INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Splitting of Actions, Higher-Dimensional Automata, and Net Synthesis

Eric Badouel

N° 3013

Octobre 1996

THÈME 2

Rapport de recherche

ISSN 0249-6399

Splitting of Actions, Higher-Dimensional Automata, and Net Synthesis

Eric Badouel

Thème 2 — Génie logiciel
et calcul symbolique
Projet Micas

Rapport de recherche n ° 3013 — octobre 1996 — 51 pages

Abstract: The behaviour of pure Petri nets (i.e. without side condition) is given by ordinary automata because all information about concurrency is encoded in the structure of the marking graphs. By contrast, the behaviour of (possibly) impure nets requires higher-dimensional automata: independence should in that case be made explicit. These higher-dimensional automata are step transition systems in the case of general Petri nets and asynchronous transition systems if we restrict to safe Petri nets. The aim of this report is to show that the synthesis problem for nets can reduce to the synthesis problem of pure nets. For that purpose, we discretize an higher-dimensional automaton by splitting its actions and we prove that it is the behaviour of some Petri net if and only if its discretized automaton is the marking graph of some pure Petri net.

Key-words: Synthesis Problem for Nets, Higher-Dimensional Automata, Splitting of Actions

(Résumé : tsvp)

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

* Email: Eric.Badouel@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Scindage d'actions, automates de dimensions supérieures et synthèse de réseaux

Résumé : Le comportement d'un réseau de Petri pur (i.e. sans condition de bord) est parfaitement décrit par son graphe de marquage. En effet, la structure d'indépendance entre les actions du réseau se retrouve (sous forme d'hypercubes) dans la structure de cet automate. La modélisation du comportement concurrent d'un réseau de Petri général nécessite par contre l'introduction d'un automate de dimension supérieure: on doit rendre explicite sa structure d'indépendance. Il pourra s'agir de systèmes d'hyper-transitions dans le cas le plus général ou de systèmes de transitions asynchrones si on se limite aux réseaux unaires. Ehrenfeucht et Rozenberg ont caractérisés les automates qui sont graphes de marquage de réseaux élémentaires (qui sont des réseaux unaires purs) la solution a été ensuite étendue au cas des réseaux de Petri purs. Suivant la même méthode, Nielsen et Winskel ont caractérisés les systèmes de transitions asynchrones qui sont des comportements de réseaux de Petri unaires et Mukund les systèmes d'hyper-transitions qui sont comportements de réseaux de Petri généraux. Le but de ce rapport est de montrer que la synthèse de réseaux non purs peut se ramener à la synthèse de réseaux purs, dans le cas unaire comme dans le cas général. Pour ceci on associe à tout automate de dimension supérieure un automate dit discrétisé obtenu en scindant chaque action en deux et on montre que l'automate de dimension supérieure est le comportement d'un réseau de Petri si et seulement si son discrétisé est graphe de marquage d'un réseau de Petri pur. Comme corollaire, on déduit un algorithme polynomial qui décide si un système d'hyper-transitions est comportement d'un réseau de Petri, et qui construit ce réseau lorsqu'il existe.

Mots-clé : Synthèse de réseaux, automate de dimension supérieure, scindage d'actions

1 Introduction

The problem of net synthesis is often presented as a graph-theoretic problem: given a labelled graph representing the sequential behaviour of some distributed system, the synthesis problem consists in deciding whether it is isomorphic to the marking graph of some Petri net. The synthesis problem was originally solved for the class of elementary net systems by Ehrenfeucht and Rozenberg [ER90]. Their approach was based on the notion of *regions in graphs* defined as sets of nodes liable to represent extensions of places with boolean values: these are set of nodes uniformly entered, exited or left invariant by all transitions bearing the same label. This solution was extended to pure Petri nets [BDPV96] using the variant notion of *generalized regions in graphs*, seen as multisets of nodes representing extensions of places with values given by whole numbers. Ehrenfeucht and Rozenberg gave a characterization of those automata which are isomorphic to the marking graphs of elementary nets in term of two *separation axioms* the first of which states that there exists sufficiently many regions to distinguish every pair of distinct states in the automaton. The second axiom of separation states that for every action and every state at which this action is not enabled there exists a region which "inhibits" this action in this state. All variants of net synthesis have a representation theorem based on the same separation axioms but with variant notions of region. An interpretation of these separation axioms is the following: a region viewed as an abstract place can be associated with a quotient automaton representing the projection of the automaton relative to the "content" of that place, the automaton is then isomorphic to the marking graph of some net if and only if it is isomorphic to the synchonized product of those automata (associated with regions). The first separation axiom expresses that states of the original automaton can injectively be encoded as vectors of local states (each local state giving its value relative to some region). The second separation axiom states that an action is not enabled in some state of the automaton if and only if there exists some component automaton such that this action is not enabled in the corresponding local state. The synthesis problem then reduces to deciding whether an automaton can be presented as a synchronized product of automata of a given type. This observation leads to a uniform presentation of the net synthesis problem [BD95] parametric on the type on nets: regions appear as morphisms from the underlying transition system of the automaton to a classifying transition system, called the *type of nets*, which characterizes the behaviour of the considered class of nets.

A characteristic feature of the graph theoretic approach of Ehrenfeucht and Rozenberg is its *extensionality*: regions are fully characterized by the way they are altered (or not) by actions in the system. This explains that we usually restrict ourself to event-simple nets and also that we rule out impure nets (i.e. nets having places which are at the same time input and output condition for some action –the so-called *side conditions*–).

Applications of this approach to the net synthesis problem may be thought in the engineering of distributed software where it can be used as a technique of parallelization of sequential programs. Actually, net synthesis gives means to build parallel systems realizing behaviours specified by finite transition systems (e.g. communication protocols). More precisely the net synthesized from an automaton exhibits the maximal parallelism compatible with the structure of the automaton in the sense that for any n-uple of actions labelling an n-dimensional hypercube in the automaton the associated n-uple of transitions in the synthesized net are concurrently enabled at the marking associated with the origin of the hypercube. Since the synthesized net is more compact than the original transition system, this technique might be used to evade the state explosion problem when verifying properties of discrete event systems. However, for the time being this technique suffers from some limitations that restrict its use in practical applications. The first limitation is the complexity issue, the synthesis problem for elementary net systems is NP-complete [BBD95b], and although the synthesis problem for pure Petri nets is solved in polynomial time [BBD95a], its solution is based on Khachiyan's ellipsoid method (see [Sch86]) which in practice is usually replaced by the, theoretically inefficient, simplex method: actually, even though this latter is not a polynomial-time method it has a better average complexity than the ellipsoid method. The second drawback is the restriction to pure nets: side conditions are often useful for modelling systems with Petri nets and we definitely loose expressivity by ruling them out.

Nielsen and Winskel [NW94] and Mukund [Muk92] solved the synthesis problem for respectively safe Petri nets and (general) Petri nets, thus allowing side conditions. With those impure nets the concurrency exhibited by the

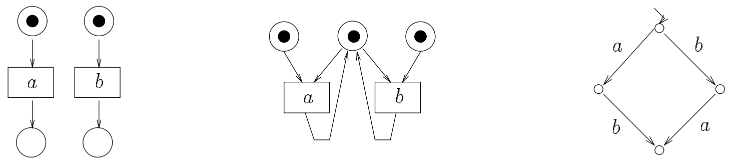


Figure 1: two safe Petri nets with identical marking graph but different independence relations

synthesized net is no longer reflected in the structure of the automaton. The simplest illustration of this phenomenon appears in Fig. 1 where the two safe Petri nets have the same marking graph shown on the right, but the actions a and b are independent in the first case and not in the second case. Actions in a safe net are independent if and only if they have disjoint domains, where the domain of an action is the set of conditions which are connected to it (i.e. its input and output conditions). The information about the independence of transitions in the marking graph of a safe net is then totally captured by a binary relation of independence on the set of actions (which are the labels of the transitions). Enriched with this relation, the marking graph becomes what is known as an *Asynchronous Transition System* [Bed88, Shi85]. For the converse direction, Nielsen and Winskel have defined a variant of region, called *condition*, in asynchronous transition systems (with the restriction that for every pair of independent actions there exists at least one state in which these actions are both enabled). Conditions of such an asynchronous transition system constitute the places of a safe net of which the asynchronous transition system is the marking graph as soon as both separation axioms are satisfied. By supplying safe nets and asynchronous transition systems with adequate notions of morphisms they established a coreflection between the subcategory of asynchronous transition systems verifying the separation axioms and the category of safe nets.

Actions in a (general) Petri net are independent in a given marking if there exists enough tokens in their input places so that they can fire concurrently. For instance the three Petri nets in Fig. 2 have the same marking graph shown on the right; however the three actions a, b and c are independent in the first net (at the indicated marking) whereas they are pairwise independent but not independent in the second case. The third example (borrowed from [HKT96])

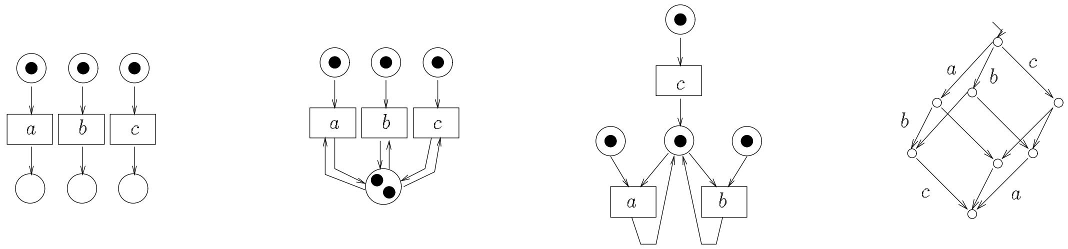


Figure 2: three Petri nets with identical marking graph but different independence relations

is even more involved: at the indicated marking the maximal sets of independently enabled actions are $\{a, c\}$ and $\{b, c\}$; a and b are not independent but they become independent once action c has been fired. Thus the independence relation of actions is now state dependent; moreover even at a given marking the information about the independence of actions does not reduce to a binary relation: actions as in the second example can be pairwise independent but not globally independent. Moreover it seems natural to allow *autoconcurrency*, i.e. several instances of the same action can fired concurrently if sufficiently many ressources are available in their input places. Enriched with the data of the multisets of actions independently enabled at each marking, the marking graph of a Petri net becomes a so-called *step transition system*. This counterpart of asynchronous transition systems introduced by Mukund in [Muk92] are transition systems in which one state can be transformed to another one in a single step consisting of a finite multiset of concurrently enabled actions. Mukund has introduced a variant of regions in step transition systems that we shall called here *generalized conditions*. The generalized conditions of a step transition system constitute the places of a Petri net of which this step transition system is the marking graph as soon as both separation axioms are satisfied. By supplying Petri nets and step transition systems with morphisms between them, Mukund has obtained a coreflection between the subcategory of step transition systems verifying the separation axioms and the category of Petri nets.

The aim of this paper is to relate the two previous approaches to the net synthesis problem. It emerges from the discussion above that the behaviour

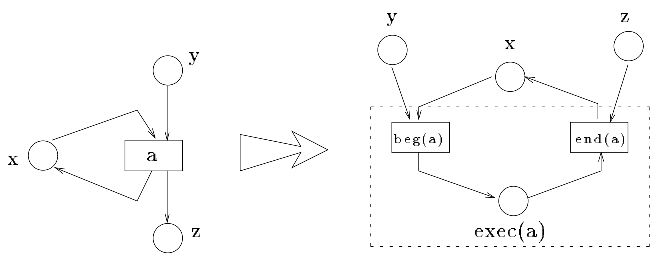


Figure 3: transformation of a net with side conditions into a pure one by splitting of actions

of pure nets is given by ordinary automata (all information about concurrency is encoded in the structure of the automata) whereas the behaviour of (possibly) impure nets is given by *Higher-Dimensional Automata*. Higher-dimensional automata are geometrical models for automata with concurrency relations on transitions. There were first proposed by Pratt and van Glabbeek [Pra91, Gla91] (see also [Shi85b]) and are extensively studied in Goubault's thesis [Gou95]. An higher-dimensional automaton consists of states of various dimensions, where an n-dimensional state is interpreted as a situation in which n independent events are concurrently executed. Asynchronous transition systems and step transition systems are indeed higher-dimensional automata. For step transition systems however axioms should be added to Mukund's original definition so that they can be viewed as higher-dimensional automata. These extra axioms (one of which is reminiscent to the definition of *distributed transition system* in [LRT88]) have however no incidence on Mukund's result. This notion of higher-dimensional automata is closely connected to the usual transformation of an impure net into a pure one by splitting the actions of the original net (see e.g. [GV87]). This transformation which is illustrated in Fig. 3 consists in splitting each action a into the "begining of a" $\boxed{beg(a)}$ and the "end of a" $\boxed{end(a)}$. Each input place of action a becomes an input place of $beg(a)$ with the same weight, similarly each ouput place of a becomes an output place of $end(a)$. The place $exec(a)$ which is an output place of $beg(a)$ and an input place of $end(a)$ (with single weights) witnesses the activity of action a in case of a safe net, or counts the number of instances of actions a which are currently executed in case of a (general) Petri net. A marking of

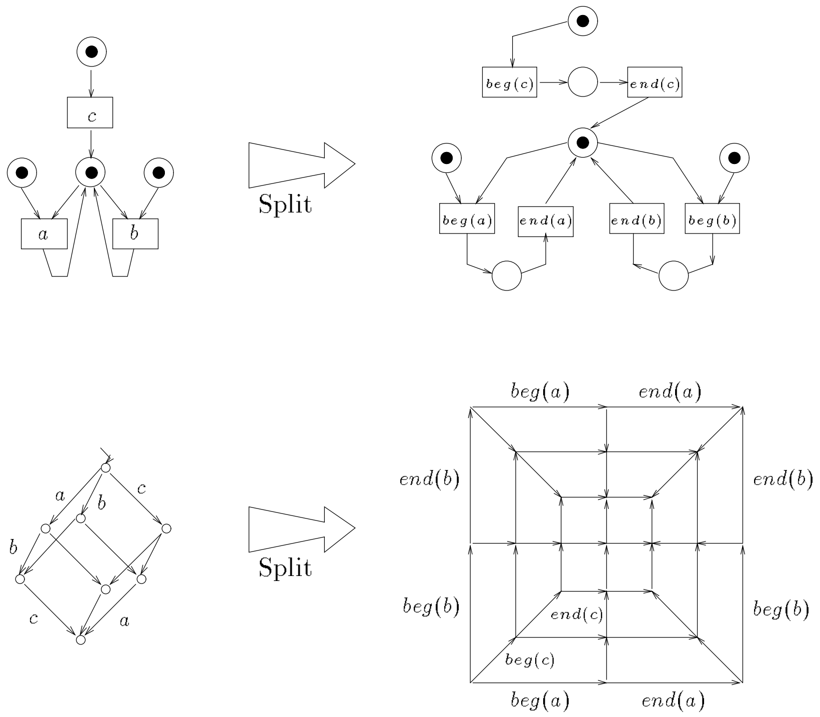


Figure 4: the marking graph of the splitted net as a discretization of the higher-dimensional automaton associated with the original net

the splitted net can then be viewed as an n-dimensional state where n is the total number of tokens in places of the form $exec(a)$. This interpretation of a safe net or of a general Petri net as a higher-dimensional automaton via the splitting of actions is consistent with their usual interpretations as respectively asynchronous transition systems and step transition systems. More precisely, the marking graph of the splitted net, which is an ordinary automaton on the splitted alphabet, appears as a discretization of the higher-dimensional automaton associated with the original net (see Fig. 4).

In this paper we show that an asynchronous transition system is the behaviour of some safe Petri net (i.e. satisfies the separation axioms w.r.t. its set of conditions) if and only if its corresponding discretization is the marking graph of some elementary net system (i.e. is separated by regions). Similarly a step transition system is the behaviour of some Petri net (i.e. is separated by generalized conditions) if and only if its discretization is the marking graph of some pure Petri net (i.e. is separated by generalized regions). Moreover in both cases one can easily proceed from one net representation to the other one. On the one direction we split the net to obtain a pure net. On the converse direction we first normalize the net so that for every place X (except those of the form $exec(a)$) one has $X^\bullet \subseteq beg(A)$ and ${}^\bullet X \subseteq end(A)$. Then we abstract each subnet $\boxed{beg(a)}\longrightarrow\bigcirc\longrightarrow\boxed{end(a)}$ into an atomic action a. In this manner Nielsen and Winskel's synthesis of safe nets from asynchronous transition systems and Mukund's synthesis of Petri nets from step transition systems reduce respectively to the synthesis of elementary net systems and pure Petri nets from ordinary automata. In particular, using the algorithm of [BBD95a] one obtains a polynomial solution to the synthesis of Petri nets from step transition systems.

The paper is organized as follows. In section 2 we fix terminology and notations for various kind of nets and for the automata used to describe their concurrent behaviours. We describe the splitting of nets and the discretization of higher-dimensional automata and show that these operations are compatible w.r.t. the behaviours of nets. In section 3 we give a brief survey of the net synthesis problem, and in section 4 we prove that the synthesis problem of nets from higher-dimensional automata reduces to the synthesis problem of pure nets from ordinary automata.

2 Petri Nets and their Concurrent Behaviours

In this section we recall the terminology and notations concerning condition/event nets and place/transition nets (also known as Petri nets). Their sequential behaviours are given by their marking graphs. However, modelling their concurrent behaviours requires higher-dimensional automata as information on the independence of transitions should be added to their marking graphs. In the particular case of pure nets (i.e. nets free from side conditions) these higher-dimensional automata are canonical which means that all the information about concurrency is already encoded in the structure of the underlying automata. In order to reduce the study of Petri nets to the study of their subclass of pure Petri nets, one can transform each Petri net into a pure one by splitting the actions. We introduce a representation of an higher-dimensional automaton by an ordinary automaton (its discretization) based on the same idea of splitting the actions. These two operations are shown to correspond in the sense that the marking graph of the splitted net is the splitting (i.e. discretization) of the higher-dimensional automaton giving the concurrent behaviour of the original net.

2.1 Elementary Net Systems

Definition 1 (C/E Net)
A condition/event net is a triple (P, A, F) where P is a set of places or conditions, A is a set of actions or events disjoint from P, and $F \subset A \times P \cup P \times A$ is a bipartite relation between places and events called the flow relation. The flow graph is assumed to have no isolated element, in the sense that

$$\forall x \in A \cup P \;\; \exists y \in A \cup P \;\; [(x, y) \in F \;\vee\; (y, x) \in F]$$

In the graphical representation of nets, places are depicted by circles and actions by boxes. We adopt the notations ${}^\bullet x = \{y/F(y, x)\}$ and $x^\bullet = \{y/F(x, y)\}$ for the respective pre-set and post-set of an element $x \in P \cup A$. A C/E net is said to be *simple* when

$$\forall x, y \in P \cup A \quad ({}^\bullet x = {}^\bullet y \text{ and } x^\bullet = y^\bullet) \Rightarrow x = y$$

A *marking* of a C/E Net is a set of places representing a *state* in the evolution of the net by the set of conditions it satisfies. All the possible evolutions are described in a transition system $(\mathcal{M}, A, T)$ whose set of states is the set $\mathcal{M}$ of markings, and whose transitions are given by

$$M \stackrel{a}{\rightarrow} M' \;\; \textbf{iff} \;\; {}^\bullet a \subseteq M \text{ and } (M \setminus {}^\bullet a) \cap a^\bullet = \emptyset \text{ and } M' = (M \setminus {}^\bullet a) \cup a^\bullet$$

Places in ${}^\bullet a \cap a^\bullet$, called the *side conditions* of a, are just tested upon: these conditions are necessary for firing event a and they still hold thereafter. The conditions in ${}^\bullet a$ (*preconditions* of a) which are not side conditions of a are also necessary for firing event a but they no longer hold after it has been fired. Symmetrically, the conditions in $a^\bullet$ (*postconditions* of a) which are not side conditions of a hold never in markings giving concession to event a, and hold always after a's executions. A net free from side conditions ($\forall a \in A \; {}^\bullet a \cap a^\bullet = \emptyset$) is said to be *pure*. The transition relation of pure C/E nets simplifies to

$$M \xrightarrow{a} M' \quad \textbf{iff} \quad {}^\bullet a \subseteq M \;\text{ and }\; M \cap a^\bullet = \emptyset \;\text{ and }\; M' = (M \setminus {}^\bullet a) \cup a^\bullet$$

A *C/E net system* is a C/E net together with an initial marking M_0. A marking M is *reachable* if there exists a finite sequence of transitions from M_0 to M. A C/E net system is *contact-free* if for every reachable marking M, and every event a the following implication holds: ${}^\bullet a \subseteq M \Rightarrow (M \setminus {}^\bullet a) \cap a^\bullet = \emptyset$, i.e. a contact situation is given by an event which cannot be fired at some reachable marking because of one of its postconditions. The transition relation between reachable markings of a contact-free C/E net simplifies to

$$M \xrightarrow{a} M' \quad \textbf{iff} \quad {}^\bullet a \subseteq M \;\text{ and }\; M' = (M \setminus {}^\bullet a) \cup a^\bullet$$

Definition 2 (Transition System, Automaton)
A transition system is a triple (S, A, T) where S is a set of states, A is a set of actions, and $T \subseteq S \times A \times S$ is a set of (labelled) transitions. $(s, a, s') \in T$ is also denoted by $s \xrightarrow{a} s'$. An automaton (S, A, T, s_0) is a transition system with a distinguished state $s_0 \in S$, called the initial state.

Definition 3 (Marking Graph)
The marking graph of a C/E net system $\mathcal{N} = (P, A, F, M_0)$ is the automaton $\mathcal{N}^*$ $= (S, A, T, s_0)$ whose initial state s_0 is the initial marking M_0 of $\mathcal{N}$ and whose underlying transition system (S, A, T) is the induced restriction of the transition system generated from the net (P, A, F) on the set of markings reachable from M_0.

We can always avoid contact situations by adding new places to the net: the *complement* of a condition x is the condition $\overline{x}$ such that $\overline{x} \in M_0 \Leftrightarrow x \notin M_0$ and for every event a, $\overline{x} \in {}^\bullet a \Leftrightarrow x \in a^\bullet \setminus {}^\bullet a$ and $\overline{x} \in a^\bullet \Leftrightarrow x \in {}^\bullet a \setminus a^\bullet$; the complemented net system is contact-free and has, up to isomorphism, the same marking graph as the original net system.

Definition 4 (Independence Relation)
Events in a C/E net are independent if they have no condition in common:

$$a \| b \quad \textbf{iff} \quad ({}^\bullet a \cup a^\bullet) \cap ({}^\bullet b \cup b^\bullet) = \emptyset$$

In any transition system the following predicate may be defined:

$$\diamond(s,a,b) \quad \Leftrightarrow \quad (\exists s_1, s_2, s_3 \;:\; s \xrightarrow{a} s_1 \;\wedge\; s_1 \xrightarrow{b} s_3 \;\wedge\; s \xrightarrow{b} s_2 \;\wedge\; s_2 \xrightarrow{a} s_3)$$

when $\diamond(s,a,b)$ we say that the events a and b form a diamond in s. The independence relation of a pure C/E net is encoded in its marking graph in the sense that *(1)* every pair of independent events form a diamond in every marking at which they are both enabled, and *(2)* every pair of events that form a diamond in some marking are independent. As shown in the introduction (Fig. 1) this is not true for C/E net with side conditions. Enriched with the independence relation between events, the marking graph of a C/E net system is an asynchronous transition system.

Definition 5 (Asynchronous Transition System)
An asynchronous transition system $(S, A, \|, T, s_0)$ *is an automaton whose set of events comes equipped with a symmetric and irreflexive binary relation of independence* $\| \subseteq A \times A$ *such that* (i) *it is deterministic:* $s \xrightarrow{a} s_1 \wedge s \xrightarrow{a} s_2 \Rightarrow s_1 = s_2$, (ii) *for every pair of independent events there exists at least one state at which they are both enabled and they form a diamond in each such state:* $(a\|b \wedge s \xrightarrow{a} s_1 \wedge s \xrightarrow{b} s_2) \Rightarrow \exists s_3 \,.\, s_1 \xrightarrow{b} s_3 \wedge s_2 \xrightarrow{a} s_3$, *and* $a\|b \Rightarrow \exists s \quad \diamond(s,a,b)$, *and* (iii) *every pair of independent events form a diamond in every state at which they are enabled in some order:* $(a\|b \wedge s \xrightarrow{a} s_1 \wedge s_1 \xrightarrow{b} s_3) \Rightarrow \exists s_2 \,.\, s \xrightarrow{b} s_2 \wedge s_2 \xrightarrow{a} s_3$

An asynchronous transition system verifying condition *(2)* above is said to be *canonical*. Therefore the concurrent behaviour of pure C/E net system is a canonical asynchronous transition system. Of particular interest are the elementary net systems for which the synthesis problem was first studied.

Definition 6 (Elementary Net System)
An elementary net is a pure and simple C/E net with no isolated element. An elementary net system $\mathcal{N} = (P, A, F, M_0)$ *is an elementary net together with an initial marking* M_0 *such that every event* $a \in A$ *may be fired at some marking reachable from* M_0.

2.2 Petri Nets

Definition 7 (Petri Net) *A Petri net (also called place/transition net) is a triple (P, A, F) where P is a set of places, A is a set of events or actions disjoint from P and $F : (A \times P) \cap (P \times A) \rightarrow I\!N$. The graph of the (multi valued) flow relation F is assumed to have no isolated element, in the sense that*

$$\forall x \in A \cup P \ \ \exists y \in A \cup P \ \ [(x, y) \in F \ \vee \ (y, x) \in F]$$

In the usual representation for Petri nets places are depicted by circles and events by boxes, and an arc with positive weight n drawn from x to y means $F(x, y) = n$. Any place or event $x \in P \cup A$ has a pre-set ${}^\bullet x$ and a post-set $x^\bullet$ which are the multi-sets given by: ${}^\bullet x(y) = F(y, x)$ and $x^\bullet(y) = F(x, y)$. Since multisets on a given set form a commutative monoid, an additive representation may be used for multisets. A *marking* is a multiset of places interpreted as a distribution of resources (each place in a Petri net is viewed as a particular type of resource); for instance the multiset $3x + 2y$ means that three resources of type x and two resources of type y are available. The pre-set ${}^\bullet a$ of an event $a \in A$ indicates the set of resources consumed by a and its post-set $a^\bullet$ the set of resources produced by a. Therefore, the evolution of a net is described by a transition system $(\mathcal{M}, A, T)$ whose set of states $\mathcal{M}$ is the set of markings of the net, and whose transitions are given by

$$M \stackrel{a}{\rightarrow} M' \ \ \textbf{iff} \ \ {}^\bullet a \leq M \ \text{ and } \ M' = (M - {}^\bullet a) + a^\bullet$$

A Petri net with no side condition ($\forall a \in A \ {}^\bullet a \cap a^\bullet = \emptyset$) is said to be *pure*.

A Petri net system is a Petri net together with an initial marking. As for C/E nets, the marking graph of a Petri net system $\mathcal{N} = (P, A, F, M_0)$ is the automaton $\mathcal{N}^* = (S, A, T, s_0)$ whose initial state s_0 is the initial marking M_0 of $\mathcal{N}$ and whose underlying transition system (S, A, T) is the induced restriction of the transition system generated from the net (P, A, F) on the set of markings reachable from M_0. A Petri net system is *safe* if the value of its places are bounded by 1; we shall make no distinction between safe Petri net systems and contact-free C/E net systems.

Before describing the independence relation in a Petri net system, we introduce further notations. In any deterministic transition system we define $s \stackrel{u}{\rightarrow} s'$ by induction on the length of $u \in A^*$: *(i)* $s \stackrel{\epsilon}{\rightarrow} s$ and *(ii)* $s \stackrel{u \cdot a}{\rightarrow} s'$ if and only if there exists (a necessarily unique) s'' such that $s \stackrel{u}{\to} s''$ and $s'' \stackrel{a}{\to} s'$; let $s \cdot u$ denote the state s' such that $s \stackrel{u}{\to} s'$ when it exists; we let $\pi(u)$ denote the *Parikh image* of word u, i.e. it is the multiset such that $\pi(u)(a)$ counts the number of occurrences of the letter a in u. We say that the transition system satisfies the *Parikh property* if the following holds:

$$(s \stackrel{u}{\to} s' \quad \wedge \quad s \stackrel{v}{\to} s'' \quad \wedge \quad \pi(u) = \pi(v)) \;\Rightarrow s' = s''$$

If a transition system satisfies the Parikh property we let $s \stackrel{\alpha}{\Rightarrow} s'$, for α a multiset of events, whenever $s \stackrel{u}{\to} s'$ for every word whose Parikh image is α. We call $s \stackrel{\alpha}{\Rightarrow} s'$ an *hyper-transition* and we denote $s \cdot \alpha$ the state s'.

Definition 8 (Independence Relation)
A multiset of events α (or step) is enabled at a (reachable) marking M (notation $M[\alpha>$) if there exists enough tokens in the input places of the events of α so that they can fire concurrently:

$$M[\alpha> \quad \textbf{iff} \quad [\forall x \in P \quad \sum_{a \in A} \alpha(a) \cdot {}^{\bullet}a(x) \leq M(x)]$$

Clearly the marking graph of a Petri net satisfies the Parikh property. The structure of independence of a pure Petri net is encoded in the structure of its marking graph in the following sense.

Observation 9 *If M is a (reachable) marking of a pure Petri net and α is a step then $M[\alpha>$ if and only if there exists an hyper-transition $M \stackrel{\alpha}{\Rightarrow} M'$ in its marking graph.*

As shown in the introduction (Fig. 2) this is not true for arbitrary Petri nets. Enriched with the data of the finite multisets of actions independently enabled at each marking, the marking graph of a Petri net is a step transition system.

Definition 10 (Step Transition System)
A step transition system $(S, A, [>, T)$ is a transition system equipped with a binary relation between states and finite multisets of actions such that (i) *it satisfies the Parikh condition: $(s \stackrel{u}{\to} s_1 \wedge s \stackrel{v}{\to} s_2 \wedge \pi(u) = \pi(v)) \Rightarrow s_1 = s_2$,* (ii) *every step enabled in some state forms an hyper-transition: $s[\alpha> \Rightarrow \exists\ s \stackrel{\alpha}{\Rightarrow} s'$, and* (iii) *the enabling relation is downward-closed and persistent: $(s[\alpha> \;\wedge\; \beta \subseteq \alpha) \Rightarrow (s[\beta> \;\wedge\; (s \cdot \beta)[\alpha - \beta>)$.*

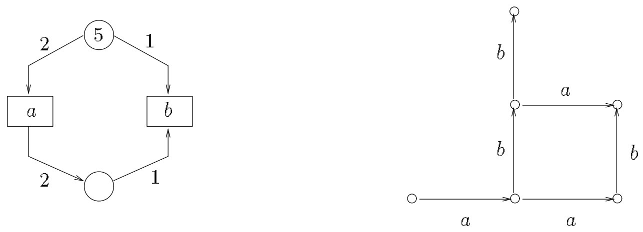


Figure 5: a Petri net whose behaviour is not an asynchronous transition system

Condition *(iii)* was not required in Mukund's original definition, however since it is satisfied by marking graphs of Petri nets, his representation theorem still holds with this new definition. This condition which takes its inspiration from the definition of *distributed transition system* [LRT88] implies that *(iv)* every sequence of transitions whose Parikh image is initially enabled takes part in an hyper-transition:($s \stackrel{u}{\to} s' \wedge \alpha = \pi(u) \wedge s[\alpha\!>) \Rightarrow \exists\ s \stackrel{\alpha}{\Rightarrow} s'$. We observe that conditions *(i)*, *(ii)* and *(iv)* generalize the analogue conditions for asynchronous transition systems, in particular the Parikh property entails determinism. More precisely an asynchronous transition system is a step transition system where a step α is enabled in state s if and only if α is a set (there is no auto-concurrency in asynchronous transition systems) of pairwise independent events enabled in state s; i.e. asynchronous transition systems are those step transition systems whose independence relation is structural (is not state dependent) and is given by a binary conflict between events. Figure 5 represents a (pure) Petri net whose behaviour is an example of a (canonical) step transition system which is not an asynchronous transition system. If we let $s \stackrel{\alpha}{\to} s'$ when α is enabled in state s and $s \stackrel{\alpha}{\Rightarrow} s'$ with the convention that $s \stackrel{0}{\to} s'$ if and only if $s = s'$, then the step transition systems of Def. 10 are the step transition systems over the commutative monoid $<\!A\!>$ freely generated by the alphabet A, where:

Definition 11 (Step Transition System over a Commutative Monoid) *A step transition system (S, M, T) over a commutative monoid M consists of a set of states S and a deterministic transition relation $T \subseteq S \times M \times S$ such that* (i) $s \stackrel{0}{\to} s'$ *if and only if* $s = s'$, *and* (ii) *for every* $s \stackrel{\alpha+\beta}{\to} s'$ *there exists* $s'' \in S$ *with* $s \stackrel{\alpha}{\to} s''$ *and* $s'' \stackrel{\beta}{\to} s'$. *A step automaton is a step transition system* *with an initial state $s_0 \in S$. If $M = <A>$ is the commutative monoid freely generated by the alphabet A, the skeleton of a step automaton (S, M, T, s_0) is its restriction $(S, A, T \cap (S \times A \times S), s_0)$ to its 1-dimensional transitions.*

Notation 12 *We let $\mathcal{N}^\circ$ denote the concurrent marking graph of the Petri net $\mathcal{N}$, i.e. it is the step transition automaton whose skeleton is the (sequential) marking graph $\mathcal{N}^*$ of $\mathcal{N}$ and whose independence relation is given by the step firing rule of Petri nets given in Def. 8.*

2.3 Higher-Dimensional Automata

We call higher-dimensional state of a step transition system a pair $< s, \alpha >$ where s is a state and α a step enabled in s. It is an n-dimensional state where n is the cardinality of α, and is interpreted as a situation in which the n independent actions in α are concurrently executed. We shall identify the states of a step transition system with its 0-dimensional states and its transitions with its 1-dimensional states. All together the higher-dimensional states constitute a (deterministically) labelled higher dimensional automaton [Gou95]. The same information can be represented by an ordinary automaton on the splitted alphabet in which each action of the step transition system is represented by a pair of events denoting the starting and the termination of the action.

Definition 13 (Splitting of a Step Transition System)
The splitting of a step transition system $\mathcal{A} = (S, A, [>, T, s_0)$ is the automaton $\mathbf{Split}(\mathcal{A}) = (\overline{S}, A + A, \overline{T}, s_0)$ *defined as follows.* $\overline{S} = \{< s, \alpha >$ *s.t.* $s[\alpha>\}$ *is the set of higher-dimensional states. An event of the splitted alphabet is denoted beg(a) or a^+ and respectively end(a) or a^- if it belongs to the first resp. the second component of the sum. The set $\overline{T}$ consists of the transitions $< s, \alpha > \xrightarrow{a^+} < s, \alpha + a >$ for $s[\alpha + a>$, together with the transitions $< s, \alpha > \xrightarrow{a^-} < s', \alpha - a >$ for $s[\alpha>$, $a \in \alpha$ and $s \xrightarrow{a} s'$.*

As asynchronous transition systems are particular instances of step transition systems the above serves also as a definition of the splitting of an asynchronous transition system. As mentioned in the introduction the notion of Higher-dimensional automata is closely connected to the transformation of a Petri net into a pure one by splitting its actions (see Fig. 3).

Definition 14 (Splitting of a Petri Net)
The splitting of a Petri net system $\mathcal{N} = (P, A, F, M_0)$ is the (pure) Petri net system $\mathbf{Split}(\mathcal{N}) = (P + A, A + A, \overline{F}, \overline{M_0})$ *defined as follows. The places of* $\mathbf{Split}(\mathcal{N})$ *are those of $\mathcal{N}$ plus a place denoted $exec(a)$ for each $a \in A$. An event of the splitted alphabet of actions is denoted $beg(a)$ or a^+ and respectively $end(a)$ or a^- if it belongs to the first resp. the second component of the sum. The flow relation $\overline{F}$ is given by: $\overline{F}(x, beg(a)) = F(x, a)$, $\overline{F}(end(a), x) = F(a, x)$, $\overline{F}(beg(a), x) = \overline{F}(x, end(a)) = 0$, $\overline{F}(e, exec(a)) = 1$ if $e = beg(a)$ and 0 otherwise, and $\overline{F}(exec(a), e) = 1$ if $e = end(a)$ and 0 otherwise. Finally, $\overline{M_0}(x) = M_0(x)$ and $\overline{M_0}(exec(a)) = 0$.*

Proposition 15 *The splitting of the step transition system associated to a Petri net system is isomorphic to the marking graph of its splitted net system:*

$$\mathbf{Split}(\mathcal{N}^\circ) \cong \mathbf{Split}(\mathcal{N})^*$$

Proof: Let Q denote the set of pairs $< M, \alpha >$ where M is a marking of $\mathcal{N}$ and α a step enabled at M; thus the states of $\mathbf{Split}(\mathcal{N}^\circ)$ are these elements $< M, \alpha >$ of Q for which marking M is reachable from the initial marking M_0. We define a transition system $(Q, A + A, T)$ on Q consisting of the transitions $< M, \alpha > \xrightarrow{a^+} < M, \alpha + a >$ for $M[\alpha + a>$and the transitions $< M, \alpha > \xrightarrow{a^-} < N, \alpha - a >$ for $M[\alpha>$, $a \in \alpha$ and $M \xrightarrow{a} N$ in the transition relation associated with the net $\mathcal{N}$ (i.e. ${}^\bullet a \leq M$ and $N = (M - {}^\bullet a) + a^\bullet$). Thus if $< M, \alpha > \xrightarrow{x} < N, \beta >$ in T, then marking N is reachable from M and every state of Q reachable from $< M_0, 0 >$ is a state of $\mathbf{Split}(\mathcal{N}^\circ)$. The converse holds because $M \xrightarrow{a} N$ entails $< M, \alpha > \xrightarrow{a^+} < M, \alpha + a > \xrightarrow{a^-} < N, \alpha >$ for all step $\alpha + a$ enabled in marking M. Thus $\mathbf{Split}(\mathcal{N}^\circ)$ is the restriction of transition system $(Q, A+A, T)$ on the set of elements reachable from $< M_0, 0 >$. For $< M, \alpha >$ an element of Q, we let $M \cdot \alpha$ denote the marking of $\mathbf{Split}(\mathcal{N})$ given by $M \cdot \alpha(x) = M(x) - \sum_{a \in A} \alpha(a) \cdot {}^\bullet a(x)$ for all $x \in P$ and $M \cdot \alpha(exec(a)) = \alpha(a)$ for all $a \in A$. in particular the initial state $< M_0, 0 >$ of $\mathbf{Split}(\mathcal{N}^\circ)$ is associated with $M_0 \cdot 0 = \overline{M_0}$, i.e. the initial state of $\mathbf{Split}(\mathcal{N})^*$. This correspondence between the set Q and the set of markings of $\mathbf{Split}(\mathcal{N})$ is a bijection: in the converse direction if M' is a marking of $\mathbf{Split}(\mathcal{N})$ we let α be the step given by $\alpha(a) = M'(exec(a))$ and M the marking of $\mathcal{N}$ whose value on place $x \in P$ is given by $M(x) = M'(x) - \sum_{a \in A} {}^\bullet a(x)$, then step α is enabled at M and $M \cdot \alpha = M'$.

Now $M \cdot \alpha \stackrel{a^+}{\to} M'$ in $\mathbf{Split}(\mathcal{N})^\circ$ for some M' if and only if for all $x \in P$, $M \cdot \alpha(x) \geq {}^\bullet a(x)$ i.e. $M(x) \geq \sum_{b \in A} \alpha(b) \cdot {}^\bullet b(x) + {}^\bullet a(x)$ i.e. $M[\alpha + a\!>$ i.e. $< M, \alpha > \stackrel{a^+}{\to} < M, \alpha + a >$ in $\mathbf{Split}(\mathcal{N}^*)$; and M' is given by: $M'(x) = M(x) - {}^\bullet a(x)$ for all $x \in P$ and $M'(exec(b)) = M(exec(b))$ if $b \neq a$ and $M'(exec(a)) = M(exec(a)) + 1$ thus $M' = M \cdot (\alpha + a)$. Hence

$$< M, \alpha > \stackrel{a^+}{\to} < M, \alpha + a > \text{ in } \mathbf{Split}(\mathcal{N}^\circ) \quad \textbf{iff} \quad M \cdot \alpha \stackrel{a^+}{\to} M \cdot (\alpha + a) \text{ in } \mathbf{Split}(\mathcal{N})^*$$

Now $M \cdot \alpha \stackrel{a^-}{\to} M'$ in $\mathbf{Split}(\mathcal{N})^*$ for some M' if and only if $M \cdot \alpha(exec(a)) \geq 1$ i.e. $\alpha(a) \geq 1$, i.e. $a \in \alpha$. This entails that a is enabled in marking M, i.e. $M \stackrel{a}{\to} N$ with $N = M - {}^\bullet a + a^\bullet$. Now, marking M' is given by $M'(exec(b)) = M(exec(b))$ if $b \neq a$ and $M'(exec(a)) = M(exec(a)) - 1$ and for all $x \in P$, $M'(x) = M \cdot \alpha(x) + a^\bullet(x)$ i.e. $M'(x) = (N \cdot (\alpha - a))(x)$. Hence

$$< M, \alpha > \stackrel{a^-}{\to} < N, \alpha - a > \text{ in } \mathbf{Split}(\mathcal{N}^\circ) \quad \textbf{iff} \quad M \cdot \alpha \stackrel{a^-}{\to} N \cdot (\alpha - a) \text{ in } \mathbf{Split}(\mathcal{N})^*$$

And the required isomorphism follows. ∎

This result does not hold for the class of C/E net systems as illustrated in Fig. 6; of course it holds for the subclass of contact-free C/E nets systems for which the previous result applies since this class can be identified with the class of safe Petri net systems.

Corollary 16 *If a step transition system is the behaviour of a Petri net system, then its splitting is the marking graph of a pure Petri net system. If an asynchronous transition system is the behaviour of a safe Petri net system, then its splitting is the marking graph of an elementary net system.*

Proof: If an asynchronous transition system is the behaviour of a safe Petri net system this latter can be chosen place-simple (two places with the same flow relations coincide); then its splitting is simple and pure and since an asynchronous transition system and thus its splitting is reduced, the splitting net system is an elementary net system. ∎

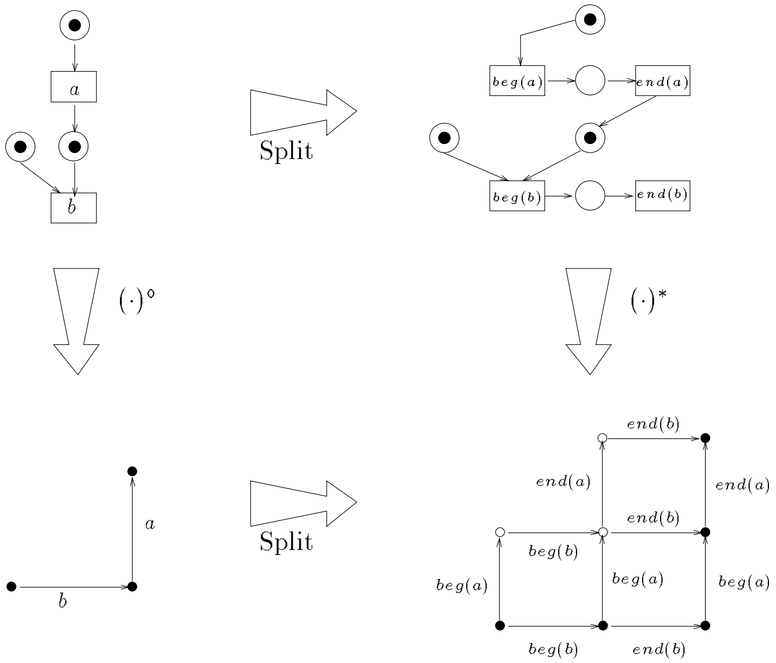


Figure 6: a C/E net for which $\mathbf{Split}(\mathcal{N}^{\circ})$ is not isomorphic to $\mathbf{Split}(\mathcal{N})^{*}$

3 Petri Net Synthesis

In the next section we establish the converse of Cor. 16, i.e. we prove that *(i)* a step transition system is the behaviour of a Petri net system if and only if its splitting is the marking graph of a pure Petri net system, and *(ii)* an asynchronous transition system is the behaviour of a safe Petri net system if and only if its splitting is the marking graph of an elementary net system. For that purpose we need to relate the synthesis of a Petri net from its concurrent behaviour to the synthesis of a pure Petri net from its marking graph. In this section we give a brief survey of the problem of net synthesis necessary to the developments of the next section. More detailed accounts on this subject can be found in [BD95] or in [DR96] (for the particular case of elementary net systems).

3.1 Regions in Transition Systems

The synthesis problem for nets was originally solved for elementary net systems using the notion of *region* ([ER90, NRT92, DR96]). The solution was then extended to pure Petri nets [BDPV96] using the related notion of *generalized region*. The solution can easily be generalized to various kinds of nets; the key observation [BD95] is that the regions of a transition system can be identified with the morphisms from that transition system into a classifying transition system, called a *type of nets*, which characterizes the behaviour of the considered class of nets. More precisely, if I is a set of so-called *incidence values*, we let $\mathbf{Nets}(I)$ denote the set of nets (P, A, C) consisting of a set of *places* P, a disjoint set of *events* A, and a *connectivity matrix* $C : P \times A \to I$.

Definition 17 (Type of Nets) *If I is a set of incidence values, a type $\tau \in \mathbf{Types}(I)$ is any deterministic transition system $\tau = (L, I, \tau)$ with labels in I; i.e. $\tau \subseteq L \times I \times L$.*

L, called the set of local states, gives the range of values of any place in a net. Nets are purely syntactic objects; the purpose of types is to define their behaviours. More precisely a type in $\mathbf{Types}(I)$ describes the sequential firing rule of nets.

Definition 18 (State Graph) *The (sequential) state graph –or marking graph– of a net* $\mathcal{N} = (P, A, C) \in$**Nets**$(I)$ *relative to a type* $\tau = (L, I, \tau) \in$ **Types**(I) *is the transition system* $\mathbf{SG}(\mathcal{N}, \tau) = (L^P, A, T)$ *whose transition relation is given by:*

$$M \xrightarrow{a} M' \in T \quad \textbf{iff} \quad \forall x \in P \;\; M(x) \xrightarrow{C(x,a)} M'(x) \in \tau \qquad (1)$$

The states $M, M' \in L^P$ *are called markings. The state graph of a net system* $\mathcal{N} = (P, A, C, M_0)$ *is the automaton* $\mathcal{N}^*$ *with initial state* M_0 *and whose underlying transition system is the part of* $\mathbf{SG}(\mathcal{N}, \tau)$ *accessible from* M_0*.*

The sequential firing rule of general Petri nets is given by the type $\tau_{Petri} = (\mathbb{N}, \mathbb{N} \times \mathbb{N}, \tau_{Petri})$ with $\mathbb{N}$ as set of local states, $\mathbb{N} \times \mathbb{N}$ as set of incidence values, and set of transitions given by

$$(n \xrightarrow{(i,j)} m) \in \tau_{Petri} \quad \Leftrightarrow \quad n \geq i \;\; \wedge \;\; m = (n - i) + j$$

$C(x, a) = (i, j)$ stands for ${}^\bullet a(x) = i$ and $a^\bullet(x) = j$; i.e. the connectivity matrix is given by $C(x, a) = (F(x, a), F(a, x))$. For pure Petri nets, the type simplifies to $\tau^{pure}_{Petri} = (\mathbb{N}, \mathbb{Z}, \tau^{pure}_{Petri})$ where

$$n \xrightarrow{p} m \in \tau^{pure}_{Petri} \quad \Leftrightarrow \quad m = n + p$$

with the convention that if $C(x, a) = p$ then ${}^\bullet a(x) = max(0, p)$ and $a^\bullet(x) = max(0, -p)$, i.e. $C(x, a) = F(a, x) - F(x, a)$: τ^{Pure}_{Petri} is viewed as the sub transition system of τ_{Petri} obtained by encoding positive integer n by the pair $(0, n)$ and negative integers $-n$ by the pair $(n, 0)$. Similarly, the type $\tau_{C/E} = (\mathbf{2}, \mathbf{2} \times \mathbf{2}, \tau_{C/E})$ of C/E nets has $\mathbf{2} = \{0, 1\}$ as its set of local states, $\mathbf{2} \times \mathbf{2}$ as its set of incidence values and its transitions are given by

$$(n \xrightarrow{(i,j)} m) \in \tau_{C/E} \quad \Leftrightarrow \quad n \geq i \;\; \wedge \;\; m = (n - i) + j$$

$C(x, a) = (i, j)$ stands for $x \in {}^\bullet a$ (respectively $x \in a^\bullet$) if and only if $i = 1$ (resp. $j = 1$). The type of pure C/E nets $\tau^{pure}_{C/E} = (\mathbf{2}, \{0, +1, -1\}, \tau^{pure}_{C/E})$ is given by

$$n \xrightarrow{p} m \in \tau^{pure}_{C/E} \quad \Leftrightarrow \quad m = n + p$$

where $C(x, a) = p$ means that $x \in {}^\bullet a$ (respectively $x \in a^\bullet$) if and only if $p = 1$ (resp. $p = -1$). Equation (1) in Def. 18 means that for every place $x \in P$ of $\mathcal{N}$, the mapping $ext(x)$ that takes the transition $M \xrightarrow{a} M'$ to the transition

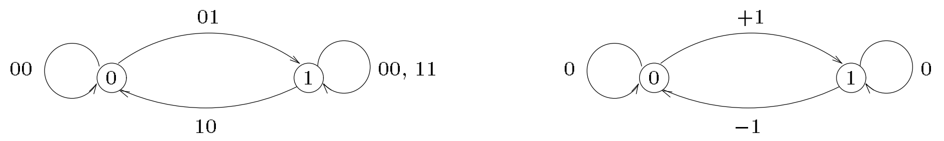


Figure 7: the type of C/E nets and of pure C/E nets

$M(x) \stackrel{C(x,a)}{\longrightarrow} M'(x)$ is a morphism of transition systems from the state graph of $\mathcal{N}$ to the type τ where a morphism of transition systems from $\mathcal{T} = (S, A, T)$ to $\mathcal{T}' = (S', A', T')$ is a pair (σ, η) of mappings $\sigma : S \to S'$ and $\eta : A \to A'$ taking states (respectively actions) of $\mathcal{T}$ to states (resp. actions) of $\mathcal{T}'$, such that $s \stackrel{a}{\to} s' \in T$ implies $\sigma(s) \stackrel{\eta(a)}{\to} \sigma(s') \in \mathcal{T}'$. We call this morphism the *extension* of place x.

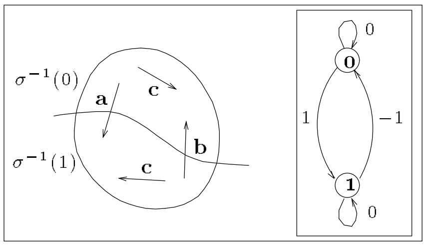
Definition 19 (Region) *The set of regions of a transition system $\mathcal{T}$ relative to a type $\tau = (L, I, \tau) \in$ **Types**(I) is the set of morphisms of transition systems from $\mathcal{T}$ to τ.*

For instance, see Fig. 8, regions in $\mathcal{T} = (S, A, T)$ relative to the type $\tau_{C/E}^{pure}$ of pure C/E nets, i.e. morphisms $(\sigma, \eta) : \mathcal{T} \to \tau_{C/E}^{pure}$, are in bijective correspondence with the subsets of states $X \subseteq S$ satisfying the predicate:

$$
\begin{array}{rrcl}
\mathbf{Region}(X) & \equiv & \multicolumn{2}{l}{\textit{for every event } a \in A :} \\
 & s \stackrel{a}{\to} s' & \Rightarrow & (s \in X \ \text{ and } \ s' \notin X) \\
\textit{or} & s \stackrel{a}{\to} s' & \Rightarrow & (s \notin X \ \text{ and } \ s' \in X) \\
\textit{or} & s \stackrel{a}{\to} s' & \Rightarrow & (s \in X \ \text{ iff } \ s' \in X)
\end{array}
$$

This was the original definition of region given by Ehrenfeucht and Rozenberg. In fact, if **Region**(X) then the characteristic function $\sigma : S \to \{0, 1\}$ of X admits a (unique) companion map $\eta : A \to \{-1, 0, 1\}$ such that $\eta(a) = \sigma(s')$ - $\sigma(s)$ for every transition $s \stackrel{a}{\to} s'$, and conversely if $(\sigma, \eta) : \mathcal{T} \to \tau_{C/E}^{pure}$ then **Region**(X) holds for $X = \sigma^{-1}(1)$.

We call *generalized region* a region relative to the type τ_{Petri}^{pure} of pure Petri nets, i.e. a morphism $(\sigma, \eta) : \mathcal{T} \to \tau_{Petri}^{pure}$. As illustrated in Fig. 8, a generalized region assigns each state to non negative integer (the "potential" of the state) such that each occurrence of a given event corresponds to a uniform



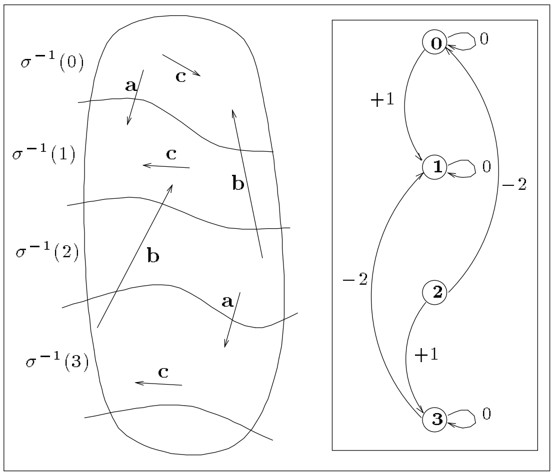


Figure 8: an ordinary region (region for C/E nets) and a generalized region (region for Petri nets) as morphisms

modification of the potential:

$$s \xrightarrow{a} s' \quad \Rightarrow \quad \sigma(s') - \sigma(s) = \eta(a)$$

3.2 Representation Theorem

The notion of region is the abstraction to arbitrary transition systems of the notion of extension of places in the marking graph of nets. The places of the net synthesized from a transition system will indeed be regions of that transition system. More precisely, a region $x = (\sigma, \eta) : \mathcal{T} \rightarrow \tau$ of a transition system $\mathcal{T} = (S, A, T)$ relative to a type τ can be associated with an atomic net $(\{x\}, A, C_x)$ where $C_x(a, x) = \eta(a)$. By definition of regions, the mapping that takes a state $s \in S$ to the marking M such that $M(x) = \sigma(s)$ induces a morphism of transition systems acting as the identity on events from $\mathcal{T}$ to $\mathcal{N}_x^*$, the marking graph of $\mathcal{N}_x$. If $\mathcal{T}$ comes equipped with an initial state s_0, we can supply $\mathcal{N}_x$ with the corresponding initial marking: $M_0(x) = \sigma(s_0)$. More generally if P is a subset of regions we form the net $\mathcal{N}_P$ obtained by assembling the atomic nets $\mathcal{N}_x$ for $x \in P$.

Definition 20 (Synthesized Net System) *The net system synthesized from a subset P of regions of the underlying transition system of an automaton $\mathcal{A} = (S, A, T, s_0)$ is $\mathcal{A}^*_P = (P, A, C, M_0)$ where $C(x, a) = \eta(a)$ and $M_0(x) = \sigma(s_0)$ for any event $a \in A$ and region $x = (\sigma, \eta) \in P$. If $R_\tau(\mathcal{A})$ denotes the set of regions of $\mathcal{A}$ relative to type τ, we let $\mathcal{A}^* = \mathcal{A}^*_{R_\tau(\mathcal{A})}$ denote the net system synthesized from $\mathcal{A}$.*

Two places x and y are said to be *redundant* in net system $\mathcal{N} = (P, A, C, M_0)$ if they have the same initial value and identical flow relations:

$$x \equiv y \quad \Leftrightarrow \quad M_0(x) = M_0(y) \text{ and } \forall a \in A \;:\; C(x, a) = C(y, a)$$

$\mathcal{N}$ is said to be *place-simple* if it has no redundant places. From now we shall assume that there is no *redundant* places in nets; similarly we shall make no distinction between two net systems which are identical up to a bijective renaming of their sets of places. Therefore we consider the set $\mathbf{Nets}(\mathrm{A})$ of isomorphism classes of place-simple net systems with alphabet of events A. One can define an order relation on $\mathbf{Nets}(\mathrm{A})$ by setting $\mathcal{N}_1 \leq \mathcal{N}_2$ if there exists a mapping $\beta : P_1 \to P_2$ between their respective sets of places such that $\forall x \in P_1\ \forall a \in A \quad C_2(\beta x, a) = C_1(x, a)$, and $M_1(x) = M_2(\beta(x))$ (where M_1 and M_2 are the respective initial markings of $\mathcal{N}_1$ and $\mathcal{N}_2$). Since there is no redundant places such a map when it exists is necessarily unique and injective. More precisely because we don't distinguish between redundant places and abstract from their identity, one can assume that each place x in net $\mathcal{N} = (P, A, C, M_0)$ be represented by the pair $(M_0(x), C(x, -)) \in L \times I^A$ i.e. one can assume $P \subseteq L \times I^A$; then the order $\mathcal{N}_1 \leq \mathcal{N}_2$ is given by the corresponding inclusion $P_1 \subseteq P_2$ between their respective set of places. Therefore $\mathbf{Nets}(\mathrm{A})$ is a complete lattice where join $\bigvee_{k \in K} \mathcal{N}_k$ and meet $\bigwedge_{k \in K} \mathcal{N}_k$ are respectively given by the unions and the intersections of the corresponding sets of places; the least element has an empty set of places and the greatest element is the "universal net" $\top = (P, A, C, M_0)$ where $P = L \times I^A$, and $M_0(x) = \ell$ and $C(x, a) = \varphi(a)$ for each place $x = (\ell, \varphi) \in P$ and $a \in A$. For instance a net system $\mathcal{N} = (P, A, C, M)$ is (isomorphic to) the least upper bound of the atomic net systems $\mathcal{N}_x = (\{x\}, A, C_x, M_x)$ induced by places $x \in P$, i.e. $C_x(x,a) = C(x,a)$ for all $a \in A$ and $M_x(x) = M(x)$.

Observation 21 *Let $R_\tau(\mathcal{A})$ denotes the set of regions of $\mathcal{A}$ relative to type τ, then $\mathcal{A}_P^* = \bigvee_{x\in P}\mathcal{A}_{\{x\}}^*$ for all $P \subseteq R_\tau(\mathcal{A})$ and therefore if $P_1 \subseteq P_2 \subseteq R_\tau(\mathcal{A})$, $\mathcal{A}_{P_1}^* \leq \mathcal{A}_{P_2}^*$.*

Let $\mathbf{Aut}(\mathrm{A})$ denote the set of isomorphism classes of deterministic and accessible automata with alphabet of events A. As for net systems we shall make no distinction between two automata which are identical up to a bijective renaming of their sets of states. One can define an order relation on $\mathbf{Aut}(\mathrm{A})$ by setting $\mathcal{A}_1 \leq \mathcal{A}_2$ if there exists a morphism $(\sigma, 1_A)$ from $\mathcal{A}_1$ to $\mathcal{A}_2$ which acts as the identity on the set of events. By accessibility of the automata, when such a mapping σ exists it is necessarily unique, therefore if $\mathcal{A}_1 \leq \mathcal{A}_2$ and $\mathcal{A}_2 \leq \mathcal{A}_1$ the automata $\mathcal{A}_1$ and $\mathcal{A}_2$ are isomorphic. The greatest lower bound $\bigwedge_{i\in I}\mathcal{A}_i$ of a family of automata $\mathcal{A}_i = (S_i, A, T_i, s_{0,i})$ is their *synchronized product*, i.e. the restriction of the product automaton $\mathcal{A} = (S, A, T, s_0)$ with set of states $S = \prod_{k\in K} S_k$, initial state $s_0 = (s_{0,k})_{k\in K}$ and transition relation given by the product $\prod_{k\in K} T_k$:

$$(s_i)_{k\in K} \xrightarrow{a} (s'_k)_{k\in K} \in T \quad \textbf{iff} \quad \forall k \in K \quad s_k \xrightarrow{a} s'_k \in T_k$$

to the set of states accessible from the initial state s_0 and to the set of transitions whose extremities are accessible states. For instance $(\mathcal{A}_P^*)^* \cong \bigwedge_{x\in P} (\mathcal{A}_x^*)^*$ and more generally the marking graph of a net system is, by definition, the synchronized product of the marking graphs of the atomic net systems associated with each of its places.

Observation 22 *There exists a bijective correspondence between regions $x \in R_\tau(\mathcal{A})$ and (isomorphic classes of) atomic net systems $\mathcal{N}$ such that $\mathcal{A} \leq \mathcal{N}^*$.*

Proof: If $x \in R_\tau(\mathcal{A})$, then as already noticed $\mathcal{A} \leq \mathcal{N}_x^*$ where $\mathcal{N}_x = \mathcal{A}_{\{x\}}^*$ is the atomic net associated to x. Conversely if $\mathcal{A} \leq \mathcal{N}^*$ for some atomic net $\mathcal{N} = (\{x\}, A, C, M)$, since $\mathcal{N}^*$ is (isomorphic) to the part of transition system (L, A, τ_C) accessible from initial state ℓ_0 where $\ell \xrightarrow{a} \ell' \in \tau_C$ if and only if $\ell \xrightarrow{i} \ell'$ $\in \tau$ where $C(x,a) = i$ and $\ell_0 = M(x)$. Therefore a morphism of transition systems $(\sigma, 1) : \mathcal{A} \to \mathcal{N}^*$ is the same data as a morphism $(\sigma, \eta) : \mathcal{A} \to \tau$ where

$\eta(a) = C(x,a)$, i.e. a region $x = (\sigma,\eta) \in R_\tau(\mathcal{A})$ whose state graph of the induced atomic net $\mathcal{A}_x = (\mathcal{N}_x)^*$ is (isomorphic to) $\mathcal{N}^*$. ∎

Observation 23 *The correspondences that take an automaton to its synthesized net system and a net system to its state graph constitute a Galois connection between the ordered sets* $\mathbf{Nets}(A)$ *and* $\mathbf{Aut}(A)$*:*

1. $\mathcal{A}_1 \leq \mathcal{A}_2 \quad \Rightarrow \quad \mathcal{A}_2^* \leq \mathcal{A}_1^*$ *for* $\mathcal{A}_1,\ \mathcal{A}_2 \in \mathbf{Aut}(A)$*,*
2. $\mathcal{N}_1 \leq \mathcal{N}_2 \quad \Rightarrow \quad \mathcal{N}_2^* \leq \mathcal{N}_1^*$ *for* $\mathcal{N}_1,\ \mathcal{N}_2 \in \mathbf{Nets}(A)$*,*
3. $\mathcal{A} \leq \mathcal{N}^* \quad \Leftrightarrow \quad \mathcal{N} \leq \mathcal{A}^*$ *for* $\mathcal{A} \in \mathbf{Aut}(A)$ *and* $\mathcal{N} \in \mathbf{Nets}(A)$*.*

Proof: A Galois connection is in fact characterized by the third condition the other two follow therefrom, however it is instructive to directly check that both correspondences are decreasing mappings. Let $\mathcal{A}_1 \leq \mathcal{A}_2$, by composition with $(\sigma,1) : \mathcal{A}_1 \to \mathcal{A}_2$ any region (σ_2,η_2) of $\mathcal{A}_2$ induces a region $(\sigma_2 \circ \sigma, \eta_2)$ of $\mathcal{A}_2$ with the same flow relations, it follows that $\mathcal{A}_2^* \leq \mathcal{A}_1^*$. Let $\mathcal{N}_1 \leq \mathcal{N}_2$, we can assume without loss of generality that $P_1 \subseteq P_2$ and then $\mathcal{N}_1^* = \bigwedge_{x\in P_1} \mathcal{A}_x \geq \bigwedge_{x\in P_2} \mathcal{A}_x = \mathcal{N}_2^*$ where $\mathcal{A}_x$ is the state graph of the atomic net induced by region x. By Obs. 22, one has $\mathcal{A} \leq \mathcal{N}^* \Leftrightarrow \mathcal{N} \leq \mathcal{A}^*$ for $\mathcal{A} \in \mathbf{Aut}(\mathrm{A})$ and $\mathcal{N}$ an *atomic* net. Now $\mathcal{A} \leq \mathcal{N}^*$ for a general net $\mathcal{N} = \bigvee_{x\in P} \mathcal{N}_x$ if and only if $\mathcal{A} \leq \bigwedge_{x\in P} \mathcal{N}_x^*$ if and only if $\forall x \in P\ \mathcal{A} \leq \mathcal{N}_x^*$ if and only if $\forall x \in P\ \mathcal{N}_x \leq \mathcal{A}^*$ (because $\mathcal{N}_x$ is atomic) if and only if $\mathcal{N} = \bigvee_{x\in P} \mathcal{N}_x \leq \mathcal{A}^*$. ∎

The mapping that takes an automaton $\mathcal{A}$ to its double dual $c(\mathcal{A}) = \mathcal{A}^{**}$ is a *closure operator* i.e. it is increasing ($\mathcal{A}_1 \leq \mathcal{A}_2 \to c(\mathcal{A}_1) \leq c(\mathcal{A}_2)$), extensive ($\mathcal{A} \leq c(\mathcal{A})$) and idempotent ($c(c(\mathcal{A})) = c(\mathcal{A})$). An automaton equal to its closure is said to be *τ-separated*; i.e. $\mathcal{A}$ is τ-separated if it is isomorphic to the product of the $\mathcal{A}_x$ associated with regions ($\mathcal{A} \cong \bigwedge_{x\in R_\tau(\mathcal{A})} \mathcal{A}_x$ where $\mathcal{A}_x = (\mathcal{A}_{\{x\}}^*)^*$ is the marking graph of the net system associated with region x). Similarly the mapping that takes a net system to its double dual is a closure operator on $\mathbf{Nets}(\mathrm{A})$, a net system which coincides with its closure is said to be *τ-saturated.* From the Galois connection we deduce an order-isomorphism between the complete lattices of separated automata and saturated net systems. Of course, if $P \subseteq P' \subseteq R_\tau(\mathcal{A})$ then $\bigwedge_{x\in P} \mathcal{A}_x \geq \bigwedge_{x\in P'} \mathcal{A}_x$. A set of regions $P \subseteq R_\tau(\mathcal{A})$ is *admissible* if

$\mathcal{A} \cong \bigwedge_{x\in P} \mathcal{A}_x$ and $P \subseteq R_\tau(\mathcal{A})$ is *complete* if $\bigwedge_{x\in P} \mathcal{A}_x \cong \bigwedge_{x\in R_\tau} \mathcal{A}_x$. Equation (1) states that the state graph of a net system is separated: extensions of places constitute an admissible set of regions. Conversely:

Proposition 24 (Representation Theorem)
Let R_τ be the set of regions, relative to a type τ, of an accessible automaton $\mathcal{A}$, then $P \subseteq R_\tau$ is admissible if and only if the following two axioms of separation are satisfied

1. $s \neq s' \;\Rightarrow\; \exists(\sigma,\eta) \in P \;\cdot\; \sigma(s) \neq \sigma(s')$
2. $s \overset{a}{\not\rightarrow} \;\Rightarrow\; \exists(\sigma,\eta) \in P \;\cdot\; \sigma(s) \overset{\eta(a)}{\not\rightarrow} \quad (in\;\; \tau)$

and in that case $\mathcal{A} \cong (\mathcal{N}_P)^*$; i.e. $\mathcal{A}$ is isomorphic to the marking graph of the net synthesized from an admissible set of regions.

Proof: $\mathcal{A} \leq \bigwedge_{x\in P} \mathcal{A}_x = (\mathcal{N}_P)^*$; and the corresponding morphism takes any state s of $\mathcal{A}$ to the vector $(\sigma(s))_{x=(\sigma,\eta)\in P}$ of local states (each local state gives its value relative to some region). The first separation axiom expresses that this encoding is injective. The second separation axiom states that an action is not enabled in some state of the automaton if and only if there exists some component automaton such that this action is not enabled in the corresponding local state. Altogether they state that the automaton is isomorphic to the synchronized product of the automata $\mathcal{A}_x$ for regions $x \in P$. ∎
When x is a region which satisfies the instance of the first separation property relative to the pair of states s, s', we say that the region x *distinguishes* these states. If x is a region which satisfies the instance of the second separation property relative to the pair consisting of state s and event a not enabled in s, we say that the region x *inhibits* event a in state s.

For (general) Petri nets, a (generalized) region $x = (\sigma,\eta)$ inhibits an event a in a state s when it is an input place of that event (${}^\bullet a(x) > 0$) which does not contain enough tokens in the marking associated to state s in order to allow the event to take place ($\sigma(s) < {}^\bullet a(x)$). For C/E nets, a region $x = (\sigma,\eta)$ inhibits an event a in a state s either when it is an input place of that event ($x \in {}^\bullet a$) which does not contain state s, or when it is an output place of that event ($x \in a^\bullet$) which contains state s. In the former case we speak of *forward inhibition* and the latter of *backward inhibition*. To every region $x = (\sigma,\eta)$ may be associated a *complementary* region $\overline{x} = (\overline{\sigma}, \overline{\eta})$ where $\overline{\sigma}(s) = 1 \Leftrightarrow \sigma(s) = 0$ and

$$\overline{\eta}(a) = \begin{cases} (0,1) & \textbf{if} \quad \eta(a) = (1,0) \\ (1,0) & \textbf{if} \quad \eta(a) = (0,1) \\ (0,0) & \textbf{if} \quad \eta(a) \in \{(0,0),(1,1)\} \end{cases}$$

A region forwardly inhibits an event in a state if and only if the complementary region backwardly inhibits this event in this state; and vice versa. Therefore if an automaton is separated we can find an admissible set of regions for which all instances of separation may be solved without using backward inhibitions. Moreover the C/E net generated from this set of admissible regions is *contact-free*.

3.3 Conditions in Step Transition Systems

Definition 25 (Extended Type of Nets) *A type $\tau = (L, I, \tau)$ is an extended type if I comes equipped with a structure of a commutative monoid so that it endows τ with a structure of a step transition system.*

As we saw a type in $\mathbf{Types}(I)$ describes the sequential firing rule of nets; now every structure of commutative monoid for which it is an extended type induces a corresponding step firing rule.

Definition 26 (Concurrent State Graph) *The concurrent state graph of a net $\mathcal{N} = (P, A, C) \in$$\mathbf{Nets}(I)$ relative to an extended type $\tau = (L, I, \tau) \in$ $\mathbf{Types}(I)$ is the step transition system $\mathbf{SG}(\mathcal{N}, \tau) = (L^P, <A>, T)$ whose transition relation is given by:*

$$M \stackrel{\alpha}{\to} M' \in T \quad \textbf{iff} \quad \forall x \in P \;\; M(x) \stackrel{C(x,\alpha)}{\longrightarrow} M'(x) \in \tau \tag{2}$$

where $C(x, \alpha) = \sum_{i=1}^{n} C(x, a_i)$ if $\alpha = a_1 + \cdots + a_n$. The states $M, M' \in L^P$ are called markings. The state graph of a net system $\mathcal{N} = (P, A, C, M_0)$ is the step automaton $\mathcal{N}^\circ$ with initial state M_0 and whose underlying transition system is the part of $\mathbf{SG}(\mathcal{N}, \tau)$ accessible from M_0.

The step firing rule of a general Petri net is given by the type τ_{Petri} where $\mathbb{N} \times \mathbb{N}$ is endowed with the structure of commutative monoid given by the componentwise addition. Equation 2 reads as follows: for every multiset $\alpha = a_1 + \cdots a_n$.

$$M \xrightarrow{\alpha} N \;\;\Leftrightarrow\;\; \forall x \in P \quad M(x) \geq \sum_{i=1}^{n} {}^\bullet a_i(x) \;\wedge\; N(x) = M(x) + \sum_{i=1}^{n}(a_i{}^\bullet(x) - {}^\bullet a_i(x))$$

We let ${}^\bullet\alpha$ and $\alpha^\bullet$ be the multisets of places given by $C(x,\alpha) = ({}^\bullet\alpha(x), \alpha^\bullet(x))$; i.e. ${}^\bullet\alpha(\mathrm{x}) = \sum_{i=1}^{n} {}^\bullet a_i(x)$ and $\alpha^\bullet(\mathrm{x}) = \sum_{i=1}^{n} a_i{}^\bullet(x)$. The type τ_{Petri}^{pure} of pure Petri nets can be interpreted as an extended type where $\mathbb{Z}$ is equipped with the structure of commutative monoid given by the addition, however this is not consistent with the previous step firing rules given for general Petri net. Similarly, the type $\tau_{C/E} = (\mathbf{2}, <\mathbf{2} \times \mathbf{2}>, \tau_{C/E})$ can be interpreted as an extended type when $\mathbf{2}$ is equipped with the structure of commutative monoid given by the disjunction, but here again this is not consistent with the usual step firing rule of C/E nets. If we interpret $\tau_{C/E}$ as an extended type (i.e. as a step transition system) over the commutative monoid $\mathbb{N} \times \mathbb{N}$ then we obtain the usual step firing rule of C/E nets.

We let a morphism of step transition systems from $\mathcal{T} = (S, M, T)$ to $\mathcal{T}$' $= (S', M', T')$ consists of a mapping $\sigma : S \to S'$ between their respective sets of states together with a morphism of monoids $\eta : M \to M'$ such that $s \stackrel{\alpha}{\to} s'$ implies $\sigma(s) \stackrel{\eta(\alpha)}{\to} \sigma(s')$. Equation 2 in Def. 18 means that for every place $x \in P$ of $\mathcal{N}$, the mapping $ext(x)$ that takes the transition $M \stackrel{\alpha}{\to} M'$ to the transition $M(x) \stackrel{C(x,\alpha)}{\longrightarrow} M'(x)$ is a morphism of step transition systems from the (concurrent) state graph of $\mathcal{N}$ to the extended type τ. We call this morphism the *extension* of place x.

Definition 27 (Condition) *The set of conditions of a step transition system $\mathcal{T}$ relative to an extended type $\tau = (L, I, \tau)$ is the set of morphisms of step transition systems from $\mathcal{T}$ to τ.*

We recall that the *skeleton* of a step transition system $\mathcal{A} = (S, <A>, T)$ over a free commutative monoid is its restriction to the set of "one-dimensional" transitions, namely $|\mathcal{A}| = (S, A, T \cap (S \times A \times S))$.

Observation 28 *The conditions of a step transition system $\mathcal{A} = (S, <A>, T)$ over a free commutative monoid are those regions $C_\tau(\mathcal{T}) \subseteq R_\tau(|\mathcal{T}|)$ of its* *skeleton such that for all $s \in S$ and $\alpha = a_1 + \cdots a_n \in \,<\!A\!>$*

$$s \xrightarrow{\alpha} \quad \Rightarrow \sigma(s) \xrightarrow{\eta(\alpha)}$$

where $\eta(\alpha) = \eta(a_1) + \cdots + \eta(a_n) \in I$.

By analogy with regions we shall call generalized condition a condition relative to the (extended) type of general Petri nets, and (ordinary) condition, a condition relative to the (extended) type of C/E nets. The preceding observation specialized as follows.

Observation 29 *A generalized condition of a step transition system is a region $x = (\sigma, \eta)$ of its skeleton relative to the type τ_{Petri} of Petri nets such that*

$$s[\alpha\!> \quad \Rightarrow \quad \sum_{a \in A} \alpha(a) \cdot {}^{\bullet}a(x) \leq \sigma(s)$$

if we let $\eta(a) = ({}^{\bullet}a(x), a^{\bullet}(x))$.

A region $x = (\sigma, \eta)$ with $\eta(a) = (i, j)$ provides a place in the synthesized net with ${}^{\bullet}a(x) = i$ and $a^{\bullet}(x) = j$, thus the above condition says that there exists enough tokens in place x at marking M (associated with state s) so that this place does not inhibit any step α enabled in s.

Observation 30 *A condition of an asynchronous transition system is a generalized condition of this asynchronous transition system viewed as a step transition system. It is a region (σ, η) of its skeleton relative to the type $\tau_{C/E}$ of condition/event nets compatible with the independence relation in the sense that*

$$a \| b \quad \Rightarrow \quad x \notin ({}^{\bullet}a \cup a^{\bullet}) \cap ({}^{\bullet}b \cup b^{\bullet})$$

where a region $x = (\sigma, \eta)$ with $\eta(a) = (i, j)$ is viewed as a place (in the synthesized net) with $x \in {}^{\bullet}a$ iff $i \neq 0$ and $x \in a^{\bullet}$ iff $j \neq 0$.

Actually if α is a step enabled in state s, then it is a set $\alpha = a_1 + \cdots + a_n$ (i.e. $i \neq j \Rightarrow a_i \neq a_j$) of pairwise independent events enabled in s (i.e. $\forall i \quad x^{\bullet}(a_i) \leq \sigma(s)$). Since they are pairwise independent, there exists at most one index i for which $x^{\bullet}(a_i) \neq 0$ and then $\sum_i x^{\bullet}(a_i) \leq \sigma(s)$. ∎

Notation 31 *We let $\mathcal{A}^\circ$ denote the net system synthesized from the set $C_\tau(\mathcal{A}) \subseteq R_\tau(|\mathcal{A}|)$ of conditions of a step automaton $\mathcal{A} = (S, <A>, T, s_0)$ over a free commutative monoid:*

$$\mathcal{A}^\circ = |\mathcal{A}|^*_{C_\tau(\mathcal{A})}$$

We let $\mathbf{SAut}(A)$ denote the set of isomorphism classes of accessible step automata over the commutative monoid generated by the alphabet A. It is a subset of $\mathbf{Aut}(<A>)$: we let $U\mathcal{A}$, the underlying automaton of $\mathcal{A}$, denote the step automaton $\mathcal{A}$ viewed as an ordinary automaton over the alphabet $<A>$. The order relation inherited from $\mathbf{Aut}(<A>)$ is equivalently given as: $\mathcal{A}_1 \leq \mathcal{A}_2$ if there exists a morphism of step automata $(\sigma, 1_A)$ from $\mathcal{A}_1$ to $\mathcal{A}_2$ which acts as the identity on the set of events. Moreover the synchronized product of step automata computed in $\mathbf{Aut}(<A>)$ is a step automaton and it provides their greatest lower bound. We then can prove the analog of Obs. 22

Observation 32 *There exists a bijective correspondence between conditions $x \in C_\tau(\mathcal{A})$ and (isomorphic classes of) atomic net systems $\mathcal{N}$ such that $\mathcal{A} \leq \mathcal{N}^\circ$.*

and of Obs. 23

Observation 33 *The correspondences that take a step automaton to its synthesized net system and a net system to its concurrent state graph constitute a Galois connection between the ordered sets* $\mathbf{Nets}(A)$ *and* $\mathbf{SAut}(A)$*:*

1. $\mathcal{A}_1 \leq \mathcal{A}_2 \quad \Rightarrow \quad \mathcal{A}_2^\circ \leq \mathcal{A}_1^\circ$ *for* $\mathcal{A}_1,\ \mathcal{A}_2 \in \mathbf{SAut}(A)$*,*
2. $\mathcal{N}_1 \leq \mathcal{N}_2 \quad \Rightarrow \quad \mathcal{N}_2^\circ \leq \mathcal{N}_1^\circ$ *for* $\mathcal{N}_1,\ \mathcal{N}_2 \in \mathbf{Nets}(A)$*,*
3. $\mathcal{A} \leq \mathcal{N}^\circ \quad \Leftrightarrow \quad \mathcal{N} \leq \mathcal{A}^\circ$ *for* $\mathcal{A} \in \mathbf{SAut}(A)$ *and* $\mathcal{N} \in \mathbf{Nets}(A)$*.*

A step automaton is τ-separated if it is isomorphic to the product of the $\mathcal{A}_x$ associated with conditions ($\mathcal{A} \cong \bigwedge_{x \in C_\tau(\mathcal{A})} \mathcal{A}_x$ where $\mathcal{A}_x = (\mathcal{A}^\circ_{\{x\}})^\circ$ is the marking graph of the net system associated with condition x). Of course, if $P \subseteq P' \subseteq C_\tau(\mathcal{A})$ then $\bigwedge_{x \in P} \mathcal{A}_x \geq \bigwedge_{x \in P'} \mathcal{A}_x$. A set of conditions $P \subseteq C_\tau(\mathcal{A})$ is *admissible* if $\mathcal{A} \cong \bigwedge_{x \in P} \mathcal{A}_x$ and $P \subseteq C_\tau(\mathcal{A})$ is *complete* if $\bigwedge_{x \in P} \mathcal{A}_x \cong \bigwedge_{x \in C_\tau} \mathcal{A}_x$. Equation (2) states that the concurrent state graph of a net system is separated: extensions of places constitute an admissible set of conditions. Conversely:

Proposition 34 (Representation Theorem)
Let $C_\tau(\mathcal{A})$ be the set of conditions, relative to an extended type τ, of an accessible step automaton $\mathcal{A}$, then $P \subseteq C_\tau(\mathcal{A})$ is admissible if and only if the following two axioms of separation are satisfied

1. $s \neq s' \;\Rightarrow\; \exists(\sigma,\eta) \in P \;\cdot\; \sigma(s) \neq \sigma(s')$
2. $s \stackrel{\alpha}{\not\to} \;\Rightarrow\; \exists(\sigma,\eta) \in P \;\cdot\; \sigma(s) \stackrel{\eta(\alpha)}{\not\to} \quad (in\ \tau)$

and in that case $\mathcal{A} \cong (\mathcal{N}_P)^*$; i.e. $\mathcal{A}$ is isomorphic to the marking graph of the net synthesized from an admissible set of regions.

As a particular case one has

Corollary 35 *An accessible step transition system $\mathcal{A}$ is the concurrent behaviour of a Petri net system if and only if it admits a set $P \subseteq R_{Petri}(\mathcal{A})$ of regions (relative to the type τ_{Petri} of Petri nets) such that*

1. $s = s' \Leftrightarrow \forall x = (\sigma,\eta) \in P \;:\; \sigma(s) = \sigma(s')$,
2. $s[\alpha\!> \Leftrightarrow \forall x = (\sigma,\eta) \in P \;:\; \sum_{a\in A} \alpha(a) \cdot x^\bullet(a) \leq \sigma(s)$.

and in that case $\mathcal{A}$ is the concurrent behaviour of the net system $\mathcal{N}_P$ synthesized from this set of (admissible) regions. An accessible asynchronous transition system (viewed as a step transition system) is the concurrent behaviour of a safe Petri net system if and only if it admits a set of admissible regions which are 1*-bounded (i.e. $\forall x = (\sigma,\eta) \in P \;:\; \sigma(S) \subseteq \{0,1\}$).*

4 Reducing the Synthesis of Petri Nets to the Synthesis of Pure Petri Nets

In this section we reduce Nielsen and Winskel' synthesis of safe Petri nets from asynchronous transition, and Mukund' synthesis of Petri nets from step transition systems to the synthesis of respectively elementary net systems and pure Petri nets from ordinary automata.

4.1 Synthesis of Safe Nets from Asynchronous Transition Systems

For nets with simple type, like elementary nets or pure Petri nets, regions $x = (\sigma, \eta)$ are fully characterized by their component on states. This is not the case for C/E nets: let $S_0(x) = \sigma^{-1}(1)$ stand for the set of states (markings) where the condition (place x) holds, this set does not suffice to characterize the region because we cannot make a difference between a place tested by an action ($x \in {}^\bullet a \cap a^\bullet$) and a place which is not related to that action ($x \notin {}^\bullet a \cup a^\bullet$). Otherwise stated we need to decide for each transition $s \xrightarrow{a} s'$ where s and s' are both in $S_0(x)$ whether $\eta(a) = (0,0)$ (i.e. $x \notin {}^\bullet a \cup a^\bullet$) or $\eta(a) = (1,1)$ (i.e. $x \in {}^\bullet a \cap a^\bullet$). We shall say that the property x holds true during the transition $s \xrightarrow{a} s'$ in the former case and does not hold true in the latter case. This is consistent with the splitting of the C/E net system: the property x holds before and after the execution of a but not during the execution of this action when $x \in {}^\bullet a \cap a^\bullet$. Let $S_1(x)$ denote the set of transitions (one-dimensional states) along which condition x holds true, i.e. $S_1(x) = \{s \xrightarrow{a} s' \in T \mid \eta(a) = 00 \ \wedge \ \sigma(s) = 1\}$. The data consisting of $S_0(x)$ and $S_1(x)$ characterizes the region $x = (\sigma, \eta)$:

1. $\sigma(s) = 1 \Leftrightarrow s \in S_0(x)$,
2. if $s \xrightarrow{a} s' \notin S_1(x)$ then $\eta(a) = (\sigma(s), \sigma(s'))$, and
3. if $s \xrightarrow{a} s' \in S_1(x)$ then $\eta(a) = (0,0)$.

This suggests an alternative definition of regions relative to the type $\tau_{C/E}$ of C/E nets and therefore of conditions in asynchronous transition systems in terms of higher-dimensional states. The following was the original definition of conditions given in [NW94], another definition appears in [MN92].

Proposition 36 *A set of states $S_0(x)$ and a set of transitions $S_1(x)$ of an asynchronous transition system induce a condition $x = (\sigma, \eta)$ if the following properties hold*

1. $s \xrightarrow{a} s' \in S_1(x) \Rightarrow (s \in S_0(x) \ \wedge \ s' \in S_0(x))$,
2. *If we let $s \xrightarrow{a} s' \in {}^\bullet x$ (respectively $s \xrightarrow{a} s' \in x^\bullet$) when this transition does not belong to $S_1(x)$ and $s' \in S_0(x)$ (resp. $s \in S_0(x)$) then*

(a) $(s \xrightarrow{a} s' \in {}^{\bullet}x \ \wedge \ u \xrightarrow{a} u') \Rightarrow u \xrightarrow{a} u' \in {}^{\bullet}x$, *and*

(b) $(s \xrightarrow{a} s' \in x^{\bullet} \ \wedge \ u \xrightarrow{a} u') \Rightarrow u \xrightarrow{a} u' \in x^{\bullet}$.

3. $s \xrightarrow{a} s' \in ({}^{\bullet}x \cup x^{\bullet}) \ \wedge u \xrightarrow{b} u' \in ({}^{\bullet}x \cup x^{\bullet}) \Rightarrow not(a \| b)$.

and then $a \in {}^{\bullet}x \Leftrightarrow \exists s, s' \ \cdot \ s \xrightarrow{a} s' \in {}^{\bullet}x$ *and* $a \in x^{\bullet} \Leftrightarrow \exists s, s' \ \cdot \ s \xrightarrow{a} s' \in x^{\bullet}$.

Proof: By Obs. 29 we have to verify that if $S_0(x)$ and $S_1(x)$ satisfy the conditions *1* and *2* of the proposition, then they induce a region $x = (\sigma, \eta)$ relative to the type of C/E nets for which $a \in {}^{\bullet}x \Leftrightarrow \exists s, s' \ \cdot \ s \xrightarrow{a} s' \in {}^{\bullet}x$ and $a \in x^{\bullet} \Leftrightarrow \exists s, s' \ \cdot \ s \xrightarrow{a} s' \in x^{\bullet}$. Suppose $S_0(x)$ and $S_1(x)$ satisfy these conditions and let $\sigma : S \to \mathbf{2}$ and $\tilde{\eta} : T \to \mathbf{2}\times\mathbf{2}$ be given by $\sigma(s) = 1 \Leftrightarrow x \in S_0(x)$ and $\tilde{\eta}(t) = (\sigma(s), \sigma(s'))$ if $t = s \xrightarrow{a} s' \notin S_1(x)$ and $\tilde{\eta}(t) = (0,0)$ if $t \in S_1(x)$. We first show that $\tilde{\eta}(t)$ depends only upon the label of t i.e. it induces a mapping $\eta : A \to \mathbf{2}\times\mathbf{2}$ such that $\tilde{\eta}(s \xrightarrow{a} s') = \eta(a)$. We proceed by case analysis

1. If $\exists s, s' \quad s \xrightarrow{a} s' \in {}^{\bullet}x \cap x^{\bullet}$, then by 2.$(a)$ and 2.(b), $u \xrightarrow{a} u' \in {}^{\bullet}x \cap x^{\bullet}$ for every a-labelled transition and then for every a-labelled transition $u \xrightarrow{a} u'$ one has $u \xrightarrow{a} u' \notin S_1(x)$, $u \in S_0(x)$ and $u' \in S_0(x)$ and then $\tilde{\eta}(u \xrightarrow{a} u') = (1,1)$.
2. Similarly if $\exists s, s' \quad s \xrightarrow{a} s' \in {}^{\bullet}x \setminus x^{\bullet}$, then for every a-labelled transition $u \xrightarrow{a} u'$ one has $\tilde{\eta}(u \xrightarrow{a} u') = (0,1)$.
3. Similarly if $\exists s, s' \quad s \xrightarrow{a} s' \in x^{\bullet} \setminus {}^{\bullet}x$, then for every a-labelled transition $u \xrightarrow{a} u'$ one has $\tilde{\eta}(u \xrightarrow{a} u') = (1,0)$.
4. if every a-labelled transition do not belong to ${}^{\bullet}x \cup x^{\bullet}$. Then for any such $s \xrightarrow{a} s'$ one has either $s \xrightarrow{a} s' \in S_1(x)$ or ($s \xrightarrow{a} s' \notin S_1(x)$ and $s \notin S_0$ and $s' \notin S_0$); then in both case one has $\tilde{\eta}(s \xrightarrow{a} s') = (0,0)$.

Now the resulting pair (σ, η) is a region for C/E nets because either $s \xrightarrow{a} s' \in S_1$ and then $\eta(a) = (0,0)$ and $\sigma(s) = \sigma(s') = 1$ or $s \xrightarrow{a} s' \notin S_1$ and $\eta(a) = (\sigma(s), \sigma(s')$, and $\tau_{C/E}$ consists exactly of the transition $1 \xrightarrow{(0,0)} 1$ together with all transitions of the form $n \xrightarrow{(n,m)} m$ for $n, m \in \mathbf{2}$. Now this region $x = (\sigma, \eta)$ satisfies $a \in {}^{\bullet}x \Leftrightarrow \exists s, s' \quad s \xrightarrow{a} s' \in {}^{\bullet}x$ and $a \in x^{\bullet} \Leftrightarrow \exists s, s' \quad s \xrightarrow{a} s' \in x^{\bullet}$. ∎

Thus a condition of an asynchronous transition system is a set of states (of dimension at most one) of its splitting.

Proposition 37 *The restriction to the set of states of dimension at most one of a region X of the splitting* $\mathbf{Split}(\mathcal{A})$ *of an asynchronous transition system* $\mathcal{A} = (S, A, \|, T, s_0)$ *such that* ${}^\bullet X \subseteq end(A)$ *and* $X^\bullet \subseteq beg(A)$ *is a condition of* $\mathcal{A}$*. Conversely any condition x of $\mathcal{A}$ is the restriction to the set of states of dimension at most one of a unique region* $X = \mathbf{Split}(x)$ *of* $\mathbf{Split}(\mathcal{A})$ *such that* $x \in {}^\bullet a \Leftrightarrow X \in {}^\bullet beg(a)$ *and* $x \in a^\bullet \Leftrightarrow X \in beg(a)^\bullet$.

Proof: Let X a region of $\mathbf{Split}(\mathcal{A})$ such that $x \in {}^\bullet a \Leftrightarrow X \in {}^\bullet beg(a)$ and $x \in a^\bullet \Leftrightarrow X \in beg(a)^\bullet$, we prove that the pair $S_0(x) = \{s \in S| <s, 0> \in X\}$ and $S_1(x) = \{s \xrightarrow{a} s' \in T| <s, a> \in X\}$ satisfies the conditions of Prop. 36 and therefore induces a condition of $\mathcal{A}$. Let $s \xrightarrow{a} s' \in S_1(x)$, since $x \in {}^\bullet a \Leftrightarrow X \in {}^\bullet beg(a)$ and $x \in a^\bullet \Leftrightarrow X \in beg(a)^\bullet$ we deduce $beg(a) \notin {}^\bullet X$ and thus $s \in S_0(x)$ and $end(a) \notin X^\bullet$ and thus $s' \in S_0(x)$. If there exists an a-labelled transition $s \xrightarrow{a} s' \notin S_1(x)$ such that $s' \in S_0(x)$ then $end(a) \in X^\bullet$ and the same condition holds for any other a-labelled transition: $u \xrightarrow{a} u' \Rightarrow (u \xrightarrow{a} u' \notin S_1(x)$ and $u' \in S_0(x))$. Condition 2.(b) is verified similarly. Now suppose $a \| b$ with $a, b \in {}^\bullet x \cup x^\bullet$ where these sets are defined as in Prop. 36 then let s be a state in which these actions form a diamond: $\exists s', s'', s'''$ $\;s \xrightarrow{a} s',\;\; s'' \xrightarrow{a} s''',\;\; s \xrightarrow{b} s'',$ and $s' \xrightarrow{b} s'''$. Then necessarily the 0-dimensional states $<s, 0>$, $<s', 0>$, $<s'', 0>$ and $<s''', 0>$ are in X and the 1-dimensional states $<s, a>$, $<s, b>$, $<s', b>$, and $<s'', a>$ are not in X this contradicts that X a region of $\mathbf{Split}(\mathcal{A})$ because of the 2-dimensional state $<s, a+b>$. For the converse direction let x be a condition of $\mathcal{A}$. We notice that a set X is a region of $\mathbf{Split}(\mathcal{A})$ such that $x \in {}^\bullet a \Leftrightarrow X \in {}^\bullet beg(a)$ and $x \in a^\bullet \Leftrightarrow X \in beg(a)^\bullet$, and whose flow relations are compatible with those of x in the sense that $x \in {}^\bullet a \Leftrightarrow X \in {}^\bullet beg(a)$ and $x \in a^\bullet \Leftrightarrow X \in beg(a)^\bullet$ if and only if the following conditions are satisfied for every transition $s \xrightarrow{a} s'$ of $\mathcal{A}$ and step α such that $a \notin \alpha$ and $\alpha + a$ is enabled in state s.

1. $x \in {}^\bullet a \cap a^\bullet \Rightarrow (<s, \alpha> \in X \;\wedge\; <s, \alpha + a> \notin X \;\wedge\; <s', \alpha> \in X)$
2. $x \in {}^\bullet a \setminus a^\bullet \Rightarrow (<s, \alpha> \in X \;\wedge\; <s, \alpha + a> \notin X \;\wedge\; <s', \alpha> \notin X)$
3. $x \in a^\bullet \setminus {}^\bullet a \Rightarrow (<s, \alpha> \notin X \;\wedge\; <s, \alpha + a> \notin X \;\wedge\; <s', \alpha> \in X)$

4. $x \notin {}^\bullet a \cup a^\bullet \Rightarrow (<s,\alpha>\in X \;\Leftrightarrow <s,\alpha+a>\in X \;\Leftrightarrow <s',\alpha>\in X)$

We show by induction on the size of α that there exists a unique such X whose restriction to the set of states of dimension at most 1 is the condition x. This latter condition characterizes the predicate $<s,\alpha>\in X$ for $|\alpha| \leq 1$ and the above conditions are indeed satisfied (for $\alpha = 0$). Suppose now that X is defined on states of dimensions $\leq n+1$ and the previous conditions hold for $|\alpha| \leq n$. Let α of size $n+1$ and $s \xrightarrow{a} s'$ a transition of $\mathcal{A}$ with $a \notin \alpha$. Then clearly the predicate $<s,\alpha+a>\in X$ should be defined as $<s,\alpha+a>\in X \Leftrightarrow$ $(x \notin {}^\bullet a \cup a^\bullet$ and $<s,\alpha>\in X)$ so that the conditions 1 to 4 above are satisfied. In order this definition to be consistent one has to check that if $\alpha + a = \beta + b$ with $a \neq b$ (which happens when $\alpha = \gamma + b$ and $\beta = \gamma + a$ and then necessarily $a \| b$) then $<s,\alpha+a>\in X$ if and only if $<s,\beta+b>\in X$. Now $<s,\alpha+a>\in X$ $\Leftrightarrow$ [by definition] $x \notin {}^\bullet a \cup a^\bullet$ and $<s,\alpha>\in X$ $\Leftrightarrow$ [by induction hypothesis] $x \notin {}^\bullet a \cup a^\bullet$ and $(x \notin {}^\bullet b \cup b^\bullet$ and $<s,\gamma>\in X)$ $\Leftrightarrow$ [by symmetry] $<s,\beta+b>\in X$.
∎

On the other direction we can associate each region X of $\mathbf{Split}(\mathcal{A})$ with a condition $x = \mathbf{Shrink}(X)$ of $\mathcal{A}$ which coincides with X on zero-dimensional states.

Proposition 38 *For every region X of $\mathbf{Split}(\mathcal{A})$ the pair $S_0(x) = \{s \in S|\ <s,0> \in X\}$ and $S_1(x) = \{s \xrightarrow{a} s' |\ <s,0> \in X \;\wedge\; <s,a> \in X \;\wedge\; <s',0> \in X\}$ induces a condition $x = \mathbf{Shrink}(X)$ of the asynchronous transition system $\mathcal{A}$. The associated region $\overline{X} = \mathbf{Split}(x)$ coincides with X on zero-dimensional states and satisfies ${}^\bullet\overline{X} \subseteq end(A)$, $\overline{X}^\bullet \subseteq beg(A)$ and*

$$\begin{array}{lcl} beg(a) \in \overline{X}^\bullet & \Leftrightarrow & beg(a) \in X^\bullet \;\; or \;\; end(a) \in X^\bullet \\ end(a) \in {}^\bullet\overline{X} & \Leftrightarrow & beg(a) \in {}^\bullet X \;\; or \;\; end(a) \in {}^\bullet X \end{array}$$

Note: the above disjunctions are exclusive in both cases, because there is no region of $\mathbf{Split}(\mathcal{A})$ which is an input region of both $beg(a)$ and $end(a)$ for some event a (otherwise the sequence of transitions $beg(a); end(a)$ would never be firable) for the same reason no region of $\mathbf{Split}(\mathcal{A})$ is an output region of both $beg(a)$ and $end(a)$.
Proof: Let $S_0(x) = \{s \in S|\ <s,0> \in X\}$ and $S_1(x) = \{s \xrightarrow{a} s' |\ <s,0> \in X \;\wedge\; <s,a> \in X \;\wedge\; <s',0> \in X\}$ for X some region of $\mathbf{Split}(\mathcal{A})$. We

prove that these sets satisfy the conditions of Prop. 36 and therefore induce a condition x of $\mathcal{A}$.

1. Let $s \xrightarrow{a} s' \in S_1(x)$, then by definition of $S_1(x)$ and $S_0(x)$ one deduces $s \in S_0$ and $s' \in S_0$.

2. let $s \xrightarrow{a} s' \in {}^\bullet x$ when this transition does not belong to $S_1(x)$ and $s' \in S_0(x)$, i.e. $s \xrightarrow{a} s' \in {}^\bullet x$ if and only if $s' \in S_0(x) \wedge (s \notin S_0(x) \vee <s,a> \notin X)$. Similarly if we let $s \xrightarrow{a} s' \in x^\bullet$ when this transition does not belong to $S_1(x)$ and $s \in S_0(x)$, then $s \xrightarrow{a} s' \in x^\bullet \Leftrightarrow s \in S_0(x) \wedge (s' \notin S_0(x) \vee <s,a> \notin X)$.

   (a) Suppose $s \xrightarrow{a} s' \in {}^\bullet x$ and let $u \xrightarrow{a} u'$ be some other a-labelled transition. Three cases are possible:

      i. $<s,0> \in X \quad \wedge \quad <s,a> \in X \quad \wedge \quad <s',0> \notin X$.
      Then $beg(a) \notin {}^\bullet X \cup X^\bullet$ and $end(a) \in X^\bullet$.

      ii. $<s,0> \in X \quad \wedge \quad <s,a> \notin X \quad \wedge \quad <s',0> \in X$.
      Then $beg(a) \in X^\bullet$ and $end(a) \in {}^\bullet X$.

      iii. $<s,0> \in X \quad \wedge \quad <s,a> \notin X \quad \wedge \quad <s',0> \notin X$.
      Then $beg(a) \in X^\bullet$ and $end(a) \notin {}^\bullet X \cup X^\bullet$.

   and in each of these cases $<u,0> \in X \Leftrightarrow <s,0> \in X$, $<u,a> \in X \Leftrightarrow <s,a> \in X$, and $<u',0> \in X \Leftrightarrow <s',0> \in X$ from which it follows that $u \xrightarrow{a} u' \in {}^\bullet x$.

   (b) Similarly $(s \xrightarrow{a} s' \in x^\bullet \;\wedge\; u \xrightarrow{a} u') \Rightarrow u \xrightarrow{a} u' \in x^\bullet$.

3. Let a and b be independent events and s a state in which they are both enabled. If $s \xrightarrow{a} s' \in {}^\bullet x$ then one of the three situations enumerated in item 2.(a) above should occur. But in all three cases one has $<s,0> \in X \Leftrightarrow <s,b> \in X \Leftrightarrow <s'',0> \in X$, $<s,a> \in X \Leftrightarrow <s,a+b> \in X \Leftrightarrow <s'',a> \in X$, and $<s',0> \in X \Leftrightarrow <s',b> \in X \Leftrightarrow <s''',0> \in X$ where s'' and s''' are the states such that $s \xrightarrow{b} s''$ and $s \xrightarrow{a+b} s'''$; therefore $s \xrightarrow{b} s' \notin ({}^\bullet x \cup x^\bullet)$. Symmetrically if $s \xrightarrow{a} s' \in x^\bullet$ one deduces $s \xrightarrow{b} s' \notin ({}^\bullet x \cup x^\bullet)$.

The result follows from the fact that the induced condition x satisfies

$$\begin{array}{lcl} a \in {}^\bullet x & \Leftrightarrow & beg(a) \in {}^\bullet X \text{ or } end(a) \in {}^\bullet X \\ a \in x^\bullet & \Leftrightarrow & beg(a) \in X^\bullet \text{ or } end(a) \in X^\bullet \end{array}$$

$\blacksquare$

Conditions of an asynchronous transition system may be identified with the regions X of its splitting such that ${}^\bullet X \subseteq end(A)$ and $X^\bullet \subseteq beg(A)$. Moreover, for every event a the set $exec(a) = \{< s, \alpha > \in \overline{S} \mid \ \ a \in \alpha\}$ is a region of $\mathbf{Split}(\mathcal{A})$ such that ${}^\bullet exec(a) = \{beg(a)\}$ and $exec(a)^\bullet = \{end(a)\}$. We let $exec(A) = \{exec(a) \mid a \in A\}$ denote the set of these regions. In view of Prop. 38, we can more generally define the shrinking $\mathbf{Shrink}(\mathcal{N})$ of a pure 1-safe Petri net system $\mathcal{N}$ on the splitted alphabet as follows.

Definition 39 *The shrinking of a pure 1-safe Petri net system* $\mathcal{N} = (P', A + A, F', M_0')$ *on the splitted alphabet –for which we can assume without loss of generality that* $P' = P + exec(A)$ *(i.e.* $exec(A) \subseteq P'$*)– is the Petri net system* $\mathbf{Shrink}(\mathcal{N}) = (P, A, F, M_0)$ *where* M_0 *is the restriction of* M_0' *and flow relation is given as follows*

$$\begin{array}{llcll} a \in {}^\bullet X & \textit{in } \mathbf{Shrink}(\mathcal{N}) & \Leftrightarrow & beg(a) \in {}^\bullet X \ \textit{ or } \ end(a) \in {}^\bullet X & \textit{in } \mathcal{N} \\ a \in X^\bullet & \textit{in } \mathbf{Shrink}(\mathcal{N}) & \Leftrightarrow & beg(a) \in X^\bullet \ \textit{ or } \ end(a) \in X^\bullet & \textit{in } \mathcal{N} \end{array}$$

In order not to confuse the places in $\mathbf{Shrink}(\mathcal{N})$ *with those in* $\mathcal{N}$ *we shall denote* $x = \mathbf{Shrink}(X)$ *the place of* $\mathbf{Shrink}(\mathcal{N})$ *associated with the place* X *in* $\mathcal{N}$*.*

This transformation can be decomposed into two stages. The first stage, called *normalisation* corresponds to the composition $\mathbf{Norm} = \mathbf{Split} \circ \mathbf{Shrink}$ which transforms the net system $\mathcal{N}$ so that every place X (except those in $exec(A)$) satisfies the conditions ${}^\bullet X \subseteq end(A)$ and $X^\bullet \subseteq beg(A)$; a net system is said to be *normalized* if it satisfies this condition. The second stage, called *abstraction*, consists in abstracting each subnet $\boxed{beg(a)} \longrightarrow \bigcirc \longrightarrow \boxed{end(a)}$ into an atomic action a. This operation defined on normalized net systems is the converse of the splitting operation; more precisely the shrinking operation is a left inverse of the splitting operation ($\mathbf{Shrink} \circ \mathbf{Split}$ is the identity).

Observation 40 *The net system synthesized from the set of regions associated with conditions of* $\mathcal{A}$ *together with the set* $exec(A)$ *is the splitting of the net system synthesized from* $\mathcal{A}$*; therefore one has an embedding of* $\mathbf{Split}(\mathcal{A}^\circ)$ *into* $\mathbf{Split}(\mathcal{A})^*$ *i.e.* $\mathbf{Split}(\mathcal{A}^\circ) \leq \mathbf{Split}(\mathcal{A})^*$*.*

The next proposition and its corollary show that these two net systems have isomorphic marking graphs.

Proposition 41 *If the splitting* $\mathbf{Split}(\mathcal{A})$ *of an asynchronous transition system* $\mathcal{A} = (S, A, \|, T, s_0)$ *is separated (by regions) then the regions of* $\mathbf{Split}(\mathcal{A})$ *associated with conditions of* $\mathcal{A}$ *together with the set of regions* $exec(A)$ *constitute an admissible set of regions for* $\mathbf{Split}(\mathcal{A})$. *More precisely if* $\mathbf{Split}(\mathcal{A}) \cong \mathcal{N}^*$ *for some contact-free elementary net system then* $\mathbf{Split}(\mathcal{A}) \cong [\mathbf{Norm}(\mathcal{N})]^*$ *and the elementary net system* $\mathbf{Norm}(\mathcal{N})$ *is also contact-free.*

Proof: Let $<s, \alpha>$ and $<s', \beta>$ states of $\mathbf{Split}(\mathcal{A})$. If $\alpha \neq \beta$ then the region $exec(a)$ for a any event in the symmetric difference $\alpha \triangle \beta = (\alpha \setminus \beta) \cup (\beta \setminus \alpha)$ distinguishes these states: $<s, \alpha> \in exec(a) \Leftrightarrow a \in \alpha \Leftrightarrow a \notin \beta \Leftrightarrow <s', \beta> \notin exec(a)$. Suppose now that $\alpha = \beta$. Let $\mathbf{Dom}(\alpha)$, the domain of step α, denote the set of events of the form $beg(a)$ or $end(a)$ for some $a \in \alpha$. First we observe that since states $<s, \alpha>$ and $<s', \alpha>$ are both targets of transitions labelled by $beg(a)$ and origins of transitions labelled by $end(a)$ for any $a \in \alpha$, any region X that distinguishes these states satisfies $({}^\bullet X \cup X^\bullet) \cap \mathbf{Dom}(\alpha) = \emptyset$. Therefore a region X distinguishes states $<s, \alpha>$ and $<s', \alpha>$ if and only if it distinguishes $<s, 0>$ and $<s', 0>$. Now, the condition $\overline{X} = \mathbf{Norm}(X)$ coincides with X on zero-dimensional states therefore one has the following sequence of equivalent statements:

1. X distinguishes states $<s, \alpha>$ and $<s', \alpha>$,
2. X distinguishes states $<s, 0>$ and $<s', 0>$,
3. $\overline{X}$ distinguishes states $<s, 0>$ and $<s', 0>$,
4. $\overline{X}$ distinguishes states $<s, \alpha>$ and $<s', \alpha>$.

Observe that event $end(a)$ is enabled in state $<s, \alpha>$ if and only if $a \in \alpha$ i.e. $<s, \alpha> \in exec(a)$; thus regions in $exec(A)$ are sufficient to inhibit events of the form $end(a)$ in states in which these events are not enabled. Therefore we consider now an event of the form $beg(a)$ and a state $<s, \alpha>$ in which this event is not enabled. We shall prove that there exists a region X, extension of some place of $\mathcal{N}$, such that the region $\overline{X} = \mathbf{Norm}(X)$ is an input region of $beg(a)$ which does not contain state $<s, \alpha>$; from which it follows that $\mathbf{Split}(\mathcal{A}) \cong [\mathbf{Norm}(\mathcal{N})]^*$ and that the net system $\mathbf{Norm}(\mathcal{N})$ is contact-free. First we recall that $X \in {}^\bullet beg(a)$ implies $\overline{X} \in {}^\bullet beg(a)$. Event $beg(a)$ is not enabled in state $<s,\alpha>$ either because event a is not enabled in state s (in asynchronous transition system $\mathcal{A}$) or because a belongs to α.

1. *event a does not belong to α.*
   Then event a is not enabled in state s in the asynchronous transition system $\mathcal{A}$, and event $beg(a)$ is not enabled in state $<s,0>$ either. Let X be an extension of some place in $\mathcal{N}$ which is an input region of $beg(a)$ that does not contain state $<s,0>$. Since X and $\overline{X}$ coincide on zero-dimensional states, $<s,0> \notin \overline{X}$; and since $beg(A) \cap {}^\bullet\overline{X} = \emptyset$, $<s,\alpha> \notin \overline{X}$.

2. *event a belongs to α.*
   Then we have the transition $<s,0> \stackrel{beg(a)}{\longrightarrow} <s,a>$, and event $beg(a)$ is not enabled is state $<s,a>$ and there exists a sequence of transitions from $<s,a>$ to $<s,\alpha>$ labelled in $beg(A)$. Let X be an extension of some place in $\mathcal{N}$ which is an input region of $beg(a)$ that does not contain state $<s,a>$. Since X and $\overline{X}$ coincide on zero-dimensional states, $<s,0> \in \overline{X}$; since $\overline{X} \in {}^\bullet beg(a)$, $<s,a> \notin \overline{X}$; and since $beg(A) \cap {}^\bullet\overline{X} = \emptyset$, $<s,\alpha> \notin \overline{X}$.

∎

Corollary 42 *If $\mathcal{A}$ is an asynchronous transition system with* $\mathbf{Split}(\mathcal{A}) \cong \mathcal{N}^*$ *for some pure 1-safe Petri net system (on the splitted alphabet), then* $\mathcal{A} \cong (\mathbf{Shrink}\ (\mathcal{N}))^\circ$.

Proof: Let $\mathbf{Split}(\mathcal{A}) \cong \mathcal{N}^*$, then $\mathbf{Split}(\mathcal{A}) \cong [\mathbf{Norm}(\mathcal{N})]^* = [\mathbf{Split}(\mathbf{Shrink}(\mathcal{N}))]^*$ $\cong \mathbf{Split}([\mathbf{Shrink}(\mathcal{N})]^\circ)$, hence $\mathcal{A} \cong [\mathbf{Shrink}(\mathcal{N})]^\circ$. ∎

By the above corollary and Cor. 16 we deduce the following.

Theorem 43 *An asynchronous transition system is the behaviour of a safe Petri net system if and only if its splitting is the marking graph of an elementary net system.*

We can even be more precise, by Cor. 16 if an asynchronous transition system $\mathcal{A}$ is the behaviour of a safe Petri net system $\mathcal{N}$ then $\mathbf{Split}(\mathcal{A})$ is the marking graph of $\mathbf{Split}(\mathcal{N})$ and by (the proof of) the above proposition if $\mathbf{Split}(\mathcal{A})$ is the marking graph of an elementary net system $\mathcal{N}$ then $\mathcal{A}$ is the behaviour of $\mathbf{Shrink}(\mathcal{N})$.

4.2 Synthesis of Petri Nets from Step Transition Systems

In this section we show that Mukund' synthesis of Petri net systems from step transition systems reduces to the synthesis of pure Petri net systems from ordinary automata. We shall follow the same path as for the 1-safe case studied in the previous section. In fact the main results of the previous section can be recovered as restrictions of the corresponding results of the present section to the class of 1-safe net systems and the class of asynchronous transition systems.

Proposition 44 *The generalized conditions of a step transition system $\mathcal{A}$ coincide with the generalized regions of* $\mathbf{Split}(\mathcal{A})$ *such that* ${}^\bullet X \subseteq end(A)$ *and* $X^\bullet \subseteq beg(A)$ [1]*; more precisely for each generalized condition $x = (\sigma, \eta)$ of $\mathcal{A}$ there exists a unique generalized region $X = \mathbf{Split}(x) = (\tilde{\sigma}, \tilde{\eta})$ which coincides with x on zero-dimensional states (i.e. $\tilde{\sigma}(< s, 0 >) = \sigma(s)$) and such that $\tilde{\eta}(beg(a)) = -x^\bullet(a)$, and $\tilde{\eta}(end(a)) = {}^\bullet x(a)$.*

Proof: Let $x = (\sigma, \eta)$ be a generalized condition of the step transition system $\mathcal{A}$; i.e. x is a pair of mappings $\sigma : S \to I\!N$ and $\eta : A \to I\!N \times I\!N$ such that if we let $\eta(a) = (x^\bullet(a), {}^\bullet x(a))$ then for every step α enabled in state s inducing hypertransition $s \stackrel{\alpha}{\to} s'$ one has $\sum_{a \in A} \alpha(a) \cdot x^\bullet(a) \leq \sigma(s)$ and $\sigma(s') = \sigma(s) + \sum_{a \in A} \alpha(a) \cdot [{}^\bullet x(a) - x^\bullet(a)]$. The mapping η induces a mapping $\tilde{\eta} : A + A \to Z\!\!\!Z$ given by $\tilde{\eta}(beg(a)) = -x^\bullet(a)$ and $\tilde{\eta}(end(a)) = {}^\bullet x(a)$. This is consistent with the definition of the splitting of the Petri net system $\mathcal{A}^\circ$: each place x of $\mathcal{A}^\circ$ is associated with the place of $\mathbf{Split}(\mathcal{A}^\circ)$ which is an input place of $beg(a)$ with weight $x^\bullet(a)$ and an output place of $end(a)$ with weight ${}^\bullet x(a)$. We show there exists a unique $\tilde{\sigma} : \overline{S} \to I\!N$ which coincides with σ on zero-dimensional states and such that $X = (\tilde{\sigma}, \tilde{\eta})$ is a generalized region of $\mathbf{Split}(\mathcal{A}^\circ)$. If such a map exists we necessarily have $\tilde{\sigma}(< s, \alpha >) = \sigma(s) - \sum_{a \in A} \alpha(a) \cdot x^\bullet(a)$ for every state s and step α enabled at s; and the pair $X = (\tilde{\sigma}, \tilde{\eta})$ is indeed a generalized region because $< s, \alpha > \stackrel{a^+}{\to} < s, \alpha + a >$ entails $\tilde{\sigma}(< s, \alpha + a >) = \sigma(s) - [\sum_{b \in A} \alpha(b) \cdot x^\bullet(b) + x^\bullet(a)] = [\sigma(s) - \sum_{b \in A} \alpha(b) \cdot x^\bullet(b)] - x^\bullet(a)) = \tilde{\sigma}(< s, \alpha >) + \tilde{\eta}(beg(a))$ and $< s, \alpha + a > \stackrel{a^-}{\to} < s', \alpha >$ entails $s \stackrel{a}{\to} s'$ in $\mathcal{A}$ hence $\sigma(s') = \sigma(s) - x^\bullet(a) + {}^\bullet x(a)$

[1] where the expressions ${}^\bullet X$ and $X^\bullet$ for a generalized condition $X = (\sigma, \eta)$ stand for the following sets: ${}^\bullet X = \{a \in A \mid \eta(a) > 0\}$ and $X^\bullet = \{a \in A \mid \eta(a) < 0\}$.

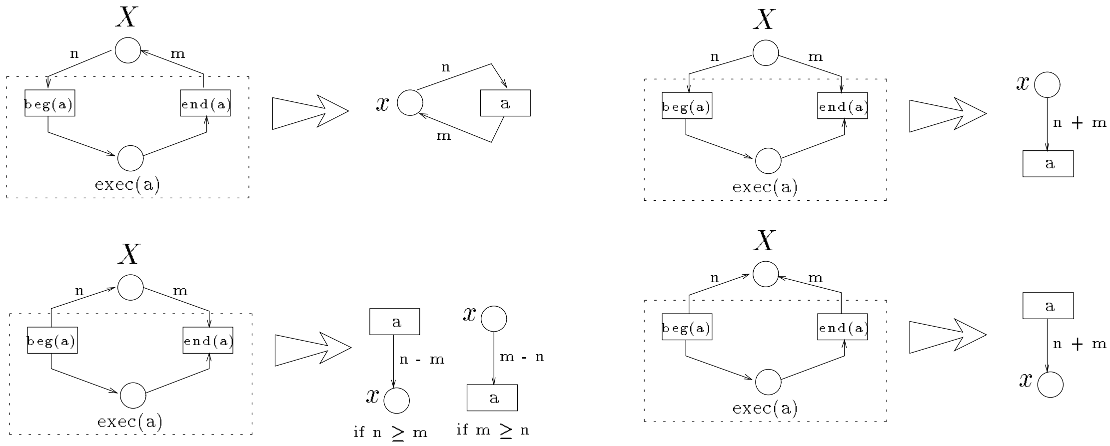


Figure 9: the shrinking $x = \mathbf{Shrink}(X)$ of a generalized region X of $\mathbf{Split}(\mathcal{A})$

and thus $\tilde{\sigma}(< s', \alpha >) = \sigma(s') - \sum_{b \in A} \alpha(b) \cdot x^\bullet(b) = \sigma(s) - x^\bullet(a) + {}^\bullet x(a) - \sum_{b \in A} \alpha(b) \cdot x^\bullet(b) = [\sigma(s) - (\sum_{b \in A} \alpha(b) \cdot x^\bullet(b) + x^\bullet(a))] + {}^\bullet x(a) = \tilde{\sigma}(< s, \alpha + a >) + \tilde{\eta}(end(a))$. By construction this generalized region $X = (\tilde{\sigma}, \tilde{\eta})$ is such that ${}^\bullet X \subseteq end(A)$ and $X^\bullet \subseteq beg(A)$. Conversely if X is such a generalized region, we let $\sigma : S \to I\!N$ be the restriction of $\tilde{\sigma}$ to the set of zero-dimensional states and $\eta : A \to I\!N \times I\!N$ be given by $\eta(a) = (-\tilde{\eta}(beg(a)), \tilde{\eta}(end(a)))$; then the pair $x = (\sigma, \eta)$ is a generalized condition of step transition system $\mathcal{A}$. Indeed if α is a step enabled at state s and $u = a_1 \cdot a_n \in A^*$ is a word with Parikh image α, then we deduce the sequence of transitions

$$< s, 0 > \stackrel{a_1^+}{\to} < s, a_1 > \stackrel{a_2^+}{\to} < s, a_1 + a_2 > \cdots \stackrel{a_n^+}{\to} < s, \alpha > \stackrel{a_n^-}{\to} < s_1, \alpha - a_n > \cdots \stackrel{a_1^-}{\to} < s_n, 0 >$$

from which it follows that $\sigma(s) \geq \sum_{a \in A} \alpha(a) \cdot x^\bullet(a)$ and $\sigma(s') = \sigma(s) + \sum_{a \in A} \alpha(a) \cdot [{}^\bullet x(a) - x^\bullet(a)]$. ∎

On the other direction we can associate each generalized region X of $\mathbf{Split}(\mathcal{A})$ with a generalized condition $x = \mathbf{Shrink}(X)$ of $\mathcal{A}$ which coincides with X on zero-dimensional states (see Fig. 9).

Proposition 45 *If $X = (\sigma, \eta)$ is a generalized region of* $\mathbf{Split}(\mathcal{A})$*, a generalized condition $x = \mathbf{Shrink}(X)$ of the step transition system $\mathcal{A}$ is given as*

Table 1: $x = \mathbf{Shrink}(X)$

	$(^\bullet beg(a)(X), beg(a)^\bullet(X))$	$(^\bullet end(a)(X), end(a)^\bullet(X))$	$(^\bullet a(x), a^\bullet(x))$
I	(n,0)	(0,m)	(n,m)
II	(n,0)	(m,0)	(n+m,0)
III	(0,n)	(0,m)	(0,n+m)
IV	(0,n)	(n+k,0)	(k,0)
V	(0,n+k)	(n,0)	(0,k)

the pair $x = (\hat{\sigma}, \hat{\eta})$ where $\hat{\sigma}$ is the restriction of σ to the set of zero dimensional states and $\hat{\eta}$ is given by $\hat{\eta}(a) = (^\bullet a(x), a^\bullet(x))$ with $a^\bullet(x) = max\{0, p\}$ and $^\bullet a(x) = max\{0, -p\}$ where $p = \eta(beg(a)) + \eta(end(a))$ if $beg(a) \notin X^\bullet$ or $end(a) \notin {}^\bullet X$ and $^\bullet a(x) = -\eta(beg(a))$ and $a^\bullet(x) = \eta(end(a))$ if $beg(a) \in X^\bullet$ and $end(a) \in {}^\bullet X$.

Proof : In order to ease notations we let $^\bullet e(X) = \max\{0, -\eta(e)\}$ and $e^\bullet(X) = \max\{0, \eta(e)\}$ for any event e in the splitted alphabet $beg(A) \cup end(A)$ and any generalized region $X = (\sigma, \eta)$ of the automaton $\mathbf{Split}(\mathcal{A})$. $\eta(e)$ is then encoded by the pair $e(X) = (^\bullet e(X), e^\bullet(X))$, it corresponds to the encoding of $\mathbb{Z}$ into $\mathbb{N} \times \mathbb{N}$ already mentionned that takes a positive integer n to the pair $(0, n)$ and a negative integer $-n$ to the pair $(n, 0)$. Table 1 gives the value of $a(x) = (^\bullet a(x), a^\bullet(x))$ for $x = \mathbf{Shrink}(X)$ in term of $beg(a)(X)$ and $end(a)(X)$ (see Fig. 45). Moreover we let $\delta_e(X) = e^\bullet(X) - {}^\bullet e(X)$ for e an event of the splitted alphabet $beg(A) \cup end(A)$ and X a generalized region of $\mathbf{Split}(\mathcal{A})$; i.e. $\delta_a(x) = \eta(e)$ if $x = (\sigma, \eta)$. Similarly we let $\delta_a(x) = a^\bullet(x) - {}^\bullet a(x)$ for a an event of A and x defined as above.

In order to prove that $x = \mathbf{Shrink}(X)$ is a generalized condition of step transition system $\mathcal{A}$ we have first to verify that $s \xrightarrow{a} s'$ implies $\hat{\sigma}(s') = \hat{\sigma}(s) - {}^\bullet a(x) + a^\bullet(x)$ and second that for every step α enabled in state s one has $\hat{\sigma}(s) \geq {}^\bullet\alpha(x)$ where $^\bullet\alpha(x) = \sum_{a \in A} \alpha(a) \cdot {}^\bullet a(x)$. For the first point, we verify that for each cases enumerated in Table 1 one has $\delta_a(x) = \delta_{beg(a)}(X) + \delta_{end(a)}(X)$ if $x = \mathbf{Shrink}(X)$. Thus $s \xrightarrow{a} s'$ implies $\hat{\sigma}(s') - \hat{\sigma}(s) = \sigma(<s', 0>) - \sigma(<s, 0>) = (\sigma(<s', 0>) - \sigma(<s, a>)) + (\sigma(<s, a>) - \sigma(<s, 0>)) = \delta_{beg(a)}(X) + \delta_{end(a)}(X) = \delta_a(x) = a^\bullet(x) - {}^\bullet a(x)$. Concerning the second point, we observe that if $u \in (beg(A) \cup end(A))^*$ is a word labelling some sequence of transitions starting from state $<s,0>$, then $\hat{\sigma}(s) = \sigma(<s,0>) \geq \delta_\beta(X)$ for any generalized region X of $\mathbf{Split}(\mathcal{A})$ where β is the parikh image of word u and where function δ is naturally extended to multisets by $\delta_\beta(x) = \sum_e \beta(e)\cdot\delta_e(x)$. Thus one has to find a mapping that takes each step α enabled in state s to a word $\tilde{\alpha}$ labelling some sequence of transitions starting from state $<s,0>$ and whose parikh image $\overline{\alpha}$ is such that ${}^\bullet\alpha(x) = -\delta_{\overline{\alpha}}(X)$. For α of length one and by looking up Table 1 we obtain the solution given by

$$\tilde{a} = \begin{cases} beg(a) & \text{for case (I)} \\ beg(a)\cdot end(a) & \text{for cases (II) and (IV)} \\ \varepsilon & \text{for cases (III) and (V)} \end{cases}$$

We then obtain the required mapping by choosing an enumeration $\alpha = a_1 + \ldots + a_n$ for each step α and then letting $\tilde{\alpha} = \tilde{a_1}\cdot\tilde{a_2}\cdots\tilde{a_n}$. ∎

Generalized conditions of a step transition system $\mathcal{A}$ may be identified with the generalized regions X of its splitting such that ${}^\bullet X \subseteq end(A)$ and $X^\bullet \subseteq beg(A)$. Moreover, for every event a, the mapping $exec(a) : \overline{S} \to I\!N$ given by $exec(a)(<s,\alpha>) = \alpha(a)$ induces a generalized region of $\mathbf{Split}(\mathcal{A})$ (also denoted $exec(a)$) such that ${}^\bullet exec(a) = beg(a)$ and $exec(a)^\bullet = end(a)$. We let $exec(A) = \{exec(a) \mid a \in A\}$ denote this set of regions. In view of Prop. 45, we can more generally define the shrinking $\mathbf{Shrink}(\mathcal{N})$ of a pure Petri net system $\mathcal{N}$ on the splitted alphabet as follows.

Definition 46 *The shrinking of a pure Petri net system* $\mathcal{N} = (P', A{+}A, F', M_0')$ *on the splitted alphabet –for which we can assume without loss of generality that* $P' = P + exec(A)$ *(i.e.* $exec(A) \subseteq P'$*)– is the Petri net system* $\mathbf{Shrink}(\mathcal{N}) = (P, A, F, M_0)$ *where* M_0 *is the restriction of* M_0' *and flow relation* F *is given as follows: if* $a \in A$ *is such that* $beg(a) \in X^\bullet$ *and* $end(a) \in {}^\bullet X$ *then* $F(X,a) = X^\bullet(beg(a))$ *and* $F(a,X) = {}^\bullet X(end(a))$*, otherwise* $F(X,a) = max\{0,-p\}$ *and* $F(a,X) = max\{0,p\}$ *where* $p = C'(X,beg(a)) + C'(X,end(a))$ *with* C' *the connectivity matrix associated with flow relation* F'*, i.e.* $C'(X,a) = F'(a,X) - F'(X,a)$*. In order not to confuse the places in* $\mathbf{Shrink}(\mathcal{N})$ *with those in* $\mathcal{N}$ *we shall denote* $x = \mathbf{Shrink}(X)$ *the place of* $\mathbf{Shrink}(\mathcal{N})$ *associated with the place* X *in* $\mathcal{N}$*.*

This transformation can be decomposed into two stages (as illustrated in Fig. 10) the first stage, called *normalisation* corresponds to the composition

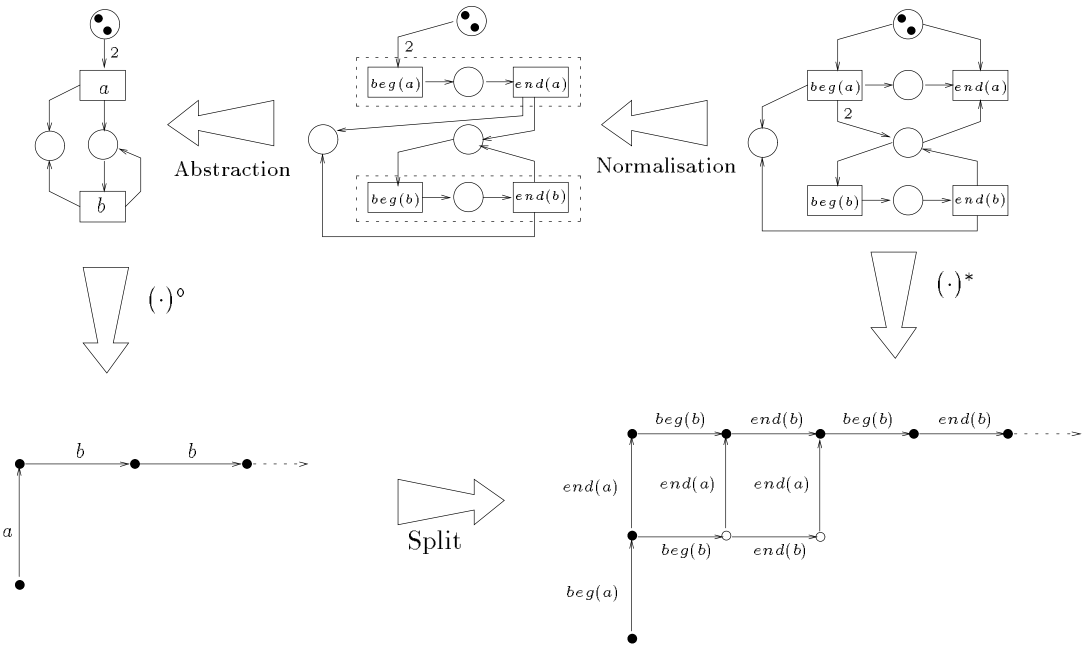


Figure 10: a net on the splitted alphabet, its shrinking and their respective states graphs

Split$\circ$**Shrink** which transforms the net system $\mathcal{N}$ so that every place X (except those in $exec(A)$) satisfies the conditions $^\bullet X \subseteq end(A)$ and $X^\bullet \subseteq beg(A)$; a net system is said to be *normalized* if it satisfies this condition. The second stage, called *abstraction*, consists in abstracting each subnet $\boxed{beg(a)}\longrightarrow\bigcirc\longrightarrow\boxed{end(a)}$ into an atomic action a. This operation defined on normalized net systems is the converse of the splitting operation; more precisely the shrinking operation is a left inverse of the splitting operation (**Shrink**$\circ$**Split** is the identity).

Observation 47 *The net system synthesized from the set of generalized regions associated with generalized conditions of $\mathcal{A}$ together with the set $exec(A)$ is the splitting of the net system synthesized from $\mathcal{A}$; therefore one has an embedding of* $\mathbf{Split}(\mathcal{A}^\circ)$ *into* $\mathbf{Split}(\mathcal{A})^*$ *i.e.* $\mathbf{Split}(\mathcal{A}^\circ) \leq \mathbf{Split}(\mathcal{A})^*$.

Table 2: $\overline{X} = \mathbf{Norm}(X)$

	$\eta(beg(a))$	$\eta(end(a))$	$\overline{\eta}(beg(a))$	$\overline{\eta}(end(a))$
I	-n	m	-n	m
II	-n	-m	-n-m	0
III	n	m	0	n+m
IV	n	-n-k	-k	0
V	n+k	-n	0	k

The next proposition shows that these two net systems have isomorphic marking graphs.

Lemma 48 *Let $X = (\sigma, \eta)$ be a generalized region of the splitting* $\mathbf{Split}(\mathcal{A})$ *of some step automaton $\mathcal{A}$, then the generalized region $\overline{X} = (\overline{\sigma}, \overline{\eta})$ defined as the normalization of X:* $\mathbf{Norm}(X) = \mathbf{Split}(\mathbf{Shrink}(X))$ *is such that (i) for every event a in A, $\overline{\eta}(beg(a)) \leq \eta(beg(a))$ and (ii) for every state $<s, \alpha>$ of* $\mathbf{Split}(\mathcal{A})$ *$\overline{\sigma}(<s, \alpha>) \leq \sigma(<s, \alpha>)$.*

Proof: From Table 1 we derive Table 2 giving the flow relations of $\overline{X} = \mathbf{Norm}(X)$ in terms of the flow relations of X. Property *(i)* is then verified by looking up Table 2. We prove property *(ii)* by induction on the size of α. We know that σ and $\overline{\sigma}$ coincide on zero-dimensional states. Suppose $\overline{\sigma}(<s, \alpha>) \leq \sigma(<s, \alpha>)$ for $|\alpha| \leq n$ and consider β of size $n+1$, then $\beta = \alpha + a$ for some $a \in A$ and α of size n and $\sigma(<s, \beta>) = \sigma(<s, \alpha>) + \eta(beg(a)) \geq$ [by inductive hypothesis] $\overline{\sigma}(<s, \alpha>) + \eta(beg(a)) \geq$ [by property *(i)*] $\overline{\sigma}(<s, \alpha>) + \overline{\eta}(beg(a)) = \overline{\sigma}(<s, \beta>)$. ∎

Proposition 49 *If $\mathcal{A}$ is a step automaton with* $\mathbf{Split}(\mathcal{A}) \cong \mathcal{N}^*$ *for some pure Petri net system (on the splitted alphabet), then $\mathcal{A} \cong$* $(\mathbf{Shrink}\ (\mathcal{N}))^\circ$.

Proof: As for Prop. 41 we show that if $\mathcal{A}$ is a step transition system with $\mathbf{Split}(\mathcal{A}) \cong \mathcal{N}^*$ for some pure Petri net system then $\mathbf{Split}(\mathcal{A}) \cong [\mathbf{Norm}(\mathcal{N})]^*$ and the result follows by the same argument as in Cor. 42. Let $<s, \alpha>$ and $<s', \beta>$ states of $\mathbf{Split}(\mathcal{A})$. If $\alpha \neq \beta$ then the generalized region $exec(a)$ for a any event such that $\alpha(a) \neq \beta(a)$ distinguishes these states: $exec(a)(<s, \alpha>) =$

$\alpha(a) \neq \beta(a) = exec(a)(<s, \beta>)$. Suppose now that $\alpha = \beta$. We observe that a generalized region $X = (\sigma, \eta)$ distinguishes states $<s, \alpha>$ and $<s', \alpha>$ if and only if it distinguishes $<s, 0>$ and $<s', 0>$ because $\sigma(<s, \alpha>) - \sigma(<s, 0>) = \sum_{a \in A} \alpha(a) \cdot {}^{\bullet}beg(a)(X) = \sigma(<s', \alpha>) - \sigma(<s', 0>)$. Now, the generalized region $\overline{X} = \mathbf{Norm}(X)$ coincides with X on zero-dimensional states therefore one has the following sequence of equivalent statements:

1. X distinguishes states $<s, \alpha>$ and $<s', \alpha>$,
2. X distinguishes states $<s, 0>$ and $<s', 0>$,
3. $\overline{X}$ distinguishes states $<s, 0>$ and $<s', 0>$,
4. $\overline{X}$ distinguishes states $<s, \alpha>$ and $<s', \alpha>$.

Observe that event $end(a)$ is enabled in state $<s, \alpha>$ if and only if $a \in \alpha$ i.e. $exec(a)(<s, \alpha>) \neq 0$; thus regions in $exec(A)$ are sufficient to inhibit events of the form $end(a)$ in states in which these events are not enabled. Therefore we consider now an event of the form $beg(a)$ and a state $<s, \alpha>$ in which this event is not enabled. Let $X = (\sigma, \eta)$ be a generalized region inhibiting event $beg(a)$ in state $<s, \alpha>$ i.e. X is such that $\sigma(<s, \alpha>) + \eta(beg(a)) < 0$ then by Lem. 48 the generalized region $\overline{X}$ inhibits also event $beg(a)$ in state $<s, \alpha>$. ∎

Therefore if $\mathbf{Split}(\mathcal{A}) \cong \mathcal{N}^*$ one can deduce $\mathbf{Split}(\mathbf{Shrink}(\mathcal{N}))^\circ \cong \mathcal{N}^*$; however as illustrated in Fig. 10 this isomorphism does not hold true for arbitrary Pure Petri net systems (on the splitted alphabet). By the above proposition and Cor. 16 we deduce the following.

Theorem 50 *A step transition system is the behaviour of a Petri net system if and only if its splitting is the marking graph of a pure Petri net system.*

The **Shrink** operation does not preserve the property of being 1-safe, however if $\mathcal{N}$ is a 1-safe Petri net system whose marking graph is isomorphic to the splitting of some asynchronous transition system $\mathcal{A}$ then there is no place in $\mathcal{N}$ which is an input place of both $beg(a)$ and $end(a)$ for some event a (otherwise the sequence of transitions $beg(a); end(a)$ would never be firable) for the same reason no place of $\mathcal{N}$ is an output place of both $beg(a)$ and $end(a)$. The operation **Shrink** then coincides for those 1-safe Petri net systems with the homonymous transformation defined in the previous section. In particular the normalisation operation then consists in anticipating the consumption of ressources (input places) when initiating the action and in delaying the production of ressources (output places) until the termination of the action.

5 Conclusion

In this paper we have shown that the synthesis problem of Petri nets can reduce to the synthesis of pure Petri nets via the splitting of the alphabet of events. Using the algorithm of [BBD95a] one can derive a polynomial algorithm for the synthesis of Petri nets from step automata; where the size of a step automaton is given by the cardinality of it set of transitions (of various dimensions) i.e. by the size of its discretization. However a step automaton can also be presented by its skeleton (restriction to the set of 1-dimensional transitions) together with the data of the steps enabled in each state, this representation can be significantly smaller. In a separate paper we shall present a variant of the algorithm of [BBD95a] which synthesizes Petri nets from step automata in time polynomial in the size of its presentation by skeleton and enabling relation. By the way we shall also present a polynomial algorithm deciding whether an automaton is isomorphic to the *sequential* state graph of a Petri net and which construct such a net when it exists.

We have used a framework allowing a uniform presentation of the constructions presented respectively by Ehrenfeucht and Rozenberg [ER90], Bernardinello and al [BDPV96], Nielsen and Winskel [NW94], and Mukund [Muk92]. In particular we have introduced a variant of step transition systems which encompasses asynchronous transition systems. This framework is essentially the one introduced in [BD95], however we have simplified the presentation by observing that for a fixed alphabet one ends up with a Galois connection (between ordered sets) thus avoiding all the machinery of category theory. However we can use the fact that the category of transition systems is fibred and co-fibred over the category of sets [NW94] (it is a category indexed by the alphabets) and similarly for Petri nets, in order to "glue" together these fibred Galois connections and thus recovering an adjunction. We have also observed that the Galois connection between automata and pure Petri nets lifts trivially to give rise to a Galois connection between step automata and Petri nets, this is due to the fact that the additional structure is compatible with the class of morphisms in each fibre (a morphism acting as the identity on the alphabet between two step automata is the same thing as a morphism acting as the identity on the alphabet between their underlying automata) and the synchronized product (the greatest lower bound of step automata is the synchronized product of their underlying automata). This suggest that we would also obtain Galois connections between Petri nets and automata whose alphabet comes equipped with some action structure related to some firing policy (sequential, step, ...) as soon as the above conditions are satisfied. These issues will be considered in a forthcomming paper.

References

[BBD95a] Badouel, E., Bernardinello, L. and Darondeau, Ph., *Polynomial algorithms for the synthesis of bounded nets*, Proccedings Caap 95, Lecture Notes in Computer Science 915 (1995) 647–679.

[BBD95b] Badouel, E., Bernardinello, L. and Darondeau, Ph., *The synthesis problem for elementary net systems is NP-complete*, Inria Research Report 2558 (1995).

[BD95] 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, volume 953 of Lecture Notes in Computer Science (1995) 24–43.

[Bed88] Bednarczyk, M. A., *Categories of Asynchronous Systems.* Ph. D. Thesis, University of Sussex (1988).

[Ber70] Berge, C., *Graphes et hypergraphes.* Dunod, Paris (1970). English translation: *Graphs and Hypergraphs*, North Holland, Amsterdam (1973).

[BDPV96] Bernardinello, L., De Michelis, G., Petruni, K., and Vigna, S., *On the Synchronic Structure of Transition Systems*, in J. Desel (Ed.), Structures in Concurrency Theory (strict), May 1995, Springer (1996) 11–31.

[DR96] Desel, J., Reisig, W., *The Synthesis Problem of Petri Nets.* Acta Informatica vol. 33 (1996) 297–315.

[ER90] 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).

[Gla91] Van Glabbeek, R., *Bisimulation Semantics for Higher Dimensional Automata.* Technical report, Stanford University (1991).

[GV87] Van Glabbeek, R., and Vaandrager, F., *Petri net models for algebraic theories of concurrency.* In Proceedings PARLE'87, vol. II: Parallel Languages, Eindhoven, volume 259 of Lecture Notes in Computer Science (1987) 224–242.

[GM85] Gondran, M., and Minoux, M., *Graphes et algorithmes.* Eyrolles, Paris (1985).

[Gou95] Goubault, E., *The Geometry of Concurrency.* Ph. D. Thesis, Ecole Normale Supérieure, Paris (1995).

[HKT96] 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.

[LRT88] Lodaya, K., Ramanujam, R., and Thiagarajan, P.S., *A Logic for Distributed Transition Systems.* J.W. de Bakker, W.-P. de Roever, and G. Rozenberg (Eds.) *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency,* School/Workshop, Noordwijkerhout, volume 354 of Lecture Notes in Computer Science (1988) 508–522.

[Muk92] Mukund, M., *Petri Nets and Step Transition Systems.* International Journal of Foundation of Computer Science, vol 3, n^o 4 (1992) 443–478.

[MN92] Mukund, M., and Nielsen, M., *CCS, locations and asynchronous transition systems.* In proceedings of 12^{th} Conference on Foundations of Software Technology and Theoretical Computer Science, New Delhi, India, volume 652 of Lecture Notes in Computer Science (1992) 328–341.

[NRT92] Nielsen, M., Rozenberg, G., and Thiagarajan, P.S., *Elementary Transition Systems.* Theoretical Computer Science, vol. 96 (1992) 3–33.

[NW94] Nielsen, M., and Winskel, G., *Models for Concurrency*, Handbook of Logic for Computer Science, vol. 3, Oxford University Press (1994) 100–200. Appeared also as Research Report DAIMI PB-463, Aarhus University, November 1993.

[Pra91] Pratt, V., *Modelling Concurrency with Geometry.* Proceedings of the 18^{th} ACM Symposium on Principles of Programming Languages, Orlando, ACM Press (1991) 311–322.

[Sch86] SCHRIJVER, A., *Theory of Linear and Integer Programming*. John Wiley (1986).

[Shi85] SHIELDS, M.W., *Concurrent machines*, Computer Journal, vol. 28 (1985) 449–465.

[Shi85b] SHIELDS, M.W., *Deterministic asynchronous automata*. In *Formal Methods in Programming*, North Holland (1985).

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399