Theory of Regions

Eric Badouel and Philippe Darondeau

Irisa, Campus de Beaulieu, F-35042 Rennes Cedex, France
E-mail : {Eric.Badouel,Philippe.Darondeau}@irisa.fr

Abstract. The synthesis problem for nets consists in deciding whether a given graph is isomorphic to the marking graph of some net and then constructing it. This problem has been solved in the literature for various types of nets ranging from elementary nets to Petri nets. The general principle for the synthesis is to inspect regions of graphs representing extensions of places of the likely generating nets. We follow in this survey the gradual development of the theory of regions from its foundation by Ehrenfeucht and Rozenberg, with a particular insistence on the abstract meaning of the theory, which is a general product decomposition of graphs into atomic components determined by a parameter called a type of nets, and on the derivation of efficient algorithms for net synthesis based on linear algebra.

Table of Contents:

1 Terminology of Graphs

Since the terminology on graph theory varies a lot from one author to the other, we found it necessary to begin by defining the terminology used in this document.

1.1 Graphs

A *graph* $G = (X, E)$ is a collection X of *vertices* or *nodes* together with a collection E of *edges*. The graph is said to be *finite* if it has finitely many vertices and edges. Each edge has either one endpoint: $\text{end}(e) = \{x\}$ in which case e is termed a *loop* at vertex x, or two endpoints: $\text{end}(e) = \{x, y\}$ in which case e is termed a *link* between vertices x and y. A graph is *simple* if it is *loop-free*: each edge is a link, and has no multiple edge: $\text{end}(e_1) = \text{end}(e_2) \Rightarrow e_1 = e_2$. Therefore an edge of a simple graph may be identified with the pair of its endpoints. The *incidence matrix* of a graph G is a matrix A with elements 0 and 1, where each row corresponds to a vertex, each column corresponds to an edge, and $A(x, e)$ is 1 if and only if x is an endpoint of e. A *chain* of length $n \geq 1$ with endpoints $\{x_1, x_{n+1}\}$ is a finite sequence $(x_1, e_1, x_2, \ldots, x_n, e_n, x_{n+1})$ of vertices and edges such that $\text{end}(e_i) = \{x_i, x_{i+1}\}$ for all $1 \leq i \leq n$. We say that the chain *connects* its endpoints. For convenience, we consider that every vertex is connected to itself by an empty chain. The *connected component* of a vertex is the set of vertices connected to this vertex by some chain; the graph is *connected* if it has only one connected component. A non empty chain is said to be *simple* if all edges are distinct, a chain is said to be *elementary* if all the vertices but possibly the endpoints are pairwise distinct. A *cycle* is a simple chain whose endpoints coincide: $x_1 = x_{n+1}$. A *tree* is a graph with no cycle or alternatively a graph in which any two vertices are connected by a unique chain. $G' = (X', E')$ is a *subgraph* of $G = (X, E)$ if $X' \subseteq X$, $E' \subseteq E$, and the mappings that send an edge $e \in E'$ to its endpoints in G' and in G coincide. G' *spans* G if $X' = X$; a *spanning tree* of G is a subgraph which is a tree spanning G.

1.2 Directed Graphs

An *orientation* of an edge e is an ordered pair of vertices (x, y) such that $\text{end}(e) = \{x, y\}$, thus a loop at x has only one possible orientation: (x, x), while a link between x and y has two possible orientations: (x, y) and (y, x). We let $e : (x, y)$ denote the assignement of the orientation (x, y) to the edge e; the vertices $x = \partial^0(e)$ and $y = \partial^1(e)$ are respectively called the *source* and *target* of edge e. An oriented edge is sometimes called an *arc*. A *directed* graph is a graph whose edges are given an orientation. A directed graph is *simple* if it is loop-free and has no multiple arc in the sense that two edges with the same endpoints are necessarily given opposite orientations: $(e_1 : (x, y) \;\; \wedge \;\; e_2 : (x, y)) \Rightarrow e_1 = e_2$. Therefore an edge of a simple directed graph may be identified with the ordered pair of its endpoints, and in that case we write $e = (x, y)$ when $\partial^0(e) = x$ and $\partial^1(e) = y$. Notice that the underlying graph of a simple oriented graph may not be simple as we can find two edges with the same endpoints but with opposite orientations. A subgraph of a directed graph G is a subgraph of the underlying graph with the orientations of edges inherited from G. The notions of chain, cycle, tree, spanning subgraph and spanning tree do not depend on the orientation of edges; therefore a chain (cycle, tree, ...) of a directed graph is a chain (cycle, tree, ...) of the underlying graph. The specific notions that take the orientation into account are the following. A *path* of length $n \geq 1$ from x_1 to x_{n+1} is a finite sequence $(x_1, e_1, x_2, \ldots, x_n, e_n, x_{n+1})$ of vertices and edges such that $\partial^0(e_i) = x_i$ and $\partial^1(e_i) = x_{i+1}$ for all $1 \leq i \leq n$. For convenience, we consider that there exists an empty path from any vertex to itself. A non empty path is said to be *simple* if all edges are distinct. A path is said to be *elementary* if all the vertices but possibly the endpoints are distinct. A *circuit* is a simple path whose endpoints coincide: $x_1 = x_{n+1}$. Thus paths and circuits are respectively chains and cycles of the underlying graph whose edges have compatible orientations. The *incidence matrix* of a directed graph is the matrix $A : X \times E \to \{-1; 0; 1\}$ given by $A(x,e) = \begin{cases} 1 & \text{if} \quad \partial^0(e) = x \\ -1 & \text{if} \quad \partial^1(e) = x \\ 0 & \text{otherwise} \end{cases}$.

2 Regional Representation of Partial 2-Structures

The theory of regions was founded by Ehrenfeucht and Rozenberg in [22] with the aim to obtain a set-theoretic representation of directed graphs (X, E), enriched with an equivalence $\equiv$ on edges. The resulting structures $(X, E, \equiv)$ are termed *partial 2-structures*. The representation problem for partial 2-structures consists in attaching properties p to nodes x so that the Kripke structure so obtained may be abstracted without loss of information to the data $\{x^* | \, x \in X\}$ and $\{e^* | \, e \in E\}$, where a node is encoded by the set $x^* = \{p \mid x \models p\}$ of properties it satisfies and an edge by the pair $e^* = (x^* \setminus y^*, y^* \setminus x^*)$ where x and y are the respective source and target of e. The main difficulty is to reconstruct the equivalence relation $\equiv$, and this cannot be done unless the considered properties are altered uniformly when passing along every edge in each equivalence class. These specific properties, seen as sets of nodes when identified with their extensions $\{x \mid x \models p\}$, are called *regions* in [22]. The presentation of regions in partial 2-structures given below is directly inspired from [22], where the proofs of the results may be found. The algorithmic aspects of elementary net synthesis will be examined in the next section.

2.1 Partial 2-Structures and their Regions

Definition 2.1 *A partial 2-structure is a triple $G = (X, E, \equiv)$ where X is a finite non empty set of nodes, $E \subseteq E_2(X) = \{(x_1, x_2) \in X \times X | \, x_1 \neq x_2\}$ is a set of 2-edges over X, and $\equiv$ is an equivalence relation on E. When $E = E_2(X)$ is the whole set of 2-edges over X, G is called a 2-structure.*

Partial 2-structures may be viewed as equivalence classes of labelled simple directed graphs, where two graphs are equivalent if their labelling functions have the same kernel. Of particular interest are the partial set 2-structures defined as follows.

Definition 2.2 *A partial set 2-structure of a finite set B is a partial 2-structure $G = (X, E, \equiv_\delta)$ where $X \subseteq \mathcal{P}(B)$ and $\equiv_\delta$ is the kernel of the function $\delta((M, M'))$ $= (M \setminus M', M' \setminus M)$ for $M, M' \in X$. Let $S2S(B)$ denote the (full) set 2-structure of B; i.e. when $X = \mathcal{P}(B)$ and $E = E_2(X)$.*

Thus in particular, any partial set 2-structure G of B is a substructure of $S2S(B)$. In notation, $G \leq S2S(B)$ where $(X_1, E_1, \equiv_1) \leq (X_2, E_2, \equiv_2)$ if $X_1 \subseteq X_2$, $E_1 \subseteq E_2$ and $\equiv_1$ is the restriction of $\equiv_2$ on $E_1 \times E_1$. The representation problem for partial 2-structures may be stated as follows.

> *Which partial 2-structures are isomorphic to substructures of $S2S(B)$ for some finite set B (of tokens)?*

The best way to grasp this problem is to examine the extents R_b of representation tokens $b \in B$ in the structure $S2S(B)$ itself, let $R_b = \{M \in \mathcal{P}(B) | \; b \in M\}$. So, $b \in M$ if and only if $M \in R_b$. The following may be observed.

1. For every pair of equivalent 2-edges (M_1, M_1') and (M_2, M_2'), and for every $b \in B$, $b \in M_1 \setminus M_1'$ entails $b \in M_2 \setminus M_2'$ and symmetrically $b \in M_1' \setminus M_1$ entails $b \in M_2' \setminus M_2$. This can also be expressed as follows:
   - $(M_1 \in R_b \wedge M_1' \notin R_b) \Rightarrow (M_2 \in R_b \wedge M_2' \notin R_b)$;
   - $(M_1 \notin R_b \wedge M_1' \in R_b) \Rightarrow (M_2 \notin R_b \wedge M_2' \in R_b)$.

   Thus, all the 2-edges in an equivalence class are incident to R_b outwards, or they are incident to R_b inwards, or they are not incident to R_b.
2. $\forall M_1, M_2 \in \mathcal{P}(B) \quad M_1 \neq M_2 \Rightarrow (\exists b \in B \; M_1 \in R_b \Leftrightarrow M_2 \notin R_b)$.
3. For every pair of inequivalent 2-edges (M_1, M_1') and (M_2, M_2'), there exists some token $b \in B$ such that one 2-edge is incident to R_b and the other is not, or one 2-edge is incident to R_b inwards and the other is incident to R_b outwards.

These properties are also valid for substructures $(X, E, \equiv_\delta)$ of $S2S(B)$, where R_b is the set $\{M \in X | \; b \in M\}$.

Definition 2.3 *A region in a partial 2-structure $G = (X, E, \equiv)$ is a subset of nodes $R \subseteq X$ such that for every pair of equivalent 2-edges (x_1, x_1') and (x_2, x_2') in E: $(x_1 \in R \wedge x_1' \notin R) \Rightarrow (x_2 \in R \wedge x_2' \notin R)$, and $(x_1 \notin R \wedge x_1' \in R) \Rightarrow (x_2 \notin R \wedge x_2' \in R)$. Let $\mathcal{R}_G$ denote the set of (non trivial) regions of G, and for $x \in X$, let $\mathcal{R}_G(x) = \{R \in \mathcal{R}_G | \; x \in R\}$.*

It is worth noting that the complement $X \setminus R$ of a region R is a region. In particular X and $\emptyset$ are regions (the trivial regions). Now the non trivial regions may serve as representation tokens for states, that is nodes, and at the same time for events, that is classes of equivalent 2-edges. One obtains in this way regional versions of partial 2-structures defined as follows.

Definition 2.4 *Given a partial 2-structure $G = (X, E, \equiv)$, the regional version of G is the partial set 2-structure* $\mathbf{regv}(G) = (X', E', \equiv_\delta)$ *with components $X' = \{\mathcal{R}_G(x) | \; x \in X\}$ and $E' = \{(\mathcal{R}_G(x), \mathcal{R}_G(x')) | \; (x, x') \in E\}$.*

In this construction, illustrated in Fig. 1, a node x is mapped to the set $\mathcal{R}_G(x)$ of the regions which include x. It appears from Fig. 1, where equivalent edges bear a common label, that the map **regv** is not an equivalence of partial 2-structures . The following theorem states when **regv** maps a partial 2-structure isomorphically to a partial set 2-structure (the regional representation of the latter).

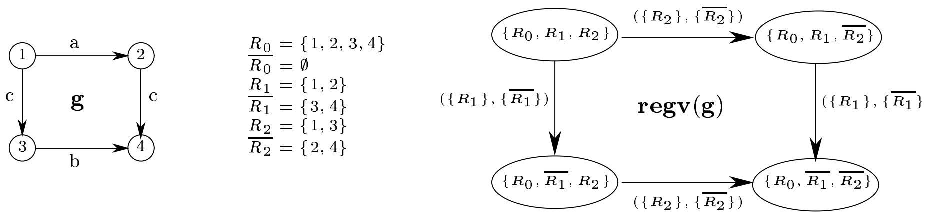


Fig. 1. a partial 2-structure and its regional version

Theorem 2.5 *A partial 2-structure $G = (X, E, \equiv)$ is isomorphic to a substructure of some set 2-structure if and only if $G \cong \mathbf{regv}(G)$ (with $\mathcal{R}_G(\cdot)$ as the isomorphism) if and only if the following two axioms of separation are satisfied:*

1. STATES SEPARATION*:* $\forall x_1, x_2 \in X \quad x_1 \neq x_2 \Rightarrow \exists R \in \mathcal{R}_G \ (x_1 \in R \Leftrightarrow x_2 \notin R)$.
2. EVENTS SEPARATION*: for all $(x_1, x_1'), (x_2, x_2') \in E$ with $(x_1, x_1') \not\equiv (x_2, x_2')$ there exists some region $R \in \mathcal{R}_G$ such that either (x_1, x_1') is incident to R outwards and (x_2, x_2') is not or (x_2, x_2') is incident to R outwards and (x_1, x_1') is not.*

There may exist nodes x_1, x_2, x_3 and x_4 such that $(x_1, x_2) \in E$, $\delta(x_1^*, x_2^*) = \delta(x_3^*, x_4^*)$, and $(x_3, x_4) \notin E$. Therefore $\mathbf{regv}(G)$ is not characterized by the sets $\{x^* | \ x \in X\}$ and $\{e^* | \ e \in E\}$. In order to reduce the mismatch, one should impose the additional axiom: $\forall (x_1, x_2) \in E \ \forall x_3, x_4 \in X \ \delta(\mathcal{R}_G(x_1), \mathcal{R}_G(x_2)) = \delta(\mathcal{R}_G(x_3), \mathcal{R}_G(x_4)) \Rightarrow (x_3, x_4) \in E$. Further on this way, one can even impose one or two stronger axioms:
FORWARD CLOSURE: $\forall (x_1, x_2) \in E \quad \forall x_3 \in X \quad (\mathcal{R}_G(x_1) \setminus \mathcal{R}_G(x_2) \subseteq \mathcal{R}_G(x_3) \ \wedge \ \mathcal{R}_G(x_3) \cap \mathcal{R}_G(x_2) \setminus \mathcal{R}_G(x_1) = \emptyset) \Rightarrow \exists x_4 \in X \ (x_3, x_4) \in E \wedge \delta(\mathcal{R}_G(x_1), \mathcal{R}_G(x_2)) = \delta(\mathcal{R}_G(x_3), \mathcal{R}_G(x_4))$.

BACKWARD CLOSURE: $\forall (x_1, x_2) \in E \quad \forall x_4 \in X \quad (\mathcal{R}_G(x_2) \setminus \mathcal{R}_G(x_1) \subseteq \mathcal{R}_G(x_4) \ \wedge \ \mathcal{R}_G(x_4) \cap \mathcal{R}_G(x_1) \setminus \mathcal{R}_G(x_2) = \emptyset) \Rightarrow \exists x_3 \in X \ (x_3, x_4) \in E \wedge \delta(\mathcal{R}_G(x_1), \mathcal{R}_G(x_2)) = \delta(\mathcal{R}_G(x_3), \mathcal{R}_G(x_4))$.

Partial 2-structures may be considered too general from a practical point of view, and one may prefer focusing on *reachable* partial 2-structures, such that all nodes can be reached by paths with a common origin. A familiar example of reachable partial set 2-structures is the class of sequential case graphs of elementary net systems.

Definition 2.6 *An elementary net is a directed bipartite graph $N = (P, E, F)$ such that $\mathbf{dom}(F) \cup \mathbf{ran}(F) = P \cup E$. Elements of P, respectively E, are called conditions (or places), resp. events. Let $x \in {}^\bullet y$ and $y \in x^\bullet$ be alternative notations of $(x, y) \in F$. A case (or marking) of N is a subset of conditions $M \in \mathcal{P}(P)$. An event e has concession in case M (noted $M[e\!>$) if and only if $({}^\bullet e, e^\bullet) = \delta(M, M')$ for some case M' (thus uniquely defined). The event e*

may then fire at M*, resulting in the step* $M[e{>}M'$*. Thus,* $M[e{>}$ *if and only if* ${}^\bullet e \subseteq M \;\wedge\; M \cap e^\bullet = \emptyset$*, and then* $M[e{>}M'$ *where* $M' = (M \setminus {}^\bullet e) \cup e^\bullet$*.*

A net is *pure* if $\forall x \in P \cup E \quad x^\bullet \cap {}^\bullet x = \emptyset$; it is *simple* if $\forall x, y \in P \cup E \quad (x^\bullet = y^\bullet \;\wedge\; {}^\bullet x = {}^\bullet y) \Rightarrow x = y$. The elementary nets considered from now on are assumed to be pure and simple.

Definition 2.7 *An elementary net system is a structure* $\mathcal{N} = (P, E, F, M_0)$ *where* $N = (P, E, F)$ *is the underlying net and* M_0 *(in* $\mathcal{P}(P)$*) is the initial case. The sequential case graph of* $\mathcal{N}$ *is the partial set 2-structure* $\mathbf{scg}(\mathcal{N}) = (X', E', \equiv_\delta)$ *where* $X' \subseteq \mathcal{P}(P)$ *is the smallest set of cases reachable from* M_0 *by sequences of steps* $M[e{>}M'$ *and* E' *is the set of corresponding pairs* (M, M')*.*

Lemma 2.8 *A partial set 2-structure* $G = (X, E, \equiv_\delta)$ *is the sequential case graph of an elementary net system if and only if it is reachable and the following property is satisfied:* $\forall (x_1, x_2) \in E \quad \forall x_3 \in X \quad (x_1 \setminus x_2 \subseteq x_3 \;\wedge\; x_3 \cap x_2 \setminus x_1 = \emptyset) \Rightarrow \exists x_4 \in X \;\; ((x_3, x_4) \in E \;\wedge\; \delta(x_1, x_2) = \delta(x_3, x_4))$*.*

From Theo. 2.5 and Lem. 2.8, one obtains the following.

Corollary 2.9 *A partial 2-structure* $G = (X, E, \equiv)$ *is isomorphic to the sequential case graph of an elementary net system if and only if it is reachable and satisfies the axioms of states separation, events separation, and forward closure.*

The elementary net system in the above corollary is essentially the set of the ordered symmetric differences $\delta(\mathcal{R}_G(x), \mathcal{R}_G(y))$ for 2-edges $(x, y) \in E$. The representation problem for partial 2-structures set at the beginning of the section has in fact been given the solution $x^* = \mathcal{R}_G(x)$. The places of the net are the regions $r \in \mathcal{R}(G)$, the events are the equivalence classes of edges, and the flow relation is such that: $F([e]_\equiv, r) \Leftrightarrow r \in \mathcal{R}_G(y) \setminus \mathcal{R}_G(x)$ for some $(x, y) \in E$; and $F(r, [e]_\equiv) \Leftrightarrow r \in \mathcal{R}_G(x) \setminus \mathcal{R}_G(y)$ for some $(x, y) \in E$. The initial case of the net system is defined as $\mathcal{R}_G(x_0)$ for some $x_0 \in X$ such that every node of G is reachable from x_0.

2.2 Elementary Automata

The second part of the section paves the way for the algorithmic analysis of the region based correspondence between reachable graphs and elementary net systems. With this objective in mind, we recast the results obtained so far into the framework of transition systems, and illustrate the modified correspondence on a complete example.

Definition 2.10 *A (labelled) transition system is a triple* $A = (S, E, T)$ *with a set of states* S*, a set of events* E*, and a set of transitions* $T \subseteq S \times E \times S$*. Let* $s \xrightarrow{e} s'$ *be an equivalent notation for* $(s, e, s') \in T$*. An event* e *is enabled at state* s *(noted* $s \xrightarrow{e}$*) if* $s \xrightarrow{e} s'$ *for some* s'*. An event* e *is co-enabled at* s' *(noted* $\xrightarrow{e} s'$*) if* $s \xrightarrow{e} s'$ *for some* s*. An automaton is a structure* $\mathcal{A} = (S, E, T, s_0)$ *consisting of an underlying transition system* $A = (S, E, T)$ *and an initial state* $s_0 \in S$*.*

A partial 2-structure $G = (X, E, \equiv)$ may be identified with the transition system $(X, E/\equiv, T)$ where $x \stackrel{[e]_\equiv}{\rightarrow} x'$ if and only if $(x, x') \equiv e$. This transition system is *loopfree*: $s \stackrel{e}{\rightarrow} s' \Rightarrow s \neq s'$, *has no multiple arc*: $s \stackrel{e_1}{\rightarrow} s' \wedge s \stackrel{e_2}{\rightarrow} s' \Rightarrow e_1 = e_2$, and *it is reduced*: $\forall e \in E \quad \exists s, s' \in S \quad s \stackrel{e}{\rightarrow} s'$. The sequential case graphs of the reduced net systems defined hereafter fall in this subclass of transition systems.

Definition 2.11 *An elementary net system $\mathcal{N} = (P, E, F, M_0)$ is reduced if every event $e \in E$ has concession at some case M reachable from M_0, and for every two distinct conditions $p, p' \in P$ there exists some case M reachable from M_0 such that $p \in M \Leftrightarrow p' \notin M$. The dual of a reduced elementary net system $\mathcal{N}$ is the automaton $\mathcal{N}^* = (S, E, T, M_0)$ where S is the set of cases reachable from M_0 by sequences of steps $M[e\!>\!M'$ and T is the set of the corresponding transitions (M, e, M').*

Thus $\mathcal{N}^*$ is essentially the image of $\mathbf{scg}(\mathcal{N})$ through the map which sends the equivalence class of 2-edges $\{(M, M')|\ \delta(M, M') = ({}^\bullet e, e^\bullet)\}$ to the event e. Since $\mathcal{N}$ is simple and reduced, this map is one to one and onto. By construction, $\mathcal{N}^*$ is *reachable* from M_0, *deterministic*: $M \stackrel{e}{\rightarrow} M' \wedge M \stackrel{e}{\rightarrow} M'' \Rightarrow M' = M''$, and *co-deterministic*: $M' \stackrel{e}{\rightarrow} M \wedge M'' \stackrel{e}{\rightarrow} M \Rightarrow M' = M''$. The definition of regions may be carried to automata in the following form.

Definition 2.12 *A region in an automaton $\mathcal{A} = (S, E, T, s_0)$, or in the underlying transition system (S, E, T), is a subset of states $R \subseteq S$ such that*
$$\forall e \in E\ \forall s_1, s_2, s_3, s_4 \in S \quad s_1 \stackrel{e}{\rightarrow} s_2 \wedge s_3 \stackrel{e}{\rightarrow} s_4 \Rightarrow \begin{cases} s_1 \in R \wedge s_2 \notin R \Rightarrow s_3 \in R \wedge s_4 \notin R \\ s_1 \notin R \wedge s_2 \in R \Rightarrow s_3 \notin R \wedge s_4 \in R \end{cases}$$
Let $\mathcal{R}_\mathcal{A}$ denote the set of (non trivial) regions of $\mathcal{A}$, and for $s \in S$ let $\mathcal{R}_\mathcal{A}(s) = \{R \in \mathcal{R}_\mathcal{A}|\ s \in R\}$.

Thus, R is a region if and only if the label e of a transition suffices to determine whether the transition is incident to R inwards (R is then termed an output region for e, noted $e^\bullet R$), or it is incident to R outwards (R is then termed an input region for e, noted $R^\bullet e$), or it is not incident to R (it is internal to R or external to R). In particular, if $\mathcal{A}$ is reachable and reduced, the non trivial regions of $\mathcal{A}$ may be represented as maps $\eta_R : E \rightarrow \{-1, 0, 1\}$ such that $\eta_R(e) = 1$ if $e^\bullet R$, $\eta_R(e) = -1$ if $R^\bullet e$, and $\eta_R(e) = 0$ otherwise; the characteristic function of R, let $\chi_R : S \rightarrow \{0, 1\}$ where $\chi_R(s) = 1 \Leftrightarrow s \in R$, is then the unique map such that $s \stackrel{e}{\rightarrow} s' \Rightarrow \eta_R(e) = \chi_R(s') - \chi_R(s)$.

It is easily seen that for every condition p of a net system $\mathcal{N}$, the set of the reachable cases M that contain p is a region of $\mathcal{N}^*$. This region, denoted by p^* and called the extension of p, is such that $e^\bullet p^* \Leftrightarrow e \in {}^\bullet p$ and $p^{*\bullet} e \Leftrightarrow e \in p^\bullet$. Reversing the process which leads from net systems to sequential case graphs, let us recast the definition of regional versions in terms of nets and net systems.

Definition 2.13 *Given an automaton $\mathcal{A} = (S, E, T, s_0)$, the dual of $\mathcal{A}$ is the (reduced) elementary net system $\mathcal{A}^* = (\mathcal{R}_\mathcal{A}, (E/_\sim) \setminus \{\varepsilon\}, F, s_0^*)$ where: $\sim$ is the equivalence on E induced by regions, let*
$$e_1 \sim e_2 \Leftrightarrow (\forall R \in \mathcal{R}_\mathcal{A} \quad e_1{}^\bullet R \Leftrightarrow e_2{}^\bullet R \quad \wedge \quad R^\bullet e_1 \Leftrightarrow R^\bullet e_2);$$

ε is the equivalence class of the events which are inputless and outputless i.e. which are internal or external to all regions, if such events exist; F is the flow relation such that $F([e]_\sim, R) \Leftrightarrow e^\bullet R$ and $F(R, [e]_\sim) \Leftrightarrow R^\bullet e$; and $s_0^* = \{R \in \mathcal{R}_\mathcal{A} |\ s_0 \in R\}$.

The net system $\mathcal{A}^*$ is also called the saturated net version of $\mathcal{A}$ (for reasons explained in the sequel). The counterpart of Cor. 2.9 for automata is the following.

Theorem 2.14 *An automaton $\mathcal{A} = (S, E, T, s_0)$ is isomorphic to the dual $\mathcal{N}^*$ of an elementary net system if and only if $\mathcal{A} \cong \mathcal{A}^{**}$ if and only if $\mathcal{A}$ is simple (it has neither loop nor multiple arc), reduced, reachable and it satisfies the following properties of separation:*
SSP *(States Separation Property):*

$$\forall s, s' \in S \quad s \neq s' \Rightarrow \exists R \in \mathcal{R}_\mathcal{A} \quad (s \in R \Leftrightarrow s' \notin R)$$

ESP *(Events Separation Property):*

$$\forall e, e' \in E \quad e \neq e' \Rightarrow \exists R \in \mathcal{R}_\mathcal{A} \quad (R^\bullet e \ \wedge \ not(R^\bullet e')) \quad \vee \quad (e^\bullet R \ \wedge \ not(e'^\bullet R))$$

ESSP *(Events-States Separation Property):*

$$\forall e \in E \quad \forall s \in S \quad not(s \overset{e}{\to}) \quad \Rightarrow \exists R \in \mathcal{R}_\mathcal{A} \ (R^\bullet e \ \wedge \ s \notin R) \ \vee \ (e^\bullet R \ \wedge \ s \in R)$$

An automaton satisfying these conditions is termed an elementary automaton.

Observe that every event in an elementary automaton has input regions and output regions (from SSP), hence the map sending e to $[e]_\sim$ is a bijection between E and $(E/\sim) \setminus \{\varepsilon\}$ (from ESP). The isomorphism from $\mathcal{A}$ to $\mathcal{A}^{**}$ (the sequential case graph of the saturated net version of $\mathcal{A}$) maps e to $[e]_\sim$ and s to $s^* = \{R \in \mathcal{R}_\mathcal{A} |\ s \in R\}$. This isomorphism applies in particular to sequential case graphs, whence $\mathcal{N}^* \cong \mathcal{N}^{***}$ for every elementary net system. However, $\mathcal{N} = (P, E, F, M_0)$ is generally not isomorphic to its double dual $\mathcal{N}^{**}$. In fact, every condition p of $\mathcal{N}$ induces a corresponding region p^* of $\mathcal{N}^*$ which includes the reachable cases in which condition p holds, and $\mathcal{N}$ is isomorphic to the full subnet system of $\mathcal{N}^{**}$ with set of events $E/_\sim$ $(= E)$ and set of places $\{p^* |\ p \in P\}$. Thus, whenever $\mathcal{N}'^* \cong \mathcal{N}^*$, $\mathcal{N}'$ is isomorphic to a subnet system of $\mathcal{N}^{**}$ which is for that reason termed the saturated version of $\mathcal{N}$. Now, for an elementary automaton $\mathcal{A}$, $\mathcal{A} \cong \mathcal{A}^{**}$ entails that $\mathcal{A}^* \cong \mathcal{A}^{***}$, hence $\mathcal{A}^*$ is always a saturated net system. The aim of the next section is to optimize the synthesis process by looking at admissible subnets $\mathcal{N}$ of $\mathcal{A}^*$ such that $\mathcal{A} \cong \mathcal{N}^*$.

Before tackling the synthesis problem, we proceed to simplifying the presentation of elementary automata, and retrieve the usual presentation given in [11, 19, 34].

Proposition 2.15 *Let automaton $\mathcal{A}$ be simple, reduced and reachable, then $\mathcal{A}$ is elementary if and only if the separation properties* SSP *and* ESSP *are satisfied.*

Proof: Let $\mathcal{A} = (S, E, T, s_0)$, and assume for contradiction $e \neq e'$ and $\forall R \in \mathcal{R}_{\mathcal{A}} \quad (R^\bullet e \Leftrightarrow R^\bullet e') \quad \wedge \quad (e^\bullet R \Leftrightarrow e'^\bullet R)$. We show that $s \xrightarrow{e} s'$ entails $s \xrightarrow{e'} s'$ contradicting the assumption that $\mathcal{A}$ is simple. Assume $s \xrightarrow{e} s'$ and not $s \xrightarrow{e'}$, then by ESSP: $\exists R \in \mathcal{R}_{\mathcal{A}} \; (R^\bullet e' \wedge s \notin R) \vee (e'^\bullet R \wedge s \in R)$ and the contradiction of $s \xrightarrow{e} s'$ follows from the definition of regions. Let $s'' \in S$ such that $s \xrightarrow{e'} s''$, then $\mathcal{R}_{\mathcal{A}}(s'') = \mathcal{R}_{\mathcal{A}}(s) \setminus {}^\bullet e' \cup e'^\bullet = \mathcal{R}_{\mathcal{A}}(s) \setminus {}^\bullet e \cup e^\bullet = \mathcal{R}_{\mathcal{A}}(s')$ and $s' = s''$ follows from ESP. ∎

For complete proofs of the results which have been stated in this subsection, the reader is referred to [19] where partial 2-structures are by-passed.

As an illustration, let us consider the elementary net system and the case graph given in Fig. 2. In Fig. 3 are displayed some of the non trivial regions of

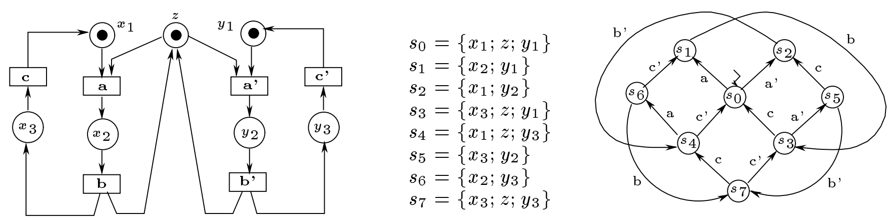


Fig. 2. an elementary net system and its case graph

this automaton. The missing items can be obtained by symmetry. Each drawing

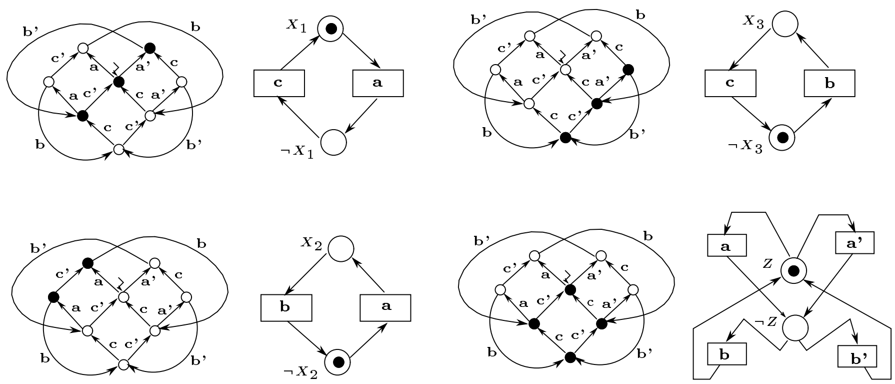


Fig. 3. some regions of the case graph of the elementary net system of Fig. 2 and their associated atomic net systems

represents a region R consisting of black states. The flow relations for the region R and for its complement $\neg R = S \setminus R$ are also represented pictorially; finally one token indicates which of these complementary regions contains the initial state.

We end up with the elementary net system of Fig. 4, which is the original net of Fig. 2 enriched with additional places (indicated by dashed lines) but with unchanged behaviour. The original net system is embedded into its saturated

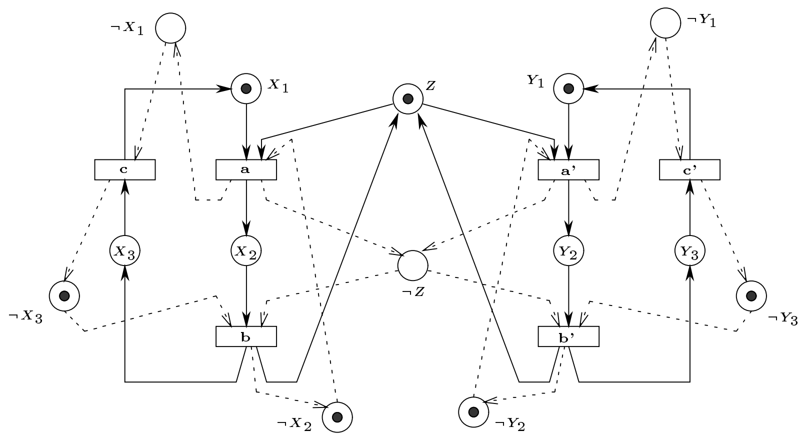


Fig. 4. the embedding of the elementary net system of Fig. 2 into its double dual

version by the map that sends a place x to its *extension* in the state graph i.e. the set of markings $\{M \in S | \; x \in M\}$.

3 The Synthesis of Elementary Net Systems

All automata considered in this section are assumed to be pre-elementary, i.e. simple, reachable and reduced. The synthesis problem of elementary net systems [19] is as follows:

> *Given a finite automaton $\mathcal{A} = (S, E, T, s_0)$, decide whether $\mathcal{A} \cong \mathcal{N}^*$ for some elementary net system $\mathcal{N}$ with the same set of events E, and if so, construct $\mathcal{N}$.*

Since the set $\mathcal{R}_\mathcal{A}$ of all the regions of $\mathcal{A}$ is finite, we already know from Prop. 2.15 that this problem can be decided in exponential time by simultaneously exploring $\mathcal{R}_\mathcal{A}$, for checking satisfaction of the separation properties ESP and ESSP, and constructing $\mathcal{N} = \mathcal{A}^*$. The aim of this section is to improve on this brute force solution. We review first Desel and Reisig's study of admissible sets of regions and their techniques for eliminating redundant regions. Next we account for Bernardinello's results on the synthesis of state machine decomposable net systems, based on the crucial remark that the minimal regions of an automaton form an admissible set, and for subsequent work by Cortadella et al. on the realization of automata by elementary nets up to some quotient of automata. We finally report the results obtained on the complexity of the synthesis problem in [25, 3].

3.1 Admissible sets of regions

In an elementary net system $\mathcal{N} = (P, E, F, M_0)$, each condition $p \in P$ determines an atomic subnet system of $\mathcal{N}$, let $\mathcal{N}_p = (\{p\}, E, F_p, M_{0,p})$ where F_p is the restriction of F and $M_{0,p}(p) = M_0(p)$. If we do not care about the isolated events in $\mathcal{N}_p$, these atomic subnet systems are elementary and $\mathcal{N}$ is just their sum $\sum_{p\in P} \mathcal{N}_p$, where nets are glued together on events $e \in E$. This decomposition may be used to isolate the contribution of each condition $p \in P$ to the global structure of the sequential case graph $\mathcal{N}^*$. This automaton may be seen as a deterministic recognizer of finite sequences, in which every state (i.e. case) is accepting. An automaton of this type is characterized up to isomorphism by the language $\mathcal{L}$ it accepts plus the equivalence $\equiv$ on $\mathcal{L}$ which identifies these sequences that lead to a common (accepting) state. Now in the case of $\mathcal{N}^*$, $\mathcal{L}$ and $\equiv$ are the intersections for p ranging over P of the respective languages and equivalences characteristic of $\mathcal{N}_p^*$: $\mathcal{L} = \cap_{p\in P}\mathcal{L}_p$ and $\equiv = \cap_{p\in P} \equiv_p$. Thus the role of each condition p is twofold: on the one hand, p cuts off sequences $u \cdot e$ such that $u \in \mathcal{L}$ but $u \cdot e \notin \mathcal{L}_p$, and on the other hand p separates pairs of words $u, v \in \mathcal{L}$ such that $u \not\equiv_p v$.

Returning to the synthesis problem, let us now clarify the relationship between automata and atomic net systems. Let $\mathcal{A} = (S, E, T, s_0)$ be a finite deterministic automaton, with language $\mathcal{L}$ and equivalence $\equiv$, and let $\mathcal{N}_p = (\{p\}, E, F_p, M_{0,p})$ be an atomic net system, inducing a dual automaton $\mathcal{N}_p^*$ with language $\mathcal{L}_p$ and equivalence $\equiv_p$. The automaton $\mathcal{N}_p^*$ has two states, $\emptyset$ and $\{p\}$, one of which is $M_{0,p}$, and it has transitions $\emptyset \xrightarrow{e} \{p\}$ if $F_p(e,p)$, $\{p\} \xrightarrow{e} \emptyset$ if $F_p(p,e)$, and otherwise $\emptyset \xrightarrow{e} \emptyset$ and $\{p\} \xrightarrow{e} \{p\}$. Suppose $\mathcal{L} \subseteq \mathcal{L}_p$ and $\equiv \subseteq \equiv_p$. Let R_p be the subset of states $s \in S$ such that $s_0 \xrightarrow{u} s$ in $\mathcal{A}$ and $M_{0,p} \xrightarrow{u} \{p\}$ in $\mathcal{N}_p^*$ for some sequence of events $u \in E^*$. Then R_p is a region of $\mathcal{A}$, $s_0 \in R_p \Leftrightarrow M_{0,p} = \{p\}$, and for every $e \in E$: $R_p{}^\bullet e \Leftrightarrow F_p(p,e)$ and $e^\bullet R_p \Leftrightarrow F_p(e,p)$. Conversely, for any region R_p of $\mathcal{A}$, the elementary net system $\mathcal{N}_p$ defined by the above relations induces a dual automaton $\mathcal{N}_p^*$ such that $\mathcal{L} \subseteq \mathcal{L}_p$ and $\equiv \subseteq \equiv_p$. Moreover, R_p separates two distinct states s' and s'' such that $s_0 \xrightarrow{u} s'$ and $s_0 \xrightarrow{v} s''$ in $\mathcal{A}$ if and only if $u \not\equiv_p v$, and R_p separates a state s such that $s_0 \xrightarrow{u} s$ from an event e such that not$(s \xrightarrow{e})$ if and only if $u \cdot e \notin \mathcal{L}_p$. Therefore, given a net system $\mathcal{N} = (P, E, F, M_0) = \sum_{p\in P} \mathcal{N}_p$, the dual automaton $\mathcal{N}^*$ is isomorphic to the automaton $\mathcal{A}$ if and only if $\mathcal{L} = \cap_{p\in P}\mathcal{L}_p$ and $\equiv = \cap_{p\in P} \equiv_p$, if and only if for all $p \in P$, $\mathcal{N}_p$ is an atomic net system defined from some corresponding region R_p in $\mathcal{A}$ and the following properties are satisfied:
SSP': $\forall u, v \in \mathcal{L} \quad u \not\equiv v \Rightarrow \exists p \in P \quad u \not\equiv_p v$,
ESSP': $\forall u \in \mathcal{L} \quad \forall e \in E \quad u \cdot e \notin \mathcal{L} \Rightarrow \exists p \in P \quad u \cdot e \notin \mathcal{L}_p$,
if and only if the family of regions $\{R_p | \; p \in P\}$ is admissible according to the following definition.

Definition 3.1 *Given an automaton $\mathcal{A} = (S, E, T, s_0)$, a subset of regions $\{R_p| \; p \in P\} \subseteq \mathcal{R}_\mathcal{A}$ is admissible if and only if it includes witnesses for the satisfaction of every instance of the following separation problems where $e \in E$ and $s, s', s'' \in S$ are such that $s' \neq s''$ and not$(s \xrightarrow{e})$:*

$\text{SSP}(s', s'')$: $\exists p \in P \quad s' \in R_p \Leftrightarrow s'' \notin R_p$,
$\text{ESSP}(s, e)$: $\exists p \in P \ (R_p{}^\bullet e \ \wedge \ s \notin R_p) \ \vee \ (e^\bullet R_p \ \wedge \ s \in R_p)$.

It is easily seen that problem $\text{SSP}(s', s'')$ cannot be solved positively in a non-deterministic automaton $\mathcal{A}$ where $s \xrightarrow{e} s'$ and $s \xrightarrow{e} s''$ for $s' \neq s''$. One rediscovers in this way a basic result established in [19].

Theorem 3.2 *An automaton $\mathcal{A} = (S, E, T, s_0)$ is isomorphic to $\mathcal{N}^*$ for $\mathcal{N} = (P, E, F, M_0)$ if and only if for every $p \in P$, the atomic subnet system $\mathcal{N}_p$ of $\mathcal{N}$ may be defined from some corresponding region R_p of $\mathcal{A}$, and the set of regions $\{R_p | \ p \in P\}$ is admissible.*

In view of Def. 3.1 and Theo. 3.2, the synthesis problem for $\mathcal{A} = (S, E, T, s_0)$ may be solved by considering at most $|S| \times (|S| + |E|)$ regions of $\mathcal{A}$. Nevertheless, this does not indicate how to select these regions from $\mathcal{R}_\mathcal{A}$. The purpose is to construct a subset of regions $\mathcal{R} \subseteq \mathcal{R}_\mathcal{A}$ as small as possible such that $\mathcal{R}$ is admissible if and only if the whole set of regions $\mathcal{R}_\mathcal{A}$ is admissible. Some structural rules are proposed in [19] for the stepwise elimination of redundant regions, starting from $\mathcal{R}_\mathcal{A}$.

Definition 3.3 *Let $\mathcal{R} \subset \mathcal{R}_\mathcal{A}$ be a set of regions. A region $R \in \mathcal{R}$ is redundant in $\mathcal{R}$ if the following assertions are equivalent:* (i) *$\mathcal{R}$ is admissible* (ii) *$\mathcal{R} \setminus \{R\}$ is admissible.*

Proposition 3.4 *Let $\mathcal{A} = (S, E, T, s_0)$ and $R \in \mathcal{R} \subset \mathcal{R}_\mathcal{A}$. In each of the following cases R is redundant in $\mathcal{R}$.*

1. $S \setminus R \in \mathcal{R}$,
2. $\exists R_1, R_2, R_3, R_4 \in \mathcal{R} \quad R = R_1 \cap R_2 \ \wedge \ S \setminus R = R_3 \cap R_4$,
3. $\exists R_1, R_2, R_3, R_4 \in \mathcal{R} \quad R = R_1 \cup R_2 \ \wedge \ S \setminus R = R_3 \cup R_4$,
4. $\exists R_1, R_2 \in \mathcal{R} \quad R = R_1 \cap R_2 \quad \wedge \quad \forall s \in R \quad \forall e \in E \quad \forall s' \in S \setminus R \quad s \xrightarrow{e} s' \Rightarrow s' \notin R_1 \cup R_2$.

Once a reduced set of regions $\mathcal{R}$ has been obtained from $\mathcal{R}_\mathcal{A}$, one can check directly from Def. 3.1 whether it is admissible, proving that $\mathcal{A}$ is elementary, and then extract from $\mathcal{R}$ a minimal subset $\{R_p | \ p \in P\}$ such that $\mathcal{A} \cong (\sum_{p \in P} \mathcal{N}_p)^*$. It is worth noting that there exists in general no least admissible set of regions. This fact is illustrated in Fig. 5 by the so-called “four seasons” example reproduced from [19]. The “four seasons” automaton may be realized by two minimal subnet systems of the dual saturated net system: one has four conditions and is contact-free while the other one has three conditions but is not contact-free.

Definition 3.5 *An elementary net system $\mathcal{N} = (P, E, F, M_0)$ is contact-free if ${}^\bullet e \subset M \Rightarrow M \cap e^\bullet = \emptyset$ for every event e and for every reachable case M.*

Thus, the subclass of elementary net systems which are contact-free and reduced coincides with the subclass of the reduced and one-safe Petri nets. Now, every saturated net system $\mathcal{N} = (P, E, F, M_0)$ is contact-free: every condition $p \in P$ induces two complementary regions R_p and $\overline{R_p}$ in $\mathcal{N}^*$, and since $\mathcal{N} \cong \mathcal{N}^{**}$ there

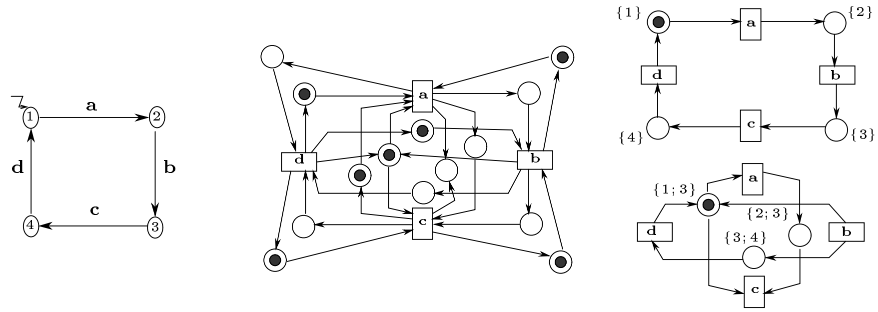


Fig. 5. the four seasons example: the automaton (on the left), the saturated net system (on the middle) and two elementary net systems corresponding to minimal sets of regions (on the right)

should exist some condition $\overline{p} \in P$ such that $R_{\overline{p}} = \overline{R_p}$. Therefore, every elementary automaton may be realized by a one-safe Petri net. The following adaptation of Theo. 3.2, based on the use of complementary regions, is established in [19]

Proposition 3.6 *An automaton $\mathcal{A} = (S, E, T, s_0)$ is isomorphic to $\mathcal{N}^*$ for a contact-free net system $\mathcal{N} = (P, E, F, M_0) = \sum_{p \in P} \mathcal{N}_p$ if and only if every atomic subnet system $\mathcal{N}_p$ of $\mathcal{N}$ may be defined from a corresponding region $R_p \in \mathcal{R}_\mathcal{A}$ and the following properties of separation are satisfied:*
SSP : $\quad \forall s, s' \in S \quad s \neq s' \Rightarrow \exists p \in P \quad s \in R_p \Leftrightarrow s' \notin R_p$
ESSP$^\sharp$: $\quad \forall e \in E \quad \forall s \in S \quad not(s \stackrel{e}{\to}) \quad \Rightarrow \exists p \in P \quad R_p{}^\bullet e \ \wedge \ s \notin R_p$.

3.2 Minimal Regions

Among the admissible sets of regions of an elementary automaton, the set of minimal regions plays a distinguished role because it leads naturally, as shown in [11], to a state machine decomposable (and hence contact-free) net system realizing the automaton.

Definition 3.7 *An elementary net system $\mathcal{N} = (P, E, F, M_0)$ is a state machine if its initial case is a singleton and every event has one precondition and one postcondition. A state machine component of $\mathcal{N} = (P, E, F, M_0)$ is a state machine $\mathcal{N}' = (P', E', F', M_0')$ such that $P' \subseteq P$, $E' = \{e \in E | ({}^\bullet e \cup e^\bullet) \cap P' \neq \emptyset\}$, $F' = F \cap (E' \times P' \cup P' \times E')$, and $M_0' = M_0 \cap P'$. A state machine decomposition of $\mathcal{N} = (P, E, F, M_0)$ is a family of state machines, let $\mathcal{N}_i = (P_i, E_i, F_i, M_{0,i})$, such that $P = \cup_i P_i$, $E = \cup_i E_i$, $F = \cup_i F_i$, and $M_0 = \cup_i M_{0,i}$.*

A state machine is nothing else than a reachable automaton, as can be seen from Fig. 6 where the elementary net system given in Fig. 2 is decomposed into three state machine components. The respective state machine components model sequential processes which are synchronized on their common events. In this example, the synchronization prevents the leftmost and rightmost processes from

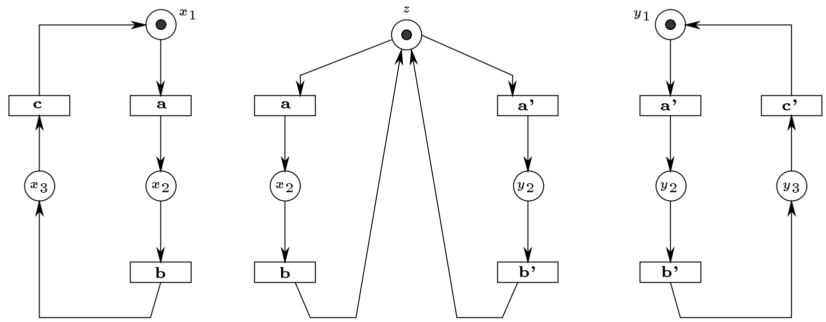


Fig. 6. three state machine components of the net system of Fig. 2

entering simultaneously the critical section figured by the mutually exclusive conditions x_2 and y_2. Each state machine component $\mathcal{N}_i$ of a net system $\mathcal{N} = \sum_i \mathcal{N}_i$ may be seen as a sequential observer of $\mathcal{N}^*$, projecting cases of $\mathcal{N}$ on observable conditions $p \in P_i$. By definition of state machine components, each case of $\mathcal{N}$ projects to one and exactly one condition $p \in P_i$, hence each case of $\mathcal{N}$ belongs to exactly one region R_p of $\mathcal{N}^*$ such that $p \in P_i$.

Proposition 3.8 *Every state machine component $\mathcal{N}_i = (P_i, E_i, F_i, M_{0,i})$ of an elementary net system $\mathcal{N} = \sum_i \mathcal{N}_i$ determines a regional partition $\{R_p | \ p \in P_i\}$ of the sequential case graph $\mathcal{N}^*$. Conversely, every regional partition $\{R_p | \ p \in P\}$ of $\mathcal{N}^*$ determines a state machine component of the saturated net system $\mathcal{N}^{**}$.*

Returning to the example, the regional partitions of $\mathcal{N}^*$ (Fig. 3) which determine the three state machine components shown in Fig. 6 are respectively $\{X_1, X_2, X_3\}$, $\{X_2, Z, Y_2\}$, and $\{Y_1, Y_2, Y_3\}$ where:

$$\begin{aligned}
Z &= \{s_0; s_3; s_4; s_7\} \\
X_1 &= \{s_0; s_2; s_4\} \\
X_2 &= \{s_1; s_6\} \\
X_3 &= \{s_3; s_5; s_7\} \\
Y_1 &= \{s_0; s_1; s_3\} \\
Y_2 &= \{s_2; s_5\} \\
Y_3 &= \{s_4; s_6; s_7\}
\end{aligned}$$

It may be observed that all these regions are minimal w.r.t. set inclusion in $\mathcal{R}_{\mathcal{N}^*}$. The particular interest of minimal regions for the net system realization of elementary automata is shown by the following proposition and corollaries.

Proposition 3.9 *Given an automaton $\mathcal{A} = (S, E, T, s_0)$, the following properties are satisfied by the set $\mathcal{R}_\mathcal{A}$ of regions of $\mathcal{A}$:*

1. *If R_1 and R_2 are disjoint regions then $R_1 \cup R_2$ is a region with*

$$\begin{aligned}
{}^\bullet(R_1 \cup R_2) &= ({}^\bullet R_1 \cup {}^\bullet R_2) \setminus (({}^\bullet R_1 \cap R_2{}^\bullet) \cup ({}^\bullet R_2 \cap R_1{}^\bullet)) \\
(R_1 \cup R_2)^\bullet &= (R_1{}^\bullet \cup R_2{}^\bullet) \setminus (({}^\bullet R_1 \cap R_2{}^\bullet) \cup ({}^\bullet R_2 \cap R_1{}^\bullet)).
\end{aligned}$$

2. *If R and R' are regions and $R' \subseteq R$ then $R \setminus R'$ is a region. If moreover R' is minimal then $e^\bullet(R \setminus R')$ for every event $e \in R'^\bullet$ which is not incident to R (i.e. such that $e \notin {}^\bullet R \cup R^\bullet$).*
3. *If R is a region and $s \in R$, then $s \in R'$ for some minimal region $R' \subseteq R$.*
4. *If R is a region and e an event such that $R^\bullet e$, then $R'^\bullet e$ for some minimal region $R' \subseteq R$; symmetrically if e is an event such that $e^\bullet R$, then $e^\bullet R'$ for some minimal region $R' \subseteq R$.*
5. *Every region is a disjoint union of minimal regions.*

Corollary 3.10 *A pre-elementary automaton is elementary if and only if its set of minimal regions is admissible.*

It may be further observed that the set of minimal regions of a pre-elementary automaton $\mathcal{A}$ is admissible w.r.t. the separation properties SSP and ESSP if and only if it is admissible w.r.t. the separation properties SSP and ESSP$^\sharp$. In fact, let $\{R_1, \ldots, R_n\}$ be any partition of the set of states of $\mathcal{A}$ into minimal regions, then each instance of the problem ESSP(s, e) solved by a region R_i such that $e^\bullet R_i$ and $s \in R_i$ can also be solved by a region R_j such that $R_j{}^\bullet e$ and $s \notin R_j$. Since the set of all partitions of the set of states of $\mathcal{A}$ into minimal regions induces a state machine decomposition of the net system $\sum_p \mathcal{N}_p$ defined from the set of all minimal regions R_p of $\mathcal{A}$, one deduces also the following.

Corollary 3.11 *Every elementary automaton may be realized by a state machine decomposable (and hence contact-free) elementary net system.*

An algorithm based on minimal regions has been proposed in [14] for a variant problem of realization of automata by net systems which may be stated as follows.

> *Given a pre-elementary automaton $\mathcal{A}$, decide whether exists and construct a (minimal) elementary net system $\mathcal{N}$ such that $\mathcal{N}^* \cong \mathcal{A}'$ for some quotient $\mathcal{A}'$ of $\mathcal{A}$.*

We recall that $\mathcal{A}' = (S', E, T', s'_0)$ is a quotient of $\mathcal{A} = (S, E, T, s_0)$ if $s_1 \xrightarrow{e} s_2$ in $\mathcal{A}$ if and only if $\sigma(s_1) \xrightarrow{e} \sigma(s_2)$ in $\mathcal{A}'$ for some surjective map $\sigma : S \to S'$ such that $s'_0 = \sigma(s_0)$. This problem is similar to the original synthesis problem, up to the fact that the states separation property SSP is ignored. Now the events-states separation property ESSP$^\sharp$ is valid in $\mathcal{A}$ if and only if for every event e the set of states $\{s \in S |\ s \xrightarrow{e}\}$ coincides with the intersection of the minimal regions R such that $R^\bullet e$. The algorithm starts from the sets $\{s \in S |\ s \xrightarrow{e}\}$ and increases them into minimal regions, which are generated until the validity of ESSP$^\sharp$ can be decided upon. The net $\mathcal{N}$ is then constructed from a minimal set of minimal regions admissible with respect to ESSP$^\sharp$. A variant form of this algorithm has been integrated to a software tool for the synthesis of asynchronous circuits [15].

It should be noted that the problem of realizing automata by nets up to a quotient differs significantly from the problem of realizing automata by nets up to behavioural equivalence (equality of the accepted languages). In order to make the difference visible, let us focus on finite and deterministic automata.

In this context, behavioural equivalence coincides with bisimilarity. Given a finite deterministic automaton $\mathcal{A}$, with language $\mathcal{L}$ and characteristic equivalence $\equiv$ on $\mathcal{L}$, the problem of realizing $\mathcal{A}$ up to behavioural equivalence consists in constructing an elementary net system $\mathcal{N}$ such that $\mathcal{N}^*$ recognizes $\mathcal{L}$. For the problem of realizing $\mathcal{A}$ up to a quotient, it is set as a further requirement that any two equivalent sequences in $\mathcal{L}$ lead to the same case when they are fired from the initial case of $\mathcal{N}$. In orther words, it is asked that $\equiv \subseteq \equiv_{\mathcal{N}^*}$. The reason why this constraint makes a notable difference is that the elementary automata are not closed under quotient. This counterfact is illustrated in Fig. 7: the automaton shown on the middle is isomorphic to the case graph of the net displayed on the left, but its minimized version shown on the right is not elementary (any region R such that $R^\bullet c$ must include state 3, hence the problem ESSP$(3, c)$ cannot be solved).

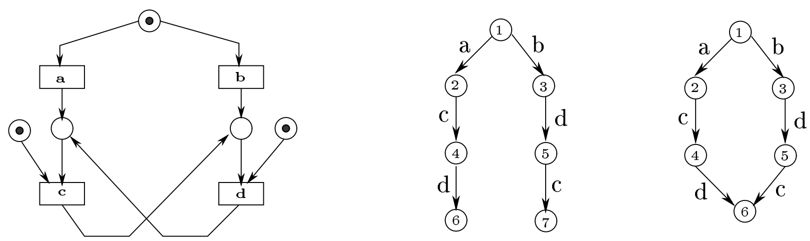


Fig. 7. elementary automata are not closed under quotient

3.3 Complexity Results

Hiraishi proved in [25] that the separation problems SSP(s, s') and ESSP$^\sharp(s, e)$ are NP-complete in the respective data $(\mathcal{A}, s, s')$ and $(\mathcal{A}, s, e)$. Since regions in $\mathcal{A}$ are closed under complementation, the problem ESSP(s, e) is also NP-complete. It does not follow therefrom that the synthesis problem for elementary net systems is NP-complete; however this is the case. The synthesis problem is obviously in NP since the total number of instances of separation problems in an automaton $\mathcal{A}$ is quadratic in the size of $\mathcal{A}$, and it can be checked in polynomial time whether a non-deterministically chosen subset of states is a region solving a fixed separation problem. Now a polynomial reduction of 3-SAT to the synthesis problem of elementary net systems was established in [3], showing NP-hardness since 3-SAT is NP-complete (see e.g. [23]). Recall that 3-SAT is the problem whether, given a finite set of boolean clauses over V, with three litterals per clause, there exists some truth assignment for V validating each clause. Each clausal system of this form is associated in [3] with an automaton such that the clausal system is satisfiable if and only if the automaton is elementary if and only if the separation property ESSP$^\sharp$ is valid. Therefore, the synthesis problem for elementary net systems is NP-complete, and so is the problem of realizing automata by nets up to a quotient. The problems of realizing automata by nets up to behavioural equivalence, or up to an unfolding (given $\mathcal{A}$ find $\mathcal{N}$ such that $\mathcal{A}$ is isomorphic to a quotient of $\mathcal{N}^*$) have unknown complexity.

4 Cutset Representation of Finite Graphs

We have seen that the region based synthesis of elementary net systems from initialized partial 2-structures $(X, E, \equiv, x_0)$ is a NP-complete problem. Nevertheless, this problem is trivial when the labelling equivalence is discrete: in that case, the partial 2-structure is essentially a state machine with set of places X; even better, this state machine is equivalent to a net system with $|X|-1$ places, whose case graph is a partial set 2-structure isomorphic to the given partial 2-structure. There exists a large variety of set-theoretic representations for an unlabelled graph (X, E), all of which using at most $|X|-1$ tokens. These representations, based on cuts and cutsets, may be computed by linear algebraic methods which are quite standard in applied graph theory. The purpose of this section is to review these methods, and thereby shed light on regions in two respects. First, we examine the close relationship between regions and cuts (this analogy was first pointed out to us by T. Murata). Second, we indicate the obstacles to using linear algebraic methods for the region based representation of labelled graphs. On account of this analysis, a variant definition of regions is proposed in the next section.

4.1 Cuts and Cutsets

Let $G = (X, E)$ be a finite, connected and simple directed graph with set of nodes $X = \{x_1, \ldots, x_n\}$ and set of 2-edges $E = \{e_1, \ldots, e_m\}$. So, G is free of loops multiple arcs, although a 2-edge $e = (x_l, x_k)$ may have an inverse $e^{-1} = (x_k, x_l)$ in E. A cutset of G is a *minimal* set of 2-edges whose removal increases the number of connected components by one. A cut of G is a cutset or an edge disjoint union of cutsets. Since G is connected, every cut or cutset $C \subseteq E$ determines two complementary subsets of nodes p and $X \setminus p$, both non empty, such that for every 2-edge $e = (x_k, x_l)$, $e \in C$ if and only if $x_k \in p \Leftrightarrow x_l \notin p$. Conversely, every non trivial subset $p \subseteq X$ determines a cut between p and $X \setminus p$, which is a cutset when both p and $X \setminus p$ are connected. An *orientation* of the cut C results from the choice of one of the two complementary subsets of nodes determined by the cut, let p. An oriented cut C may be coded by a vector $C \in \mathbb{R}^m$ such that for every 2-edge $e_i = (x_k, x_l)$, $C(i) = 1$ if $x_k \notin p$ and $x_l \in p$, $C(i) = -1$ if $x_k \in p$ and $x_l \notin p$, and $C(i) = 0$ if $x_k \in p \Leftrightarrow x_l \in p$.

Let $X = \{x_1, \ldots, x_n\}$ and $E = \{e_1, \ldots, e_m\}$. We will address the problem of constructing a variety of sets of properties $\{p_1, \ldots, p_{n-1}\}$ where $p_i \subseteq X$ such that the partial 2-structure $(\{x^* | \ x \in X\}, \{e^* | \ e \in E\}, \equiv_\delta)$ where $x^* = \{p_i | \ x \in p_i\}$, $(x_k, x_l)^* = (x_k^*, x_l^*)$, and $\delta(x_k^*, x_l^*) = (x_k^* \setminus x_l^*, x_l^* \setminus x_k^*)$ is isomorphic to G viewed as a partial 2-structure: $G = (X, E, id_E)$. Each family of tokens $\{p_1, \ldots, p_{n-1}\}$

will determine a corresponding set of (oriented) cuts $\{C_1, \ldots C_{n-1}\}$ which are linearly independent as vectors $C_i \in I\!R^m$.

The interesting fact here is that one can easily construct linear bases of cuts, given as sets of fundamental cutsets of G with respect to arbitrary spanning trees. Recall that a spanning tree is a set of edges $U \subseteq E$, free of cycles and connecting X. The fundamental cutsets w.r.t. U are the cuts which include exactly one branch of U. Each branch of U determines two connected components of U (and thus of G), with set of nodes p and $X \setminus p$, such that every other branch of U is internal either to p or to $X \setminus p$. The fundamental cutsets w.r.t. U may be computed by classical methods of linear algebra. These methods are recalled below, following the notations of [16].

4.2 Computing Cutsets

The graph $G = (X, E)$ is characterized up to isomorphism by its incidence matrix. We recall that this matrix $A = [a_{i,j}]$ is an $n \times m$ matrix with entries in $\{-1, 0, 1\}$, with $a_{i,j} = 0$ if edge e_j is not incident to node x_i, $a_{i,j} = 1$ if x_i is the source of e_j, and $a_{i,j} = -1$ if x_i is the target of e_j. Since every column contains exactly two non zero entries (1 and -1) every row can be computed from the other rows, and the matrix A has the same rank as the matrix A_1 obtained by erasing its last row. Let $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ where A_1 is an $(n-1) \times m$ matrix and A_2 is an $1 \times m$ matrix. Actually A_1 and A have rank $n-1$. Assume w.l.o.g. that the $(n-1)$ branches of the spanning tree U are the edges $e'_j = e_{j+(m-n+1)}$ for $j \in \{1, \ldots, n-1\}$. Then $A_1 = \begin{bmatrix} A_{11} \; A_{12} \end{bmatrix}$ where A_{12} is the $(n-1) \times (n-1)$ matrix corresponding to the edges of the tree (the *branches*) and A_{11} is the $(n-1) \times (m-n+1)$ matrix corresponding to the other edges (the *chords*).

The fundamental cutset C_i of G determined by the edge e'_i of the spanning tree is given by the i^{th} row of the *fundamental cutset matrix* $Q_f = A_{12}^{-1} \cdot A_1$. This $(n-1) \times m$ matrix has the form $\begin{bmatrix} Q_{f11} \; I_{n-1} \end{bmatrix}$ where I_k is the identity matrix of rank k. The i^{th} row of Q_f associated with the fundamental cutset C_i is an m vector with entries in $\{-1, 0, 1\}$. Let p_i and $X \setminus p_i$ be the two connected components of G separated by C_i, such that e'_i has its source in $X \setminus p_i$ and its target in p_i. Then for every $j \in \{1, \ldots, m\}$, $C_i(j) = 0$ if e_j is not in C_i, $C_i(j) = 1$ if e_j is oriented from $X \setminus p_i$ to p_i and $C_i(j) = -1$ if e_j is oriented from p_i to $X \setminus p_i$. A complete example is shown in Fig. 8.

It is worth noting that the matrix A_{12}^{-1} can be computed directly from G without inverting matrix A_{12}, for it coincides with the path matrix $P = [p_{i,j}]$ defined as follows. For each $j \in \{1, \ldots, n-1\}$, let Π_j be the unique chain (in the tree U) connecting x_j and the reference node x_n; then for $1 \leq i, j \leq n-1$, let $p_{i,j} = 0$ if e'_i does not belong to Π_j, $p_{i,j} = 1$ if e'_i belongs to Π_j and is oriented towards the reference node x_n, and $p_{i,j} = -1$ if e'_i belongs to Π_j and is oriented towards node x_j.

4.3 Cutset Representation of Graphs

The nodes of G may be coded injectively by $\{0,1\}$ vectors according to their membership to the properties p_j determined by the cuts C_j, resulting in an $n \times (n-1)$ matrix $S_a = [s_{i,j}]$, called the *state matrix*, such that $s_{i,j} = 1$ if $x_i \in p_j$, and $s_{i,j} = 0$ if $x_i \notin p_j$. Let $S_a = [X_1 \ \cdots \ X_n]^t$, where the X_i are column vectors. The set $\{X_i^t | \ i \leq n\}$ of rows of S_a, representing nodes x_i, together with the set $\{C_i | \ i < n\}$ of rows of Q_f, representing fundamental cutsets, provide a representation of G. These data are also sufficient for retrieving the spanning tree. Actually, there is exactly one way to assemble the row vectors C_i into a matrix of the form $Q_f = \left[Q_{f11} \ I_{n-1} \right]$; and an ordered pair of vectors (X_k, X_l) represents an edge $e_j = (x_k, x_l)$ if and only if $X_l - X_k = Q_f(\cdot, j)$.

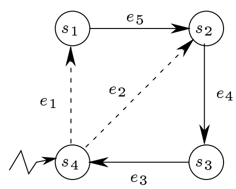
4.4 Variant Representations

A variant representation of G is given by the pair of matrices P and Q_{f11}. As a matter of fact, the reduced incidence matrix $A_1 = \left[A_{11} \ A_{12} \right]$ may be computed by $A_{12} = P^{-1}$ and $A_{11} = P^{-1} \cdot Q_{f11}$. The path matrix P can in turn be reconstructed from X_n and the reduced state matrix $S = [X_1, \ldots, X_{n-1}]^t$. Actually, for every $j < n$, $X_n = X_j + P_j$ where P_j is the j^{th} column of P (coding the chain Π_j connecting x_j and x_n), hence the path matrix P and the reduced state matrix S are connected by the identity $S^t = \overbrace{[X_n, \ldots, X_n]}^{(n-1) \text{ times}}$ - P. In particular, $S = -P^t$ if all edges e'_j of U are oriented away from the reference node x_n.

4.5 Fundamental Cycles

It has some importance for the sequel to note that the information provided by the fundamental matrix Q_f is exactly the same as the information provided by the *fundamental cycle matrix* [1] B_f, defined as follows from the spanning tree U. Each chord (i.e. edge in $E \setminus U$) determines a cycle in G, consisting of this edge and the unique chain in U that connects its endpoints. This cycle may be represented by an m vector B_i with entries in $\{-1, 0, 1\}$ as follows: $B_i(j) = 0$ if e_j is not contained in the cycle, else $B_i(j) = 1$ or -1 depending on whether the orientation of e_j agrees with, or is opposite to the orientation of e_i within this cycle. The fundamental cycle matrix B_f is the $(m - n + 1) \times m$ matrix defined by $B_f(i,j) = B_i(j)$. This matrix is of the form $B_f = \left[I_{m-n+1} \ B_{f12} \right]$, where $B_{f12} = -Q_{f11}^t$ (in particular, a branch belongs to the fundamental cycle defined by a chord if and only if the chord belongs to the fundamental cutset defined by the branch). Therefore, $B_f \cdot Q_f^t = 0$, and the vector spaces $\mathcal{V}_B$ and $\mathcal{V}_Q$ respectively generated over $\mathbb{R}$ by the fundamental cycles (rows of B_f) and by the fundamental cutsets (rows of Q_f) are orthogonal. These two vector spaces, which do not depend on the choice of the spanning tree, are indeed orthogonal complements of $\mathbb{R}^m$.

[1] called *fundamental circuit matrix* in [16]



Incidence Matrix:

$A =$

	e_1	e_2	e_3	e_4	e_5
s_1	−1	0	0	0	1
s_2	0	−1	0	1	−1
s_3	0	0	1	−1	0
s_4	1	1	−1	0	0

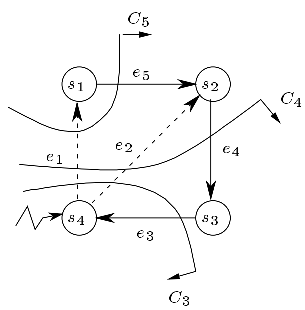
$$= \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline \multicolumn{2}{c}{A_2} \end{array} \right]$$

$n = 4$ vertices
$m = 5$ edges
$r = n - 1 = 3$ rank

Path Matrix:

$P = A_{12}^{-1} =$

	s_1	s_2	s_3
e_3	1	0	0
e_4	1	1	0
e_5	1	1	1



Fundamental Cutset Matrix:

$Q_f =$

	e_1	e_2	e_3	e_4	e_5
C_3	−1	−1	1	0	0
C_4	−1	−1	0	1	0
C_5	−1	0	0	0	1

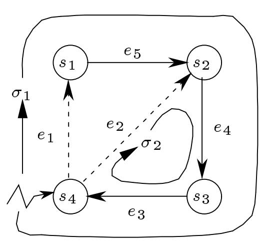
$$= \left[Q_{f11} \ I_r \right]$$

$$\boxed{Q_f = P \cdot A_1}$$

State Matrix:

$S_a =$

	C_3	C_4	C_5
s_1	0	0	0
s_2	1	0	0
s_3	1	1	0
s_4	1	1	1



Fundamental Cycle Matrix:

$B_f =$

	e_1	e_2	e_3	e_4	e_5
σ_1	1	0	1	1	1
σ_2	0	1	1	1	0

$$= \left[I_{m-r} \ B_{f12} \right]$$

$$\boxed{B_{f12} = -Q_{f11}^t}$$

Fig. 8. fundamental cutsets and cycles

Every non null vector in $\mathcal{V}_B$ with entries -1, 0, and 1 is a sum of fundamental cycles and/or inverses of fundamental cycles, hence it is either a cycle or an edge disjoint union of cycles in vector form. Similarly, every non null vector C in $\mathcal{V}_Q$ with entries -1, 0, and 1 defines a cut $\{e_j | C(j) \neq 0\}$, but C may differ by the sign of its components from the vector which represents this cut (and also from the opposite of this vector). For a counterexample, let $C = (-1, 1)$ where e_1 and e_2 have the same target and distinct sources.

4.6 Back to set 2-Structures

We saw that G may be represented by a set of $\{0,1\}$ vectors expressing the set of properties of its nodes ($x_j \in p_i \Leftrightarrow X_j(i) = 1$), plus the set of the fundamental cutsets which define these properties (the cutset C_i defining p_i is given by the i^{th} row of Q_f). A node x_j is then identified with the set of tokens $x_j^* = \{i | X_j(i) = 1\}$; similarly, an edge $e_j = (x_k, x_l)$ is identified with the ordered pair $e_j^* = (x_k^*, x_l^*)$. We show that the resulting partial 2-structure $G^* = (\{x^* | x \in X\}, \{e^* | e \in E\}, \equiv_\delta)$ is actually isomorphic to the given graph $G = (X, E, id_E)$. It is easily seen that the above representation is injective on nodes, since two different nodes of the spanning tree are always separated by a fundamental cutset. In order to prove that $G^* \cong G$, it suffices therefore to show that $\delta(e_j^*) = \delta(e_l^*)$ entails $e_j = e_l$. We establish a stronger property, namely:

Lemma 4.1 *Let $e_j = (x_k, x_l)$ be an edge of G, then for every pair of nodes x_p and x_q, $\delta(x_k^*, x_l^*) = \delta(x_p^*, x_q^*)$ entails that $x_p = x_k$ and $x_l = x_q$.*

Proof: Assuming the premises, let π be a chain connecting x_p and x_q in the spanning tree U, represented by a vector $\pi \in \{-1, 0, 1\}^m$ by "orienting" the chain from x_p to x_q. Suppose $\pi(j) = -1$, thus the edge e_j is oriented away from x_q and towards x_p in that chain. Let p_j be the property defined by the fundamental cutset which includes e_j, then necessarily $x_p, x_l \in p_j$ and $x_q, x_k \notin p_j$, hence $x_k^* \setminus x_l^* \neq x_p^* \setminus x_q^*$, contradicting our assumptions. Therefore, if we let 1_j denote the vector with a 1 at position j and 0 elsewhere, the vector $\pi - 1_j$ has all entries in $\{-1, 0, 1\}$. Since $Q_f \cdot \pi$ measures variations of properties along π, the assumption $\delta(x_k^*, x_l^*) = \delta(x_p^*, x_q^*)$ reads as $Q_f \cdot \pi = Q_f \cdot 1_j$. Thus the vector $\pi - 1_j$ lies in $\mathcal{V}_B$, and it is either a cycle or a disjoint union of cycles in vector form. Since there is no cycle in U, it follows that $\pi - 1_j$ is a cycle, hence $x_p = x_k$ and $x_q = x_l$ as was to show. ∎

Now, any set $\{p'_1, \ldots, p'_{n-1}\}$ of non trivial subsets of X determines a corresponding 2-structure $G^* = (\{x^* | x \in X\}, \{e^* | e \in E\}, \equiv_\delta)$, defined as above by setting $X_j^* = \{i | x_j \in p'_i\}$ and $(x_k, x_l)^* = (x_k^*, x_l^*)$. For $1 \leq i \leq n-1$, let C'_i denote the cut separating the complementary subsets $X \setminus p'_i$ and p'_i. We will show that $G^* \cong G$ whenever the corresponding vectors $C'_1, \ldots, C'_{n-1}$ are linearly independent. This is for instance the case when $p'_i = \{x_i\}$. Beware of the fact that G^* may be isomorphic to G even though $C'_1, \ldots, C'_{n-1}$ are not linearly independent. For an illustration, let $p'_1 = \{x_2, x_3\}$, $p'_2 = \{x_2, x_4\}$ and $p'_3 = \{x_1, x_3\}$ in

$G = (X, E)$ where $X = \{x_1, x_2, x_3, x_4\}$ and $E = \{e_1, e_2, e_3\}$ with $e_i = (x_1, x_{i+1})$, then $G^* \cong G$ but $C'_2 + C'_3 = 0$. Notice that in this representation of G the vectors e_1^*, e_2^*, and e_3^* are not linearly independent: $e_1^* + 2 \cdot e_2^* + e_3^* = 0$ even though there is no cycle in G.

Assuming that $C'_1, \ldots, C'_{n-1}$ are linearly independent, let us prove that $G = (X, E, id_E)$ and $G^* = (\{x^* |\ x \in X\}, \{e^* |\ e \in E\}, \equiv_\delta)$ are isomorphic partial 2-structures. Let $x_k \neq x_l$ and assume for contradiction $x_k^* = x_l^*$. Let π be the chain in U connecting the vertices x_k and x_l. By construction of the cuts C'_i, $\pi \cdot C'_i = 0$ for every $i \leq n-1$. Since $C'_1, \ldots, C'_{n-1}$ are linearly independent, they span the vector space $\mathcal{V}_Q$ and π is a cycle, thus $x_k = x_l$. It remains to show that $\delta(e_j^*) = \delta(e_i^*)$ entails $e_j = e_i$.

Lemma 4.2 *Let $e_j = (x_k, x_l)$ be an edge of G, then for every pair of nodes x_p and x_q, $\delta(x_k^*, x_l^*) = \delta(x_p^*, x_q^*)$ entails that $x_p = x_k$ and $x_l = x_q$.*

Proof: Let π be a chain connecting x_p and x_q in the spanning tree, represented by a vector $\pi \in \{-1, 0, 1\}^m$ by "orienting" the chain from x_p to x_q. Suppose $\pi(j) = -1$, thus the edge e_j is in π and it is oriented away from x_q and towards x_p in that chain. From the assumption $\delta(x_k^*, x_l^*) = \delta(x_p^*, x_q^*)$ and by construction of the cuts C'_i, it follows that $\pi \cdot C'_i = 1_j \cdot C'_i$ for all $i \leq n-1$, where 1_j denotes the vector with a 1 at position j and 0 elsewhere. Thus $(\pi - 1_j) \cdot C'_i = 0$ for all i, and since $C'_1, \ldots, C'_{n-1}$ form a basis of the vector space $\mathcal{V}_Q$, it follows that $(\pi - 1_j) \cdot C_k = 0$ for all $k \leq n-1$ and in particular for $h = j - (m - n + 1)$. Now the first $m - n + 1$ entries of the vector $\pi - 1_j$ are zeros and the last $n - 1$ entries of C_h are zeros but $C_h(j)$ which is 1. Therefore, $\pi(j) = 1$ and we have reached a contradiction. Thus the vector $\pi - 1_j$ has all entries in $\{-1, 0, 1\}$. Since $(\pi - 1_j) \cdot C'_i = 0$ for all i, the vector $\pi - 1_j$ lies in $\mathcal{V}_B$, and it is either a cycle or a disjoint union of cycles in vector form. Since there is no cycle in U, it follows that $\pi - 1_j$ is a cycle, hence $x_p = x_k$ and $x_q = x_l$ as was to show. ∎

We now give an example (see Fig. 9) showing that the computation of cuts and cutsets cannot lead directly to a net representation of $G = (X, E)$. Let

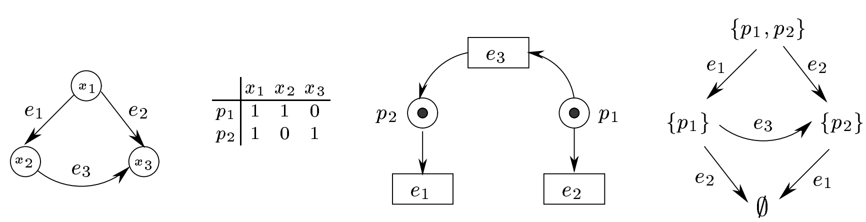


Fig. 9. elementary net system associated with a basis of cuts

$X = \{x_1, x_2, x_3\}$ and $E = \{e_1, e_2, e_3\}$ with $e_1 = (x_1, x_2)$, $e_2 = (x_1, x_3)$ and $e_3 = (x_2, x_3)$. A basis of cuts for G is given by the vectors $C_1 = (0, -1, -1)$

and $C_2 = (-1,0,1)$, inducing respective properties $p_1 = \{x_1, x_2\}$ and $p_2 = \{x_1, x_3\}$ such that $x_1^* = \{p_1, p_2\}$ and $x_2^* = \{p_1\}$ and $x_3^* = \{p_2\}$. Now let $\mathcal{N} = (\{p_1, p_2\}, E, F, x_1^*)$ be the elementary net system such that ${}^\bullet e_1 = x_1^* \setminus x_2^* = \{p_2\}$, $e_1{}^\bullet = x_2^* \setminus x_1^* = \emptyset$, ${}^\bullet e_2 = x_1^* \setminus x_3^* = \{p_1\}$, $e_2{}^\bullet = x_3^* \setminus x_1^* = \emptyset$, and ${}^\bullet e_3 = x_2^* \setminus x_3^* = \{p_1\}$, $e_3{}^\bullet = x_3^* \setminus x_2^* = \{p_2\}$. The case graph of $\mathcal{N}$ is not isomorphic to the initialized partial set 2-structure $(X^*, E^*, \equiv_\delta, x_1^*)$, due to the presence of two additional transitions $x_2^*[e_2 > \emptyset$ and $x_3^*[e_1 > \emptyset$.

4.7 Cuts and Regions

A non trivial region p of a partial 2-structure $(X, E, \equiv)$ always determines and is determined by a cut C of (X, E), which we may therefore call a *regional cut*. If we identify cuts C with the corresponding vectors $C : E \to \{-1, 0, 1\}$, then a cut is regional if and only if it is compatible with the equivalence $\equiv$ in the sense that $e \equiv e' \Rightarrow C(e) = C(e')$ for all $e, e' \in E$. In particular, all cuts are regional when $\equiv$ is the identity relation.

Let us adapt the above to transition systems. We saw that a non trivial region R of (S, E, T) is always determined from a corresponding map $\eta : E \to \{-1, 0, 1\}$ such that $\eta(e) = -1$ if $R^\bullet e$, $\eta(e) = 1$ if $e^\bullet R$, and $\eta(e) = 0$ otherwise ($\eta(e) = \chi_R(s') - \chi_R(s)$ when $s \xrightarrow{e} s'$). Let $\ell : T \to E$ be the labelling function such that $\ell(s \xrightarrow{e} s') = e$. Then a map $\eta : E \to \{-1, 0, 1\}$ determines a region in a pre-elementary transition system (S, E, T) if and only if the map $C : T \to \{-1, 0, 1\}$ defined by $C(t) = \eta(\ell(t))$ is a cut of the underlying graph (S, T).

On that basis, let us try to point out the obstacles to a polynomial synthesis of elementary net systems. On one hand, one can compute in polynomial time a linear basis for the real vector space $\mathcal{V}_Q$ which contains all cuts, but also elements which are not cuts even though all their entries are in $\{-1, 0, 1\}$. On the other hand, abstract regions are quotients of cuts, but it is not possible to derive a basis of abstract regions from a basis of cuts since abstract regions are not closed under summation. A well known recipe for getting rid of the first problem is to replace the real field $\mathbb{R}$ by the boolean field $\mathbf{2}$ in the definition of the vector space $\mathcal{V}_Q$. The second problem will then be overcome by a slight adaptation of the definition of regions, amounting to embed the elementary nets in a wider class of one-safe nets which have actually a polynomial time synthesis.

5 Flip Flop Nets and their Synthesis

We examine in this section extended regions in automata, defined as sets of states R such that all transitions with the same label are incident jointly to R, possibly inwards for some transitions and outwards for the others, or are not incident to R. A class of one-safe nets based on these regions, called *flip flop nets* and extending elementary nets, has been defined in [39]. We show that the synthesis problem for flip flop nets may be solved in polynomial time, following techniques of linear algebra based on cutsets. Pairs of complementary regions in an automaton may be identified with vectors $\eta : E \to \mathbf{2}$; these maps form a vector space over **2**, a basis of which is easily derived from any set of fundamental cutsets of the (undirected) graph underlying the automaton.

5.1 The Vector Space of Cuts

Let $\mathcal{A} = (S, E, T, s_n)$ be a loop-free, reachable and reduced finite automaton (not necessarily simple), with $S = \{s_1, \ldots, s_n\}$ and $T = \{t_1, \ldots, t_m\}$. Let $\partial^0(t) = s$, $\partial^1(t) = s'$ and $\ell(t) = e$ denote the respective source, target and label of a transition $t = s \xrightarrow{e} s'$. Let $A = [a_{i,j}]$ be the incidence matrix of the (undirected) graph (S, T). All definitions and results from section 4 carry to (undirected) graphs up to the replacement of $I\!R$ by **2**, see e.g. [32, 17]. Recall that a cut is a cutset or an edge-disjoint union of cutsets, where a cutset is a minimal set of edges whose removal increases the number of connected components by one. Cuts and cutsets are now represented as boolean vectors in $\mathbf{2}<T> \cong \mathbf{2}^m$, [2] and similarly for cycles and for edge-disjoint unions of cycles. The (pointwise) sum of two cuts is a cut, and similarly for two edge-disjoint unions of cycles. In other words, cuts and edge-disjoint union of cycles form vector spaces over **2**, let $\mathcal{V}_Q$ and $\mathcal{V}_B$. These vector spaces are respectively spanned by the rows of the fundamental cutset matrix Q_f and by the rows of the fundamental cycle matrix B_f, jointly computed from any spanning tree by the algorithms described in section 4, interpreted over the boolean field. Thus Q_f and B_f are $(n-1) \times m$ and $(m-n+1) \times m$ matrices with boolean entries such that $B_f \cdot Q_f^t = 0$. Therefore $\mathcal{V}_Q$ and $\mathcal{V}_B$ form orthogonal complements in the boolean vector space $\mathbf{2}<T> \cong \mathbf{2}^m$.

5.2 The Vector Space of Abstract Regions

Our purpose is to transport the linear algebraic methods from the vector space $\mathbf{2}<T>$ to the vector space $\mathbf{2}<E>$ through the labelling function $\ell : T \to E$, which maps transitions to their labelling events.

Definition 5.1 *A cut $C = [c_j]$ is a regional cut if $\ell(t_j) = \ell(t_k) \Rightarrow c_j = c_k$ for all $1 \leq j, k \leq m$. An abstract region is a map $\eta : E \to \mathbf{2}$ such that $c_j = \eta(\ell(t_j))$ defines a regional cut $C = \eta \circ \ell$.*

By an abuse of notation, we make no distinction between vectors $C \in \mathbf{2}<T> \cong \mathbf{2}^m$ and the corresponding maps $C : T \to \mathbf{2}$. We make a similar confusion between maps $\eta : E \to \mathbf{2}$ and vectors $\eta \in \mathbf{2}<E> \cong \mathbf{2}^l$, where $E = \{e_1, \ldots, e_l\}$. Because $\mathcal{A}$ is reduced, regional cuts and abstract regions are in a bijective correspondence. Moreover, given any pair of regional cuts $C = \eta \circ \ell$ and $C' = \eta' \circ \ell$, their sum is a regional cut $C + C' = (\eta + \eta') \circ \ell$. Thus abstract regions form a subspace of the vector space $\mathbf{2}<E> \cong \mathbf{2}^l$. A method for computing a basis of abstract regions is indicated below.

[2] If K is a ring (or a field) and X a set (of generators) we let $K<X>$ denote the K-module (or vector space) freely generated by X.

Since the vector spaces $\mathcal{V}_Q$ and $\mathcal{V}_B$ are orthogonal complements, η is an abstract region if and only if $\eta \circ \ell \in \mathcal{V}_Q$ if and only if $C = \eta \circ \ell$ is orthogonal to all fundamental cycles B_i (rows of B_f). For any cycle $B = [b_j]$, let $\Pi(B) = [\pi_k] \in \mathbf{2} <E>$ be the *Parikh image* of B given by $\pi_k = \sum\{b_j | \ \ell(t_j) = e_k\}$. Otherwise stated $\pi_k = \sum_{j=1}^{m} \varphi_j(e_k)$ where $\varphi_j(e_k) = 1$ if $\ell(t_j) = e_k$ and $b_j = 1$, else 0. Then $C \cdot B = 0$ if and only if $\sum_{j=1}^{m} \eta(\ell(t_j)) \cdot b_j = 0$ if and only if $\sum_{k=1}^{l} (\eta(e_k) \cdot \sum_{j=1}^{m} \varphi_j(e_k)) = 0$ if and only if $\eta \cdot \Pi(B) = 0$. Since $\Pi(B + B') = \Pi(B) + \Pi(B')$, it follows that η is an abstract region if and only if η is orthogonal to the linear subspace of $\mathbf{2} <E>$ spanned by the Parikh images $\Pi(B_i)$ of the fundamental cycles B_i. Let $l - p$ be the dimension of the linear space $\Pi(\mathcal{V}_B)$. A basis of abstract regions $\{\eta_1, \ldots, \eta_p\}$ follows, e.g. by Gauss resolution.

5.3 Flip Flop Regions and Flip Flop Nets

Definition 5.2 *A flip flop region in* $\mathcal{A} = (S, E, T, s_n)$ *is a non trivial subset of states* $R \subset S$ *whose characteristic function* $\chi_R : S \to \mathbf{2}$ *satisfies* $\forall i \in \{1, \ldots, m\} \quad \chi_R(\partial^1(t_i)) = \chi_R(\partial^0(t_i)) + \eta(\ell(t_i))$ *for some abstract region* η.

Since $\mathcal{A}$ is reachable and reduced, abstract regions η are in bijective correspondence with pairs of complementary regions R and $S \setminus R$. Flip flop regions, represented as vectors $\chi_R \in \mathbf{2} <S> \cong \mathbf{2}^n$, form a linear subspace of $\mathbf{2}^n$, closed under the complementation operation $\chi_R + \mathbf{1} = \chi_{S \setminus R}$.

The definition of flip flop nets stems from the analysis of the possible crossing relations between a flip flop region R and the transitions bearing an identical label. All possible cases are covered by four relations:

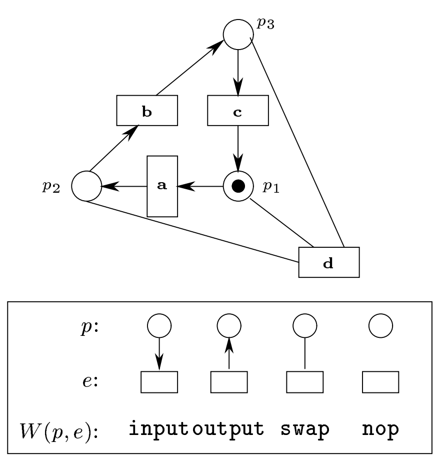
$$
\begin{array}{ll}
R^\bullet e \ : \forall t \in T & \ell(t) = e \Rightarrow (\partial^0(t) \in R \ \wedge \ \partial^1(t) \notin R) \\
e^\bullet R \ : \forall t \in T & \ell(t) = e \Rightarrow (\partial^0(t) \notin R \ \wedge \ \partial^1(t) \in R) \\
e^\perp R : \forall t \in T & \ell(t) = e \Rightarrow (\partial^0(t) \in R \Leftrightarrow \partial^1(t) \in R) \\
e^\times R : \forall t \in T & \ell(t) = e \Rightarrow (\partial^0(t) \in R \Leftrightarrow \partial^1(t) \notin R)
\end{array}
$$

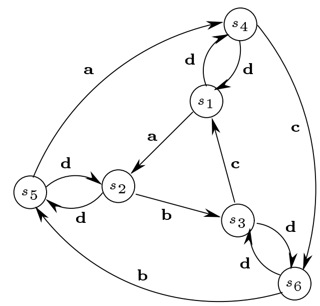
Conversely, any non trivial subset of states $R \subset S$ satisfying $R^\bullet e \vee e^\bullet R \vee e^\perp R \vee e^\times R$ for all events $e \in E$ is a flip flop region, associated with an abstract region η such that $\eta(e) = 0$ if and only if $e^\perp R$. It is now patent that flip flop regions are an extension of elementary regions, which must satisfy $R^\bullet e \ \vee \ e^\bullet R \ \vee \ e^\perp R$. Observe that $e^\bullet R \Rightarrow e^\times R$ and $R^\bullet e \Rightarrow e^\times R$. However these three relations play incomparable roles in flip flop nets, where they are called respectively *input* ($R^\bullet e$), *output* ($e^\bullet R$), and *swap* ($e^\times R$).

Definition 5.3 *A flip flop net is a triple* $N = (P, E, W)$ *where* P *is the set of places or conditions,* E *is the set of events, and* $W : P \times E \to \{\texttt{input}, \texttt{output}, \texttt{nop}, \texttt{swap}\}$ *is a matrix such that* $\forall e \in E \quad \exists p \in P \quad W(p, e) \neq \texttt{nop}$. *A case of* N *is a map* $M : P \to \mathbf{2}$. *An event* e *has concession at* M *if and only if* $\forall p \in P$ $(W(p, e) = \texttt{input} \Rightarrow M(p) = 1) \wedge (W(p, e) = \texttt{output} \Rightarrow M(p) = 0)$. *The event* e *may then fire, resulting in a transition* $M[e\!>\!M'$ *where for every condition* p: $W(p, e) = \texttt{nop} \Rightarrow M'(p) = M(p)$ *and* $W(p, e) \neq \texttt{nop} \Rightarrow M'(p) = 1 + M(p)$.

Definition 5.4 *A flip flop net system is a structure $\mathcal{N} = (P, E, W, M_0)$, where M_0 is a case of the underlying flip flop net $N = (P, E, W)$. The sequential case graph of $\mathcal{N}$ is the automaton $\mathcal{N}^* = (S, E, T, M_0)$ where S is the set of cases reachable from M_0 by sequences of steps $M[e{>}M'$ and T is the subset of these steps in $S \times E \times S$.*

It follows that for every condition $p \in P$, the sets $\{M \in S |\ M(p) = 1\}$ and $\{M \in S |\ M(p) = 0\}$ are complementary flip flop regions of $\mathcal{N}^*$. So the sets of states $\{s_1, s_5, s_6\}$ and $\{s_2, s_3, s_4\}$ are flip flop regions in the example shown in Fig. 10.





	s_1	s_2	s_3	s_4	s_5	s_6
p_1	1	0	0	0	1	1
p_2	0	1	0	1	0	1
p_3	0	0	1	1	1	0

Fig. 10. a flip flop net and its sequential case graph

5.4 Representation Result

The following result was established in [39].

Proposition 5.5 *A finite loop-free automaton $\mathcal{A} = (S, E, T, s_n)$, reachable from s_n and reduced, is isomorphic to the sequential case graph of a flip flop net system if and only if the following conditions are satisfied for R ranging over the set $\mathcal{R}_{\mathrm{FFN}}(\mathcal{A})$ of flip flop regions of $\mathcal{A}$:*
SSP: $\forall s, s' \in S \quad s \neq s' \Rightarrow \exists R \quad (s \in R \Leftrightarrow s' \notin R)$.
ESSP: $\forall s \in S \quad \forall e \in E \quad$ *not* $s \stackrel{e}{\rightarrow} \quad \Rightarrow \exists R \quad (R^{\bullet}e \wedge s \notin R) \vee ({e^{\bullet}}R \wedge s \in R)$.

A synthesis algorithm follows easily. Let $\{\eta_1, \ldots, \eta_p\}$ be a basis of abstract regions of $\mathcal{A}$, computed from some spanning tree $U \subseteq T$. For each state $s_i \in S$, let p_i be the chain connecting s_i and s_n in the spanning tree. An instance $\mathrm{SSP}(s_i, s_j)$ of the states separation problem can be solved if and only if $\eta_k \cdot (\Pi(p_i) + \Pi(p_j)) \neq 0$ for some $k \in \{1, \ldots, p\}$, where $\Pi(p) \in \mathbf{2}{<}E{>}$ is the

Parikh image of the chain $p \in \mathbf{2} <T>$. An instance of ESSP(s_i, e) can be solved if and only if there exists a linear combination $\eta = \sum_{k=1}^{p} \alpha_k \cdot \eta_k$, where $\alpha_k \in \mathbb{Z}$, satisfying $\eta \cdot [\Pi(p_i) + \Pi(p_j)] = 1$ for every state s_j in which event e is enabled. When these conditions are satisfied, a net system $\mathcal{N} = (P, E, W, M_0)$ such that $\mathcal{A} \cong \mathcal{N}^*$ may be constructed by assembling the atomic net systems $\mathcal{N}_p$ defined from conditions p as follows:

1. for each instance SSP(s_i, s_j) solved by η_k, let p be the condition such that $W(p, e) = \texttt{swap}$ if $\eta_k(e) = 1$ and $W(p, e) = \texttt{nop}$ if $\eta_k(e) = 0$, with $M_0(p)$ fixed arbitrarily to 0 or 1;
2. for each instance ESSP(s_i, e) solved by an abstract region $\eta = \sum_{k=1}^{p} \alpha_k \cdot \eta_k$, let p be the condition such that $W(p, e) = \texttt{input}$, $M_0(p) = \eta \cdot \Pi(p_i)$, and for $e' \neq e$, $W(p, e') = \texttt{swap}$ if $\eta(e') = 1$, and $W(p, e') = \texttt{nop}$ if $\eta(e') = 0$.

A minimal system $\mathcal{N}'$ such that $\mathcal{A} \cong \mathcal{N}'^*$ may be obtained by eliminating from $\mathcal{N}$ redundant places. The following is proved in [39].

Proposition 5.6 *The synthesis problem for flip flop nets may be solved in time $O(|S|^2 \times |E|^3)$, where S and E are the respective sets of states and events of the automaton.*

The synthesis algorithm which has been suggested here is a simplified form of the synthesis algorithm for Petri nets proposed in [2] and presented in section 7 of this survey. The case of Petri nets is significantly more complex, to a limited extent because the integer module $\mathbb{Z} <E>$ is more complicated that the boolean vector space $\mathbf{2} <E>$, and to a large extent because combinatorial approximation techniques are needed for the synthesis of Petri nets, while they are useless for flip flop nets.

Before tackling the synthesis problem for Petri nets, we make a detour to show that the striking similarity of the representation results for elementary net systems and flip flop net systems is not incidental, and does not depend on the type of nets.

6 Regions for Arbitrary Types of Nets

The automata $\mathcal{A} = (S, E, T, s_0)$ considered in this section are always assumed to be reachable and deterministic, but they may not be simple, nor reduced, and they are not necessarily finite. The transition systems (S, E, T) are always assumed to be deterministic, but they are not necessarily connected. Recall that a morphism of transition systems $(\sigma, \eta) : (S, E, T) \to (S', E', T')$ is a pair of maps $\sigma : S \to S'$ and $\eta : E \to E'$ such that $s \xrightarrow{e} s'$ in T entails $\sigma(s) \xrightarrow{\eta(e)} \sigma(s')$ in T'; morphisms of automata are morphisms of the underlying transition systems which map the initial state to the initial state.

The extension of the concept of regions to arbitrary types of nets stems from the following observation. Let τ_{FFN} be the transition system given in Fig. 11. Solving the synthesis problem for $\mathcal{A} = (S, E, T, s_0)$ w.r.t. flip flop nets amounts

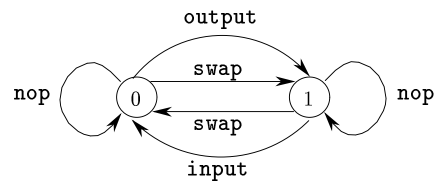


Fig. 11. the type $\tau_{\rm FFN}$ of flip flop nets

to amalgamate on E a set of atomic net systems $\mathcal{N}_p = (\{p\}, E, W, M_0)$, defined from morphisms $p = (\sigma, \eta) : (S, E, T) \to \tau_{\rm FFN}$ such that $W((\sigma, \eta), e) = \eta(e)$ and $M_0((\sigma, \eta)) = \sigma(s_0)$. The resulting net system $\mathcal{N} = \sum_{p \in P} \mathcal{N}_p$ has a case graph $\mathcal{N}^*$ isomorphic to $\mathcal{A}$ if and only if the family $\{\mathcal{N}_p |\ p = (\sigma, \eta) \in P\}$ is *admissible* in the sense that the following two separation conditions are satisfied:
SSP: $\forall s, s' \in S \quad s \neq s' \Rightarrow \exists (\sigma, \eta) \in P \quad \sigma(s) \neq \sigma(s')$.
ESSP: $\forall s \in S \quad \forall e \in E \quad \text{not } (s \stackrel{e}{\to}) \Rightarrow \exists (\sigma, \eta) \in P \quad \text{not } (\sigma(s) \stackrel{\eta(e)}{\to}) \quad \text{in} \quad \tau_{\rm FFN}$.
Thus the concept of regions as sets of states may profitably be replaced by the richer concept of regions as morphisms, which is actually the central concept for the synthesis of net systems. The two concepts are not strictly equivalent for flip flop nets: several morphisms $(\sigma, \eta) : (S, E, T) \to \tau_{\rm FFN}$ may actually determine the same set of states $\sigma^{-1}(\{1\})$, i.e. the same set theoretic region, because the component σ on states does not determine the component η (even though the transition system is connected and reduced).

Our aim is to show that the representation results which have been stated so far for elementary nets and for flip flop nets may be established at once for all possible types of nets, using the concept of regions as morphisms.

6.1 Types of Nets

For the sake of a uniform presentation, we depart here from the traditional definition of nets and adopt a parametric definition covering elementary nets, flip flop nets, and Petri nets as particular instances. The parameters of this general definition are called *types of nets* [6].

Definition 6.1 *A type of nets is a deterministic transition system* $\tau = (LS, LE, \tau)$, *where* LS *and* LE *are the respective sets of local states and local events, and* $\tau \subseteq LS \times LE \times LS$ *defines the partial action of local events on local states.*

Definition 6.2 *A net of type* τ *is a triple* $N = (P, E, W)$ *where* P *is a set of places,* E *is a set of events, and* $W : P \times E \to LE$ *is the weight matrix. A marking is a mapping* $M : P \to LS$. *A net system of type* τ *is a structure* $\mathcal{N} = (P, E, W, M_0)$ *where* M_0, *the initial marking, is a marking of the underlying net* $N = (P, E, W)$.

A net or net system is place simple if all rows of the weight matrix are different; it is event simple if all columns of the weight matrix are different. All nets and net systems considered in this section are assumed to be place simple but not necessarily event simple.

A net may be seen as an undirected complete bipartite graph whose edges are weighted by local events. As such, nets are of a static nature, but types (of nets) define their dynamics: the partial actions of events on markings may be inferred from the partial actions of local events on local states, using the weight matrix to control products of local events. The following definition extends in this way the usual *sequential firing rule.*

Definition 6.3 *Given a net $N = (P, E, W)$, of type $\tau = (LS, LE, \tau)$, the (sequential) marking graph of N is the transition system (LS^P, E, T) with set of transitions T defined by $(M \xrightarrow{e} M') \in T$ if and only if $\forall p \in P \;\; (M(p) \xrightarrow{W(p,e)} M'(p)) \in \tau$. Given a net system $\mathcal{N} = (P, E, W, M_0)$, the (sequential) marking graph of $\mathcal{N}$ is the (dual) automaton $\mathcal{N}^* = (S, E, T_S, M_0)$ where S is the inductive closure of $\{M_0\}$ w.r.t. forward transitions in T, and $T_S = T \cap (S \times E \times S)$.*

Thus an event has concession at marking M if and only if for every place p, the local event $W(p, e)$ is enabled at the local state $M(p)$ in the transition system τ (defining the type of the net). A net system is reduced if every event e has concession at some reachable marking and if for every pair of distinct places p and p', there exists some reachable marking M such that $M(p) \neq M(p')$. The net systems which we consider are generally not reduced. We now illustrate the above definitions on two classical examples, namely elementary nets and Petri nets.

Let τ_{EN} be the transition system shown in Fig. 12. The elementary nets

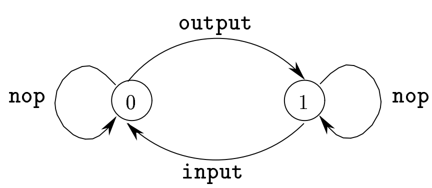


Fig. 12. the type τ_{EN} of elementary nets

(P, E, F) correspond bijectively with nets (P, E, W) of type τ_{EN}, with $W(p, e) =$ `input` $\Leftrightarrow F(p, e)$ and $W(p, e) =$ `output` $\Leftrightarrow F(e, p)$, and $W(p, e) =$ `nop` otherwise. One may easily verify that the corresponding nets have identical marking graphs.

Let us now recall the classical definition of Petri nets.

Definition 6.4 *A Petri net is a triple $N = (P, E, F)$ where P and E are disjoint sets of places and events, and F is a function, $F : (P \times E) \cup (E \times P) \to \mathbb{N}$. A marking of N is a map $M : P \to \mathbb{N}$. An event e has concession at M if and only if $\forall p \in P \quad F(p, e) \leq M(p)$. An event e which has concession at M may fire, resulting in a transition $M[e\!>\!M'$ where $\forall p \in P \quad M'(p) = M(p) - F(p, e) + F(e, p)$. A Petri net N is said to be pure if $\forall p \in P \; \forall e \in E \;\; F(p, e) \times F(e, p) = 0$.*

Let the type of pure Petri nets be the transition system $\tau_{PPN} = (\mathbb{N}, \mathbb{Z}, T)$ such that $n \xrightarrow{z} n'$ if and only if $n' = n + z$, i.e. τ_{PPN} is the full subgraph of the Cayley graph of $\mathbb{Z}$ induced by the restriction on the subset of nodes in $\mathbb{N}$. Pure Petri nets (P, E, F) are linked by a marking graph preserving bijective correspondence with nets (P, E, W) of type τ_{PPN} given by $W(p, e) = F(e, p) - F(p, e)$.

Let the type of Petri nets be the transition system $\tau_{PN} = (\mathbb{N}, \mathbb{N} \times \mathbb{N}, T)$ such that $n \xrightarrow{(p,q)} n'$ if and only if $n \geq p$ and $n' = (n - p) + q$. Petri nets are set in bijective correspondence with nets of type τ_{PN} by the relation $W(p, e) = (F(p, e), F(e, p))$. With this correspondence, the firing rule stated in Def. 6.3 reads actually as

$$M[e{>}M' \quad \Leftrightarrow \quad \forall p \in P \;\; M(p) \geq F(p, e) \;\wedge\; M'(p) = M(p) - F(p, e) + F(e, p)$$

The above Petri nets are a particular instance of the generalized Petri nets studied in [20]. In this paper, Droste and Shortt parametrize the classical definition of Petri nets (Def. 6.4), in which $\mathbb{N}$ is substituted for by the positive part G^+ of a partially ordered abelian group G. These authors further classify types of Petri nets over a fixed group G by the set of pairs $((F(p, e), F(e, p)) \in G^+ \times G^+$ occurring in associated subclasses of nets. For instance, condition-event nets are obtained by restricting nets over $\mathbb{Z}$ to the pairs $((F(p, e), F(e, p)) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. Note that all types of nets which we have defined so far can similarly be obtained from Cayley graphs (G, G, T) (i.e. $g' \xrightarrow{g} g''$ in T if and only if $g'' = g' + g$) by eliminating nodes and/or by restricting group actions to partially defined group actions. For instance, τ_{EN} is the Cayley graph of $\mathbb{Z}/_{3\mathbb{Z}}$ restricted on nodes 0 and 1, with $\texttt{nop} = 0$, $\texttt{output} = 1$, and $\texttt{input} = 2$. Similarly, τ_{FFN} is obtained from the Cayley graph of $\mathbb{Z}/_{2\mathbb{Z}}$ by identifying action 0 with `nop`, action 1 with `swap`, and the partial action 1 defined at node 0 (resp. at node 1) with `output` (resp. with `input`). Therefore, all nets considered so far are *reversible* in the sense that they have co-deterministic sequential marking graphs ($M_1[e{>}M$ and $M_2[e{>}M$ entail $M_1 = M_2$). Nevertheless, flip flop nets are not Petri nets over a group according to the definition of Droste and Shortt. The main reason why we do not stick here to types of nets based on groups is that we want to cover also non reversible nets, such as trace nets (see 6.5).

6.2 Regions as Morphisms

The firing rule for nets stated in Def. 6.3 tells us that for every place p in a net of type τ, the pair of maps (σ_p, η_p) defined by $\sigma_p(M) = M(p)$ and $\eta_p(e) = W(p, e)$ is a morphism of transition systems from the marking graph of the net to the type τ. Therefore, if we forget the internal structure of states in the marking graph, identified with any isomorphic transition system (S, E, T), and if we identify a place p with its extension (σ_p, η_p), we can rediscover the places of the net (and also discover implicit places) as morphisms $(\sigma, \eta) : (S, E, T) \to \tau$. This motivates the following definition of regions for arbitrary types of nets.

Definition 6.5 *Given a transition system $A = (S, E, T)$ and a type of nets $\tau = (LS, LE, \tau)$, the set $\mathcal{R}_\tau(A)$ of τ-type regions in A is the set of morphisms from A to τ.*

By an abuse of notation, we extend the above definition to automata by letting $\mathcal{R}_\tau(\mathcal{A}) = \mathcal{R}_\tau(A)$ where A is the transition system underlying the automaton $\mathcal{A}$. We now illustrate this definition on elementary nets and on Petri nets.

An elementary region in $A = (S, E, T)$ is a morphism $(\sigma, \eta) : A \to \tau_{EN}$. The map η classifies events $e \in E$ into three families according to their relationship with the property $R = \sigma^{-1}(\{1\})$: all events e such that $\eta(e) = \texttt{input}$ take R as an input condition and falsify R ($s \stackrel{e}{\to} s' \Rightarrow s \in R \ \wedge \ s' \notin R$), all events e such that $\eta(e) = \texttt{output}$ take the falsity of R as a precondition and establish R ($s \stackrel{e}{\to} s' \Rightarrow s \notin R \ \wedge \ s' \in R$), and the remaining events such that $\eta(e) = \texttt{nop}$ do not modify R ($s \stackrel{e}{\to} s' \Rightarrow (s \in R \Leftrightarrow s' \in R)$). One recognizes in $R = \sigma^{-1}(\{1\})$ a region according to the original definition of Ehrenfeucht and Rozenberg.

A pure Petri region in $A = (S, E, T)$ is a morphism $(\sigma, \eta) : A \to \tau_{PPN}$ (see Fig. 13 for an illustration). Here the map σ measures the availability of a

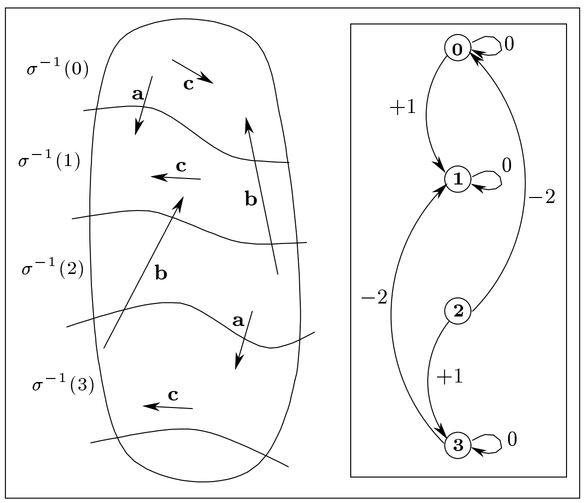


Fig. 13. a pure Petri region as a morphism: $A \stackrel{(\sigma,\eta)}{\longrightarrow} \tau_{PPN}$

resource at each state $s \in S$, and the map η classifies events $e \in E$ according to the amount of resource which they produce (when $\eta(e) > 0$) or consume (when $\eta(e) < 0$) at each firing. When A is a finite transition system, the abstract regions η defined in this way are in bijective correspondence with weighted synchronic distances in A, measuring the relative degree of freedom of the respective subsets of events e such that $\eta(e) < 0$, resp. $\eta(e) > 0$ [12].

A Petri region in $A = (S, E, T)$ is a morphism $(\sigma, \eta) : A \to \tau_{PN}$. Here again the map σ measures the availability of a resource at each state. The map η

classifies events according to associated pairs $\eta(e) = ({}^\bullet\eta(e), \eta^\bullet(e))$ where ${}^\bullet\eta(e)$ measures the amount of resource consumed for triggering e while $\eta^\bullet(e)$ measures the amount of resource produced by e, amounting to a neat variation of resource $\eta^\bullet(e) - {}^\bullet\eta(e)$. These Petri regions coincide with the regions which have been defined by Mukund [33] (in the larger framework of step transition systems) and which have been adapted by Droste and Shortt [20] to Petri nets over partially ordered abelian groups.

6.3 A Galois Connection between Automata and Nets

We saw that regions may serve to reverse the production of marking graphs. The reversing process may also be applied to arbitrary transition systems, leading to the following definitions.

Definition 6.6 *Given a transition system $A = (S, E, T)$ and a type of nets τ, the dual of A is the net $A^* = (\mathcal{R}_\tau(A), E, W)$ with weights defined by $W((\sigma, \eta), e) = \eta(e)$. For any subset $\mathcal{R}$ of $\mathcal{R}_\tau(A)$, let $A^*_{\mathcal{R}}$ denote the subnet of A^* with restricted set of places $\mathcal{R}$.*

Definition 6.7 *Given an automaton $\mathcal{A}$ composed of a transition system A and an initial state s_0, and a type of nets τ, the dual of $\mathcal{A}$ is the net system $\mathcal{A}^*$ composed of the underlying net A^* and of the initial marking M_0 defined by $M_0(\sigma, \eta) = \sigma(s_0)$ for every $(\sigma, \eta) \in \mathcal{R}_\tau(\mathcal{A})$. For any subset $\mathcal{R}$ of $\mathcal{R}_\tau(\mathcal{A})$, let $\mathcal{A}^*_{\mathcal{R}}$ denote the subnet system of $\mathcal{A}^*$ with restricted set of places $\mathcal{R}$.*

We will show that the two $()^*$ operators mapping the automaton $\mathcal{A}$ to the net system $\mathcal{A}^*$ and the net system $\mathcal{N}$ to its marking graph $\mathcal{N}^*$ form a Galois connection: $\mathcal{A} \leq \mathcal{N}^* \Leftrightarrow \mathcal{N} \leq \mathcal{A}^*$. The main difficulty is to construct the appropriate order relations. One expects in particular $\mathcal{A} \leq \mathcal{N}_p^* \Leftrightarrow \mathcal{N}_p \leq \mathcal{A}^*$ for every region $p = (\sigma_p, \eta_p) \in \mathcal{R}_\tau(\mathcal{A})$ where $\mathcal{N}_p$ is the *atomic* subnet system of $\mathcal{A}^*$ with sole place p (i.e. $\mathcal{N}_p = \mathcal{A}^*_{\{p\}}$) and $\mathcal{N}_p^*$ is its marking graph. This particular case will help us to find out the order relation on automata. Since $\mathcal{N}_p$ is a subnet system of $\mathcal{A}^*$, both $\mathcal{N}_p \leq \mathcal{A}^*$ and $\mathcal{A} \leq \mathcal{N}_p^*$ are expected; by definition of regions, if E is the set of events of $\mathcal{A}$ then $(\sigma_p, 1_E)$ is an *event preserving* morphism from $\mathcal{A}$ to $\mathcal{N}_p^*$. Moreover, if there exists an event preserving morphism $(\sigma, 1_E) : \mathcal{A}_1 \to \mathcal{A}_2$ between two automata with set of events E, this morphism is necessarily unique owing to the strong properties of determinism and reachability we have assumed from all automata; therefore, if there exist morphisms $(\sigma_1, 1_E) : \mathcal{A}_1 \to \mathcal{A}_2$ and $(\sigma_2, 1_E) : \mathcal{A}_2 \to \mathcal{A}_1$, then $\mathcal{A}_1$ and $\mathcal{A}_2$ are identical up to the identity of states ($\mathcal{A}_1 =_E \mathcal{A}_2$). So let $\mathbf{Aut}(E)$ be the set of (deterministic and reachable) automata with fixed set of events E, quotiented by $=_E$, then

$$\mathcal{A}_1 \leq \mathcal{A}_2 \quad \textbf{if} \quad \exists \sigma : \ (\sigma, 1_E) : \ \mathcal{A}_1 \to \mathcal{A}_2$$

is a partial order on $\mathbf{Aut}(E)$, such that $\mathcal{A} \leq \mathcal{N}_p^*$ for every region $p \in \mathcal{R}_\tau(\mathcal{A})$. This partial order is a complete lattice, with greatest lower bounds computed as synchronized products. We remind the reader that the synchronized product $\bigwedge_{i\in I}\mathcal{A}_i$ of a family of automata $\mathcal{A}_i = (S_i, E, T_i, s_{0,i})$ indexed by $i \in I$ is the automaton (S, E, T, s_0) with components as follows: $s_0 = (s_{0,i})_{i\in I}$, S is the inductive closure of the set $\{s_0\}$ w.r.t. the *synchronized transition rule*

$$(s_i)_{i\in I} \xrightarrow{e} (s'_i)_{i\in I} \quad \textbf{iff} \quad \forall i \in I \quad (s_i \xrightarrow{e} s'_i) \in T_i$$

and T is the set of occurrences of this rule at states $(s_i)_{i\in I} \in S$. By definition of marking graphs, the automaton $\mathcal{N}^*$ dual to a net system $\mathcal{N} = (P, E, W, M_0)$ is actually the synchronized product $\bigwedge_{p\in P}\mathcal{N}_p^*$ of the marking graphs of its atomic subnet systems.

Concerning the order relation on net systems, the central assumption that $\mathcal{N}_p \le \mathcal{A}^*$ for every region p of $\mathcal{A}$ leads to choose something akin to the substructure ordering: $\mathcal{N}_1 \le_{sub} \mathcal{N}_2$ if $\mathcal{N}_1$ is $\mathcal{N}_2$ restricted on a subset of places. However *replicated* places may occur in a net system $\mathcal{N} = (P, E, W, M_0)$, i.e. places which the initial marking M_0 and the weight function W do not distinguish from one another, and we do not care about their degree of multiplicity nor about their identities. Let morphisms of net systems with fixed set of events be defined as follows: a morphism from $\mathcal{N}_1 = (P_1, E, W_1, M_{0,1})$ to $\mathcal{N}_2 = (P_2, E, W_2, M_{0,2})$ is a map $\beta : P_1 \to P_2$ such that $M_{0,1}(p) = M_{0,2}(p)$ and $W_1(p, e) = W_2(\beta(p), e)$ for all $p \in P_1$ and $e \in E$. Two net systems connected by morphisms in both directions are henceforth declared equivalent. Let $\mathbf{Nets}(E)$ denote the set of equivalence classes of net systems with set of events E (replication free nets are canonical representatives). One can equip $\mathbf{Nets}(E)$ with the partial order relation defined as:

$$\mathcal{N}_1 \le \mathcal{N}_2 \quad \textbf{iff} \quad \exists\beta : \mathcal{N}_1 \to \mathcal{N}_2$$

This partial order is a complete lattice, with least upper bounds $\bigvee_{i\in I}\mathcal{N}_i$ of families of net systems computed by amalgamation of sets of places. Told in another way, if we identify a place p in a net system $\mathcal{N} = (P, E, W, M_0)$ with the pair $(M_0(p), \eta_p)$ such that $\eta_p(e) = W(p, e)$ for $e \in E$ then $\bigvee_{i\in I}(P_i, E, W_i, M_{0,i})$ $= (\bigcup_{i\in I} P_i, E, W, M_0)$ where $W(p, e) = W_i(p, e)$ and $M_0(p) = M_{0,i}(p)$ for $p \in P_i$. A net system $\mathcal{N}$ with set of places P is now the least upper bound $\bigvee_{p\in P}\mathcal{N}_p$ of its atomic subnet systems $\mathcal{N}_p$. In the particular case where $\mathcal{N} = \mathcal{A}^*$ is dual to the automaton $\mathcal{A}$, its set of places is the set of regions $\mathcal{R}_\tau(\mathcal{A})$, where τ is the type of $\mathcal{N}$, hence its atomic subnet systems $\mathcal{N}_p$ have the form $\mathcal{A}^*_{\{p\}}$ and we get the following.

Proposition 6.8 *Let $\mathcal{R} \subseteq \mathcal{R}_\tau(\mathcal{A})$ then $\mathcal{A}^*_{\mathcal{R}} = (\bigvee_{p\in\mathcal{R}}\mathcal{A}^*_{\{p\}})$.*

The key for the Galois connection between the ordered sets $(\mathbf{Aut}(E), \le)$ and $(\mathbf{Nets}(E), \le)$ is the following proposition, proved in [7]

Proposition 6.9 *Let $\mathcal{N} = (\{p\}, E, W, M_0)$ be an atomic net system of type τ, then $\mathcal{A} \le \mathcal{N}^*$ if and only if $M_0(p) = \sigma(s_0)$ and $\forall e \in E \quad W(p, e) = \eta(e)$ for some region $(\sigma, \eta) \in \mathcal{R}_\tau(\mathcal{A})$, where s_0 is the initial state of the automaton $\mathcal{A}$.*

Thus any atomic net system $\mathcal{N}$ such that $\mathcal{A} \leq \mathcal{N}^*$ is isomorphic to $\mathcal{N}_p = \mathcal{A}^*_{\{p\}}$ for some region $p = (\sigma, \eta) \in \mathcal{R}_\tau(\mathcal{A})$. Since $\mathcal{A}^* = \bigvee\{\mathcal{A}^*_{\{p\}} \mid p \in \mathcal{R}_\tau(\mathcal{A})\}$, it follows that $\mathcal{N} \leq \mathcal{A}^*$. Conversely, by definition of the order relation on net systems, any atomic net system $\mathcal{N}$ such that $\mathcal{N} \leq \mathcal{A}^*$ is isomorphic to $\mathcal{N}_p = \mathcal{A}^*_{\{p\}}$ for some region $p = (\sigma, \eta) \in \mathcal{R}_\tau(\mathcal{A})$. Since $\mathcal{A} \leq \mathcal{N}^*_p$ by construction of the order relation on automata, it follows that $\mathcal{A} \leq \mathcal{N}^*$. Altogether, we obtain the following.

Proposition 6.10 *For any atomic net system $\mathcal{N}$, $\mathcal{A} \leq \mathcal{N}^* \Leftrightarrow \mathcal{N} \leq \mathcal{A}^*$.*

We are ready to establish the expected Galois connection between automata and net systems.

Proposition 6.11 *The two $()^*$ operators, mapping respectively the automaton $\mathcal{A}$ to the dual net system $\mathcal{A}^*$ and the net system $\mathcal{N}$ to its marking graph $\mathcal{N}^*$, constitute a Galois connection between the ordered sets* $\mathbf{Nets}(E)$ *and* $\mathbf{Aut}(E)$*: $\mathcal{A} \leq \mathcal{N}^* \Leftrightarrow \mathcal{N} \leq \mathcal{A}^*$ for $\mathcal{A} \in \mathbf{Aut}(E)$ and $\mathcal{N} \in \mathbf{Nets}(E)$.*

Proof: By Prop. 6.10, $\mathcal{A} \leq \mathcal{N}^* \Leftrightarrow \mathcal{N} \leq \mathcal{A}^*$ if $\mathcal{N}$ is an *atomic* net system. Now for a net system $\mathcal{N} = \bigvee_{p \in P} \mathcal{N}_p$, where $\mathcal{N}_p$ is the atomic subnet system of $\mathcal{N}$ with the unique place p, $\mathcal{N}^* = \bigwedge_{p \in P} \mathcal{N}^*_p$ by definition of marking graphs. Thus $\mathcal{A} \leq \mathcal{N}^*$ *if and only if* $\mathcal{A} \leq \mathcal{N}^*_p$ for all $p \in P$ *if and only if* $\mathcal{N}_p \leq \mathcal{A}^*$ for all $p \in P$ (because $\mathcal{N}_p$ is atomic) *if and only if* $\mathcal{N} \leq \mathcal{A}^*$. ∎

The relations $\mathcal{A}_1 \leq \mathcal{A}_2 \Rightarrow \mathcal{A}^*_2 \leq \mathcal{A}^*_1$ (for $\mathcal{A}_1, \mathcal{A}_2 \in \mathbf{Aut}(E)$) and $\mathcal{N}_1 \leq \mathcal{N}_2 \Rightarrow \mathcal{N}^*_2 \leq \mathcal{N}^*_1$ (for $\mathcal{N}_1, \mathcal{N}_2 \in \mathbf{Nets}(E)$) follow immediately from the Galois connection. Another property of Galois connections is to produce *closure operators* by conjugated composition of the dual operators. Recall that an operator $\overline{()}$ on $(X, \leq)$, mapping x to $\overline{x}$, is a closure operator if it is increasing ($x_1 \leq x_2 \Rightarrow \overline{x_1} \leq \overline{x_2}$), extensive ($x \leq \overline{x}$), and idempotent ($\overline{\overline{x}} = x$). The double dual operators $()^{**}$ acting respectively on the ordered sets $(\mathbf{Aut}(E), \leq)$ and $(\mathbf{Nets}(E), \leq)$ are therefore closure operators. An automaton $\mathcal{A}$ equal to its closure $\mathcal{A}^{**}$ is said to be *separated* with respect to the fixed type of nets τ, while a net system $\mathcal{N}$ equal to its closure $\mathcal{N}^{**}$ is said to be *saturated*. Owing to the Galois connection, the lattices of separated automata and saturated net systems are dually order-isomorphic (i.e. isomorphic up to reversing the order).

6.4 Representation Results

By definition, an automaton separated with respect to type τ is isomorphic to the synchronized product of marking graphs $\mathcal{N}^*_p$ of atomic net systems $\mathcal{N}_p = \mathcal{A}^*_{\{p\}}$ derived from τ-regions p of $\mathcal{A}$ (in formulas: $\mathcal{A} \cong \bigwedge_{p \in \mathcal{R}_\tau(\mathcal{A})} \mathcal{N}^*_p$). Following [19], let us say that a subset of regions $\mathcal{R} \subseteq \mathcal{R}_\tau(\mathcal{A})$ is *admissible* if $\mathcal{A} \cong \bigwedge_{p \in R} \mathcal{N}^*_p$. So, $\mathcal{A}$ is separated if and only if $\mathcal{R}_\tau(\mathcal{A})$ is admissible, and of course every superset of an admissible set of regions is admissible. The marking graph $\mathcal{N}^*$ of a net system $\mathcal{N}$ is separated because $\mathcal{N}^* \cong \mathcal{N}^{***}$ follows from the Galois connection. In fact, the extensions (σ_p, η_p) of places p of $\mathcal{N}$ form an admissible set of regions of $\mathcal{N}^*$. The following criterion may be used to recognize admissible sets of regions, and consequently separated automata.

Theorem 6.12 *Given an automaton $\mathcal{A} = (S, E, T, s_0)$ and a type of nets τ, a set of regions $\mathcal{R} \subseteq \mathcal{R}_\tau(\mathcal{A})$ is admissible if and only if the following* separation properties *are satisfied for all states $s, s' \in S$ and for every event $e \in E$:*

(SSP) $s \neq s' \;\Rightarrow\; \exists(\sigma,\eta) \in \mathcal{R} : \; \sigma(s) \neq \sigma(s')$

(read: (σ,η) solves the states separation *problem at (s,s'))*

(ESSP) $s \stackrel{e}{\not\rightarrow} \;\Rightarrow\; \exists(\sigma,\eta) \in \mathcal{R} : \; \sigma(s) \stackrel{\eta(e)}{\not\longrightarrow}$ *w.r.t.* τ

(read: (σ,η) solves the event/state separation *problem at (s,e))*

When both properties are satisfied, $\mathcal{A} \cong (\mathcal{A}^*_\mathcal{R})^*$, where $\mathcal{A}^*_\mathcal{R}$ is the subnet system of $\mathcal{A}^*$ with restricted set of places $\mathcal{R}$ (also called the net synthesized *from $\mathcal{R}$).*

Proof: Let $\mathcal{N}_p = \mathcal{A}^*_{\{p\}}$ for $p \in \mathcal{R}$, and let $\mathcal{N}_\mathcal{R} = \mathcal{A}^*_\mathcal{R}$. Seeing that $\mathcal{A} \leq \mathcal{N}^*_p$ for every region p, $\mathcal{A} \leq \bigwedge_{p\in\mathcal{R}} \mathcal{N}^*_p = \mathcal{N}^*_\mathcal{R}$. Accordingly, there exists a morphism of automata $(\sigma, 1) : \mathcal{A} \rightarrow \mathcal{N}^*_\mathcal{R}$. Moreover this morphism is unique. On the other hand, every region $p = (\sigma_p, \eta_p)$ factors into $(\iota, \eta_p) \circ (\sigma_p, 1)$ where ι acts as the identity on the local states in its domain, and $(\sigma_p, 1)$ lifts to the unique event preserving morphism from $\mathcal{A}$ to $\mathcal{N}^*_p$. As $\mathcal{N}^*_\mathcal{R}$ is the synchronized product of $(\mathcal{N}^*_p)_{p\in\mathcal{R}}$, σ must be the map that sends each state s of $\mathcal{A}$ to the associated vector $\sigma(s) = (\sigma_p(s))_{p=(\sigma_p,\eta_p)\in\mathcal{R}}$ (the p-component is computed by evaluating region p at state s). Since $(\sigma, 1)$ is the unique morphism of this form from $\mathcal{A}$ to $\mathcal{N}^*_\mathcal{R}$, and seeing that all automata are accessible and deterministic, the assertion $\mathcal{A} \cong \mathcal{N}^*_\mathcal{R}$ is now equivalent to *(i)* σ is an injective map, and *(ii)* $s \stackrel{e}{\rightarrow}$ in $\mathcal{A}$ whenever $\sigma(s) \stackrel{e}{\rightarrow}$ in $\mathcal{N}^*_\mathcal{R}$. Now SSP is just another form of assertion *(i)*. By definition of the synchronized product, $\sigma(s) \stackrel{e}{\rightarrow}$ in $\mathcal{N}^*_\mathcal{R}$ entails $\sigma_p(s) \stackrel{e}{\rightarrow}$ in $\mathcal{N}^*_p$ for all $p \in \mathcal{R}$,hence $ESSP$ is just another form of assertion *(ii)*. ∎

Corollary 6.13 *Given an automaton $\mathcal{A} \in$ **Aut**(E) and a type of nets τ, $\mathcal{A} \cong \mathcal{N}^*$ for some net system $\mathcal{N} \in$ **Nets**(E) if and only if $\mathcal{A} \cong \mathcal{A}^{**}$ if and only if the conditions* SSP *and* ESSP *are valid in $\mathcal{A}$. Given an automaton $\mathcal{A} \in$ **Aut**(E), a type of nets τ, and a net system $\mathcal{N} \in$ **Nets**(E), $\mathcal{A} \cong \mathcal{N}^*$ if and only if $\mathcal{N}$ is isomorphic to subnet system of $\mathcal{A}^*$ determined from some admissible subset of regions $\mathcal{R} \subseteq \mathcal{R}_\tau(\mathcal{A})$.*

By setting $\tau = \tau_{EN}$, resp. $\tau = \tau_{FFN}$, in the above theorem and corollary, one retrieves the results of Ehrenfeucht and Rozenberg (Prop. 2.15) and Desel and Reisig (Prop. 3.6), resp. the result of Schmitt (Prop. 5.5). The application of the theorem to the types τ_{PPN} and τ_{PN} will be examined in section 7.

6.5 Some Applications to Safe Nets

We call *safe* nets all nets whose markings are defined as subsets of places $M \subseteq P$, or equivalently as maps $M : P \rightarrow \{0,1\}$. Thus, a type of safe nets is a transition system $\tau = (LS, LE, \tau)$ whose set of local states is $LS = \{0,1\}$. The largest type of safe nets, let τ_{safe}, is obtained by including in its transitions all the defined instances $s \stackrel{f}{\rightarrow} f(s)$ of partial functions $f : \{0,1\} \rightarrow \{0,1\}$.

Table 1. safe nets

	0	1	C/E nets	elementary nets	flip flop nets	trace nets
`input`	-	0	yes	yes	yes	yes
`output`	1	-	yes	yes	yes	yes
`test=1`	-	1	yes	no	no	yes
`test=0`	0	-	no	no	no	yes
`set_0`	0	0	no	no	no	yes
`set_1`	1	1	no	no	no	yes
`nop`	0	1	yes	yes	yes	yes
`swap`	1	0	no	no	yes	no

These functions, tabulated in Table 1, form a set LE_{safe}. Various types $\tau_X = (\{0,1\}, LE_X, \tau_X)$ follow as induced restrictions of τ_{safe} on particular subsets of local events $LE_X \subseteq LE_{safe}$ (see again Table 1 for some examples). Each type τ_X determines corresponding regions in transition systems $A = (S, E, T)$, defined as morphisms $(\sigma, \eta) : A \to \tau_X$. These morphisms define in turn set-theoretic regions $R = \sigma^{-1}(\{1\}) \in \mathcal{P}(S)$. We restrict our analysis of safe type to types τ_X larger than τ_{EN} and such that the complement of a region $\sigma^{-1}(\{1\})$ is a region. This amounts to set on LE_X the constraints $\{\texttt{nop}, \texttt{input}, \texttt{output}\} \subseteq LE_X$ and $\texttt{set\_0} \in LE_X \Leftrightarrow \texttt{set\_1} \in LE_X$.

We declare equivalent, resp. weakly equivalent, two safe types τ_X which determine an identical family of separated automata, resp. identical families of set-theoretic regions in automata. With the above constraints, there are four classes of weakly equivalent types τ_X, each of which splitting into two equivalence classes of types.

The four possible concepts of set-theoretic regions are determined from five forbidding patterns displayed in Fig. 14. Each pattern represents a pair transi-

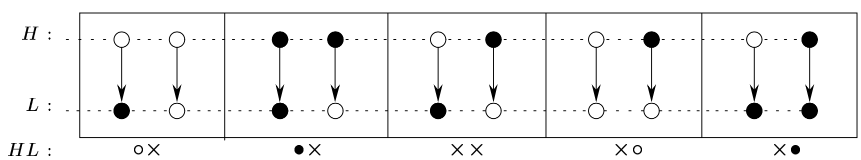


Fig. 14. five patterns for a pair of transitions with the same label $s_1 \xrightarrow{e} s_1'$ and $s_2 \xrightarrow{e} s_2'$ where $s \in R$ if and only if the corresponding node is coloured black

tions $s_1 \xrightarrow{e} s_1'$ and $s_2 \xrightarrow{e} s_2'$ with a common label e in a transition system $A = (S, E, T)$ whose states s are coloured black or white. Let $R \subseteq S$ be the subset of states coloured in black. The four possible concepts of **set-theoretic** regions are as follows.

1. R is an *elementary region* if and only if the patterns ○×, ●×, ××, ×○, and ×● do not occur in A. A safe type τ_X induces the elementary regions if and

only if $LE_X \cap \{\texttt{swap}, \texttt{set\_0}, \texttt{set\_1}\} = \emptyset$. This case is met for τ_{EN} and for the type of C/E-nets, let τ_{CEN} where $LE_{CEN} = \{\texttt{nop}, \texttt{input}, \texttt{output}, \texttt{test=1}\}$.
2. R is a *flip flop region* if and only if the patterns $\circ\times$, $\bullet\times$, $\times\circ$, $\times\bullet$ do not occur in A. A safe type τ_X induces the flip flop regions if and only if $LE_X \cap \{\texttt{set\_0}, \texttt{set\_1}\} = \emptyset$. This case is met for τ_{FFN}.
3. R is a *trace region* if and only if the patterns $\circ\times$, $\bullet\times$, $\times\times$ do not occur in A. Trace regions have been introduced independently for *trace nets* in [4, 5] and for *chart nets* in [29] where they are called chart regions. A safe type τ_X induces the trace regions if and only if $\{\texttt{set\_0}, \texttt{set\_1}\} \subseteq LE_X$ and $\texttt{swap} \notin LE_X$. This case is met for the type of trace nets, let τ_{TRN} where $LE_{TRN} = \{\texttt{nop}, \texttt{input}, \texttt{output}, \texttt{test=1}, \texttt{test=0}, \texttt{set\_0}, \texttt{set\_1}\}$.
4. R is a *safe region* if and only if the patterns $\circ\times$ and $\bullet\times$ do not occur in A. A type τ_X induces the safe regions if and only if $\{\texttt{swap}, \texttt{set\_0}, \texttt{set\_1}\} \subseteq LE_X$.

Safe types may be classified further into pure types and impure types according to whether $LE_X \cap \{\texttt{test=0}, \texttt{test=1}\}$ is empty or not. One obtains in this way 8 classes of equivalent types. These classes may be ordered according to the inclusion of the associated sets of separated automata. All inclusions are shown in Fig. 15 together with representative automata showing they are strict.

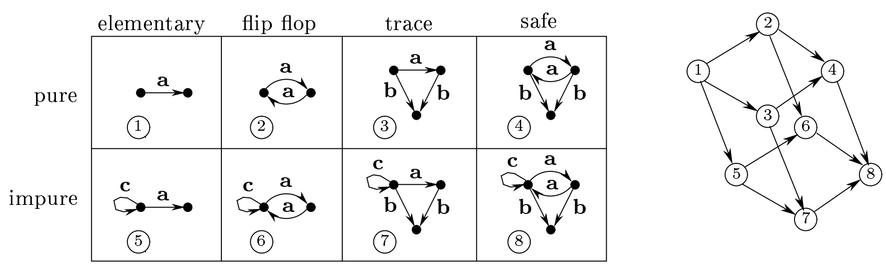


Fig. 15. classification of the equivalence classes of safe types

By specializing Theo. 6.12 to a particular type τ_X, one obtains an immediate characterization of the family of separated automata specific to its equivalence class. We have yet implicitly applied this technique to the type τ_{EN} of elementary nets and to the type τ_{FFN} of flip flop nets. Let us focus on the types τ_{CEN} (of C/E nets) and τ_{TRN} (of trace nets).

From Theo. 6.12 applied to τ_{CEN}, one retrieves Nielsen and Winskel's characterization of marking graphs of C/E nets [35]. Given an asynchronous automaton $\mathcal{A} = (S, E, \|, T, s_0)$ with an empty independence relation, the regions of $\mathcal{A}$ defined in [35] are actually in bijective correspondence with morphisms $(\sigma, \eta) : (S, E, T) \to \tau_{CEN}$ [1]. We recall that an asynchronous automaton according to the definition of Shields and Bednarczyk [9, 41] is a deterministic automaton $\mathcal{A} = (S, E, T, s_0)$, enriched with a symmetric and irreflexive relation of independence $\| \subseteq E \times E$ such that the following conditions are satisfied whenever $e_1 \| e_2$:

FORWARD DIAMOND PROPERTY: $s \xrightarrow{e_1} s_1 \ \wedge \ s \xrightarrow{e_2} s_2 \Rightarrow \exists s' \in S \ \ s_2 \xrightarrow{e_1} s' \ \wedge \ s_1 \xrightarrow{e_2} s'$.
COMMUTATION PROPERTY: $s \xrightarrow{e_1} s_1 \ \wedge \ s_1 \xrightarrow{e_2} s' \Rightarrow \exists s_2 \in S \ \ s \xrightarrow{e_2} s_2 \ \wedge \ s_2 \xrightarrow{e_1} s'$.

Following Nielsen and Winskel, let us define *asynchronous regions* in $\mathcal{A} = (S, E, \|, T, s_0)$ as the morphisms $(\sigma, \eta) : (S, E, T) \to \tau_{CEN}$ such that $\forall e_1, \ e_2 \in E \quad e_1 \| e_2 \Rightarrow (\eta(e_1) = \texttt{nop}) \ \vee \ (\eta(e_2) = \texttt{nop})$. This is consistent with the usual definition of independence of events in C/E nets, according to which $e_1 \| e_2$ if and only if $({}^\bullet e_1 \cup e_1{}^\bullet) \cap ({}^\bullet e_2 \cup e_2{}^\bullet) = \emptyset$. Nielsen and Winskel show that the asynchronous automata which are generated from C/E net systems with this definition of independence are exactly those in which the separation properties SSP and ESSP are satisfied w.r.t. the asynchronous regions.

If we apply now Theo. 6.12 to the type τ_{TRN}, we retrieve the characterization of marking graphs of trace nets established in [5]. Given a trace automaton $\mathcal{A} = (S, E, \|, T, s_0)$ with an empty relation of independence, the trace regions of $\mathcal{A}$ defined in [5] are actually the morphisms $(\sigma, \eta) : (S, E, T) \to \tau_{TRN}$. We recall that a trace automaton according to the definition of Stark [42] is like an asynchronous automaton up to the removal of the commutation constraint. A typical trace automaton is shown in Fig. 16 together with a generating trace net. This trace automaton is not an asynchronous automaton since e.g. the

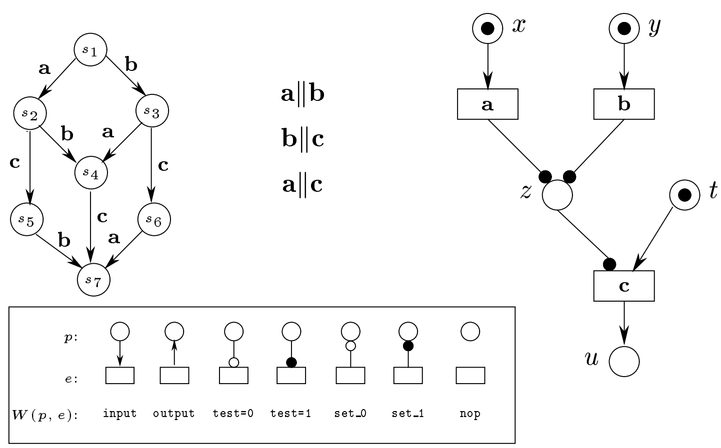


Fig. 16. a trace automaton and a generating trace net

sequence $a \cdot c$ can be fired from state s_1, but this is not the case with $c \cdot a$ although $a \| c$. The reader may verify that the separation problem ESSP(s_1, c) cannot be solved with elementary regions nor with asynchronous regions, but it is solved by the trace region corresponding to the place z of the trace net of Fig. 16. In the general case where the independence relation is not empty, a trace region of $\mathcal{A} = (S, E, \|, T, s_0)$ is defined as a trace region of the underlying automaton compatible with the independence of events in the sense that for any two independent events e_1 and e_2 one has *(i)* $\eta(e_1) \in \{\texttt{input}, \texttt{output}\} \Rightarrow \eta(e_2) = \texttt{nop}$ and *(ii)* $\eta(e_1) \in \{\texttt{test=1}, \texttt{set\_1}\} \Rightarrow \eta(e_2) \neq \texttt{set\_0}$. This is coherent with the independence of events in trace nets, defined similarly by $e_1 \| e_2$ if and only if for all places $p \in P$ *(i)* $W(p, e_1) \in \{\texttt{input}, \texttt{output}\} \Rightarrow W(p, e_2) = \texttt{nop}$ and *(ii)* $W(p, e_1) \in \{\texttt{test=1}, \texttt{set\_1}\} \Rightarrow W(p, e_2) \neq \texttt{set\_0}$. It is shown in [5] that the trace automata which are generated from trace net systems with this relation of independence are exactly those in which the separation properties SSP and ESSP are satisfied w.r.t. the trace regions. It is also shown that the finite trace automata which can be defined in the so-called *simple* format of Plotkin's Structural Operational Specification rules, with proofs of transitions as events and independence of proofs as independence of events, are exactly the finite and separated trace automata.

More will be said on the topic of independence in section 8.

6.6 Other Applications of Types

Types may serve alternatively to classify existing families of nets or to explore new families of nets. One may study hybrid types forged from existing ones by amalgamation, or by disjoint summation. One may study translations between classes of nets based on morphisms between their types. Theoretically speaking, this amounts to consider nets over a fixed set of events as a category indexed over the category of automata (their types). Compilation techniques for nets may also be defined on the following principle: let $N = (P, E, W)$ be a net of type τ, where τ is the marking graph of a net $N_\tau = (P_\tau, LE, W_\tau)$ of type τ', then N is equivalent to the net $N' = (P', E', W')$ of type τ' such that $P' = P \times P_\tau$, and $W'((p, p_\tau), e) = W_\tau(p_\tau, W(p, e))$. This amounts to consider nets as functors over automata, and composition of functors as compilation of nets.

7 Polynomial Time Algorithms for the Synthesis of Petri Nets

We present in this section the polynomial time algorithm proposed in [2] for the synthesis of pure Petri nets from finite automata. This algorithm has been implemented in the tool SYNET [13]. Next, we give a sketch of the variant algorithm for the synthesis of (general) Petri nets proposed in [7]. Finally, we indicate for both algorithms degenerated forms allowing to synthesize Petri nets from regular languages.

7.1 The Synthesis Problem for Pure Petri Nets

In the sequel, let $\mathcal{A} = (S, E, T, s_0)$ be a loop-free, reachable and reduced finite deterministic automaton, and let A denote the underlying transition system (S, E, T). The synthesis problem for pure Petri nets consists in *(i)* deciding whether an automaton $\mathcal{A}$, given as input, is isomorphic to the marking graph

$\mathcal{N}^*$ of some net system $\mathcal{N} = (P, E, W, M_0)$ of type τ_{PPN}, and if so *(ii)* producing as output a net system $\mathcal{N}$ such that $\mathcal{A} \cong \mathcal{N}^*$ and no proper subnet system of $\mathcal{N}$ satisfies this property. Recall that $\tau_{PPN} = (\mathbb{N}, \mathbb{Z}, \tau)$ with transitions $n \xrightarrow{z} n' \in \tau$ iff $n' = n + z$. On the grounds of Theo. 6.12, this amount to *(i)* deciding whether all instances of the separation problems in $\mathcal{A}$ can be solved by corresponding regions, and if so *(ii)* synthesizing the desired net system $\mathcal{N} = \mathcal{A}^*_{\mathcal{R}}$ from a minimal admissible subset of regions $\mathcal{R}$, where $\mathcal{A}^*_{\mathcal{R}} = (\mathcal{R}, E, W, M_0)$ with $W((\sigma, \eta), e) = \eta(e)$ and $M_0((\sigma, \eta)) = \sigma(s_0)$. Now, there is at most $|S|^2 - |S|$ possible inputs for the states separation problem:
$\mathrm{SSP}_{\mathcal{A}}(s, s')$: "*construct from $\mathcal{A}$ and $s \neq s'$ a region (σ, η) s.t. $\sigma(s) \neq \sigma(s')$*"
and at most $|S| \times |E|$ instances of the event/state separation problem:
$\mathrm{ESSP}_{\mathcal{A}}(s, e)$: "*construct from $\mathcal{A}$ and $(s \overset{e}{\not\rightarrow})$ a region (σ, η) s.t. $(\sigma(s) \overset{\eta(e)}{\not\longrightarrow})$*".
Part *(i)* of the problem will therefore be solved in time polynomial (in $|S|$ and $|E|$) as soon as $\mathrm{SSP}_{\mathcal{A}}(s, s')$ and $\mathrm{ESSP}_{\mathcal{A}}(s, e)$ are solved in polynomial time. Part *(ii)* consists in extracting from a set of regions with size polynomial in $|S|$ and $|E|$ a minimal admissible subset and this certainly can be done in polynomial time. So, a polynomial algorithm for the synthesis of pure Petri nets will follow if we succeed to construct procedures that solve in polynomial time $\mathrm{SSP}_{\mathcal{A}}(s, s')$ and $\mathrm{ESSP}_{\mathcal{A}}(s, e)$ with respect to the type τ_{PPN}. This is the main program of the section. The first stage of the program is to study the algebraic properties of the set of pure Petri regions of $\mathcal{A}$. The second stage of the program is to elaborate decision procedures based on these properties.

7.2 The Structure of Pure Petri Regions

Let $\mathcal{R}_{PPN}(\mathcal{A})$ denote the set of pure Petri regions of $\mathcal{A}$, i.e. the set of morphisms $(\sigma, \eta) : A \to \tau_{PPN}$. Before investigating the algebraic properties of $\mathcal{R}_{PPN}(\mathcal{A})$, let us recall some terminology borrowed from algebraic topology (see e.g. [31]). In the fixed transition system $A = (S, E, T)$, let $\partial^0, \partial^1 : T \to S$ and $\ell : T \to E$ denote the respective *source*, *target*, and *labelling* functions given by $\partial^0(t) = s$, $\partial^1(t) = s'$, and $\ell(t) = e$ for $t = s \xrightarrow{e} s' \in T$. A *0-chain* of A is a vector in the free $\mathbb{Z}$-module $C_0(A) = \mathbb{Z}{<}S{>}$ [3] . A *1-chain* of A is a vector in the free $\mathbb{Z}$-module $C_1(A) = \mathbb{Z}{<}T{>}$. The *boundaries* of the 1-chains are the 0-chains computed by the operator $\partial : C_1(A) \to C_0(A)$ such that $\partial(\Sigma z_j \cdot t_j) = \Sigma z_j \cdot (\partial^1(t_j) - \partial^0(t_j))$. The *co-boundaries* of the 0-chains are the 1-chains computed by the operator $\partial^* : C_0(A) \to C_1(A)$ such that $\partial^*(\sum z_i \cdot s_i) = \sum z_i \cdot \partial^*(s_i)$ where $\partial^*(s_i) = \sum\{t_j | \partial^1(t_j) = s_i\}$ - $\sum\{t_j | \partial^0(t_j) = s_i\}$. [4] A *cycle* of A is a 1-chain with a null boundary, and a *co-cycle* is a 0-chain with a null co-boundary. The cycles of A, resp. the

[3] we recall that the free $\mathbb{Z}$-module generated by a finite set $X = \{x_1, \ldots, x_n\}$ of generators is the set of maps α from X to $\mathbb{Z}$, viewed as vectors indexed by X with entries in $\mathbb{Z}$ and represented as formal sums $\alpha = \sum_i \alpha_i \cdot x_i$ where $\alpha(x_i) = \alpha_i$.

[4] the dual linear operators ∂ and ∂^* are associated respectively with $-A$ and its transpose $-A^t$, where A is the incidence matrix of the underlying graph. This change of sign is not technically significant and comes from different usages in the literature on graphs: the definition of the incidence matrix of a directed graph that we gave corresponds to the one used in [10, 16, 17], whereas Lefschetz [31] and Tutte [40]

co-boundaries of A, form submodules $\mathcal{V}_B$ resp. $\mathcal{V}_Q$ of $C_1(A)$ which are orthogonal complements. Linear bases for $\mathcal{V}_B$ and $\mathcal{V}_Q$ are supplied by the respective sets of fundamental cycles and fundamental cutsets of the underlying graph (S,T) w.r.t. some spanning tree $U \subseteq T$. Thus every cycle may be written as a linear combination $\sum z_i \cdot B_i$ of fundamental cycles $B_i : T \to \{-1,0,1\}$, and every co-boundary may be written as a linear combination $\sum z_i \cdot C_i$ of fundamental cutsets $C_i : T \to \{-1,0,1\}$, with integral coefficients $z_i \in \mathbb{Z}$. The Parikh images of the cycles form in turn a submodule of the free $\mathbb{Z}$-module $\mathbb{Z}<E>$, where the Parikh mapping $\pi : \mathbb{Z}<T> \to \mathbb{Z}<E>$ is the linear transformation given by $\pi(\sum z_i \cdot t_i) = \sum z_i \cdot \ell(t_i)$. In the sequel, the maps $\eta : E \to \mathbb{Z}$ are represented accordingly as formal sums $\eta = \sum z_i \cdot e_i$ where $z_i = \eta(e_i)$. For any two vectors $\alpha = \sum \alpha_i \cdot x_i$ and $\beta = \sum \beta_i \cdot x_i$ in a finite dimensional free $\mathbb{Z}$-module $\mathbb{Z}<X>$, we let $\alpha \cdot \beta$ denote the scalar product $\sum \alpha_i \cdot \beta_i \in \mathbb{Z}$.

Proposition 7.1 *$(\sigma,\eta) \in \mathcal{R}_{PPN}(A)$ if and only if $\sigma \cdot \partial(c) = \eta \cdot \pi(c)$ for all $c \in C_1(A)$.*

Proof: By linearity, the condition $\forall c \in C_1(A) \;\; \sigma \cdot \partial(c) = \eta \cdot \pi(c)$ is equivalent to the condition $\forall t \in T \;\; \sigma \cdot \partial(t) = \eta \cdot \pi(t)$ where t is identified with the chain $(1.t)$. Now the equation $\sigma \cdot \partial(t) = \eta \cdot \pi(t)$ is valid if and only if $\sigma(\partial^1(t))$ - $\sigma(\partial^0(t)) = \eta(\ell(t))$, if and only if $\sigma(\partial^0(t)) \stackrel{\eta(\ell(t))}{\longrightarrow} \sigma(\partial^1(t))$ w.r.t. the type τ_{PPN}, if and only if $(\sigma,\eta) \in \mathcal{R}_{PPN}(A)$ by definition of regions. ∎

Proposition 7.2 *A map $\eta : E \to \mathbb{Z}$ is the second projection of some region $(\sigma,\eta) \in \mathcal{R}_{PPN}(A)$ if and only if $\eta \cdot \pi(B) = 0$ for every cycle $B \in \mathcal{V}_B$; the regions $(\sigma,\eta) \in \mathcal{R}_{PPN}(A)$ which project on η are then characterized by the condition: $\sigma(s_0) + (\eta \cdot \pi(c)) \geq 0$ for every 1-chain $c \in C_1(A)$ such that $\partial(c) = s - s_0$ for some $s \in S$.*

Proof: From Prop. 7.1, the condition on η must hold and whenever it does, the scalar product $\eta \cdot \pi(c)$ takes an identical value for all 1-chains c with an identical boundary. From the definition of regions, the condition on $\sigma(s_0)$ must hold because the local states specified for the type of nets τ_{PPN} are the non negative integers. Now the two conditions taken together guarantee that one can always complete the data $(\sigma(s_0),\eta)$ to a pure region by selecting for each state $s \in S$ a corresponding 1-chain c_s such that $\partial(c_s) = s - s_0$ and then setting $\sigma(s) = \sigma(s_0) + \eta \cdot \pi(c_s)$, which is always possible since $\mathcal{A}$ is reachable. ∎

Let $\mathcal{R}_{\text{abs}}(A)$ denote the set of maps $\eta : E \to \mathbb{Z}$ characterized by Prop. 7.2, henceforth called *abstract regions*. It appears from this characterization that the abstract regions of A are in bijective correspondence with the co-boundaries of A which are compatible with the kernel of the labelling function $\ell : T \to E$. Actually, for every abstract region $\eta : E \to \mathbb{Z}$, the map $\kappa = \eta \circ \ell : T \to \mathbb{Z}$ is a

use the opposite matrix. In the same manner what we term co-boundary, following Lefschetz and Tutte and more generally those authors who identify graphs with 1-dimensional complexes, are called cocycles in many books on graph theory.

co-boundary of A, since for every cycle B, $\kappa \cdot B = \sum_t \eta(\ell(t)) \cdot B(t) = \sum_e \eta(e) \cdot \sum_{\ell(t)=e} B(t) = \eta \cdot \pi(B) = 0$. Conversely, every co-boundary $\kappa : T \to \mathbb{Z}$ such that $\ell(t) = \ell(t') \Rightarrow \kappa(t) = \kappa(t')$ determines a unique abstract region $\eta : E \to \mathbb{Z}$ such that $e = \ell(t) \Rightarrow \eta(e) = \kappa(t)$, since $\ell : T \to E$ is surjective and $\mathcal{A}$ is reduced.

An abstract region η determines a unique region (σ, η) such that $\sigma(s) = 0$ for some state s, called a *strict* region and given by $\sigma(s_0) = -\mathbf{min}\{\eta \cdot \pi(c) | \; \exists s \in S \quad \partial(c) = s - s_0\}$, and an infinite family of non strict regions $(\sigma + h, \eta)$ for $h \in \mathbb{N} \setminus \{0\}$. Now any instance of the separation problems $\mathrm{SSP}_{\mathcal{A}}(s, s')$ or $\mathrm{ESSP}_{\mathcal{A}}(s, e)$ solved by $(\sigma + h, \eta)$ is also solved by (σ, η). For this reason, let us concentrate on strict regions, or equivalently on abstract regions.

The set $\mathcal{R}_{\mathrm{abs}}(A)$ of abstract regions of A is obviously a $\mathbb{Z}$-module. From Prop. 7.2, a linear basis for this module may be computed as follows. Let $S = \{s_1, \ldots, s_n\}$, $T = \{t_1, \ldots, t_m\}$, and $E = \{e_1, \ldots, e_p\}$. Let $U \subseteq T$ be a spanning tree of the underlying graph $G = (S, T)$, and let $\{B_1, \ldots, B_{m-n+1}\}$ be the set of fundamental cycles of G w.r.t. U. Thus $\{B_1, \ldots, B_{m-n+1}\}$ is a basis for $\mathcal{V}_B$ and $\mathcal{R}_{\mathrm{abs}}(A)$ is the kernel of the linear transformation $M_A : \mathbb{Z}^p \to \mathbb{Z}^{m-n+1}$ defined by the $(m - n + 1) \times p$ matrix M_A with integral coefficients

$$M_A(i, j) = \Sigma\{B_i(t_k) | \; 1 \leq k \leq m \quad \wedge \quad \ell(t_k) = e_j\}$$

Let k be the dimension of $Ker(M_A)$. The algorithm of von zur Gathen and Sieveking (see [38]), given M_A as input, produces in time polynomial in $m - n + 1$ and p (or $|S| = n$ and $|E| = p$, because $m \leq n \times p$ follows from determinism of A) a basis $\{\eta_1, \ldots, \eta_k\}$ for $Ker(M_A) = \mathcal{R}_{abs}(A)$.

We have in hand all the elements needed for solving problems $SSP_A(s, s')$ and $ESSP_A(s, e)$ relatively to the type of pure Petri nets. The data needed are the spanning tree U, or more exactly the application $c_{(.)}$ that maps each state $s \in S$ to the unique chain c_s from s_0 to s in U, and the basis of abstract regions $\{\eta_1, \ldots, \eta_k\}$.

For the sake of illustration, let us exhibit these data for the automaton $\mathcal{A}$ shown in Fig. 17. Here $n = 8$, $m = 14$, and $p = 6$. The spanning tree U, indicated

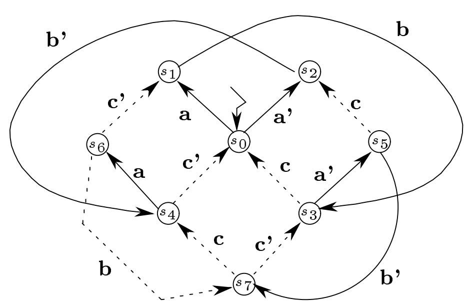


Fig. 17. an automaton with one of its spanning trees (in solid lines)

in solid lines, contains $n - 1 = 7$ transitions, The module $\mathcal{V}_B$ is generated from the $m - n + 1 = 7$ fundamental cycles B_i defined by the respective chords t_i indicated in dashed lines, let $t_1 = s_5 \xrightarrow{c} s_2$, $t_2 = s_3 \xrightarrow{c} s_0$, $t_3 = s_7 \xrightarrow{c} s_4$, $t_4 = s_6 \xrightarrow{c'} s_1$, $t_5 = s_4 \xrightarrow{c'} s_0$, $t_6 = s_7 \xrightarrow{c'} s_3$, and $t_7 = s_6 \xrightarrow{b} s_7$. For instance, the chord t_1 defines the fundamental cycle

$$B_1 = (s_0 \xrightarrow{a} s_1) + (s_1 \xrightarrow{b} s_3) + (s_3 \xrightarrow{a'} s_5) + (s_5 \xrightarrow{c} s_2) - (s_0 \xrightarrow{a'} s_2)$$

whose Parikh image is $\pi(B_1) = a + b + c$. One can verify that $\pi(B_1) = \pi(B_2) = \pi(B_3) = a + b + c$, $\pi(B_4) = \pi(B_5) = \pi(B_6) = a' + b' + c'$, and $\pi(B_7) = 0$. The $\mathbb{Z}$-module of abstract regions consists of those vectors $\eta : E \to \mathbb{Z}$ such that:

$$\eta(a) + \eta(b) + \eta(c) = 0 \quad \text{and} \quad \eta(a') + \eta(b') + \eta(c') = 0$$

It is therefore a four dimensional $\mathbb{Z}$-module with basis as follows:

$$\eta_1 = a - c \ ; \ \eta_2 = b - c \ ; \ \eta_3 = a' - c' \ \text{and} \ \eta_4 = b' - c'$$

In this example, the spanning tree U is rooted at the initial state s_0 of the automaton. Let c_s denote the branch of U from s_0 to s and let $\pi_s = \pi(c_s)$ be its Parikh image. Thus, we have:

$$\begin{array}{llll} \pi_{s_0} = 0 & \pi_{s_1} = a & \pi_{s_2} = a' & \pi_{s_3} = a + b \\ \pi_{s_4} = a' + b' & \pi_{s_5} = a + b + a' & \pi_{s_6} = a' + b' + a & \pi_{s_7} = a + b + a' + b' \end{array}$$

The corresponding scalar products $\eta_i \cdot \pi_s$ are tabulated in Table 2

Table 2. states $s \in S$ represented by vectors $(\eta_i \cdot \pi_s)_i$ indexed by the set of basic abstract regions η_i

$\eta_i \cdot \pi_s$	π_{s_0}	π_{s_1}	π_{s_2}	π_{s_3}	π_{s_4}	π_{s_5}	π_{s_6}	π_{s_7}
η_1	0	1	0	1	0	1	1	1
η_2	0	0	0	1	0	1	0	1
η_3	0	0	1	0	1	1	1	1
η_4	0	0	0	0	1	0	1	1

7.3 Solving the separation problems

Let s and s' be two distinct states. From Prop. 7.1 and Prop. 7.2, $\text{SSP}_{\mathcal{A}}(s, s')$ has a solution in $\mathcal{R}_{PPN}(\mathcal{A})$ *iff* $\eta \cdot \pi(c_s - c_{s'}) \neq 0$ for some abstract region $\eta \in \mathcal{R}_{abs}(A)$ *iff* $\eta_i \cdot \pi(c_s - c_{s'}) \neq 0$ for some $i \in \{1, \cdots, k\}$, and the strict region (σ_i, η_i) determined from the basic abstract region η_i by setting $\sigma_i(s_0) = -\mathbf{min}\{\eta_i \cdot \pi(c_s) \mid s \in S\}$ is then a solution. Therefore, deciding whether $\text{SSP}_{\mathcal{A}}(s, s')$ has a solution and producing it takes time polynomial in $|S|$ and $|E|$.

In our running example all instances of the separation problem $\text{SSP}_{\mathcal{A}}(s, s')$ can be solved, because all the columns of table 2 are different.

Given $s' \in S$ and $e \in E$ such that $s' \stackrel{e}{\not\rightarrow}$, let us now consider the separation problem $\text{ESSP}_{\mathcal{A}}(s', e)$. From Prop. 7.2, this problem has a solution in $\mathcal{R}_{PPN}(\mathcal{A})$ *iff* there exists $\sigma(s_0) \in I\!N$ and $\eta \in \mathcal{R}_{abs}(A)$ such that

$$\forall s \in S \quad \sigma(s_0) + \eta \cdot \pi(c_s) \geq 0 \tag{1}$$

$$\sigma(s_0) + \eta \cdot \pi(c_{s'}) + \eta(e) < 0 \tag{2}$$

iff there exists $\eta \in \mathcal{R}_{abs}(A)$ satisfying the condition

$$\forall s \in S \quad \eta \cdot (\pi(c_{s'}) - \pi(c_s)) + \eta(e) < 0 \tag{3}$$

Whenever η satisfies condition 3, the strict region (σ, η) defined from η satisfies actually conditions 1 and 2 and therefore solves $\text{ESSP}_{\mathcal{A}}(s', e)$. Let $\eta = \sum_{i=1}^{k} x_i \cdot \eta_i$ where $\{\eta_1, \ldots, \eta_k\}$ is the basis of abstract regions, and $x_i \in Z\!\!\!Z$. For every $s \in S$, let $\alpha_i^s = \eta_i \cdot (\pi(c_{s'}) - \pi(c_s)) + \eta_i(e)$. With these notations, condition 3 may be rewritten to the system of linear inequations $\{\sum_{i=1}^{k} \alpha_i^s \cdot x_i < 0 |\ s \in S\}$ in the variables $x_i \in Z\!\!\!Z$. Now a system of linear inequations

$$MX \leq (-1)^n \tag{4}$$

where M is an integral matrix and $(-1)^n = < -1, \ldots, -1 >$ $(\in Z\!\!\!Z^n)$ has an integral solution iff it has a rational solution. The method of Khachiyan (see [38] p.170) may be used to decide upon the feasability of (4) and to compute a rational solution, if it exists, in polynomial time. Thus, every instance of the problem $\text{ESSP}_{\mathcal{A}}(s', e)$ is solved up to a multiplicative factor, or shown unfeasible, in time polynomial in $|S|$ and $|E|$. In our running example, the system of linear inequations which express the separation problem $\text{ESSP}_{\mathcal{A}}(s_2, a)$ is the following:

$$\begin{array}{lr}
\eta \cdot (\pi_{s_2} - \pi_{s_0}) + \eta(a) < 0 : & x_1 + x_3 < 0 \\
\eta \cdot (\pi_{s_2} - \pi_{s_1}) + \eta(a) < 0 : & x_3 < 0 \\
\eta \cdot (\pi_{s_2} - \pi_{s_2}) + \eta(a) < 0 : & x_1 < 0 \\
\eta \cdot (\pi_{s_2} - \pi_{s_3}) + \eta(a) < 0 : & x_3 - x_2 < 0 \\
\eta \cdot (\pi_{s_2} - \pi_{s_4}) + \eta(a) < 0 : & x_1 - x_4 < 0 \\
\eta \cdot (\pi_{s_2} - \pi_{s_5}) + \eta(a) < 0 : & -x_2 < 0 \\
\eta \cdot (\pi_{s_2} - \pi_{s_6}) + \eta(a) < 0 : & -x_4 < 0 \\
\eta \cdot (\pi_{s_2} - \pi_{s_7}) + \eta(a) < 0 : & -x_2 - x_4 < 0
\end{array}$$

This system is solvable, and admits in particular the solution $x_1 = x_3 = -1$ and $x_2 = x_4 = 1$. Therefore, $\eta = -\eta_1 + \eta_2 - \eta_3 + \eta_4 = -a + b - a' + b'$ satisfies condition 3, and (σ, η) solves $\text{ESSP}_{\mathcal{A}}(s_2, a)$ with $\sigma(s_0) = 1$. The automaton of Fig. 17 is actually separated by the set of strict regions (σ_η, η) which are indicated in Table 3, computed from SYNET. The pure Petri net synthesized from this set of admissible regions is shown in Fig. 18. For full precision, it should be said that SYNET [13] does not relie on the method of Khachiyan but on the simplex method which has cubic complexity in the average (see [38]). A quite different solution to the synthesis problem of pure Petri nets from finite automata up to a quotient is described in [30]. This solution is based on the investigation of minimal regions.

Table 3. values taken on states by strict regions

$\sigma_\eta(s)$	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
$-\eta_1$	1	0	1	0	1	0	0	0
η_2	0	0	0	1	0	1	0	1
$-\eta_3$	1	1	0	1	0	0	0	0
η_4	0	0	0	0	1	0	1	1
$\eta_1 - \eta_2$	0	1	0	0	0	0	1	0
$\eta_3 - \eta_4$	0	0	1	0	0	1	0	0
$\eta_2 + \eta_4 - \eta_1 - \eta_3$	1	0	0	1	1	0	0	1

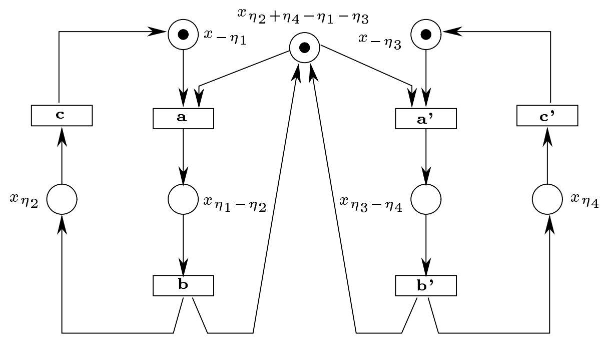


Fig. 18. the net synthesized from the admissible set of strict regions given in table 3

The key observation is the following: let $(\sigma_1, \eta_1), (\sigma_2, \eta_2) \in \mathcal{R}_{PPN}(\mathcal{A})$ such that $\forall s \in S \quad \sigma_1(s) \geq \sigma_2(s)$, then $(\sigma_1 - \sigma_2, \eta_1 - \eta_2)$ is a region of $\mathcal{A}$ and any instance of the separation problems which is solved by (σ_1, η_1) is solved either by (σ_2, η_2) or by $(\sigma_1 - \sigma_2, \eta_1 - \eta_2)$. Therefore, $\mathcal{A}$ is separated if and only if the set of its minimal regions is admissible.

7.4 The Case of General Petri Nets

A polynomial time algorithm for the synthesis of general Petri nets from finite automata was proposed in [7]. This algorithm is a modified form of the algorithm just described for pure Petri nets. We indicate below the main adaptations leading to the modified algorithm.

In the sequel, $\mathcal{A}$ is a reachable and reduced finite deterministic automaton, not necessarily simple. A region of $\mathcal{A}$ w.r.t. the type τ_{PN} of Petri nets is a morphism $(\sigma, ({}^\bullet\eta, \eta^\bullet)) : A \to \tau_{PN}$, called a Petri region, where ${}^\bullet\eta$ and $\eta^\bullet$ are maps from E to $\mathbb{N}$. As it was observed in [21], a Petri region is entirely determined from σ and ${}^\bullet\eta$ or alternatively from $\sigma(s_0)$, ${}^\bullet\eta$, and $\eta^\bullet$. Petri regions and pure Petri regions are connected by a pair of maps $J_{\mathcal{A}} : \mathcal{R}_{PN}(\mathcal{A}) \to \mathcal{R}_{PPN}(\mathcal{A})$ and $I_{\mathcal{A}} : \mathcal{R}_{PPN}(\mathcal{A}) \to \mathcal{R}_{PN}(\mathcal{A})$, such that $J_{\mathcal{A}}(\sigma, ({}^\bullet\eta, \eta^\bullet)) = (\sigma, \eta^\bullet - {}^\bullet\eta)$ and $I_{\mathcal{A}}(\sigma, \eta) = (\sigma, ({}^\bullet\eta, \eta^\bullet))$ where ${}^\bullet\eta(e) = \mathbf{max}\{0, -\eta(e)\}$ and $\eta^\bullet(e) = \mathbf{max}\{0, \eta(e)\}$.

Owing to this correspondence, an instance of the separation problem $\text{SSP}_{\mathcal{A}}(s, s')$ can be solved in $\mathcal{R}_{PN}(\mathcal{A})$ if and only if it can be solved in $\mathcal{R}_{PPN}(\mathcal{A})$. The respective solutions are actually connected by the maps $I_{\mathcal{A}}$ and $J_{\mathcal{A}}$.

The treatment of the event state separation problem is more delicate. An instance of $\text{ESSP}_{\mathcal{A}}(s', e)$ can be solved in $\mathcal{R}_{PN}(\mathcal{A})$ if and only if there exists $\sigma(s_0) \in \mathbb{N}$, ${}^\bullet\eta(e) \in \mathbb{N}$, and $\eta \in \mathcal{R}_{abs}(A)$ such that:

1. $\eta(e) + {}^\bullet\eta(e) \geq 0$
2. $\forall s \in S \quad \sigma(s_0) + \eta \cdot \pi(c_s) \geq 0$
3. $\forall s \in S \quad s \overset{e}{\to} \ \Rightarrow \sigma(s_0) + \eta \cdot \pi(c_s) \geq {}^\bullet\eta(e)$
4. $\sigma(s_0) + \eta \cdot \pi(c'_s) < {}^\bullet\eta(e)$

A solution is then given by the Petri region $(\sigma, ({}^\bullet\eta, \eta^\bullet))$ defined by ${}^\bullet\eta(e') = \mathbf{max}\{0, -\eta(e')\}$ for $e' \neq e$, and $\eta^\bullet(e') = \eta(e') + {}^\bullet\eta(e')$ for every $e' \in E$. Set $x = \sigma(s_0)$, $y = {}^\bullet\eta(e)$, and $\eta = \sum x_i \cdot \eta_i$ where $\{\eta_1, \ldots, \eta_k\}$ is the basis of $\mathcal{R}_{abs}(A)$ and $x_i \in \mathbb{Z}$. With these notations, the above conditions may be rewritten to a system of linear inequations in the $k+2$ variables x, y and x_i $(1 \leq i \leq k)$, where $w_{is} = \eta_i \cdot \pi(c_s)$:

1. $y + \sum z_i \eta_i(e) \geq 0$
2. $x + \sum z_i w_{is} \geq 0$ (one inequation for each $s \in S$)
3. $x - y + \sum z_i w_{is} \geq 0$ (one inequation for each $s \in S$ such that $s \overset{e}{\to}$)
4. $x - y + \sum z_i w_{is'} < 0$

This system, augmented with the constraints $x \geq 0$ and $y \geq 0$, is homogeneous and can therefore be solved or shown unfeasible in polynomial time following Khachiyan's method.

7.5 Synthesizing Bounded Net Systems up to Language Equivalence

A net system is termed bounded if its marking graph is finite. Given a reachable and reduced finite deterministic automaton $\mathcal{A}$, let $\mathcal{L}(\mathcal{A})$ denote the (prefix closed) language of words accepted by $\mathcal{A}$. We face now the problem of deciding whether $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{N}^*)$ for some bounded net system and if so constructing $\mathcal{N}$. This problem can be decided upon in polynomial time, for both types τ_{PPN} and τ_{PN}, when $\mathcal{A}$ is given in tree-like form.

Definition 7.3 $\mathcal{A} = (S, E, T, s_0)$ *is a tree-like automaton if there exists a spanning tree* $U \subseteq T$ *rooted at* s_0*, with all transitions in* U *directed away from* s_0*, such that for every chord* $s \overset{e}{\to} s' \notin U$*,* s' *is an ancestor of* s *in* U*.*

Suppose $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{N}^*)$, where $\mathcal{A}$ is tree-like and $\mathcal{N} = (P, E, W, M_0)$ is a bounded net system with type $\tau \in \{\tau_{PPN}, \tau_{PN}\}$. For each place $p \in P$, let $(\sigma_p, \eta_p) : \mathcal{N}^* \to \tau$ denote the associated region of $\mathcal{N}^*$. From the inclusion $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{N}^*)$ and the assumption of boundedness of $\mathcal{N}$, $\sigma'_p(M_0) = \sigma_p(M_0)$ and η_p define a region in $(\sigma'_p, \eta_p) : \mathcal{A} \to \tau$ (seeing that $s \overset{u}{\to} s$ [5] in $\mathcal{A}$ entails $\eta_p \cdot \pi(u) = 0$

[5] $s \overset{\varepsilon}{\to} s$ for every $s \in S$, where ε is the empty word, and $s \overset{u \cdot e}{\to} s' \Leftrightarrow \exists s'' \in S \quad s \overset{u}{\to} s'' \wedge s'' \overset{e}{\to} s'$.

for $\tau = \tau_{PPN}$). From the relation $\mathcal{L}(\mathcal{N}^*) = \mathcal{L}(\mathcal{A})$, every instance of the event state separation problem $\mathrm{ESSP}_{\mathcal{A}}(s', e)$ is solved by a region $(\sigma'_p, \eta_p) \in \mathcal{R}_\tau(\mathcal{A})$ defined as above from some corresponding place $p \in P$. Conversely, if the condition ESSP is valid in $\mathcal{A}$, then $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{N}^*)$ for any net system $\mathcal{N} = \sum_{p \in P} \mathcal{N}_p$ assembled from a subset of τ-regions of $\mathcal{A}$ admissible for ESSP. Therefore, in the restricted case of tree-like automata, the synthesis problem for pure or impure Petri nets up to language equivalence can be solved in polynomial time [2].

Now, every deterministic automaton $\mathcal{A} = (S, E, T, s_0)$ may be translated to an equivalent tree-like automaton $\mathcal{A}' = (S', E, T', (s_0, \varepsilon))$ with sets of states and transitions defined as follows.

- S' is the set of pairs $(s, u) \in S \times \mathcal{L}(\mathcal{A})$ such that $s_0 \xrightarrow{u} s$ in $\mathcal{A}$ and every two states of $\mathcal{A}$ visited in this path are different;
- $T' \subseteq S' \times E \times S'$ is the set of transitions $(s, u) \xrightarrow{e} (s', u')$ such that $s \xrightarrow{e} s'$ in $\mathcal{A}$ and $u' = u \cdot e$ or u' is a prefix of u.

It must be noted, however, that the size of the tree-like automaton $\mathcal{A}'$ constructed in this way is *exponential* in the size of $\mathcal{A}$ (so as the number of elementary circuits of $\mathcal{A}$, which shows that the case of general automata cannot be dealt with by polynomial algorithms).

8 Regions in Step Transition Systems

We leave now the classical frame of (sequential) transition systems for the more expressive frame of *step transition systems*, defined by Mukund so as to account fully for the independence of events in general Petri nets [33].

Definition 8.1 *A* step transition system (S, M, T) *over an abelian monoid* M *consists of a set of states* S *and a deterministic transition relation* $T \subseteq S \times M \times S$, *with distinguished* empty steps: $s \xrightarrow{0} s'$ *iff* $s = s'$. *A* step automaton $\mathcal{A}$ *is an initialized step transition system* (S, M, T, s_0) *with initial state* $s_0 \in S$, *such that every state* $s \in S$ *is reachable from* s_0 *in the underlying transition system* $A = (S, M, T)$. *The step automaton* $\mathcal{A}$ *is* finite *if the set of transitions* T *is finite. When* $M = <E>$ *is the free abelian monoid freely generated from set* E *(the elements of* M *are then finite multisets over* E*), the step automaton* $\mathcal{A}$ *is said to be* reduced *if its* skeleton $(S, E, T \cap (S \times E \times S), s_0)$ *is a reduced automaton.*

This definition of step transition systems extends slightly Mukund's original definition, which was restricted to free abelian monoids. The extension allows to accomodate the idea of regions as morphisms to step transition systems which do not necessarily present the *intermediate state* property: $s \xrightarrow{\alpha+\beta} s' \Rightarrow \exists s'' \in S \; s \xrightarrow{\alpha} s'' \wedge s'' \xrightarrow{\beta} s'$. The definition of regions in step transition systems is parametric on enriched types of nets defined as follows.

Definition 8.2 *An enriched type of nets is a (deterministic) step transition system* $\tau = (LS, LE, \tau)$, *where* LE *is an abelian monoid* $(LE, +, 0)$.

For instance, the enriched type of Petri nets is just the type $\tau_{PN} = (I\!N, I\!N \times I\!N, \tau)$, where $n \xrightarrow{(i,j)} n' \in \tau$ if and only if $n \geq i$ and $n' = n - i + j$, enriched with the operation of componentwise addition in $I\!N \times I\!N$. As a matter of fact, $(I\!N \times I\!N, +, (0,0))$ is the free abelian monoid generated from $(0,1)$ and $(1,0)$.

Each type of nets determines a specific concurrent firing rule and hence a specific construction of concurrent marking graphs.

Definition 8.3 *Given a net $N = (P, E, W)$ with (enriched) type $\tau = (LS, LE, \tau)$, the concurrent marking graph of N is the step transition system $(LS^P, <E>, T)$ with set of transitions T defined by :*

$$(M \stackrel{\alpha}{\to} M') \in T \;\Leftrightarrow\; \forall x \in P \;\; (M(x) \stackrel{W(x,\alpha)}{\longrightarrow} M'(x)) \in \tau \tag{5}$$

where $W(x, e_1 + \cdots + e_n) = W(x, e_1) + \cdots + W(x, e_n)$. Given a net system $\mathcal{N} = (P, E, W, M_0)$, the concurrent marking graph of $\mathcal{N}$ is the step automaton $\mathcal{N}^* = (S, <E>, T_S, M_0)$ where S is the inductive closure of the singleton set $\{M_0\}$ w.r.t. forward transitions in T, and $T_S = T \cap (S \times <E> \times S)$.

In order to illustrate this definition, let us inspect the relationship between the sequential and concurrent marking graphs of a Petri net. On the one hand, the sequential marking graph is the induced restriction of the concurrent marking graph on the subset of *atomic* steps, i.e. steps α such that $\Sigma_{e \in E}\, \alpha(e) = 1$. On the other hand, the concurrent marking graph cannot in general be reconstructed up to isomorphism from an arbitrary copy of the sequential marking graph, even though some additional informations are provided as in [20] by a binary relation of independence on events depending on markings, such that $e \,\|_M\, e'$ if and only if $M[\, \{e, e'\} \,>$. The example shown in Fig. 19, borrowed from [27], makes this fact clear. Regions may now be introduced, based on the following definition of

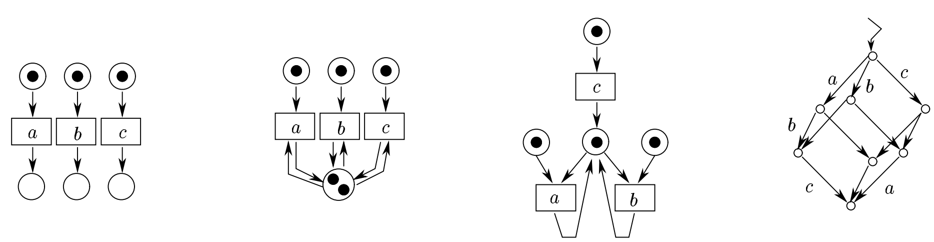


Fig. 19. three nets with an identical sequential marking graph but with different concurrent marking graphs: the three events a, b, and c are independent at the indicated marking in the first net whereas they are pairwise independent but not independent in the second net; the case of the third net is more involved: at the indicated marking the maximal sets of independent events are $\{a, c\}$ and $\{b, c\}$, but a and b become independent once c has been fired. Thus independence in Petri nets is marking dependent.

morphisms of step transition systems.

Definition 8.4 *A morphism of step transition systems from $A = (S, M, T)$ to $A' = (S', M', T')$ is a pair (σ, η), made of a map $\sigma : S \to S'$ and a monoid morphism $\eta : M \to M'$, such that $s \xrightarrow{\alpha} s' \Rightarrow \sigma(s) \xrightarrow{\eta(\alpha)} \sigma(s')$. The* morphisms of step automata *from $\mathcal{A}$ to $\mathcal{A}'$ are the morphisms from A to A' that preserve the initial state.*

Definition 8.5 *Given a step transition system $A = (S, M, T)$ and an enriched type of nets $\tau = (LS, LE, \tau)$, the set $\mathcal{R}_\tau(A)$ of τ-type (extended) regions of A is the set of morphisms of step transition systems from A to τ. The set of τ-type (extended) regions of a step automaton $\mathcal{A}$ is the set $\mathcal{R}_\tau(\mathcal{A}) = \mathcal{R}_\tau(A)$.*

By specializing this definition to the type τ_{PN}, one retrieves exactly the regions defined by Mukund in step transition systems over a free abelian monoid [33]. Special attention may be paid to the class of step transition systems $A = (S, M, T)$ derived from asynchronous transition systems $(S, E, \|, T')$ as follows: $M = <E>$ is the free abelian monoid generated by E (the elements of M are finite multisets of elements of E), $s \xrightarrow{\alpha} s'$ in T if and only if α is a sub*set* of pairwise independent events $\{e_1, \ldots, e_n\} \subseteq E$ (hence there is no auto-concurrency) and there exists in T' a sequence of transitions $s \xrightarrow{e_1} s_1 \xrightarrow{e_2} s_2 \ldots s_{n-1} \xrightarrow{e_n} s_n$ such that $s' = s_n$ (such sequences exist therefore for all permutations of $\{e_1, \ldots, e_n\}$). For this class of step transition systems, the regions $(\sigma, \eta) : A \to \tau_{PN}$ which are *safe* in the sense that $\sigma(s) \in \{0, 1\}$for all $s \in S$ are in bijective correspondence with the regions defined by Nielsen and Winskel for asynchronous transition systems [35].

The results about (ordinary) transition systems which have been presented in section 6 may be reproduced nearly intact in the richer setting of step transition systems over a free abelian monoid. In particular, Def. 6.6 and 6.7 and Prop. 6.11 may be extended to step transition systems, yielding a Galois connection $\mathcal{A} \leq \mathcal{N}^* \Leftrightarrow \mathcal{N} \leq \mathcal{A}^*$ between step automata $\mathcal{A} = (S, <E>, T, s_0)$ and net systems $\mathcal{N} = (P, E, W, M_0)$, for any enriched type of nets τ. The following counterpart to Theo. 6.12 for step transition systems appears in [7].

Theorem 8.6 *Given a step automaton over a free abelian monoid, let $\mathcal{A} = (S, <E>, T, s_0)$, and an enriched type of nets τ, a subset of extended regions $\mathcal{R} \subseteq \mathcal{R}_\tau(\mathcal{A})$ is admissible if and only if the following* separation properties *are satisfied for all states $s, s' \in S$ and for every multiset $\alpha \in <E>$:*

(SSP) $s \neq s' \Rightarrow \exists (\sigma, \eta) \in \mathcal{R} : \sigma(s) \neq \sigma(s')$

(ESSP) $s \stackrel{\alpha}{\not\to} \Rightarrow \exists (\sigma, \eta) \in \mathcal{R} : \sigma(s) \stackrel{\eta(\alpha)}{\not\longrightarrow}$ *in* τ

When both properties are satisfied, $\mathcal{A} \cong (\mathcal{A}^*_{\mathcal{R}})^*$, where $\mathcal{A}^*_{\mathcal{R}}$ is the subnet system of $\mathcal{A}^*$ with restricted set of places $\mathcal{R}$ (also called the net synthesized *from $\mathcal{R}$).*

Mukund's characterization of Petri net transition systems, established in [33], follows directly from Theo. 8.6 applied to the type τ_{PN}. Nielsen and Winskel's characterization of separated asynchronous automata, established in [35], follows therefrom as the subcase met when imposing on regions $(\sigma, \eta) \in \mathcal{R}$ the constraint that $\sigma(s) \in \{0, 1\}$ for every state s.

An algorithm for synthesizing Petri nets from finite step transition systems, based on Theo. 8.6, is proposed in [7]. This algorithm is an adaptation of the basic algorithm for pure Petri nets described in section 7. Let $\mathcal{A} = (S, <E>, T, s_0)$ be some finite and reduced step automaton. Seeing that the intermediate state property is always satisfied in the concurrent marking graph of a Petri net, we assume this property from $\mathcal{A}$. Thus, $(s \stackrel{\alpha+\beta}{\to} s') \in T \;\Rightarrow\; \exists s'' \in S \;(s \stackrel{\alpha}{\to} s'') \in T \wedge (s'' \stackrel{\beta}{\to} s') \in T$. The import is that we may assume a compact representation for $\mathcal{A}$, given by its skeleton and the set of maximal steps at each state $s \in S$. This makes sense since the set of steps of $\mathcal{A}$ is bounded, from the assumption that $\mathcal{A}$ is finite. As regards the event state separation problem, let us observe the following: if a region (σ, η) solves an instance $\text{ESSP}_{\mathcal{A}}(s, \alpha)$ of this problem, where α is a failure at s, then (σ, η) solves also every instance $\text{ESSP}_{\mathcal{A}}(s, \beta)$ such that $\alpha < \beta$. It is then sufficient to solve at each state s the instances $\text{ESSP}_{\mathcal{A}}(s, \alpha)$ such that α is a minimal failure in that state. From this remark and the assumed representation for $\mathcal{A}$, the following is proved in [7].

Theorem 8.7 *The synthesis problem for Petri net systems with the step firing rule, taking as inputs finite step transition systems, is polynomial in their numbers of states and events, in the size of the largest set of minimal failures in one state, and in the size of the largest set of maximal steps enabled in one state.*

Notice that the minimal failures are not determined at a given state by the maximal steps, as shown by the third net on Fig. 19 for which $Max\_steps(s_0) = \{a+c, b+c\}$ and $Min\_fails(s_0) = \{2a, a+b, 2b\}$ whence $Min\_fails(s) \not\subseteq \{\alpha + e | \alpha \in Max\_steps(s)\}$. Every step automaton may in fact be transformed to an ordinary automaton by splitting the alphabet of events: the states of the split automaton are the pairs $<s, \alpha>$ where α is a step with concession at s, and each transition $s \stackrel{e}{\to} s'$ gives rise to the pair of transitions $<s, \alpha> \stackrel{e^+}{\to} <s, \alpha + e>$ and $<s, \alpha + e> \stackrel{e^-}{\to} <s', \alpha>$ for every step $\beta = \alpha + e$ with concession at s. In [1] it is shown that the synthesis of Petri nets from step automata may be reduced to the synthesis of pure Petri nets from ordinary automata by splitting events, which yields a synthesis algorithm taking time polynomial in the number of higher-dimensional states. Now if $\beta \in Min\_fails(s)$ is a minimal failure, then $<s, \alpha> \stackrel{e^+}{\not\to}$ for any step α and event e such that $\beta = \alpha + e$, and the problem $\text{ESSP}_{\mathcal{A}}(s, \beta)$ is equivalent to the separation problem $\text{ESSP}_{\text{Split}(\mathcal{A})}(<s, \alpha>, e^+)$ for the event e^+ at the higher-dimensional state $<s, \alpha>$. There are (at most $|E|$ times) more instances of ESSP to be solved in the split automaton since β can be decomposed as $\beta = \alpha + e$ in several ways, but the total number of instances of the problem $\text{ESSP}_{\mathcal{A}}(s, \alpha)$ for minimal failure α in the step automaton is already exponential in the number of events.

In order to conclude with extended regions (σ, η) where η maps steps $\alpha \in <E>$ to (pairs of) integral weights, let us mention that such regions have also been used to solve different synthesis problems in the setting of generalized trace languages [26] and general event structures [27].

9 Adjunctions between Transition Systems and Nets

In section 6, a Galois connection $\mathcal{A} \leq \mathcal{N}^* \Leftrightarrow \mathcal{N} \leq \mathcal{A}^*$ between automata and net systems was established. However $\mathcal{A}^*$ and $\mathcal{N}^*$ are not constructed in a symmetrical way: $\mathcal{A}^*$ has been assembled from morphisms of transition systems from A to the type of nets τ, but $\mathcal{N}^*$ has not been constructed from net morphisms. We show in this section that $\mathcal{N}^*$ can equally well be assembled from net morphisms $\varphi : N \to \tau'$ where the places of τ' encode bijectively the transitions of τ.

Therefore types of nets are schizophrenic objects $<\tau, \tau'>$ living both in the category of transition systems and in the category of nets. Taking advantage of this fact, we adapt a work of Porst and Tholen [36] on concrete dualities induced by schizophrenic objects and construct dual adjunctions between transition systems and nets for any type of nets. We show in this way that the region based representation theorems for transition systems are a close analogue of the classical representation theorems for ordered algebras, which all arise from concrete dualities induced by schizophrenic objects based on the two element set $\mathbf{2} = \{0, 1\}$.

For the reader unfamiliar with schizophrenic objects, we review briefly some of the classical representation theorems. Birkhoff's duality between finite distributive lattices and finite partial orders relies on the schizophrenic object $\mathbf{2}$, viewed as a lattice and as an ordered set where $0 \leq 1$. The dual L^* of a distributive lattice L is the ordered set of its prime filters x whose characteristic functions are the lattice morphisms $\chi_x : L \to \mathbf{2}$. The dual X^* of an ordered set X is the lattice of its upwards closed subsets l whose characteristic functions are the morphisms of ordered sets $\chi_l : X \to \mathbf{2}$. Any ordered set is isomorphic to its double dual ($X \cong X^{**}$) where $x \in X$ is identified with $x^{**} \in X^{**}$ such that $\chi_{x^{**}}(l) = \chi_l(x)$ for any upwards closed subset $l \subseteq X$. Any distributive lattice is isomorphic to its double dual ($L \cong L^{**}$) where $l \in L$ is identified with $l^{**} \in L^{**}$ such that $\chi_{l^{**}}(x) = \chi_x(l)$ for any prime filter $x \subseteq L$. Thus both units of the dual adjunction are morphisms whose underlying maps are the evaluation maps.

Stone's duality between boolean algebras and the Stone spaces relies similarly on the schizophrenic object $\mathbf{2}$, viewed as a boolean algebra and as a discrete topological space. More instructive in the context of this paper is the duality between spatial frames and sober spaces (see [28, 18]). Recall that a frame is a complete lattice with the generalized distributivity law (finite meets distribute over arbitrary joins: $f \wedge \bigvee_i f_i = \bigvee_i (f \wedge f_i)$). For any frame F, let $pt(F)$ be the set of *points* x of F defined as frame morphisms $x : F \to \mathbf{2}$. The dual F^* of F is the topological space $(pt(F), \Omega)$ whose open sets are the sets $O_f = \{x : F \to \mathbf{2} | \ x(f) = 1\}$ for f ranging over F. Conversely, the dual X^* of a topological space (X, Ω) is the frame of its open sets $O \in \Omega$, whose characteristic functions χ_O are the continuous maps from (X, Ω) to the Sierpinski space $\mathbf{2}$ (with open sets $\{0, 1\}$, $\{1\}$, and $\emptyset$). Frames and topological spaces are connected by a dual adjunction $\mathbf{Frame}(F, X^*) \cong \mathbf{Top}(X, F^*)$.

By restricting this adjunction at both sides on its kernel, one obtains a duality $\mathbf{Top}^* \overset{op}{\cong} \mathbf{Frame}^*$ between the subcategory $\mathbf{Top}^*$ of *spatial* frames and the subcategory **Frame*** of *sober* spaces. So, a frame F is isomorphic to its double dual F^{**} if and only if F is a spatial frame. Now, spatial frames are characterized by two conditions very similar to our separation conditions for automata, when regions are replaced by morphisms $x : F \to \mathbf{2}$. Namely, a frame F is spatial if and only if the following conditions are satisfied for all $f, f' \in F$, where $f \leq f' \Leftrightarrow f = f \wedge f'$:

$$\begin{array}{ll} (i) & f \neq f' \Rightarrow \exists x : F \to \mathbf{2} \;:\; x(f) \neq x(f') \\ (ii) & f \not\leq f' \Rightarrow \exists x : F \to \mathbf{2} \;:\; x(f) = 1 \;\wedge\; x(f') = 0 \end{array}$$

Condition *(i)* is the analogue of our state separation condition SSP. Condition *(ii)* is the counterpart of our event state separation condition ESSP, when the structure of labelled transition system is replaced by the structure of partial order.

The classical dualities recalled above are concerned with points x, properties p, and a binary relation of evaluation $ev(x)(p) = p(x)$ valued in the underlying set of the schizophrenic object, i.e. $\{0, 1\}$. When this relation is given a matrix form, duality appears as matrix transposition [37]. Now, dualities between transition systems and nets fit exactly in the same pattern: the points are the transitions $s \stackrel{e}{\to} s'$, the properties are the regions (σ, η), and the evaluation matrix given by $ev(s \stackrel{e}{\to} s', (\sigma, \eta)) = \sigma(s) \stackrel{\eta(e)}{\to} \sigma(s')$ describes the local effect of the transitions on the places (σ, η) of the dual net. The technical development presented in the remaining of the section is based on the material contained in [6].

9.1 Schizophrenic Objects and Dual Adjunctions

Definition 9.1 *A* Set-category *(or category over* **Set***) is a pair* $<\mathcal{C}, U>$ *where* $\mathcal{C}$ *is a category and* $U : \mathcal{C} \to \mathbf{Set}$ *is a functor called the underlying functor. It is a* concrete category *if* U *is faithful.*

In the sequel, the underlying functor is left implicit and we use the uniform notation $|C|$ and $|f|$ for respectively the underlying set of an object C and the underlying map of an arrow f. In a Set-category $\mathcal{C}$, a *structured source* is an indexed family of pairs $\{C_i; f_i : X \to |C_i|\}$, where the C_i's are objects of $\mathcal{C}$ and the f_i's are maps from a fixed set X to the underlying sets of the C_i's. A *lift* of a structured source is an indexed family $\tilde{f}_i : C \to C_i$ of arrows of $\mathcal{C}$ such that $|\tilde{f}_i| = f_i$, and hence $|C| = X$. An *initial lift* of a structured source is a lift such that, if $g_i : C' \to C_i$ is another lift and there exists a map $f : |C'| \to X$ such that $|g_i| = f_i \circ f$ for all indices, then there exists a unique arrow $\tilde{f} : C' \to C$ such that $|\tilde{f}| = f$ and $g_i = f_i \circ \tilde{f}$ for all indices. The following definition is an adaptation from [36].

Definition 9.2 *A schizophrenic object between two Set-categories* $\mathcal{A}$ *and* $\mathcal{B}$ *is a pair of objects* $<K_{\mathcal{A}}, K_{\mathcal{B}}> \in |\mathcal{A}| \times |\mathcal{B}|$ *with the same underlying set* $K = |K_{\mathcal{A}}| = |K_{\mathcal{B}}|$ *and such that*

1. *for every object* A *in* $\mathcal{A}$*, the family* $\{K_{\mathcal{B}}; ev_A(a) : \mathcal{A}(A, K_{\mathcal{A}}) \to K\}_{a \in |A|}$ *of evaluation maps* $ev_A(a)(f) = |f|(a)$ *has an initial lift* $\{\epsilon_A(a) : A^* \to K_{\mathcal{B}}\}_{a \in |A|}$

2. *for every object B in $\mathcal{B}$, the family $\{K_{\mathcal{A}}; ev_B(b) : \mathcal{B}(B, K_{\mathcal{B}}) \to K\}_{b \in |B|}$ has an initial lift $\{\epsilon_B(b) : B^* \to K_{\mathcal{A}}\}_{b \in |B|}$.*

A^*, called the *dual* of A, is therefore an object of the category $\mathcal{B}$ whose underlying set is the set of $\mathcal{A}$-morphisms from A to the *classifying object* $K_{\mathcal{A}}$. If $K = \{0, 1\}$ and $\mathcal{A}$ is concrete, then the elements of the underlying set of the dual of A can be identified with subsets of the underlying set of A: $|A^*| \subseteq 2^{|A|}$ and $|A^{**}| \subseteq 2^{2^{|A|}}$. In any case, A and A^{**} are linked by an evaluation morphism $Ev_A : A \to A^{**}$ according to the following statement.

Lemma 9.3 *Let $<K_{\mathcal{A}}, K_{\mathcal{B}}>$ be a schizophrenic object between two Set-categories $\mathcal{A}$ and $\mathcal{B}$. The initial lift $\{\epsilon_A(a) : A^* \to K_{\mathcal{B}}\}_{a \in |A|}$ of the evaluation maps, viewed as a mapping $\epsilon_A : |A| \to \mathcal{B}(A^*, K_{\mathcal{B}})$, is the underlying map of an arrow $Ev_A : A \to A^{**}$.*

As an initial lift, the dual A^* of A is only defined up to an isomorphism. However, once an arbitrary representative A^* is fixed for each class of isomorphic objects, the operator $(-)^*$ gives rise to a functor according to the following statement.

Lemma 9.4 *Let $<K_{\mathcal{A}}, K_{\mathcal{B}}>$ be a schizophrenic object between two Set-categories $\mathcal{A}$ and $\mathcal{B}$. For every morphism $f : A_1 \to A_2$ in $\mathcal{A}$, the map "composing with f" given by $f^\bullet : \mathcal{A}(A_2, K_{\mathcal{A}}) \to \mathcal{A}(A_1, K_{\mathcal{A}})$: $g \mapsto g \circ f$ is the underlying map of an arrow $f^* : A_2^* \to A_1^*$ in $\mathcal{B}$ such that the functoriality laws $(1_A)^* = 1_{A^*}$ and $(f \circ g)^* = g^* \circ f^*$ are satisfied.*

The following proposition tells us that the two functors $(-)^*$ induced from a schizophrenic object are in fact dual adjoints.

Proposition 9.5 *Let $<K_{\mathcal{A}}, K_{\mathcal{B}}>$ be a schizophrenic object between two Set-categories $\mathcal{A}$ and $\mathcal{B}$. The following identities, where $f : A \to B^*$ and $g : B \to A^*$, define a bijective correspondence $\mathcal{A}(A, B^*) \cong \mathcal{B}(B, A^*)$:*

$$g = f^* \circ Ev_B \text{ and } f = g^* \circ Ev_A$$

i.e. the functors $(-)^*$ are adjoint to the right with the evaluations as units.

In the particular case where $\mathcal{A}$ and $\mathcal{B}$ are concrete categories, the above correspondence may be presented as matrix transposition. Actually, in this special case, $\mathcal{A}(A, B^*) \cong \mathbf{Span}_K(A, B) \cong \mathcal{B}(B, A^*)$ where $\mathbf{Span}_K(A, B)$ is the set of matrices $|A| \times |B| \to K$ whose rows, resp. columns, are underlying maps of morphisms $\varphi_a : B \to K_{\mathcal{B}}$ (for $a \in |A|$), resp. of morphisms $\varphi^b : A \to K_{\mathcal{A}}$ (for $b \in |B|$). In such a matrix, the set of rows determines a unique morphism from A to B^*, and the set of columns φ^b determines a unique morphism from B to A^*.

9.2 Application to Automata and Nets

Let **Trans** be the category of deterministic and reduced transition systems (S, E, T) free of isolated states ($\forall s \in S\ \exists t \in T : s = \partial^0(t) \ \vee\ s = \partial^1(t)$), where a morphism $(\sigma, \eta) : (S, E, T) \to (S', E', T')$ is a pair of maps $\sigma : S \to S'$ and

$\eta : E \to E'$ such that $s \xrightarrow{e} s'$ (in T) entails $\sigma(s) \xrightarrow{\eta(e)} \sigma(s')$ (in T'). **Trans** is a concrete category with forgetful functor U : **Trans** $\to$ **Sets** given by $U(S, E, T) = T$ and $U(\sigma, \eta)(s \xrightarrow{e} s') = (\sigma(s) \xrightarrow{\eta(e)} \sigma(s'))$.

Let $\tau = (LS, LE, LT)$ be an arbitrary object of **Trans**, called the type of nets. Let **Nets** be the category of event-simple nets (P, E, W) of type τ, thus $W : P \times E \to LE$ has all columns distinct, where a morphism $(\beta, \eta) : (P, E, W) \to (P', E', W')$ is a pair of maps $\beta : P \to P'$ and $\eta : E' \to E$ such that $W(p, \eta(e')) = W(\beta(p), e')$. Owing to the assumption of event-simpleness, β determines η in any morphism (β, η), and **Nets** is a concrete category with forgetful functor U : **Nets** $\to$ **Sets** given by $U(P, E, W) = P$ and $U(\beta, \eta) = \beta$.

Let $\tau' = (LT, \{\bullet\}, W) \in$ **Nets** be the net with the unique event $\bullet$ such that $W(\ell s \xrightarrow{\ell e} \ell s') = \ell e$ for every place $\ell s \xrightarrow{\ell e} \ell s' \in LT$. Thus $U\tau' = LT = U\tau$. Figure 20 displays the net τ'_{EN} corresponding to the type τ_{EN} of elementary nets.

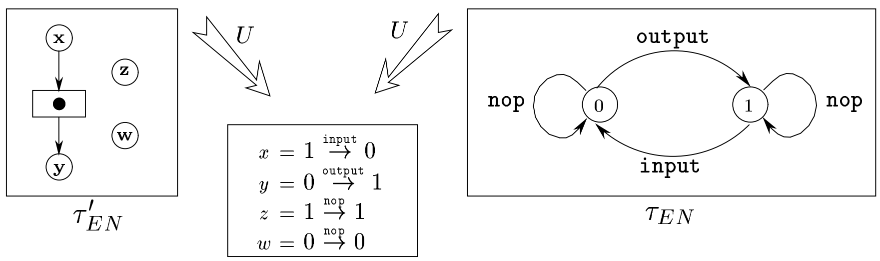


Fig. 20. the schizophrenic object for elementary nets

Proposition 9.6 *The pair (τ, τ') is a schizophrenic object between the categories* **Trans** *and* **Nets**, *inducing a dual adjunction* **Trans**$(A, N^*) \cong$ **Nets**(N, A^*).

It remains to interpret A^* and N^* in more familiar terms. For any transition system $A = (S, E, T)$, the homset **Trans**(A, τ) is the set $\mathcal{R}_\tau(A)$ of τ-regions of A. The evaluation $ev_A(s \xrightarrow{e} s')(\sigma, \eta) = (\sigma(s) \xrightarrow{\eta(e)} \sigma(s'))$ classifies therefore the transitions $t = (s \xrightarrow{e} s') \in T$ according to their local effect on each region. By definition, A^* is the net resulting from the initial lift of the family of evaluation maps $ev_A(t)$ for $t \in T$. The following proposition shows that A^* coincides with the net synthesized from the set of regions $\mathcal{R}_\tau(A)$ up to the confusion of indiscernible events.

Proposition 9.7 *A^* is isomorphic to the net $(P, E_\equiv, W)$ where $P = \mathcal{R}_\tau(A)$ is the set of regions of A, $\equiv$ is the equivalence relation on E such that $e \equiv e'$ when $\eta(e) = \eta(e')$ for every region (σ, η), and $W((\sigma, \eta), [e]_\equiv) = \eta(e)$.*

Now, for any net $N = (P, E, W)$, the homset **Nets**(N, τ') is in bijective correspondence with the set of transitions of the marking graph of N. In the sample case of elementary nets (see Fig. 21), each morphism $(\beta, \eta) : N \to \tau'_{EN}$ induces the transition $\beta^{-1}(\{x,z\}) \xrightarrow{\eta(e)} \beta^{-1}(\{y,z\})$, and conversely, each firing $M[e > M'$ induces the morphism (β, η) such that $\eta(\bullet) = e$ and for every place p,

$$\beta(p) = \begin{cases} x & \textbf{if} \quad p \in M \setminus M' \\ y & \textbf{if} \quad p \in M' \setminus M \\ z & \textbf{if} \quad p \in M \cap M' \\ w & \textbf{if} \quad p \notin M \cup M' \end{cases}$$

Therefore, the evaluation $ev_N(p)(\beta, \eta) = \beta(p)$ classifies the places of N ac-

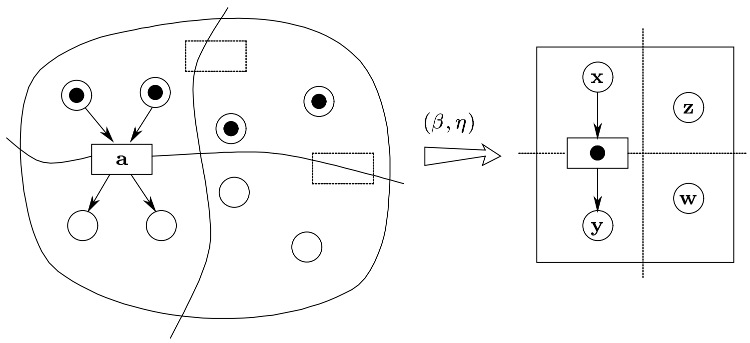


Fig. 21. firings as net morphisms

cording to the local transition they undergo in each global firing of the net. By definition, N^* is the transition system resulting from the initial lift of the family of evaluation maps $ev_N(p)$ for $p \in P$.

Proposition 9.8 *N^* is isomorphic to the marking graph of N.*

If one now specifies initial states for transition systems, and forward closed sets of markings for nets, the dual adjunction $\mathbf{Trans}(A, N^*) \cong \mathbf{Nets}(N, A^*)$ may be extended to a *Galois connection* $\mathbf{Aut}(\mathcal{A}, \mathcal{N}^*) \cong \mathbf{Netsys}(\mathcal{N}, \mathcal{A}^*)$ between automata and net systems, i.e. to a dual adjunction such that $\mathcal{A}^* \cong \mathcal{A}^{***}$ for every automaton $\mathcal{A}$, and $\mathcal{N}^* \cong \mathcal{N}^{***}$ for every net system $\mathcal{N}$. The details of the construction can be found in [6]. By restricting the Galois connection at both sides on its kernel, one finally obtains a duality $\mathbf{Netsys}^* \overset{op}{\cong} \mathbf{Aut}^*$ between separated automata and saturated net systems. As a consequence, the separated automata appear as a co-reflective subcategory of $\mathbf{Netsys}^{op}$. Similar co-reflections between separated automata and nets have been established in the literature for various categories of automata or concurrent automata, including elementary automata [34], asynchronous automata [35], automata with concurrency relations [20], and step automata [33].

10 Some Applications

Regions have come to use so far in two areas of application, asynchronous circuits and distributed protocols. A computer assisted solution to the state encoding problem for asynchronous circuits, based on elementary regions and supported by the tool `Petrify`, is described in [15]. A computer assisted solution to the distribution of protocols, based on Petri regions and supported by the tool SYNET, is described in [13]. Distributed and cooperative systems offer a wide range of problems to be solved prior to any successful application. The synthesis of stratified Petri nets [8], a weaker form of Valk's self-modifying nets [43, 44], may for instance be used for analysing cooperative systems in order to identify their normal and exceptional modes of operation, and possibly for simplifying the control of the transitions between these modes. Another goal of research is to derive systems from service specifications while decomposing large specifications into pieces. This might become feasible if one could solve the relaxed synthesis problem as follows: given a pair of rational languages L and L' such that $L \subseteq L'$, construct a (possibly not bounded) net system $\mathcal{N}$ such that $L \subseteq \mathcal{L}(\mathcal{N}) \subseteq L'$.

Acknowledgments. This work was partly supported by the H.C.M. Network *Express*.

References

[1] BADOUEL, E., *Splitting of Actions, Higher-Dimensional Automata and Net Synthesis.* Inria Research Report No 3013 (1996).

[2] BADOUEL, E., BERNARDINELLO, L. and DARONDEAU, PH., *Polynomial algorithms for the synthesis of bounded nets*, Proceedings Caap 95, Lecture Notes in Computer Science 915 (1995) 647–679.

[3] BADOUEL, E., BERNARDINELLO, L. and DARONDEAU, PH., *The synthesis problem for elementary net systems is NP-complete*, Inria Research Report 2558 (1995). To appear in Theoretical Computer Science.

[4] BADOUEL, E., and DARONDEAU, PH., *Trace Nets.* REX workshop, Beekbergen "Semantics: Foundation and Applications", Springer-Verlag Lecture Notes in Computer Science, vol. 666 (1993) 21–50.

[5] BADOUEL, E., and DARONDEAU, PH., *Trace nets and process automata*, Acta Informatica 32 (1995) 647–679.

[6] BADOUEL, E., and DARONDEAU, PH., *Dualities between Nets and Automata induced by Schizophrenic Objects*, 6^{th} International Conference on Category Theory and Computer Science, Cambridge, Lecture Notes in Computer Science, vol. 953 (1995) 24–43.

[7] BADOUEL, E., and DARONDEAU, PH., *On the Synthesis of General Petri Nets.* Inria Research Report No 3025 (1996).

[8] BADOUEL, E., and DARONDEAU, PH., *Stratified Petri Nets.* Inria Research Report No 3128 (1997).

[9] BEDNARCZYK, M. A., *Categories of Asynchronous Systems.* Ph. D. Thesis, University of Sussex (1988).

[10] Berge, C., *Graphes et hypergraphes.* Dunod, Paris (1970). English translation: *Graphs and Hypergraphs*, North Holland, Amsterdam (1973).

[11] Bernardinello, L., *Synthesis of Net Systems.* Application and Theory of Petri Nets, Lecture Notes in Computer Science 691 (1993) 89–105.

[12] Bernardinello, L., De Michelis, G., Petruni, K., and Vigna, S., *On the Synchronic Structure of Transitions Systems.* In "Structures in Concurrency Theory", J. Desel ed., Springer-Verlag (1996) 11–31.

[13] Caillaud, B., synet : *un outil de synthèse de réseaux de Petri bornés, applications* Irisa Research Report no 1101 (1997).

[14] Cortadella, J., Kishinevsky, M., Lavagno, L., and Yakovlev, A., *Synthesizing Petri Nets from State-Based Models.* Proceedings of iccad'95 (1995) 164–171.

[15] Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., and Yakovlev, A., *Complete state encoding based on the theory of regions.* Proceedings of the 2nd International Workshop on Advanced Research in Asynchronous Circuits and Systems (1996) 36–47.

[16] Chen, W.K., *Applied Graph Theory.* North-Holland (1971).

[17] Christofides, N., *Graph Theory - An Algorithmic Approach.* Academic Press (1975).

[18] Davey, B.A., and Priestley, H.A., *Introduction to Lattices and Order.* Cambridge University Press, (1990).

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

[20] Droste, M., and Shortt, R.M., *Petri Nets and Automata with Concurrency Relations – an Adjunction.* in "Semantics of Programming Languages and Model Theory", M. Droste and Y. Gurevich eds(1993) 69–87.

[21] Droste, M., and Shortt, R.M., *From Petri Nets to Automata with Concurrency.* Draft communicated to the authors.

[22] Ehrenfeucht, A., and Rozenberg, G., *Partial 2-structures ;* Part I : *Basic Notions and the Representation Problem*, and Part II : *State Spaces of Concurrent Systems*, Acta Informatica, vol 27 (1990).

[23] Garey, M.R., and Johnson, D.S., *Computer and Intractability. A guide to the theory of NP-Completeness.* W.H. Freeman and Company (1979).

[24] Gondran, M., and Minoux, M., *Graphes et algorithmes.* Eyrolles, Paris (1979). English translation by Steven Vajda: *Graphs and Algorithms*, John Wiley (1984).

[25] Hiraishi, K., *Some complexity results on transition systems and elementary net systems.* Theoretical Computer Science 135 (1994) 361–376.

[26] Hoogers, P.W., Kleijn, H.C.M., and Thiagarajan, P.S., *A Trace Semantics for Petri Nets.* Lecture Notes in Computer Science vol. 623 (1992) 595–604.

[27] Hoogers, P.W., Kleijn, H.C.M., and Thiagarajan, P.S., *An Event Structure Semantics for General Petri Nets.* Theoretical Computer Science, volume 153 (1996) 129–170.

[28] Johnstone, P.T., *Stone Spaces.* Cambridge University Press, (1982).

[29] Kishinevsky, M., Cortadella, J., Kondratyev, A., Lavagno, L., Taubin, A., and Yakovlev, A., *Place Chart Nets and their Synthesis.* Technical Report 96-2-003 Department of Computer Hardware, University of Aizu (1996).

[30] Kishinevsky, M., Cortadella, J., Kondratyev, A., Lavagno, L., and Yakovlev, A., *Synthesis of General Petri Nets.* Technical Report 96-2-004, Department of Computer Hardware, University of Aizu (1996).

[31] Lefschetz, S., *Applications of Algebraic Topology.* Applied Mathematical Science 16, Spinger-Verlag (1975).

[32] Liu, C.L., *Introduction to Combinatorial Mathematics.* Mac Graw Hill (1968).

[33] Mukund, M., *Petri Nets and Step Transition Systems.* International Journal of Foundation of Computer Science, vol 3, n^o 4 (1992) 443–478.

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

[35] Nielsen, M., and Winskel, G., *Models for Concurrency*, Handbook of Logic for Computer Science, vol. 4, Oxford University Press (1995).

[36] Porst, H.-E., and Tholen, W., *Concrete Dualities.* In "Category Theory at Work", H. Herrlich, and H.-E. Porst (eds.), Heldermann Verlag Berlin (1991) 111-136.

[37] Pratt, V.R., *The Stone Gamut: A Coordinatization of Mathematics*, Proceedings of the 10^{th} Symposium on Logics in Computer Science, IEEE Computer Society (1995) 444–454.

[38] Schrijver, A., *Theory of Linear and Integer Programming.* John Wiley (1986).

[39] Schmitt, V., *Flip-Flop Nets*, Proceedings of Stacs 96, Lecture Notes in Computer Science vol. 1046 (1996) 517–528.

[40] Tutte, W.T., *Graph Theory.* Encyclopedia of Mathematics and its Application vol. 21, Addison-Wesley (1984).

[41] Shields, M.W., *Concurrent machines*, The Computer Journal, vol. 28 (1985) 449–465.

[42] Stark, E.W., *Connections between a Concrete and an Abstract Model of Concurrent Systems.* 5th Mathematical Foundation of Programming Semantics (1989) 53–79.

[43] Valk, R., *Self-Modifying Nets, a Natural Extension of Petri Nets.* Proceedings of Icalp'78, Lecture Notes in Computer Science vol. 62 (1978) 464–476.

[44] Valk, R., *Generalizations of Petri Nets.* Proceedings of MFCS'81, Lecture Notes in Computer Science vol. 118 (1981) 140–155.

This article was processed using the LaTeX macro package with LLNCS style