due to Fact 4 every pair of complementary regions satisfies the conditions of Theorem 3 and would form a state machine component of $\mathfrak{enl}$. Furthermore, from Corollary 1 it follows that, similarly as for the class of Elementary Net Systems (see [6]), the ENL-system obtained from $\mathfrak{enl}$ by deleting all non-minimal regions following Reduction Rule 2 is also state machine decomposable as every region can be represented as a composition of minimal regions (w.r.t. $\prec_s$) and selected subsets of $\mathfrak{R}^{min}_{\mathfrak{ts}} = \mathfrak{R}^{min,s}_{\mathfrak{ts}}$ would satisfy the conditions of Theorem 3.

As an example we can take the saturated ENL-system synthesised from the ENLST-system in Figure 2(a), shown in Figure 2(b), and its minimised version shown in Figure 2(c). The state machine components of the former ENL-system are generated by the following subsets of conditions/regions:

$$\mathfrak{r}_1 \oplus_s \mathfrak{r}_3 = \mathfrak{r}_2 \oplus_s \mathfrak{r}_5 \oplus_s \mathfrak{r}_6 = \mathfrak{r}_2 \oplus_s \mathfrak{r}_4 = \mathfrak{r}_7 \oplus_s \mathfrak{r}_5 = \mathfrak{r}_8 \oplus_s \mathfrak{r}_6 = (\varnothing, Q, \varnothing).$$

The minimised ENL-system in Figure 2(c) has the first two state machine components from the components listed above.

6. A strategy to eliminate redundant regions

The three reduction rules give conditions for deleting one of the redundant regions at a time. Therefore, we need a *strategy* to delete as many redundant regions as possible to obtain a net, where all (or almost all) remaining regions are needed (essential). The regions are redundant or essential only in the context of other regions. Different strategies lead to different sets of essential (or nearly essential) regions. Such sets of regions were called in [28] *admissible*.

As an example we can take again the ENLST-system in Figure 2(a). We observe that region $\mathfrak{r}_4 = \bar{\mathfrak{r}}_2$ can be deleted according to Reduction Rule 1 (as the complement of region $\mathfrak{r}_2$) or according to Reduction Rule 2 (as a non-minimal region: $\mathfrak{r}_4 = \mathfrak{r}_5 \oplus \mathfrak{r}_6$).

When looking for a strategy for deleting redundant regions, we will take into consideration the following criteria:

- Limiting as much as possible the non-determinism in the process of computing admissible regions.
- Effectiveness of the strategy gauged in terms of the number of the removed regions.
- Efficiency of the strategy gauged in terms of time needed to compute a set of admissible regions.

Our first attempt at formulating a strategy will be based on the first criterion listed above. Reduction Rule 2 showed that all non-minimal regions are redundant [2], so we can eliminate first the non-minimal regions. After this step, for a given ENLST-system $\mathfrak{ts}$, we obtain from the unique set of regions, $\mathfrak{R}_{\mathfrak{ts}}$, the unique set of minimal regions: $\mathfrak{R}_{\mathfrak{ts}}^{min}$. The application of Reduction Rule 1 and Reduction Rule 3 might not lead to a unique resultant set of regions. We might decide to keep certain companion regions and delete other companion regions in case of Reduction Rule 3. Similarly, we can keep both or one (random one) out of two complementary regions. As, in general, the Reduction Rule 3, leads to fewer possible choices of regions to delete, and might be even irrelevant in the case of thin step transition systems, where there are no companion regions, we might decide that this rule should be applied before the Reduction Rule 1, which can lead to many possible combinations of regions to keep/delete. This strategy, called *Strategy (2,3,1)*, can be defined as follows:

1. Use Reduction Rule 2 to delete all non-minimal regions.

2. Use Reduction Rule 3 to delete any redundant companion regions that might be present among the minimal regions.

3. Use Reduction Rule 1 to delete any redundant complementary regions that might be present after the first two steps of the strategy.

To check how good this strategy is from the second criterion point of view we consider a set of ENLST-systems generated by nets composed of several sequential subsystems, where all the events are co-located. Such systems have a lot of companion regions. We will call them $\mathfrak{ts}_{i,j}^{co-loc}$, where the index i denotes the number of sequential subsystems, and the index j denotes the

[2] Reduction Rule 2 uses operator $\oplus$ and it was proved in [15] for this operator, but from Corollary 1(1) we have $\mathfrak{R}_{\mathfrak{ts}}^{min} = \mathfrak{R}_{\mathfrak{ts}}^{min,s}$, so it does not matter whether we use $\prec$ and $\oplus$, or $\prec_s$ and $\oplus_s$, to define the set of minimal regions.

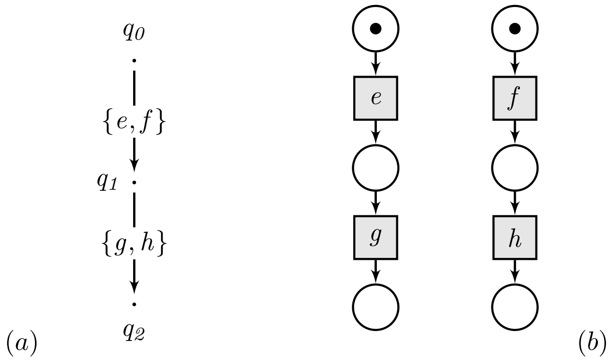

Figure 3: An ENLST-system $\mathfrak{ts}^{co-loc}_{2,2}$ with co-located events e, f, g and h (a), and one of the possible ENL-systems generating it (b).

Table 1
Comparison between the effectiveness of Strategy (3,2,1) and Strategy (2,3,1), where x - y - z in the last two columns reports the number of remaining regions after the first (x), the second (y) and the third (z) stage of the strategies.

$\mathfrak{ts}$	$\lvert\mathbf{Q}\rvert$	$\lvert\mathbf{E}\rvert$	$\lvert\mathfrak{R}_{\mathfrak{ts}}\rvert$	Strategy (3,2,1)	Strategy (2,3,1)
$\mathfrak{ts}^{co-loc}_{2,2}$	3	4	16	12 - 6 - 6	8 - 6 - 6
$\mathfrak{ts}^{co-loc}_{2,3}$	4	6	52	28 - 10 - 10	12 - 8 - 8
$\mathfrak{ts}^{co-loc}_{2,4}$	5	8	160	60 - 12 - 12	16 - 10 - 10
$\mathfrak{ts}^{co-loc}_{2,5}$	6	10	484	124 - 26 - 26	20 - 12 - 12
$\mathfrak{ts}^{co-loc}_{3,2}$	3	6	30	22 - 11 - 11	15 - 11 - 11
$\mathfrak{ts}^{co-loc}_{3,3}$	4	9	126	66 - 20 - 20	24 - 16 - 16
$\mathfrak{ts}^{co-loc}_{3,4}$	5	12	510	190 - 52 - 51	33 - 21 - 21
$\mathfrak{ts}^{co-loc}_{3,5}$	6	15	2046	546 - 147 - 143	42 - 26 - 26
$\mathfrak{ts}^{co-loc}_{4,2}$	3	8	48	36 - 18 - 18	24 - 18 - 18
$\mathfrak{ts}^{co-loc}_{4,3}$	4	12	248	140 - 48 - 47	40 - 28 - 28
$\mathfrak{ts}^{co-loc}_{4,4}$	5	16	1248	540 - 165 - 159	56 - 38 - 38
$\mathfrak{ts}^{co-loc}_{4,5}$	6	20	6248	2108 - 532 - 508	72 - 48 - 48

number of events in each of the line-like sequential subsystem. As an example of such a step transition system we can see an ENLST-system $\mathfrak{ts}^{co-loc}_{2,2}$ in Figure 3(a).

Using the set of step transition systems $\mathfrak{ts}^{co-loc}_{i,j}$ ($i = 2, \ldots, 4;\ j = 2, \ldots, 5$), we compare the effectiveness of region removal of Strategy (2,3,1) and Strategy (3,2,1) (Strategy (2,3,1) with the first two steps reversed). The result of this comparison is presented in Table 1.

The results in Table 1 are not so surprising. The removal of non-minimal regions makes only sense in the context of all non-trivial regions (as the first step of the strategy). When removing companion regions, the algorithm processes groups of companion regions (each based on a shared set of states) separately from each other. From each group some subset of regions may be removed, at random, according to Reduction Rule 3. Once some of the minimal companion regions are removed (if we remove companion regions first), some of the regions that were previously non-minimal would become minimal as it won't be possible to represent them as compositions of minimal regions using the remaining minimal regions. Therefore, they won't be deleted by the Reduction Rule 2, if it is applied after Reduction Rule 3. The results in Table 1 show how great would be the loss of effectiveness if we used Strategy (3,2,1) for ENLST-systems with a big number of companion regions.

While the Reduction Rule 2, applied first in our strategy, can be considered as a method for eliminating non-minimal regions, non-minimal from the 'state information' point of view, the Reduction Rule 3 can be understood as a method for eliminating regions that are redundant from the 'event information' point of view. However, some subsets of companion regions will remain after the application of Reduction Rule 3, because they are essential as shown below:

Fact 5. *Let $\mathfrak{r} = (in, r, out)$ be a non-trivial region of $\mathfrak{ts} = (Q, A, q_0)$ and let $\mathfrak{R}^r \subseteq \mathfrak{R}^r_{\mathfrak{ts}}$ be a set of its companion regions (including $\mathfrak{r}$) that were left after the application of the Reduction Rule 3 and let $|\mathfrak{R}^r| \geq 2$. Then all the regions of $\mathfrak{R}^r$ do not satisfy the same conditions of the Reduction Rule 3 (all do not satisfy condition (1) or all do not satisfy condition (2) or all do not satisfy both conditions: (1) and (2)).*

Corollary 2. *Let $\mathfrak{R}^r = \{\mathfrak{r}_1, \mathfrak{r}_2, \ldots, \mathfrak{r}_n\} \subseteq \mathfrak{R}^r_{\mathfrak{ts}}$ be a set of companion regions based on r that were left after the application of the Reduction Rule 3. Then, for every region $\mathfrak{r}_i = (in_i, r, out_i)$ of $\mathfrak{R}^r$ that does not satisfy condition (1) (respectively (2)) of Reduction Rule 3 there exists a unique $E_i \subseteq in_i$ (respectively $E'_i \subseteq out_i$) with events that are not present in the in (respectively out) sets of other regions from $\mathfrak{R}^r$. So, companion regions of $\mathfrak{R}^r$ are 'indexed' by the unique subsets of events of their in (respectively out) sets. We will call these subsets of events* in-indices *(respectively* out-indices*) of r for the regions of $\mathfrak{R}^r$.* ♢

Notice that sets $\mathfrak{R}^r$ in Corollary 2 (and in Fact 5) might be equal to $\mathfrak{R}^r_{\mathfrak{ts}}$. For example, for $\mathfrak{ts}$ in Figure 2(a), we have $\mathfrak{R}^{\{q_0\}} = \mathfrak{R}^{\{q_0\}}_{\mathfrak{ts}} = \{\mathfrak{r}_1, \mathfrak{r}_2\}$. Also, the indexing sets of events do not need to be singleton sets (as, for example, set $\{e_1, e_2\}$ for $\{q_0\}$ of $\mathfrak{r}_2$ of $\mathfrak{ts}$ in Figure 2(a)).

We will further illustrate the above results using the ENLST-system in Figure 3(a). Its non-trivial regions are listed below:

$$\begin{array}{lcllcl}
\mathfrak{r}_1 & = & (\varnothing, \{q_0\}, \{e\}) & \bar{\mathfrak{r}}_1 & = & (\{e\}, \{q_1, q_2\}, \varnothing) \\
\mathfrak{r}_2 & = & (\{e\}, \{q_1\}, \{g\}) & \bar{\mathfrak{r}}_2 & = & (\{g\}, \{q_0, q_2\}, \{e\}) \\
\mathfrak{r}_3 & = & (\{g\}, \{q_2\}, \varnothing) & \bar{\mathfrak{r}}_3 & = & (\varnothing, \{q_0, q_1\}, \{g\}) \\
\mathfrak{r}_4 & = & (\{e\}, \{q_1\}, \{h\}) & \bar{\mathfrak{r}}_4 & = & (\{h\}, \{q_0, q_2\}, \{e\}) \\
\mathfrak{r}_5 & = & (\{h\}, \{q_2\}, \varnothing) & \bar{\mathfrak{r}}_5 & = & (\varnothing, \{q_0, q_1\}, \{h\}) \\
\mathfrak{r}_6 & = & (\varnothing, \{q_0\}, \{f\}) & \bar{\mathfrak{r}}_6 & = & (\{f\}, \{q_1, q_2\}, \varnothing) \\
\mathfrak{r}_7 & = & (\{f\}, \{q_1\}, \{g\}) & \bar{\mathfrak{r}}_7 & = & (\{g\}, \{q_0, q_2\}, \{f\}) \\
\mathfrak{r}_8 & = & (\{f\}, \{q_1\}, \{h\}) & \bar{\mathfrak{r}}_8 & = & (\{h\}, \{q_0, q_2\}, \{f\})
\end{array}$$

The implemented tool, after applying Reduction Rule 2, will delete 8 out of 16 regions, leaving the minimal regions ($\mathfrak{r}_1$ - $\mathfrak{r}_8$). The set of minimal regions will be the same whether they are defined w.r.t. $\prec$ or $\prec_s$ strict pre-order (see Corollary 1(1)) as every non-minimal region that can be expressed as a composition of compatible regions can be also expressed as a composition of strongly compatible regions (see Proposition 2). For example, $\bar{\mathfrak{r}}_3 = \mathfrak{r}_1 \oplus_s \mathfrak{r}_2 = \mathfrak{r}_1 \oplus \mathfrak{r}_7$, where the first two regions are strongly compatible, but the second two regions are only compatible, but not strongly compatible. The algorithm that implements Reduction Rule 3, when applied to this example, would delete two out of four regions based on the set of states $\{q_1\}$ leaving either $\mathfrak{r}_2$ and $\mathfrak{r}_8$ or $\mathfrak{r}_4$ and $\mathfrak{r}_7$. The remaining pairs of companion regions, based on sets of states $\{q_0\}$,$\{q_1\}$ and $\{q_2\}$, will remain as they are essential (having different in-indices or/and out-indices for the shared sets of states; see Corollary 2). The Reduction Rule 1, the last to be used in Strategy (2,3,1), is not applicable to this example as all the complementary regions of $\mathfrak{r}_1$ - $\mathfrak{r}_8$ were already deleted as non-minimal regions (see the results for $\mathfrak{ts}^{co-loc}_{2,2}$ in Table 1).

7. Conclusions

In this paper we discussed the minimisation of the synthesised ENL-systems and the strategy to eliminate redundant regions that involves three reduction rules. Also, we investigated the properties of minimal regions that play a crucial role in the minimisation process. We showed that synthesised and minimised nets that are based on all minimal regions (after the application of the Reduction Rule 2) do not lose the property of the saturated ENL-systems of being state machine decomposable. We believe [3] that after the application of the Reduction Rule 3 this property still holds for the resultant net. However, after applying Reduction Rule 1, some synthesised ENL-systems are no longer decomposable. As an example, we can take the ENLST-system generated by the ENL-system in Figure 1. The synthesis procedure for this example will produce a saturated net that has only minimal regions (12 regions). Some of them are redundant and can be deleted according to Reduction Rule 1. Figure 1 shows one of the possible minimised versions of this net that is not state machine decomposable.

In the future work, we plan to investigate the relationship between the split of ENL-systems into state machine components (based on conditions) and the split into localities (based on events). Also, we want to develop an improved algorithm implementing Reduction Rule 1, which would allow to target certain regions for deletion from the pairs of complementary regions.

Acknowledgement

We are grateful to the anonymous reviewers for their constructive comments, which have helped us to improve the presentation of the paper and to clarify our ideas.

The first author is grateful to the National Transitional Council of Libya for funding her PhD studentship and research.

References

[1] S. Dasgupta, D. Potop-Butucaru, B. Caillaud, A. Yakovlev, Moving from weakly endochronous systems to delay-insensitive circuits, Electronic Notes in Theoretical Compututer Science 146 (2006) 81–103.

[3] This still needs to be formally proved.

[2] G. Păun, Membrane Computing: An Introduction, Natural Computing Series, Springer, 2002.

[3] J. Kleijn, M. Koutny, G. Rozenberg, Towards a Petri net semantics for membrane systems, in: R. Freund, G. Paun, G. Rozenberg, A. Salomaa (Eds.), Membrane Computing, 6th International Workshop, WMC 2005, Vienna, Austria, July 18-21, 2005, Revised Selected and Invited Papers, volume 3850 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 292–309.

[4] A. Ehrenfeucht, G. Rozenberg, Partial (set) 2-structures. Part I: Basic notions and the representation problem. Part II: State spaces of concurrent systems, Acta Informatica 27 (1990) 315–368.

[5] É. Badouel, P. Darondeau, Theory of regions, in: W. Reisig, G. Rozenberg (Eds.), Lectures on Petri Nets I: Basic Models, Advances in Petri Nets, volume 1491 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 529–586.

[6] L. Bernardinello, Synthesis of net systems, in: M. A. Marsan (Ed.), Application and Theory of Petri Nets 1993, 14th International Conference, Chicago, Illinois, USA, June 21-25, 1993. Proceedings, volume 691 of *Lecture Notes in Computer Science*, Springer, 1993, pp. 89–105.

[7] L. Bernardinello, G. D. Michelis, K. Petruni, S. Vigna, On the synchronic structure of transition systems, in: J. Desel (Ed.), Proceedings of the International Workshop on Structures in Concurrency Theory, STRICT 1995, Berlin, Germany, May 11-13, 1995, Workshops in Computing, Springer, 1995, pp. 69–84.

[8] M. Mukund, Petri nets and step transition systems, International Journal of Foundations of Computer Science 3 (1992) 443–478.

[9] M. Nielsen, G. Rozenberg, P. S. Thiagarajan, Elementary transition systems, Theoretical Computer Science 96 (1992) 3–33.

[10] M. Pietkiewicz-Koutny, The synthesis problem for elementary net systems with inhibitor arcs, Fundamenta Informaticae 40 (1999) 251–283.

[11] É. Badouel, L. Bernardinello, P. Darondeau, Petri Net Synthesis, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2015.

[12] M. Koutny, M. Pietkiewicz-Koutny, Transition systems of elementary net systems with localities, in: C. Baier, H. Hermanns (Eds.), Concurrency Theory 2006, 17th International Conference, CONCUR 2006, Bonn, Germany, August 27-30, 2006. Proceedings, volume 4137 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 173–187.

[13] M. Koutny, M. Pietkiewicz-Koutny, Synthesis of elementary net systems with context arcs and localities, Fundamenta Informaticae 88 (2008) 307–328.

[14] M. Koutny, M. Pietkiewicz-Koutny, Synthesis of Petri nets with localities, Scientific Annals of Computer Science 19 (2009) 1–23.

[15] M. Koutny, M. Pietkiewicz-Koutny, Minimal regions of ENL-transition systems, Fundamenta Informaticae 101 (2010) 45–58.

[16] A. Ahmed, M. Pietkiewicz-Koutny, Algorithms for the synthesis of elementary net systems with localities, in: M. Köhler-Bußmeier, E. Kindler, H. Rölke (Eds.), Proceedings of the International Workshop on Petri Nets and Software Engineering, PNSE 2020, Paris, France, June 24, 2020 (due to COVID-19: virtual workshop), volume 2651 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 86–107.

[17] A. Ahmed, M. Koutny, M. Pietkiewicz-Koutny, Synthesising elementary net systems with

localities, Theoretical Computer Science 908 (2022) 123–140.

[18] W. M. P. van der Aalst, B. F. van Dongen, C. W. Günther, A. Rozinat, E. Verbeek, T. Weijters, Prom: The process mining toolkit, in: A. K. A. de Medeiros, B. Weber (Eds.), Proceedings of the Business Process Management Demonstration Track (BPMDemos 2009), Ulm, Germany, September 8, 2009, volume 489 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2009.

[19] R. Bergenthum, J. Desel, R. Lorenz, S. Mauser, Synthesis of Petri nets from scenarios with viptool, in: K. M. van Hee, R. Valk (Eds.), Applications and Theory of Petri Nets, 29th International Conference, PETRI NETS 2008, Xi'an, China, June 23-27, 2008. Proceedings, volume 5062 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 388–398.

[20] R. Bergenthum, Prime miner - process discovery using prime event structures, in: International Conference on Process Mining, ICPM 2019, Aachen, Germany, June 24-26, 2019, IEEE, 2019, pp. 41–48.

[21] J. Carmona, J. Cortadella, M. Kishinevsky, Genet: A tool for the synthesis and mining of Petri nets, in: Application of Concurrency to System Design 2009, 9th International Conference, ACSD 2009, Augsburg, Germany, 1-3 July 2009, IEEE Computer Society, 2009, pp. 181–185.

[22] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, A. Yakovlev, Petrify: A tool for manipulating concurrent specifications and synthesis of asynchronous controllers, IEICE Transactions on Information and Systems E80-D (1997) 315–325.

[23] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, A. Yakovlev, Logic Synthesis of Asynchronous Controllers and Interfaces, Springer Berlin Heidelberg, 2002.

[24] L. L. Mannel, R. Bergenthum, W. M. P. van der Aalst, Removing implicit places using regions for process discovery, in: W. M. P. van der Aalst, R. Bergenthum, J. Carmona (Eds.), Proceedings of the International Workshop on Algorithms & Theories for the Analysis of Event Data 2020, ATAED 2020, Paris, France, June 24, 2020 (due to COVID-19: virtual workshop), volume 2625 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 20–32.

[25] I. Poliakov, V. Khomenko, A. Yakovlev, Workcraft - A framework for interpreted graph models, in: G. Franceschinis, K. Wolf (Eds.), Applications and Theory of Petri Nets, 30th International Conference, PETRI NETS 2009, Paris, France, June 22-26, 2009. Proceedings, volume 5606 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 333–342.

[26] Workcraft, https://workcraft.org/, 2018.

[27] M. Pietkiewicz-Koutny, Synthesis of ENI-systems using minimal regions, in: D. Sangiorgi, R. de Simone (Eds.), Concurrency Theory 1998, 9th International Conference, CONCUR 1998: Nice, France, September 8-11, 1998. Proceedings, volume 1466 of *Lecture Notes in Computer Science*, Springer, 1998, pp. 565–580.

[28] J. Desel, W. Reisig, The synthesis problem of Petri nets, Acta Informatica 33 (1996) 297–315.