Internal Report TCS-92-3
November, 1992

Transition System Models for Concurrency

Madhavan Mukund[1]
School of Mathematics
SPIC Science Foundation
92, G.N. Chetty Road, T. Nagar
Madras 600 017, India
E-mail: madhavan@ssf.ernet.in

Abstract

Labelled transition systems can be extended to faithfully model concurrency by permitting transitions between states to be labelled by a collection of actions, denoting a concurrent step. We can characterize a subclass of these *step transition systems*, called *PN*-transition systems, which describe the behaviour of Petri nets. This correspondence is formally described in terms of a coreflection between a category of *PN*-transition systems and a category of Petri nets.

In this paper, we show that we can define subcategories of *PN*-transition systems whose objects are *safe PN-transition systems* and *elementary PN-transition systems* such that there is a coreflection between these subcategories and subcategories of our category of Petri nets corresponding to safe nets and elementary net systems.

We also prove that our category of elementary *PN*-transition systems is equivalent to the category of (sequential) *elementary transition systems* defined by Nielsen, Rozenberg and Thiagarajan, thereby establishing that the concurrent behaviour of an elementary net system can be completely recovered from a description of its sequential behaviour.

Finally, we establish a coreflection between our category of safe *PN*-transition systems and a subcategory of *asynchronous transition systems* which has been shown by Winskel and Nielsen to be closely linked to safe nets.

[1] Part of the work reported here was done while the author was visiting the Computer Science Department, Aarhus University, Denmark. The author's stay in Denmark was supported by a grant from the Aarhus University Science Foundation.

This report is also published as DAIMI-PB-399, Computer Science Department, Aarhus University, Aarhus, Denmark.

1 Introduction

Labelled transition systems provide a simple and convenient framework for abstractly describing the behaviour of computing systems. Their main shortcoming from the point of view of describing *concurrent* systems is that they are inherently sequential in nature.

We can overcome this limitation by adding some structure to transition systems. One way of doing this is to permit transitions to be labelled by *steps*, consisting of more than one action [5, 9]. This *step transition relation* is intended to be read as describing how the system evolves from one state to another by performing (multi)sets of concurrent actions.

In [9], we have shown a close correspondence between a class of step transition systems, called *PN*-transition systems, and Petri nets [14]. The relationship is described in terms of a coreflection between a category of *PN*-transition systems, called $\mathcal{PN}ts$, and a category of Petri nets, called $\mathcal{PN}et$, where the morphisms in the two categories correspond to a notion of one system simulating another. This coreflection shows that we can regard *PN*-transition systems as a model which captures precisely the class of concurrent behaviours described by Petri nets, while abstracting away from the structural information associated with nets.

In this paper, we define subcategories of $\mathcal{PN}ts$ which correspond to some interesting classes of behaviours and relate these subclasses of *PN*-transition systems to other models of concurrent systems. In particular, we show that we can identify natural subclasses of *PN*-transition systems which correspond to two widely studied classes of nets, safe nets and elementary net systems.

The first observation we make in this paper is that the choice of using sequential or step transition systems to describe Petri nets depends on how detailed a description one wants of system behaviour. It turns out that we can characterize in a precise way the sequential transition systems correponding to Petri nets in terms of a coreflection between the full subcategory of $\mathcal{PN}ts$ whose objects are *sequential PN-transition systems* and the category $\mathcal{PN}et$ of Petri nets.

Next, we turn to the question of representing the behaviour of safe nets in terms of step transition systems. Petri nets are a very general model for describing concurrent systems. To obtain a tractable theory of their behaviour, one often looks at a restricted class of nets called *safe* nets. Safe nets are very "well-behaved" and have given rise to a rich theory. In particular, we note that there are strong connections between the theory of safe nets, trace languages [7] and event structures [11, 17].

To identify a subcategory of $\mathcal{PN}ts$ corresponding to safe nets, we further refine the concept of a *region*. Regions play a key role in establishing the coreflection between $\mathcal{PN}ts$ and $\mathcal{PN}et$. They were originally defined in the context of sequential transition systems by Ehrenfeucht and Rozenberg [3] as a transition system counterpart of the notion of a condition in an elementary net system. Using regions, they characterized the class of sequential transition systems which represent the behaviour of elementary net systems. To define *PN*-transition systems, the notion of a region is generalized in [9] to capture the transition system counterpart of a place of a Petri net. Here, we show that we can "tune" the notion of a region to identify a full subcategory $\mathcal{SPN}ts$ of *safe PN-transition systems* so that there is a coreflection between $\mathcal{SPN}ts$ the full subcategory $\mathcal{SN}et$ of $\mathcal{PN}et$ whose objects are safe nets.

We then turn our attention to elementary net systems. In [12], Nielsen, Rozenberg and Thiagarajan exploit the regions defined in [3] to establish a coreflection between a class of sequential transition systems called *elementary transition systems* and elementary net systems. Here we show how to describe a full subcategory of *elementary PN*-transition systems which is equivalent to the category of elementary transition systems defined in [12]. This equivalence provides an alternative proof of the result, established by Hoogeboom and Rozenberg [4], that for elementary net systems, *no* information about concurrency is lost by restricting one's attention to sequential transition systems.

Enriching the transition relation to include steps as labels is not the only way of introducing additional structure into transition systems to faithfully describe concurrency. Another possibility is to retain a sequential transition relation, and add a relation which explicitly specifies which underlying events in the system are independent of each other. This is the approach taken in defining *asynchronous transition systems* [1, 15].

In [19], Winskel and Nielsen establish a coreflection between a category $\mathcal{A}ts_0$ of asynchronous transition systems and a category of safe nets. From this, it would appear that safe *PN*-transition systems are closely related to asynchronous transition systems. In fact, we establish a coreflection between our category of safe *PN*-transition systems and the subcategory $\mathcal{A}ts_0$ of asynchronous transition systems defined in [19].

The reason this correspondence is a coreflection and not an equivalence is to do with the role played by the independence relation in an asynchronous transition system. It turns out that this relation also incorporates some "structural" information about the system, in addition to information about concurrency. So, in a sense, asynchronous transition systems are a more abstract model than nets but a more concrete model than safe *PN*-transition systems.

Another point concerning this correspondence between safe *PN*-transition systems and asynchronous transition systems is that the category of safe nets that we work with is slightly different from the category of safe nets that Winskel and Nielsen work with. However, it turns out that we can establish an adjunction between these two categories of safe nets. This correspondence can be strengthened when we restrict our attention to *saturated* nets, which are those nets constructed out of transition systems using regions.

The main results of this paper are summarized in Figures 1 and 2. In the diagrams, a double arrow $\Rightarrow$ represents a coreflection. The arrow indicates the direction of the left adjoint.

The first diagram describes the correspondence between subclasses of *PN*-transition systems and subclasses of nets. The vertical arrows represent inclusions. For each pair of categories connected by a vertical arrow, the lower category is a full subcategory of the category immediately above. In the bottom row, we have indicated that the category of elementary *PN*-transition systems is equivalent to both the subcategory of sequential elementary *PN*-transition systems and to the category of elementary transition systems defined in [12].

In the second diagram, we show the correspondence between safe *PN*-transition systems and asynchronous transition systems. We also show the adjunction between our category of safe nets and the category of safe nets described in [19], where the right adjoint is the inclusion functor.

The paper is organized as follows. In the next two sections we briefly review some terminology and basic results concerning the categories $\mathcal{PN}et$ (of Petri nets) and $\mathcal{PN}ts$ (of *PN*-transition systems) defined in [9]. In Section 4 we describe the coreflection between $\mathcal{PN}ts$ and $\mathcal{PN}et$. In the next section, we characterize the sequential behaviours of Petri nets. Section 6 describes the subcategory of $\mathcal{PN}ts$ corresponding to safe nets. In Section 7

$$
\begin{array}{ccccccc}
 & & \begin{array}{c}\text{Sequential}\\ PN\text{-transition}\\ \text{systems}\end{array} & \Rightarrow & \begin{array}{c}PN\text{-transition}\\ \text{systems}\end{array} & \Rightarrow & \begin{array}{c}\text{Petri}\\ \text{nets}\end{array} \\
 & & \uparrow & & \uparrow & & \uparrow \\
 & & \begin{array}{c}\text{Sequential}\\ \text{safe}\\ PN\text{-transition}\\ \text{systems}\end{array} & \Rightarrow & \begin{array}{c}\text{Safe}\\ PN\text{-transition}\\ \text{systems}\end{array} & \Rightarrow & \begin{array}{c}\text{Safe}\\ \text{nets}\end{array} \\
 & & \uparrow & & \uparrow & & \\
\begin{array}{c}\text{Elementary}\\ \text{transition}\\ \text{systems}\end{array} & \cong & \begin{array}{c}\text{Sequential}\\ \text{elementary}\\ PN\text{-transition}\\ \text{systems}\end{array} & \cong & \begin{array}{c}\text{Elementary}\\ PN\text{-transition}\\ \text{systems}\end{array} & \Rightarrow & \begin{array}{c}\text{Elementary}\\ \text{net}\\ \text{systems}\end{array}
\end{array}
$$

Figure 1: Subclasses of PN-transition systems

$$
\begin{tikzcd}
\begin{array}{c}\text{Safe}\\ PN\text{-transition}\\ \text{systems}\end{array} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \begin{array}{c}\text{Asynchronous}\\ \text{transition}\\ \text{systems}\end{array} \arrow[d, Rightarrow] \\
\begin{array}{c}\text{Safe}\\ \text{nets}\end{array} \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \begin{array}{c}\text{Safe nets}\\ \text{(Winskel and Nielsen)}\end{array} \arrow[l, hook, shift left]
\end{tikzcd}
$$

Figure 2: The connection to asynchronous transition systems

we describe a subcategory of $\mathcal{PN}ts$ which is equivalent to the category of elementary transition systems defined in [12]. We relate our category of safe *PN*-transition systems to Winskel and Nielsen's category of asynchronous transition systems in Section 8. The next section describes the correspondence between the two different categories of safe nets defined here and in [19]. We conclude with a discussion of the results presented here.

A word about notation—the definition of *PN*-transition systems uses multisets quite extensively. We describe the notation and terminology we use for multisets in the Appendix.

2 Petri nets

We begin with a brief introduction to Place/Transition nets, which are often simply called Petri nets. A more detailed discussion of this class of nets can be found in [14].

Definition 2.1 *A* Petri net *is a quadruple* $PN = (S, T, W, M_{in})$*, where:*

- S *is set of* places*,* T *is a set of* transitions *and* $S \cap T = \emptyset$*.* T *is assumed to be countable.*
- $W : (S \times T) \cup (T \times S) \to \mathbf{N}_0$ *is the* weight *function such that* $\forall t \in T.\ \exists s \in S.\ W(s,t) > 0$*.*
- $M_{in} : S \to \mathbf{N}_0$ *is the* initial marking*.*

For $t \in T$, let ${}^\bullet t = \{s \in S \mid W(s,t) > 0\}$ and $t^\bullet = \{s \in S \mid W(t,s) > 0\}$. Similarly, for $s \in S$, let ${}^\bullet s = \{t \in T \mid W(t,s) > 0\}$ and $s^\bullet = \{t \in T \mid W(s,t) > 0\}$. For $x \in S \cup T$, let ${}^\bullet x^\bullet = {}^\bullet x \cup x^\bullet$. Notice that we have insisted that ${}^\bullet t$ be nonempty for each $t \in T$.

The places of a Petri net intuitively correspond to local states of the system. A global state, called a *marking*, is a multiset $M : S \to \mathbf{N}_0$. If $M(s) = n$, then s is said to be assigned n tokens by M.

A transition t can occur at a marking M if for all $s \in S$, $M(s) \geq W(s,t)$. We say that t is *enabled at* M and denote this by $M[t\rangle$.

When a transition t occurs at a marking M, a new marking M' is generated according to the following rule:

$$\forall s \in S.\ M'(s) = M(s) - W(s,t) + W(t,s)$$

We denote the fact that M evolves to M' via t by $M[t\rangle M'$.

Suppose t_1 and t_2 are two transitions and M is a marking such that $\forall s \in S.\ M(s) \geq W(s,t_1) + W(s,t_2)$. Then t_1 and t_2 can both occur *independently* at M and are thus *concurrently enabled*. In such a situation, M can evolve in a single *step* by the occurrence of both t_1 and t_2 to a marking M' where

$$\forall s \in S.\ M'(s) = M(s) - W(s,t_1) - W(s,t_2) + W(t_1,s) + W(t_2,s)$$

We can thus extend the transition relation associated with a Petri net to permit steps of actions between a pair of markings. In general, such a step will be a multiset over T rather than a subset of T because a transition may be concurrent with itself (a phenomenon called *autoconcurrency*).

Let $u \in MS_{fin}(T)$. u is enabled at a marking M, denoted $M[u\rangle$, if for all $s \in S$, $M(s) \geq \sum_{t \in T} u(t) \cdot W(s,t)$. (Recall that $u(t)$ denotes the number of occurrences of t in u). When u occurs, M is transformed to M' (denoted $M[u\rangle M'$) where

$$\forall s \in S.\ M'(s) = M(s) + \sum_{t \in T} u(t) \cdot (W(t,s) - W(s,t))$$

If ${}^\bullet t = \emptyset$, it is clear that unboundedly large steps consisting of copies of t will be enabled at any reachable marking. This is a fairly undesirable phenomenon and prompts the restriction we have made that every transition have an input place. This restriction was not present in the nets considered in [9]. We shall say more on this in Section 4.

The set of all markings reachable from a marking M is denoted by $[M\rangle$. $[M\rangle$ is the smallest set of markings such that

- $M \in [M\rangle$
- If $M' \in [M\rangle$ and $\exists u \in MS_{fin}(T).\ M'[u\rangle M''$ then $M'' \in [M\rangle$.

Given a Petri net $PN = (S, T, W, M_{in})$, we can associate a transition relation $\Rightarrow_{PN} \subseteq [M_{in}\rangle \times MS_{fin}(T) \times [M_{in}\rangle$ with PN as follows.

$$\Rightarrow_{PN} = \{(M, u, M') \mid M \in [M_{in}\rangle \textit{ and } M[u\rangle M'\}.$$

Using $\Rightarrow_{PN}$, we can associate with PN an obvious transition system TS_{PN} whose states are the reachable markings of PN and whose transition relation is labelled by multisets. We shall formally define such *step* transition systems in the next section.

Here, we proceed by constructing a category of Petri nets. To do so, we have to define a suitable notion of morphism.

Definition 2.2 *Let $PN_i = (S_i, T_i, W_i, M^i_{in}), i = 1, 2$, be two Petri nets. A net morphism from PN_1 to PN_2 is a pair $\phi = (\phi_S, \phi_T)$ where:*

(i) $\phi_S : S_2 \rightharpoonup S_1$ is a partial function. (Notice that ϕ_S is a map from S_2 to S_1 and not from S_1 to S_2. Thus, in the "forward" direction, $\phi_S^{-1} \subseteq S_1 \times S_2$ is a relation. For $X \subseteq S_1$, $\phi_S^{-1}(X)$ denotes the set $\{y \in S_2 \mid \phi_S(y) \in X\}$.)

(ii) $\phi_T : T_1 \rightharpoonup T_2$ is a partial function.

(iii) $\forall s_1 \in S_1.\ \forall s_2 \in S_2$. If $s_1 = \phi_S(s_2)$ then $M^1_{in}(s_1) = M^2_{in}(s_2)$.

(iv) $\forall t_1 \in T_1$. If $\phi_T(t_1)$ is undefined then $\phi_S^{-1}({}^\bullet t_1) = \phi_S^{-1}(t_1{}^\bullet) = \emptyset$.

(v) $\forall t_1 \in T_1$. If $\phi_T(t_1) = t_2$ then:

- *$\phi_S^{-1}({}^\bullet t_1) = {}^\bullet t_2$ and $\phi_S^{-1}(t_1{}^\bullet) = t_2{}^\bullet$.*
- *$\forall s \in {}^\bullet t_2.\ W_1(\phi_S(s), t_1) = W_2(s, t_2)$.*
- *$\forall s \in t_2{}^\bullet.\ W_1(t_1, \phi_S(s)) = W_2(t_2, s)$.*

We shall denote both ϕ_S and ϕ_T by ϕ, unless it is unclear from the context which component we are referring to. Thus, normally we shall write $\phi(s)$ for $\phi_S(s)$ and $\phi(t)$ for $\phi_T(t)$.

Let $\mathcal{PN}et$ be the category whose objects are Petri nets and whose arrows are net morphisms as defined above.

We conclude this section with a result showing that net morphisms preserve concurrent behaviour in a strong way. The proof of this result is given in [9].

Lemma 2.3 *Let $PN_i = (S_i, T_i, W_i, M_{in}^i), i = 1, 2$, be two Petri nets and let ϕ be a net morphism from PN_1 to PN_2. For each $M \in [M_{in}^1\rangle$, define $M_\phi : S_2 \to \mathbf{N}_0$ as follows:*

$$\forall s \in S_2.\ M_\phi(s) = \begin{cases} M(\phi(s)) & \text{if } \phi(s) \text{ exists} \\ M_{in}^2(s) & \text{otherwise} \end{cases}$$

We then have the following:

(i) $\forall M \in [M_{in}^1\rangle.\ M_\phi \in [M_{in}^2\rangle$.

(ii) Suppose that $(M, u, M') \in \Rightarrow_{PN_1}$. *Then* $(M_\phi, \phi(u), M'_\phi) \in \Rightarrow_{PN_2}$.

3 *PN*-transition systems

A labelled transition system is usually defined as a quadruple $TS = (Q, \Sigma, \to, q_{in})$, where Q is a set of states and $\to \subseteq Q \times \Sigma \times Q$ is a (sequential) transition relation which describes how the system evolves from state to state by performing actions from Σ, beginning with the initial state q_{in}.

We enrich the transition relation by permitting one state to be transformed to another in a single *step* consisting of a finite multiset of actions. We can then define the class of *PN*-transition systems as a subclass of this new class of transition systems which satisfies some simple axioms ensuring that all the steps in the system are "consistent".

Definition 3.1 *A* step transition system *is a structure $TS = (Q, E, \to, q_{in})$, where*

- *Q is a countable set of* states, *with $q_{in} \in Q$ as the* initial state.
- *E is a countable set of* events.
- *$\to \subseteq Q \times MS_{fin}(E) \times Q$ is the* transition relation.

We shall often write $q \xrightarrow{u} q'$ instead of $(q, u, q') \in \to$.

We can extend $\to$ to a relation $\to^*$ over step sequences in the usual way. Let $\rho = u_1 u_2 \ldots u_n \in (MS_{fin}(E))^*$ be a sequence of steps. Then $(q, \rho, q') \in \to^*$ iff $\exists q_0, q_1, \ldots, q_n.$ $q_0 = q$, $q_n = q'$ and $q_{i-1} \xrightarrow{u_i} q_i$ for $1 \le i \le n$.

We put three basic restrictions on transition systems. First, we introduce idling transitions, represented by the empty multiset, as self loops at each state and demand that these special transitions occur *only* as self loops. We also ensure that all states in a transition system are reachable from the initial state. Finally, we insist that there be no unbounded autoconcurrency in the system. Formally, we have the following basic axioms.

(A1) $\forall q, q' \in Q.\ q \xrightarrow{\mathtt{0}_E} q'$ iff $q = q'$ (where $\mathtt{0}_E$ is the empty multiset over E).

(A2) $\forall q \in Q.\ \exists \rho \in (MS_{\mathit{fin}}(E))^*.\ (q_{in}, \rho, q) \in \rightarrow^*$.

(A3) $\forall q \in Q.\ \forall e \in E.\ \exists k \in \mathbf{N}_0.\ (q, u, q') \in \rightarrow$ *implies* $u(e) < k$.

Henceforth, we shall assume that every step transition system we consider satisfies axioms (A1), (A2) and (A3).

Notice that (A1) does *not* rule out the presence of non-trivial self-loops of the form $q \xrightarrow{u} q$.

(A3) corresponds to the restriction we have placed on nets that each transition have an input place. Notice that no global bound is placed on autoconcurrency—all (A3) says is that autoconcurrency is locally bounded at each state in the transition system.

To describe *PN*-transition systems, we need to introduce the notion of a region. Regions were originally defined in the context of elementary transition systems in [3] and exploited to define a coreflection between elementary transition systems and elementary net systems in [12]. Here we generalize the regions of [3, 12] to describe the transition system counterpart of a place of a Petri net.

Definition 3.2 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a step transition system. A* region *is a pair of functions $r = (r_Q, r_E)$ such that:*

(i) $r_Q : Q \rightarrow \mathbf{N}_0$.

(ii) $r_E : E \rightarrow \mathbf{N}_0 \times \mathbf{N}_0$.

For convenience, we denote the first component of $r_E(e)$ as ${}^re$ and the second component of r_E as e^r. In other words, if $r_E(e) = (n_1, n_2)$, then ${}^re = n_1$ and $e^r = n_2$.

(iii) $\forall (q, u, q') \in \rightarrow.\quad r_Q(q) \geq \sum_{e \in E} u(e) \cdot {}^re$ *and* $r_Q(q') = r_Q(q) + \sum_{e \in E} u(e) \cdot (e^r - {}^re)$.

We shall denote both r_Q and r_E by r, unless it is unclear from the context which component we are referring to. Thus, normally we shall write $r(q)$ for $r_Q(q)$ and $r(e)$ for $r_E(e)$. If ${}^re > 0$, we say that r is a *preregion* of e and if $e^r > 0$, we say that r is a *postregion* of e.

So, a region r corresponds to a place of the Petri net which we would like to associate with a given step transition system. Recall that for a Petri net PN, we can associate an "obvious" transition system TS_{PN}, with states corresponding to the reachable markings of PN, events to the transitions of PN and the step transition relation defined by $\Rightarrow_{PN}$. We specify the number of tokens on the "place" r at the "marking" q by $r(q)$. For each $e \in E$, $r(e)$ specifies the "weights" $W(r, e)$ and $W(e, r)$. The last condition in the definition of a region ensures that r_Q is consistent with the overall behaviour of the net — for every transition $q \xrightarrow{u} q'$ present in the system, $r(q)$ must have enough "tokens" to permit u to occur and $r(q')$ must contain the correct number of "tokens" as specified by the normal firing rule of a Petri net.

We disregard regions r which are "disconnected" from all the events — i.e. r such that $r(e) = (0, 0)$ for all $e \in E$. These *trivial* regions correspond to isolated places in a Petri net and do not contribute in any way to characterizing the behaviour of the system.

Definition 3.3 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a step transition system. A region r is* non-trivial *iff for some $e \in E$, $r(e) \neq (0,0)$. We denote the set of non-trivial regions of TS by $\mathcal{R}_{TS}$.*

Henceforth, whenever we make a statement referring to all regions, we assume that we are only considering non-trivial regions (unless explicitly stated otherwise).

PN-transition systems are characterized by two "regional" axioms in addition to the basic axioms (A1) and (A2):

(A4) Let $q, q' \in Q$. $q \neq q' \Rightarrow \exists r \in \mathcal{R}_{TS}.\ r(q) \neq r(q')$. *(Separation)*

(A5) $\forall q \in Q.\ \forall u \in MS_{fin}(E)$. If there does not exist $q' \in Q$ such that $q \xrightarrow{u} q'$, then $\exists r \in \mathcal{R}_{TS}.\ r(q) < \sum_{e \in E} u(e) \cdot {}^{r}e$. *(Enabling)*

Axiom (A4) says that any pair of distinct states in Q will be distinguished by at least one (non-trivial) region. Axiom (A5) captures the fundamental idea underlying the dynamic behaviour of a Petri net. It says that if the system cannot perform a step labelled by u at the state q then there must be some region r which does not have enough "tokens" at q to permit u to occur. In other words, whenever a multiset of actions u is enabled at a state q of the system by all regions, it must be possible to perform u and reach some state q' in the system.

Definition 3.4 *A PN*-transition system *is a step transition system $TS = (Q, E, \rightarrow, q_{in})$ which satisfies axioms (A4) and (A5) (in addition to the basic axioms (A1) to (A3)).*

We now state a couple of useful properties of *PN*-transition systems which are formally established in [9].

The first observation about *PN*-transition systems is that they are deterministic. $TS = (Q, E, \rightarrow, q_{in})$ is said to be a *deterministic* step transition system in case the following is true:

$$\forall q \in Q.\ \forall u \in MS_{fin}(E).\ (q, u, q') \in\ \rightarrow\ \textit{and}\ (q, u, q'') \in\ \rightarrow\ \textit{implies}\ q' = q''.$$

Proposition 3.5 *Every PN-transition system is deterministic.*

The second observation is that every step in a *PN*-transition system can be broken up into substeps in a consistent way. This shows that steps do indeed reflect concurrency in a natural way.

Proposition 3.6 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a PN-transition system and let $q \xrightarrow{u} q'$ in TS. Then*

$$\forall v \subseteq_{MS} u.\ \exists q_v \in Q.\ q \xrightarrow{v} q_v\ \textit{and}\ q_v \xrightarrow{u-v} q'.$$

To construct a category of *PN*-transition systems, we now define morphisms between *PN*-transition systems. These are standard transition system morphisms as defined, say, in [12, 19], extended to respect steps.

Definition 3.7 *Let $TS_i = (Q_i, E_i, \rightarrow_i, q^i_{in})$, $i = 1, 2$, be two PN-transition systems. A* transition system morphism *f from TS_1 to TS_2 is a pair of functions $f = (f_Q, f_E)$ where:*

(i) $f_Q : Q_1 \to Q_2$ *is a total function such that* $f_Q(q^1_{in}) = q^2_{in}$.

(ii) $f_E : E_1 \rightharpoonup E_2$ *is a partial function.*

(iii) If $(q, u, q') \in \ \to_1$ *then* $(f_Q(q), f_E(u), f_Q(q')) \in \ \to_2$.

As with regions, we shall denote both f_Q and f_E by f, unless it is unclear from the context which component we are referring to. Thus, normally we shall write $f(q)$ for $f_Q(q)$ and $f(e)$ for $f_E(e)$.

Notice that the last clause ensures that if a step u is hidden by f then every transition $(q, u, q') \in \ \to_1$ results in q and q' being mapped to the same state in Q_2; i.e. if for all $e \in u$, $f(e)$ is undefined, then $(q, u, q') \in \ \to_1$ implies $(f(q), \mathsf{0}_{E_2}, f(q')) \in \ \to_2$, which by axiom (A1) forces $f(q) = f(q')$.

PN-transition systems with transition system morphisms form a category, which we shall call $\mathcal{PN}ts$.

4 Relating Petri nets and *PN*-transition systems

There is a natural way to define a functor NT from $\mathcal{PN}et$ to $\mathcal{PN}ts$.

NT maps objects in the obvious way—each Petri net PN is mapped to the transition system associated with its "step" marking diagram. Let $PN = (S, T, W, M_{in})$ be a Petri net. Then

$$\mathsf{NT}(PN) = ([M_{in}\rangle, T, \Rightarrow_{PN}, M_{in})$$

where $[M_{in}\rangle$ is the set of markings reachable from the initial marking M_{in}, T is the set of transitions of PN and $\Rightarrow_{PN}$ is the step transition relation for Petri nets defined in Section 2.

Next we define how NT maps arrows. Let $PN_i = (S_i, T_i, W_i, M^i_{in})$, $i = 1, 2$, be two Petri nets and let ϕ be a net morphism from PN_1 to PN_2. Then, $\mathsf{NT}(\phi) = f^\phi$ is defined as follows.

- $\forall t \in T_1.\ f^\phi(t) = \phi(t)$.
- $\forall M \in [M^1_{in}\rangle.\ f^\phi(M) = M_\phi$ (where M_ϕ is the map defined in Lemma 2.3).

The main result established in [9] is the existence of a left adjoint to this functor.

Theorem 4.1 *There exists a functor* $\mathsf{TN} : \mathcal{PN}ts \to \mathcal{PN}et$ *such that* TN *is left adjoint to* NT.

The unit of the adjunction in fact turns out to be a natural isomorphism, so there is actually a *coreflection* between this pair of functors.

We shall not describe TN in any detail. The main idea is that a *PN*-transition system can be transformed into a Petri net by regarding events as the transitions of the net and regions as the places of the net.

A remark is in order at this point about the categories $\mathcal{PN}ts$ and $\mathcal{PN}et$ and the coreflection that we have defined here. In the original formulation of the coreflection between *PN*-transition systems and Petri nets in [9], no assumption was made about transitions in a net having input places. Correspondingly, the axiom (A3) that we have introduced for *PN*-transition systems was not present.

It is quite clear that for any net $PN = (S, T, W, M_{in})$, if every $t \in T$ has an input place, then TS_{PN} will satisfy axiom (A3). It is not difficult to prove the converse—if $TS = (Q, E, \rightarrow, q_{in})$ satisfies (A3), then for every $e \in E$ there exists an $r \in \mathcal{R}_{TS}$ such that ${}^{r}e > 0$ and so $\mathsf{TN}(TS)$ will have an input place for each transition.

The coreflection that is established in [9] continues to hold when we restrict TN and NT to the categories we have defined here. The reason we have chosen to work in this more restrictive framework is that here we will be dealing mainly with special classes of nets, like safe nets, which do not exhibit any autoconcurrency under "normal circumstances". So, for these classes of nets, it is reasonable to demand that we abolish the unbounded autoconcurrency generated by transitions with no input places.

5 Sequential *PN*-transition systems

PN-transition systems faithfully record the concurrent behaviour of Petri nets by means of transitions labelled with multisets of events.

However, it turns out that we can also characterize the transition systems corresponding to the purely sequential behaviour of Petri nets.

For convenience, we shall define sequential transition systems as special cases of step transition systems.

Definition 5.1 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a step transition system. TS is* sequential *iff TS satisfies axioms (A1) to (A3) and, further,*

$$\forall (q, u, q') \in \rightarrow . \ |u| \leq 1.$$

In other words, a sequential transition system can have steps labelled either by single events, or by $\mathsf{0}_E$, corresponding to the idling transition at each state.

Definition 5.2 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a transition system. TS is a* sequential *PN*-transition system *if it is sequential and, further, it satisfies the two axioms (A4) and (A5) for PN-transition systems.*

It is clear that any sequential *PN*-transition system is also a (normal) *PN*-transition system. In fact, we can define a full subcategory $\mathcal{S}eq\mathcal{PN}ts$ whose objects are sequential *PN*-transition systems and whose arrows are transition system morphisms.

It is not difficult to prove the following.

Theorem 5.3 *$\mathcal{S}eq\mathcal{PN}ts$ is a coreflective subcategory of $\mathcal{PN}ts$.*

Proof To establish this, we have to show that the inclusion functor from $\mathcal{S}eq\mathcal{PN}ts$ to $\mathcal{PN}ts$ has a right adjoint. The right adjoint is the functor which forgets concurrency.

More formally, define $\mathsf{U} : \mathcal{PN}ts \rightarrow \mathcal{S}eq\mathcal{PN}ts$ to be the functor which maps a *PN*-transition system $TS = (Q, E, \rightarrow, q_{in})$ to a sequential *PN*-transition system $TS' = (Q, E, \rightarrow', q_{in})$, where

$$\rightarrow' = \{(q, u, q') \mid q \xrightarrow{u} q' \ \textit{and} \ |u| \leq 1\}.$$

For each transition system morphism $f : TS_1 \to TS_2$ in $\mathcal{PN}ts$, $\mathsf{U}(f) : \mathsf{U}(TS_1) \to \mathsf{U}(TS_2)$ is the map such that $\mathsf{U}(f)_Q = f_Q$ and $\mathsf{U}(f)_E = f_E$.

It is straightforward to verify that U is in fact right adjoint to the inclusion functor. We omit the details. $\square$

So, by composing the inclusion functor with the functor $\mathsf{TN} : \mathcal{PN}ts \to \mathcal{PN}et$ we obtain a functor which is left adjoint to the functor $\mathsf{U} \circ \mathsf{NT}$ taking nets to their sequential marking diagrams.

We can also characterize sequential *PN*-transition systems directly in terms of regions.

Proposition 5.4 *Let $TS = (Q, E, \to, q_{in})$ be a PN-transition system. TS is sequential iff $r_{seq} \in \mathcal{R}_{TS}$, where r_{seq} is defined as follows:*

$\forall q \in Q.\ r_{seq}(q) = 1.$

$\forall e \in E.\ r_{seq}(e) = (1, 1).$

Proof It is easy to see that if TS is sequential, r_{seq} is in fact a region. Conversely, if r_{seq} is a region in $\mathcal{R}_{TS}$, clearly for each transition $q \xrightarrow{u} q'$ in TS, $|u| \leq 1$ and so TS is sequential. $\square$

In the net $\mathsf{TN}(TS)$ corresponding to the sequential *PN*-transition system TS, r_{seq} will be a place marked at the initial marking and connected to all transitions by self-loops, ensuring that the net exhibits no concurrency in its behaviour.

6 Safe nets

Petri nets can exhibit very complex behaviours which are difficult to characterize globally. To obtain a mathematically tractable theory, one often looks at restricted classes of nets.

In this regard, one very important subclass of nets is the class of *safe* nets. In general, a Petri net $PN = (S, T, W, M_{in})$ is said to be k-*safe* if $M(s) \leq k$ for every reachable marking $M \in [M_{in}\rangle$. Call PN a *safe* net if it is 1-safe.

Thus, in a safe net, every reachable marking is a set, rather than a multiset of places. Let $PN = (S, T, W, M_{in})$ be a safe net. For any transition $t \in T$, if there is a place $s \in {}^\bullet t$ such that $W(s,t) > 1$, then t will never be enabled. Similarly, if there is a place $s \in t^\bullet$ such that $W(t,s) > 0$, t can never occur because after t occurs, s would be unsafe. So, it makes sense to restrict W to values from $\{0, 1\}$ instead of the entire range $\mathbf{N}_0$.

With this in mind, we define safe nets in terms of our general definition of Petri nets as follows.

Definition 6.1 *Let $PN = (S, T, W, M_{in})$ be a Petri net. Then PN is a* safe net *provided*

(i) $\forall s \in S.\ \forall t \in T.\ W(s,t) \leq 1$ *and* $W(t,s) \leq 1$.

(ii) $\forall M \in [M_{in}\rangle.\ \forall s \in S.\ M(s) \leq 1$.

Let $\mathcal{SN}et$ be the full subcategory of $\mathcal{PN}et$ whose objects are safe nets. We can restrict the functor NT to a functor $\mathsf{SNT} : \mathcal{SN}et \to \mathcal{PN}ts$.

We now want to identify a subcategory $\mathcal{SPN}ts$ of $\mathcal{PN}ts$ such that there is a coreflection between $\mathsf{STN} : \mathcal{SPN}ts \to \mathcal{SN}et$ and $\mathsf{SNT} : \mathcal{SN}et \to \mathcal{SPN}ts$.

For this, we define $\mathsf{0/1}$-regions.

Definition 6.2 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a step transition system. Then $r = (r_Q, r_E)$ is a* 0/1-region *of TS if r is a region and*

$\forall q \in Q.\ r_Q(q) \leq 1.$

$\forall e \in E.\ {}^re \leq 1$ *and* $e^r \leq 1.$

Let $\mathcal{R}_{TS}^{0/1} = \{r \in \mathcal{R}_{TS} \mid r$ is a 0/1-region $\}$.

We can then modify the regional axioms (A4) and (A5) to refer only to 0/1-regions.

(A4') Let $q, q' \in Q$. $q \neq q' \Rightarrow \exists r \in \mathcal{R}_{TS}^{0/1}.\ r(q) \neq r(q')$. *(Separation)*

(A5') $\forall q \in Q.\ \forall u \in MS_{fin}(E)$. If there does not exist $q' \in Q$ such that $q \xrightarrow{u} q'$, then $\exists r \in \mathcal{R}_{TS}^{0/1}.\ r(q) < \sum_{e \in E} u(e) \cdot {}^re.$ *(Enabling)*

Definition 6.3 *A* safe *PN*-transtion system *is a step transition system $TS = (Q, E, \rightarrow, q_{in})$ which satisfies axioms (A4') and (A5') (in addition to the basic axioms (A1) to (A3)).*

Let $\mathcal{SPN}ts$ be the full subcategory of $\mathcal{PN}ts$ whose objects are safe *PN*-transition systems.

As we had mentioned in Section 2, in general we need to consider steps labelled by multisets rather than sets in order to deal with autoconcurrency. Clearly, a safe net cannot exhibit autoconcurrency. So, since safe *PN*-transition systems are supposed to describe the behaviour of safe nets, it is not surprising that we have the following.

Proposition 6.4 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a safe PN-transition system. Then for every transition $q \xrightarrow{u} q'$ in TS, u is a set.*

Proof Let $e \in u$. It suffices to show that there is a 0/1-region r such that ${}^re = 1$.

By (A3) we know that autoconcurrency is bounded. In other words, at each state q, there is some $k \in \mathbf{N}_0$ such that the step consisting of k occurrences of e is not enabled. By (A5') there is a 0/1-region r such that $r(q) < k \cdot {}^re$. It is clear that ${}^re$ must be 1, and so we are done. □

We now establish that the transition system TS_{PN} associated with a safe net PN is in fact a safe *PN*-transition system.

Lemma 6.5 *Let $PN = (S, T, W, M_{in})$ be a safe net. Then $TS_{PN} = ([M_{in}\rangle, T, \Rightarrow_{PN}, M_{in})$ is a safe PN-transition system.*

Proof It is straightforward to show that TS_{PN} satisfies the three basic axioms (A1) to (A3) for step transition systems. So, what we have to show is that axioms (A4') and (A5') are true as well.

For each $s \in S$, we can define a region r_s in TS_{PN} as follows.

$\forall M \in [M_{in}\rangle.\ r_s(M) = M(s).$

$\forall t \in T.\ r_s(t) = (W(s,t), W(t,s)).$

It is not difficult to establish that r_s is a region, and, in fact is a 0/1-region. r_s will be non-trivial provided s is not isolated in PN (i.e. there is some $t \in T$ such that $s \in {}^\bullet t$ or $s \in t^\bullet$.)

It then immediately follows that TS_{PN} satisfies (A4'). Given any $M, M' \in [M_{in}\rangle$, if $M \neq M'$, there must be a non-isolated place $s \in S$ such that $M(s) \neq M(s')$. Then clearly r_s is a non-trivial $\mathsf{0/1}$-region of TS_{PN} separating M from M'.

Next consider (A5'). Suppose $M \in [M_{in}\rangle$ and $u \in MS_{fin}(T)$, and there is no M' such that $M \xrightarrow{u} M'$. Then, at the marking $M \in [M_{in}\rangle$, u is not enabled. By the firing rule for Petri nets, this implies that there is some $s \in S$ such that $M(s) < \sum_{t \in u} W(s,t)$. Clearly, r_s is then a non-trivial $\mathsf{0/1}$-region such that $r_s(M) < \sum_{t \in u} {}^{r_s}t$ and we are done. □

Given a pair of safe nets $PN_i = (S_i, T_i, W_i, M_{in}^i)$, $i = 1, 2$, and a net morphism $\phi : PN_1 \to PN_2$, for each $M \in [M_{in}^1\rangle$ we can define a marking $M_\phi \in [M_{in}^2\rangle$ as in Lemma 2.3. That is

$$\forall s \in S_2 . M_\phi(s) = \begin{cases} M(\phi(s)) & \textit{if } \phi(s) \textit{ exists} \\ M_{in}^2(s) & \textit{otherwise} \end{cases}$$

We can then define $\mathsf{SNT} : \mathcal{SN}et \to \mathcal{SPN}ts$ as follows:

- Let $PN = (S, T, W, M_{in})$ be a safe net. Then $\mathsf{SNT}(PN) = ([M_{in}\rangle, T, \Rightarrow_{PN}, M_{in})$.
- Let $PN_i = (S_i, T_i, W_i, M_{in}^i)$, $i = 1, 2$, be a pair of safe nets and $\phi : PN_1 \to PN_2$ a net morphism. Then $\mathsf{SNT}(\phi) : \mathsf{SNT}(PN_1) \to \mathsf{SNT}(PN_2)$ is given by:
  - $\forall t \in T_1$. $\mathsf{SNT}(\phi)(t) = \phi(t)$.
  - $\forall M \in [M_{in}^1\rangle$. $\mathsf{SNT}(\phi)(M) = M_\phi$.

It is easy to check the following.

Proposition 6.6 $\mathsf{SNT} : \mathcal{SN}et \to \mathcal{SPN}ts$ *is a functor.*

We can construct a functor $\mathsf{STN} : \mathcal{SPN}ts \to \mathcal{SN}et$ which is left adjoint to SNT. We first define STN_0, a map on objects from $\mathcal{SPN}ts$ to $\mathcal{SN}et$.

Let $TS = (Q, E, \to, q_{in})$ be a safe *PN*-transition system. Then

$$\mathsf{STN}_0(TS) = (\mathcal{R}_{TS}^{0/1}, E, W_{TS}, M_{in}^{TS})$$

where $W_{TS}(r, e) = {}^r e$ and $W_{TS}(e, r) = e^r$ for each $r \in \mathcal{R}_{TS}^{0/1}$ and $e \in E$, and $M_{in}^{TS}(r) = r(q_{in})$ for each $r \in \mathcal{R}_{TS}^{0/1}$.

Theorem 6.7 STN_0 *extends to a functor* $\mathsf{STN} : \mathcal{SPN}ts \to \mathcal{SN}et$ *such that* STN *is left adjoint to* SNT *and the unit of the adjunction is a natural isomorphism.*

Proof We just sketch the main ideas. The details are similar to those used to establish the coreflection between TN and NT and can be filled in from [9].

We can first establish that for each safe *PN*-transition system TS, there is a transition system isomorphism $\eta_{TS} : TS \to \mathsf{SNT} \circ \mathsf{STN}_0(TS)$. This map will serve as the unit of the adjunction.

Suppose that $TS \in \mathcal{SPN}ts$ and $PN \in \mathcal{SN}et$ such that there is a transition system morphism $f : TS \to \mathsf{SNT}(PN)$. Then, we can establish that there is a unique morphism $\phi : \mathsf{STN}_0(TS) \to PN$ such that $f = \mathsf{SNT}(\phi) \circ \eta_{TS}$.

Given this, if follows (according to [6]), that STN_0 can be extended uniquely to a functor $\mathsf{STN} : \mathcal{SPN}ts \to \mathcal{SN}et$ which is left adjoint to SNT. □

7 Elementary transition systems

Next, we look at one of the basic models of net theory, elementary net systems. In [12], Nielsen, Rozenberg and Thiagarajan establish a coreflection between a class of transition systems called elementary transition systems and elementary net systems.

In many ways, that result is the starting point of the work reported here. In this section, we define a subcategory of $\mathcal{PN}ts$ whose objects are elementary *PN*-transition systems, which corresponds to the category of elementary transition systems of [12]—that is, there is an equivalence between these two categories.

We begin by describing elementary net systems. This will motivate the axioms we need to put on *PN*-transition systems to define elementary *PN*-transition systems.

Rather than try and define elementary net systems in terms of general Petri nets, we start from scratch and provide the standard definition (see, for instance, [16]).

We begin with the definition of a net.

Definition 7.1 *A* net *is a triple $N = (S, T, F)$ where:*

(i) S *is a set of* S-elements *and* T *is a set of* T-elements, *such that* $S \cap T = \emptyset$.

(ii) $F \subseteq (S \times T) \cup (T \times S)$ *is the* flow relation *such that* $\forall x \in S \cup T.\ \exists y \in S \cup T.\ [(x, y) \in F \vee (y, x) \in F]$.

Thus a net specifies the underlying structure of a system. The flow relation F corresponds to the $\{0, 1\}$-valued weight function we defined for safe nets. We use ${}^\bullet x$, $x^\bullet$ and ${}^\bullet x^\bullet$ to denote the neighbourhood of $x \in S \cup T$, as usual.

Normally, the S-elements are called *conditions* and denoted by B and the T-elements are called *events* and denoted by E. Here, we shall stick to S and T to remain consistent with the notation for nets used so far.

Definition 7.2 *An* elementary net system *is a quadruple $ENS = (S, T, F, c_{in})$ where*

(i) (S, T, F) *is a net, called the underlying net of* ENS.

(ii) $c_{in} \subseteq S$ *is the* initial case.

Thus, the initial case corresponds to an initial marking in a safe net. The essential difference between an elementary net system and a safe net is in the firing rule. Let $c, c' \subseteq S$ be *cases* of an elementary net system and $t \in T$ be an event. Then

$$c \xrightarrow{t} c' \overset{\text{def}}{=} c - c' = {}^\bullet t \wedge c' - c = t^\bullet$$

Thus, in an elementary net system, an event cannot occur at a case where its post-conditions are not empty. This means that an event which is connected to a condition by a self-loop will be permanently disabled.

Given the transition relation defined above, we can define $[c_{in}\rangle$, the set of cases reachable from c_{in}, in the same way that we defined $[M_{in}\rangle$ for Petri nets. We can then associate a sequential transition relation $\rightarrow_{ENS}$ with an elementary net system $ENS = (S, T, F, c_{in})$ in the obvious way:

$$\rightarrow_{ENS} = \{(c, t, c') \mid c, c' \in [c_{in}\rangle \textit{ and } c \xrightarrow{t} c'\}$$

We can extend this sequential transition relation to a step transition relation between cases. As in a safe net, a set of transitions is concurrently enabled at a case provided each individual transition is enabled and the neighbourhoods of the transitions are pairwise disjoint.

$$\begin{array}{rl}\Rightarrow_{ENS}= \{c, u, c' \mid & c, c' \in [c_{in}\rangle, u = \{t_1, t_2, \dots, t_n\}, \\ & \exists c_1, c_2, \dots, c_n.\ \forall i \in \{1, 2, \dots, n\}.\ c \xrightarrow{t_i} c_i,\ \ and \\ & \forall i, j \in \{1, 2, \dots, n\}.\ i \neq j\ implies\ {}^\bullet t_i{}^\bullet \cap {}^\bullet t_j{}^\bullet = \emptyset\}\end{array}$$

So, given an elementary net system $ENS = (S, T, F, c_{in})$, we can associate with it a sequential transition system $STS_{ENS} = ([c_{in}\rangle, T, \rightarrow_{ENS}, c_{in})$ and a step transition system $TS_{ENS} = ([c_{in}\rangle, T, \Rightarrow_{ENS}, c_{in})$.

In [3], Ehrenfeucht and Rozenberg gave a characterization of the sequential transition systems arising from elementary net systems. In [12], this characterization was extended to a coreflection between these *elementary transition systems* and elementary net systems.

Here we shall show how to characterize the step transition systems corresponding to elementary net systems as a suitable subclass of *PN*-transition systems. We shall then establish a categorical equivalence between our *elementary PN*-transition systems and elementary transition systems.

We begin by defining elementary regions.

Definition 7.3 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a step transition system. An elementary region of TS is a pair of functions $r = (r_Q, r_E)$ such that r is a region of TS and, in addition:*

$\forall q \in Q.\ r(q) \leq 1.$

$\forall e \in E.\ r(e) \in \{(0,1), (1,0), (0,0)\}.$

Let $\mathcal{R}^E_{TS}$ denote the set of all non-trivial elementary regions of TS.

Thus an elementary region is a 0/1-region with the constraint that $r(e) \neq (1,1)$ for any event e. As before, we modify the regional axioms (A4) and (A5) to refer only to elementary regions. We also explicitly add the condition that every $e \in E$ have an occurrence.

(A4") Let $q, q' \in Q.\ q \neq q' \Rightarrow \exists r \in \mathcal{R}^E_{TS}.\ r(q) \neq r(q')$. *(Separation)*

(A5") $\forall q \in Q.\ \forall u \in MS_{fin}(E)$. If there does not exist $q' \in Q$ such that $q \xrightarrow{u} q'$, then $\exists r \in \mathcal{R}^E_{TS}.\ r(q) < \sum_{e \in E} u(e) \cdot {}^r e$. *(Enabling)*

(A6") $\forall e \in E.\ \exists q \xrightarrow{u} q'.\ e \in u.$

Definition 7.4 *An* elementary *PN-transition system is a step transition system $TS = (Q, E, \rightarrow, q_{in})$ which satisfies axioms (A4") to (A6") in addition to the basic axioms (A1) to (A3).*

Let $\mathcal{EPN}ts$ be the full subcategory of $\mathcal{PN}ts$ whose objects are elementary *PN*-transition systems. As with safe *PN*-transition systems, it is easy to show that all steps in an elementary transition system consist of sets of events rather than multisets.

We want to establish a categorical equivalence between the subcategory $\mathcal{EPN}ts$ and the category of elementary transition systems defined in [12]. In order to do this, we first have to describe elementary transition systems.

Elementary transition systems are defined as a subclass of "conventional" sequential transition systems (as opposed to the sequential versions of step transition systems which we defined Definition 5.1).

Definition 7.5 *A* transition system *is a quadruple $STS = (Q, E, \rightarrow, q_{in})$ where*

- *Q is a set of states with $q_{in} \in Q$ as the initial state.*
- *E is a set of events.*
- *$\rightarrow \subseteq Q \times E \times Q$ is the transition relation.*

The next thing to do is to define regions on these transition systems.

Definition 7.6 *Let $STS = (Q, E, \rightarrow, q_{in})$ be a transition system. A* simple region *is a subset $\rho \subseteq Q$ such that:*

(i) $q \xrightarrow{t} q' \wedge q \in \rho \wedge q' \notin \rho \Rightarrow \forall q_1 \xrightarrow{t} q_1'$ *in* $STS.\ [q_1 \in \rho \wedge q_1' \notin \rho]$

(ii) $q \xrightarrow{t} q' \wedge q \notin \rho \wedge q' \in \rho \Rightarrow \forall q_1 \xrightarrow{t} q_1'$ *in* $STS.\ [q_1 \notin \rho \wedge q_1' \in \rho]$

ρ is non-trivial if it is not equal to Q or to $\emptyset$. Let R_{STS} denote the set of non-trivial simple regions of STS.

For $e \in E$, define $\rho \in {}^\bullet e$ if there is a transition $q \xrightarrow{e} q'$ in STS such that $q \in \rho$ and $q' \notin \rho$. Similarly, $\rho \in e^\bullet$ if there is a transition $q \xrightarrow{e} q'$ in STS such that $q \notin \rho$ and $q' \in \rho$. As usual, ${}^\bullet e = \{\rho \in \mathsf{R}_{STS} \mid \rho \in {}^\bullet e\}$, $e^\bullet = \{\rho \in \mathsf{R}_{STS} \mid \rho \in e^\bullet\}$ and ${}^\bullet e^\bullet = {}^\bullet e \cup e^\bullet$.

The class of elementary transition systems is then given by the following axioms.

(EA1) $\forall q \xrightarrow{e} q'.\ q \neq q'$.

(EA2) $\forall e \in E.\ \exists q \xrightarrow{e} q'$.

(EA3) $\forall q \in Q.\ \exists \sigma \in E^*.\ (q_{in}, \sigma, q) \in \rightarrow^*$.

(EA4) $\forall q, q' \in Q.\ q \neq q' \Rightarrow \exists \rho \in \mathsf{R}_{STS}.\ q \in \rho \Leftrightarrow q' \notin \rho$.

(EA5) $\forall q \in Q.\ \forall e \in E.$ If there does not exist q' such that $q \xrightarrow{e} q'$ then $\exists \rho \in {}^\bullet e.\ q \notin \rho$.

The first axiom rules out self loops in the transition system. The other axioms correspond to restrictions we have encountered before. In the formulation of elementary transition systems presented in [12], there is an additional axiom preventing two different transitions between the same pair of states. This amounts to requiring *simplicity* of the nets one is considering. In [9] we have pointed out that the coreflection between elementary transition systems and elementary net systems holds even without this restriction, so we avoid this additional axiom here.

A morphism between elementary transition systems is, as usual, a total function on the states and a partial function on the events that preserves the transition relation. The only complication is that we do not have idling transitions, so we have to be a bit careful in defining the simulation condition. Once again, the definition we present here is slightly different from the one presented in [12], but is equivalent to their formulation.

Definition 7.7 *Let $STS_i = (S_i, E_i, \to_i, q^i_{in})$, $i = 1, 2$, be a pair of transition systems. A morphism f from STS_1 to STS_2 is pair of maps $f = (f_Q, f_E)$ where:*

(i) $f_Q : Q_1 \to Q_2$ is a total function such that $f_Q(q^1_{in}) = q^2_{in}$.

(ii) $f_E : E_1 \rightharpoonup E_2$ is a partial function.

(iii) $\forall q \xrightarrow{e}_1 q'$. If $f_E(e)$ is defined, then $f_Q(q) \xrightarrow{f_E(e)}_2 f_Q(q')$. Otherwise, $f_Q(q) = f_Q(q')$.

Let $\mathcal{ETS}$ denote the category whose objects are elementary transition systems and whose arrows are transition system morphisms as defined above.

In an elementary transition system, every "diamond" represents concurrency. This is stated in a little more generality in the following proposition.

Proposition 7.8 *Let $STS = (Q, E, \to, q_{in})$ be an elementary transition system and $\{e_1, e_2, \ldots, e_n\} \subseteq E$, $n \geq 2$, be a subset of events in E. Then the following statements are equivalent.*

(i) For each $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$ implies ${}^\bullet e_i{}^\bullet \cap {}^\bullet e_j{}^\bullet = \emptyset$

(ii) $\forall q \in Q$. If $\exists q_0, q_1, \ldots, q_n$. $q = q_0$ and $q_{i-1} \xrightarrow{e_i} q_i, 1 \leq i \leq n$, then for each permutation $\pi : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$. $\exists q'_0, q'_1, \ldots, q'_n \in Q$, where $q_0 = q'_0, q_n = q'_n$ and $q'_{i-1} \xrightarrow{e_{\pi(i)}} q'_i, 1 \leq i \leq n$.

Proof $((i) \Rightarrow (ii))$ If the n events have pairwise disjoint neighbourhoods, then they can occur independently. The result is then straightforward, by appealing to axiom (EA5). The proof proceeds by induction on n, the number of pairwise independent events and we omit the details.

$((ii) \Rightarrow (i))$ Again, the proof is straightforward, by induction on n.

The base case is when $n = 2$. So, we have $q, q_1, q_2, q' \in Q$ and $e_1, e_2 \in E$, such that $q \xrightarrow{e_1} q_1 \xrightarrow{e_2} q'$ and $q \xrightarrow{e_2} q_2 \xrightarrow{e_1} q'$.

Consider any $\rho \in \mathsf{R}_{STS}$. Suppose $\rho \in {}^\bullet e_1$. Then, $q \in \rho$ and $q_1 \notin \rho$, so, by definition, $\rho \notin {}^\bullet e_2$. Similarly, we must have $q_2 \in \rho$ and $q' \notin \rho$, so $\rho \notin e_2{}^\bullet$ either. By a similar argument, if $\rho \in e_1{}^\bullet$, $\rho \notin {}^\bullet e_2{}^\bullet$.

A symmetric argument shows that $\rho \in {}^\bullet e_2{}^\bullet$ implies $\rho \notin {}^\bullet e_1{}^\bullet$.

The case $n > 2$ follows in a straightforward way from the induction hypothesis. We omit the details. □

We now prove some useful properties of elementary *PN*-transition systems. The first observation is that elementary regions are completely characterized by their value on states.

Proposition 7.9 *Let $TS = (Q, E, \to, q_{in})$ be an elementary PN-transition system and $r, r' \in \mathcal{R}^E_{TS}$. Then $r_Q = r'_Q$ implies $r = r'$.*

Proof Consider any $e \in E$. Then, by axiom (A6") and Proposition 3.6, we know there is some transition $q \xrightarrow{\{e\}} q'$ in TS. From the definition of an elementary region, it is easy to establish that

$$r(e) = r'(e) = \begin{cases} (1,0) & \textit{if } r(q) = 1 \textit{ and } r(q') = 0 \\ (0,1) & \textit{if } r(q) = 0 \textit{ and } r(q') = 1 \\ (0,0) & \textit{otherwise} \end{cases}$$

□

Next, we show that an elementary *PN*-transition system is free of self-loops labelled by singleton steps.

Proposition 7.10 *Let $TS = (Q, E, \to, q_{in})$ be an elementary PN-transition system. Then, there does not exist a transition of the form $q \xrightarrow{\{e\}} q$ for any $q \in Q$ and $e \in E$.*

Proof By appealing to axiom (A3) which locally bounds autoconcurrency, we can deduce that for every $e \in E$, there is an elementary region r such that ${}^{r}e = 1$. It then follows that $e^{r} = 0$, by the definition of elementary regions. So $r(q)$ cannot be defined consistently for any q such that $q \overset{\{e\}}{\to} q$, and so no such transition can exist in the system. □

We can now define functors $\mathsf{Eseq} : \mathcal{EPN}ts \to \mathcal{ETS}$ and $\mathsf{Estep} : \mathcal{ETS} \to \mathcal{EPN}ts$.
First, we describe Eseq.

- Let $TS = (Q, E, \to, q_{in})$ be an elementary *PN*-transition system. Then $\mathsf{Eseq}(TS) = (Q, E, \to', q_{in})$ where
$$\to' = \{(q, e, q') \mid q \overset{\{e\}}{\to} q'\}.$$
- Let $TS_i = (Q_i, E_i, \to_i, q^i_{in}), i = 1, 2$, be a pair of elementary *PN*-transition systems and $f : TS_1 \to TS_2$ a morphism in $\mathcal{EPN}ts$. Then $\mathsf{Eseq}(f) = \hat{f} : \mathsf{Eseq}(TS_1) \to \mathsf{Eseq}(TS_2)$ is given by:
  - $\forall q \in Q.\ \hat{f}(q) = f(q)$.
  - $\forall e \in E.\ \hat{f}(e) = f(e)$.

Definition 7.11 *Let $r \in \mathcal{R}^{E}_{TS}$ be a non-trivial elementary region of an elementary PN-transition system $TS = (Q, E, \to, q_{in})$. Then $\rho_r \overset{\text{def}}{=} \{q \in Q \mid r(q) = 1\}$.*

Proposition 7.12 *Let $TS = (Q, E, \to, q_{in})$ be an elementary PN-transition system. Then*

(i) $\forall r \in \mathcal{R}^{E}_{TS}$. ρ_r *is an simple region of* $\mathsf{Eseq}(TS)$.

(ii) $\rho_r = \rho_{r'}$ *implies* $r = r'$.

Proof Part (ii) follows from Proposition 7.9. The proof of part (i) is straightforward and we omit the details. □

Proposition 7.13 Eseq *is a functor.*

Proof Given an elementary *PN*-transition system $TS = (Q, E, \rightarrow, q_{in})$, we have to check that $\mathsf{Eseq}(TS)$ is an elementary transition system. Axiom (EA1) holds because of Proposition 7.10. (EA2) and (EA3) are satisfied because TS satisfies (A6") and (A2) respectively. Finally, we come to the regional axioms (EA4) and (EA5). We know that we can find regions in TS satisfying axioms (A4') and (A5'). By Proposition 7.12, these regions correspond uniquely to regions in $\mathsf{Eseq}(TS)$ which would enforce the regional axioms (EA4) and (EA5) in $\mathsf{Eseq}(TS)$.

It is then trivial to check that the image of a morphism in $\mathcal{EPN}ts$ is in fact a morphism in $\mathcal{ETS}$, and we are done. □

To define the functor going the other way, from $\mathcal{ETS}$ to $\mathcal{EPN}ts$, we implicitly use the result proved in Proposition 7.8.

Let $\mathsf{Estep} : \mathcal{ETS} \rightarrow \mathcal{EPN}ts$ be defined as follows:

- Let $STS = (Q, E, \rightarrow, q_{in})$ be an elementary transition system. Then $\mathsf{Estep}(STS) = (Q, E, \rightarrow', c_{in})$ where

$$\begin{array}{rcll} \rightarrow' & = & \{(q, u, q') \mid & u = \{e_1, e_2, \ldots, e_n\} \subseteq E \\ & & & \textit{and } \exists q_0, q_1, \ldots, q_n \in Q.\ q = q_0, q' = q_n \\ & & & \textit{such that } \forall i \in \{1, 2, \ldots, n\}.\ q_{i-1} \xrightarrow{e_i} q_i, \\ & & & \textit{where } \forall i, j \in \{1, 2, \ldots, n\}.\ i \neq j \Rightarrow \\ & & & \qquad {}^\bullet e_i{}^\bullet \cap {}^\bullet e_j{}^\bullet = \emptyset\} \\ & & \cup \ \{(q, \mathsf{0}_E, q) \mid q \in Q\}. & \end{array}$$

- Let $STS_i = (S_i, E_i, \rightarrow_i, q^i_{in}), i = 1, 2$, be a pair of elementary transition systems and $f : STS_1 \rightarrow STS_2$ a morphism in $\mathcal{ETS}$. Then $\mathsf{Estep}(f) = \hat{f} : \mathsf{Estep}(STS_1) \rightarrow \mathsf{Estep}(STS_2)$ is given by:
  - $\forall q \in Q.\ \hat{f}(q) = f(q)$.
  - $\forall e \in E.\ \hat{f}(e) = f(e)$.

Definition 7.14 *Let $\rho \in \mathsf{R}_{STS}$ be a non-trivial simple region of an elementary transition system $STS = (Q, E, \rightarrow, q_{in})$. Then $r_\rho = (r_{\rho_Q}, r_{\rho_E})$, where $r_{\rho_Q} : Q \rightarrow \{0, 1\}$ and $r_{\rho_E} : E \rightarrow \{(0,1), (1,0), (1,1)\}$ is defined as follows.*

$\forall q \in Q.\ r_{\rho_Q}(q) = 1$ *iff* $q \in \rho$.

$$\forall e \in E.\ r_{\rho_E}(e) = \begin{cases} (0,1) & \textit{if } \rho \in e^\bullet \\ (1,0) & \textit{if } \rho \in {}^\bullet e \\ (0,0) & \textit{otherwise} \end{cases}$$

Proposition 7.15 *Let $STS = (Q, E, \rightarrow, q_{in})$ be an elementary transition system. Then*

(i) $\forall \rho \in \mathsf{R}_{STS}$. r_ρ *is an elementary region of* $\mathsf{Estep}(STS)$.

(ii) $r_\rho = r_{\rho'}$ *implies* $\rho = \rho'$.

Proof The proof is straightforward and we omit the details. The main condition we have to check for part (i) is that r_ρ permits all the steps we have introduced, and this follows directly from Proposition 7.8. Part (ii) follows from Proposition 7.9. □

Proposition 7.16 Estep *is a functor.*

Proof For an elementary transition system $STS = (Q, E, \rightarrow, q_{in})$, we have to check that $\mathsf{Estep}(STS)$ is an elementary *PN*-transition system. Axioms (A1) follows from the definition of $\rightarrow'$. Axiom (A2) follows from the fact that STS satisfies axiom (EA3). (A3) is trivially satisfied. (A6") follows from the fact that STS satisfies (EA2). Finally, every region in STS corresponds uniquely to a region in $\mathsf{Estep}(STS)$, so we can deduce that $\mathsf{Estep}(STS)$ satisfies (A4') and (A5') from the fact that STS satisfies (EA4) and (EA5).

Once again, it is trivial to check that the image of a morphism in $\mathcal{ETS}$ is in fact a morphism in $\mathcal{EPN}ts$. □

Theorem 7.17 *The functors* Eseq *and* Estep *define a categorical equivalence between* $\mathcal{EPN}ts$ *and* $\mathcal{ETS}$.

Proof We have to show that the functor $\mathsf{Estep} \circ \mathsf{Eseq}$ is naturally isomorphic to the identity functor $\mathsf{id}_{\mathcal{EPN}ts}$ and, correspondingly, that $\mathsf{Eseq} \circ \mathsf{Estep}$ is naturally isomorphic to $\mathsf{id}_{\mathcal{ETS}}$.

In fact, we can prove something much stronger. Using the results we have proved so far, it is easy to show that for all $TS \in \mathcal{EPN}ts$, $\mathsf{Estep} \circ \mathsf{Eseq}(TS) = TS$ and for all $STS \in \mathcal{ETS}$, $\mathsf{Eseq} \circ \mathsf{Estep}(STS) = STS$. So, these two categories are not just equivalent, they are in fact isomorphic. □

So, we have established that the category of step transition systems defined by elementary net systems is equivalent to the category of sequential transition systems defined by them. This equivalence provides an alternate proof of the result, established combinatorially by Hoogeboom and Rozenberg in [4], that information about the concurrent behaviour of an elementary net system can always be *completely* recovered from a description of its sequential behaviour.

8 Asynchronous transition systems

Asynchronous transition systems were introduced by Bednarczyk [1] and Shields [15]. These are sequential transition systems equipped with information about concurrency in terms of an independence relation on the events. These transition systems are closely related to safe nets. In fact, in [19], Winskel and Nielsen establish a coreflection between a special class of asynchronous transition systems and safe nets. We now show that there is a coreflection between our category $\mathcal{SPN}ts$ of safe *PN*-transition systems and the category of asynchronous transtion systems defined in [19].

We begin by defining asynchronous transition systems. (The particular definition we use is adapted from [19]).

Definition 8.1 *An* asynchronous transition system *is a structure* $ATS = (Q, E, \Rightarrow, q_{in}, I)$ *such that*

- $(Q, E, \Rightarrow, q_{in})$ *is a sequential transition system (in the sense of Definition 5.1).*

- $I \subseteq E \times E$ *is an irreflexive, symmetric,* independence *relation satisfying the following four conditions:*

  (i) $e \in E$ *implies* $\exists q, q' \in Q.\ q \stackrel{\{e\}}{\Longrightarrow} q'$.

  (ii) $q \stackrel{u}{\Rightarrow} q'$ *and* $q \stackrel{u}{\Rightarrow} q''$ *implies* $q' = q''$.

  (iii) $e_1 I e_2$ *and* $q \stackrel{\{e_1\}}{\Longrightarrow} q_1$ *and* $q \stackrel{\{e_2\}}{\Longrightarrow} q_2$ *implies* $\exists q'.\ q_1 \stackrel{\{e_2\}}{\Longrightarrow} q'$ *and* $q_2 \stackrel{\{e_1\}}{\Longrightarrow} q'$.

  (iv) $e_1 I e_2$ *and* $q \stackrel{\{e_1\}}{\Longrightarrow} q_1$ *and* $q_1 \stackrel{\{e_2\}}{\Longrightarrow} q'$ *implies* $\exists q_2.\ q_1 \stackrel{\{e_2\}}{\Longrightarrow} q_2$ *and* $q_2 \stackrel{\{e_1\}}{\Longrightarrow} q'$.

Condition (i) in the definition above specifies that each event in E must be "used" somewhere in the system. The second condition stipulates that the system is deterministic. The third and fourth conditions capture the fact that I specifies pairs of events which are independent of each other and can thus occur concurrently if they are simultaneously enabled. Actually, the independence relation specifies more than just concurrency—for instance, two events may be independent without being enabled simultaneously anywhere in the system. We shall return to this point later.

Since we are dealing with sequential transition systems, for convenience we shall write $q \stackrel{e}{\Rightarrow} q'$ instead of $q \stackrel{\{e\}}{\Longrightarrow} q'$, where $e \in E$. We shall typically write $q \stackrel{u}{\Rightarrow} q'$ to indicate that u could either correspond to $\{e\}$ for some $e \in E$ or to the empty step $\mathtt{0}_E$.

Notice that the underlying sequential transition system is a step transition system satisfying axioms (A1) to (A3). So, we have idling transitions at each state and every state is reachable from the initial state. ((A3) is trivially satisfied in a sequential transition system).

Asynchronous transition systems are closely connected to safe nets. In [19], Winskel and Nielsen define a category $\mathbf{A}$ (which we shall call $\mathcal{A}ts$) consisting of asynchronous transition systems equipped with transition system morphisms which satisfy the additional requirement that the map on events preserve the independence relation. They then establish a coreflection between a subcategory of asynchronous transition systems, denoted $\mathbf{A}_0$ (which we shall call $\mathcal{A}ts_0$), and a category of safe Petri nets.

To identify the subcategory $\mathcal{A}ts_0$, they define a version of regions called *conditions*, using which they define axioms exactly like the regional axioms we impose on *PN*-transition systems.

Definition 8.2 *Let* $ATS = (Q, E, \Rightarrow, q_{in}, I)$ *be an asynchronous transition system. Its* conditions *are nonempty subsets* $b \subseteq\ \Rightarrow$ *such that*

(i) $(q, e, q') \in b$ *implies* $(q, \mathtt{0}_E, q) \in b$ *and* $(q', \mathtt{0}_E, q') \in b$.

(ii) *(a)* $(q_1, e, q_1') \in {}^\bullet b$ *and* $(q_2, e, q_2') \in\ \Rightarrow$ *implies* $(q_2, e, q_2') \in {}^\bullet b$

(b) $(q_1, e, q_1') \in b^\bullet$ *and* $(q_2, e, q_2') \in\ \Rightarrow$ *implies* $(q_2, e, q_2') \in b^\bullet$

where for $(q, e, q') \in\ \Rightarrow$ *we define*

$$(q, e, q') \in {}^\bullet b \stackrel{\text{def}}{=} (q, e, q') \notin b \text{ and } (q', \mathtt{0}_E, q') \in b,$$
$$(q, e, q') \in b^\bullet \stackrel{\text{def}}{=} (q, \mathtt{0}_E, q) \in b \text{ and } (q, e, q') \notin b \text{ and}$$
$${}^\bullet b^\bullet = {}^\bullet b \cup b^\bullet.$$

(iii) $(q_1, e_1, q_1') \in {}^\bullet b^\bullet$ *and* $(q_2, e_2, q_2') \in {}^\bullet b^\bullet$ *implies* $\neg e_1 I e_2$.

Let B be the set of conditions of ATS. For $e \in E$, define

$$\begin{array}{rcl} {}^\bullet e & = & \{b \in B \mid \exists q, q'.\ (q, e, q') \in b^\bullet\}, \\ e^\bullet & = & \{b \in B \mid \exists q, q'.\ (q, e, q') \in {}^\bullet b\},\ \textit{and} \\ {}^\bullet e^\bullet & = & {}^\bullet e \cup e^\bullet. \end{array}$$

Further, for $q \in Q$, define $M(q) = \{b \in B \mid (q, \mathsf{0}_E, q) \in b\}$.

Notice that a condition is really a subset of states and transitions. The information about the states is coded up in terms of the idling transitions.

We shall establish that the notion of a condition is equivalent to a natural notion of a region for this class of transition systems, defined as follows.

Definition 8.3 *Let $ATS = (Q, E, \Rightarrow, q_{in}, I)$ be an asynchronous transition system. A* region *of ATS is a pair of functions $r = (r_Q, r_E)$ where*
$r_Q : Q \to \{0,1\}$ *and*
$r_E : E \to (\{0,1\} \times \{0,1\})$ *such that*

(i) $\forall q \stackrel{e}{\Rightarrow} q'.\ {}^r e = 1$ *implies* $r_Q(q) = 1$ *and* $r_Q(q') = r_Q(q) + (e^r - {}^r e)$.

(ii) $\forall e, e' \in E.$ *If eIe' then* $({}^r e = 1$ *or* $e^r = 1)$ *implies* ${}^r e' = e'^r = 0$.

So, regions for asynchronous transition systems are very similar to the $\mathsf{0}/\mathsf{1}$-regions we define for safe *PN*-transition systems. The only additional requirement is that independent events have disjoint sets of pre and postregions. This reflects the intuition that two transitions in a safe net are independent provided their neighbourhoods are disjoint.

Definition 8.4 *Let $ATS = (Q, E, \Rightarrow, q_{in}, I)$ be an asynchronous transition system. Let B denote the set of conditions of ATS and let R denote the set of regions of ATS. We define two functions, $\hat{r} : B \to R$ and $\hat{b} : R \to B$.*

First, let $\hat{r} : B \to R$ be defined as follows.

$$\forall b \in B.\ \forall q \in Q.\ \hat{r}(b)(q) = \begin{cases} 1 & \textit{if } (q, \mathsf{0}_E, q) \in b \\ 0 & \textit{otherwise} \end{cases}$$

$$\forall b \in B.\ \forall e \in E.\ \hat{r}(b)(e) = \begin{cases} (1,0) & \textit{if } b \in {}^\bullet e \setminus e^\bullet \\ (0,1) & \textit{if } b \in e^\bullet \setminus {}^\bullet e \\ (1,1) & \textit{if } b \in {}^\bullet e^\bullet \\ (0,0) & \textit{otherwise} \end{cases}$$

Next, let $\hat{b} : R \to B$ be defined as follows.

$$\forall r \in R.\ \hat{b}(r) = \{(q, u, q') \mid \begin{array}{l} u = \mathsf{0}_E, q = q' \textit{ and } r(q) = 1,\ \textit{or} \\ u = \{e\}, r(e) = (0,0) \textit{ and } r(q) = r(q') = 1\} \end{array}$$

It is not hard to show the following result.

Proposition 8.5 *Let ATS be an asynchronous transition system, with B as its set of conditions and R as its set of regions.*

(i) $\forall b \in B.\ \hat{b}(\hat{r}(b)) = b$.

(ii) $\forall r \in R.\ \hat{r}(\hat{b}(r)) = r$.

We can now describe the subcategory $\mathcal{A}ts_0$ defined in [19]. Let $ATS = (Q, E, \Rightarrow, q_{in}, I)$ be an asynchronous transition system. Then $ATS \in \mathcal{A}ts_0$ if it satisfies the following two axioms, stated in terms of its set of conditions B:

Axiom ATS1 $M(q) = M(q')$ *implies* $q = q'$.

Axiom ATS2 ${}^\bullet e \subseteq M(q)$ *implies* $\exists q'.\ q \overset{e}{\Rightarrow} q'$, *for all* $q \in Q, e \in E$.

Clearly $M(q)$ is equivalent to the set of regions $\{r \in R \mid r(q) = 1\}$. And, ${}^\bullet e \subseteq M(q)$ is equivalent to saying $r(q) \geq {}^r e$ for all $r \in R$. So we can reformulate these two axioms in terms of regions and observe that they correspond to the axioms of separation and enabling for *PN*-transition systems (stated in the contrapositive form).

Axiom ATS1' $(\forall r \in R.\ r(q) = r(q'))$ *implies* $q = q'$.

Axiom ATS2' $(\forall r \in R.\ r(q) \geq {}^r e)$ *implies* $\exists q'.\ q \overset{e}{\Rightarrow} q'$, *for all* $q \in Q, e \in E$.

Actually, when defining the category $\mathcal{A}ts$ of *all* asynchronous transition systems in [19], Winskel and Nielsen do not assume that every state is reachable (as we have done here by requiring the underlying sequential transition system to satisfy axiom (A2)). The axiom for reachability is then introduced in [19] as a third axiom that an asynchronous transition system must satisfy to be in the subcategory $\mathcal{A}ts_0$. Since we are only interested in the subcategory $\mathcal{A}ts_0$ here, our presentation is equivalent to the one in [19].

The morphisms in $\mathcal{A}ts_0$ are transition system morphisms that preserve the independence relation. In other words, given two asynchronous transition systems $ATS_i = (Q_i, E_i, \Rightarrow_i, q^i_{in}, I_i)$, $i = 1, 2$, a morphism $f : ATS_1 \to ATS_2$ is a pair (f_Q, f_E) where:

- $f_Q : Q_1 \to Q_2$ is a total function such that $f_Q(q^1_{in}) = q^2_{in}$.
- $f_E : E_1 \rightharpoonup E_2$ is a partial function.
- $q \overset{e}{\Rightarrow}_1 q'$ implies $f_Q(q) \overset{f_E(e)}{\Longrightarrow}_2 f_Q(q')$.
- If $e_1 I_1 e'_1$ and $f_E(e_1), f_E(e'_1)$ are both defined, then $f_E(e_1) I_2 f_E(e'_1)$, for all $e_1, e'_1 \in E_1$.

We want to establish a relationship between our category of safe *PN*-transition systems $\mathcal{SPN}ts$ and the category $\mathcal{A}ts_0$. Actually, to describe the result we are after we have to make a slight restriction to our notion of a safe *PN*-transition system. Henceforth, we assume that if $TS = (Q, E, \to, q_{in})$ is a safe transition system, for every event $e \in E$ there is some transition $q \overset{u}{\to} q'$ in TS with $e \in u$—that is, every event has an occurrence. We shall discuss the need for this restriction at the end of this section.

We first prove a standard result which describes how the independence relation I in an asynchronous transition system specifies concurrency. It says that a sequence of actions which are pairwise independent corresponds to a concurrent step consisting of those actions. So, if such a sequence is enabled at a state in the system, all permutations of that sequence must also be enabled at that system and, furthermore, they should all lead to the same state as the original sequence.

Lemma 8.6 *Let* $ATS = (Q, E, \Rightarrow, q_{in}, I)$ *be an asynchronous transition system and* $\{e_1, e_2, \ldots, e_n\} \subseteq E, n \geq 2$, *be a pairwise independent subset of events in* E*—in other words,* $e_i I e_j$ *for all* $1 \leq i, j \leq n, i \neq j$.

If $q \overset{e_1}{\Rightarrow} q_1 \overset{e_2}{\Rightarrow} \ldots \overset{e_{n-1}}{\Longrightarrow} q_{n-1} \overset{e_n}{\Rightarrow} q''$ *then for all permutations* $\pi : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$, *there exist states* $\{q'_1, q'_2, \ldots, q'_{n-1}\}$ *such that* $q \overset{e_{\pi(1)}}{\Longrightarrow} q'_1 \overset{e_{\pi(2)}}{\Longrightarrow} \ldots \overset{e_{\pi(n-1)}}{\Longrightarrow} q'_{n-1} \overset{e_{\pi(n)}}{\Longrightarrow} q''$.

Proof The proof is straightforward, by induction on n, the number of pairwise independent events. The base case $n = 2$ corresponds to condition (iv) in the definition of an asynchronous transition system. We omit the details. $\square$

We can now describe a functor $\mathsf{AS} : \mathcal{A}ts_0 \to \mathcal{SPN}ts$. Given an asynchronous transition system $ATS = (Q, E, \Rightarrow, q_{in}, I)$, $\mathsf{AS}(ATS) = (Q, E, \to, q_{in})$, where

$$\begin{array}{rl} \to \ = \{(q, u, q') \mid & u = \{e_1, e_2, \ldots, e_n\} \subseteq E, \textit{ such that} \\ & e_i I e_j \textit{ for all } 1 \leq i, j \leq n, i \neq j \textit{ and} \\ & \exists q_1, q_2, \ldots, q_{n-1}.\ q \overset{e_1}{\Rightarrow} q_1 \overset{e_2}{\Rightarrow} \ldots \overset{e_{n-1}}{\Longrightarrow} q_{n-1} \overset{e_n}{\Rightarrow} q' \textit{ in } ATS\} \end{array}$$

Let $ATS_i = (Q_i, E_i, \Rightarrow_i, q^i_{in}, I_i)$, $i = 1, 2$, be a pair of asynchronous transition systems and let $f : ATS_1 \to ATS_2$ be a morphism in $\mathcal{A}ts_0$. Then the corresponding morphism $\mathsf{AS}(f) : \mathsf{AS}(ATS_1) \to \mathsf{AS}(ATS_2)$ is given by $\mathsf{AS}(f)_Q = f_Q$ and $\mathsf{AS}(f)_E = f_E$.

Lemma 8.7 AS *is a functor.*

Proof Let $ATS = (Q, E, \Rightarrow, q_{in}, I)$ be an asynchronous transition system in $\mathcal{A}ts_0$. To check that $\mathsf{AS}(ATS)$ is a safe *PN*-transition system, we just observe that every region of ATS is also a $\mathsf{0/1}$-region of $\mathsf{AS}(ATS)$. It then follows that $\mathsf{AS}(ATS)$ must satisfy the regional axioms (A4') and (A5') because ATS satisfies Axioms ATS1' and ATS2'.

Given a morphism $f : ATS_1 \to ATS_2$, where $ATS_i = (Q_i, E_i, \Rightarrow_i, q^i_{in}, I_i)$, $i = 1, 2$, and $\mathsf{AS}(ATS_i) = (Q_i, E_i, \to_i, q^i_{in})$, $i = 1, 2$, we have to check that $\hat{f} = \mathsf{AS}(f)$ satisfies condition (iii) in Definition 3.7.

In other words, if $q \overset{u}{\to}_1 q'$, we have to ensure that $\hat{f}(q) \overset{\hat{f}(u)}{\longrightarrow}_2 \hat{f}(q')$. Let $u = \{e_1, e_2, \ldots, e_n\}$. By the definition of $\to_1$, there must exist a sequence of actions $q \overset{e_1}{\Rightarrow}_1 q_1 \overset{e_2}{\Rightarrow}_1 \ldots \overset{e_{n-1}}{\Longrightarrow}_1 q_{n_1} \overset{e_n}{\Rightarrow}_1 q'$. Since f is a transition system morphism, it then follows that the f-image of this sequence exists in ATS_2. That is there is a sequence $f(q) \overset{f(e_1)}{\Longrightarrow}_2 f(q_1) \overset{f(e_2)}{\Longrightarrow}_2 \ldots \overset{f(e_{n-1})}{\Longrightarrow}_2 f(q_{n-1}) \overset{f(e_n)}{\Longrightarrow}_2 f(q')$. Since we know that the events in u are pairwise independent and f preserves independence, the events in $f(u)$ must be pairwise independent as well. It then follows, by the definition of $\to_2$, that $f(q) \overset{f(u)}{\longrightarrow}_2 f(q')$. $\square$

We now construct a functor SA which is left adjoint to the functor AS. Let $TS = (Q, E, \to, q_{in})$ be a safe *PN*-transition system. Define $\mathsf{SA}(TS) = (Q, E, \Rightarrow, q_{in}, I)$ where

- $\Rightarrow = \{(q, u, q') \mid q \overset{u}{\to} q' \textit{ and } |u| \leq 1\}$
- $I = \{(e_1, e_2), (e_2, e_1) \mid \exists q \in Q.\ q \overset{\{e_1, e_2\}}{\longrightarrow} q'\}$

Let $TS_i = (Q_i, E_i, \rightarrow_i, q_{in}^i)$, $i = 1, 2$, be a pair of safe *PN*-transition systems and $f : TS_1 \rightarrow TS_2$ be a morphism in $\mathcal{SPN}ts$. Then $\mathsf{SA}(f) : \mathsf{SA}(TS_1) \rightarrow \mathsf{SA}(TS_2)$ is given by $\mathsf{SA}(f)_Q = f_Q$ and $\mathsf{SA}(f)_E = f_E$.

Before proving that SA is a functor, it will be useful to prove a small result about $\mathsf{0/1}$-regions.

Proposition 8.8 *Let $TS = (Q, E, \rightarrow, q_{in})$ be a safe PN-transition system. Suppose that $e_1, e_2 \in E$ such that there exists a step $q \xrightarrow{\{e_1,e_2\}} q'$ in TS. Then, for all $r \in \mathcal{R}_{TS}^{0/1}$, if ${}^r e_1 = 1$ or ${e_1}^r = 1$ then ${}^r e_2 = {e_2}^r = 0$.*

Proof Suppose that $r \in \mathcal{R}_{TS}^{0/1}$ such that ${}^r e_1 = 1$. Then, since $r(q) \leq 1$, ${}^r e_2$ must be 0, otherwise the step $\{e_1, e_2\}$ would not be enabled at q. ${e_2}^r$ must be 0 as well. For, consider the state q_2 reached by the transition $q \xrightarrow{\{e_2\}} q_2$. (Such a transition must exist by Proposition 3.6). We have $r(q_2) = r(q) + ({e_2}^r - {}^r e_2)$. But $r(q) = 1$, since ${}^r e_1 = 1$ and e_1 is enabled at q. We also know that ${}^r e_2 = 0$. So, if ${e_2}^r$ were 1, we would have $r(q_2) = 2$, which is not possible.

On the other hand, if ${e_1}^r = 1$, we must have $r(q) = 0$. Then, we cannot have ${}^r e_2 = 1$, or e_2 would not be enabled at q. We cannot have ${e_2}^r = 1$ either because then $r(q') = 2$, which is not possible. $\square$

Lemma 8.9 SA *is a functor.*

Proof Let $TS = (Q, E, \rightarrow, q_{in})$ be a safe *PN*-transition system. We have to first check that $\mathsf{SA}(TS)$ is an asynchronous transition system. We basically have to check that conditions (i) to (iv) of Definition 8.1 hold.

Condition (i) holds because we have restricted the objects in $\mathcal{SPN}ts$ appropriately. Condition (ii) follows from Proposition 3.5 which says that *PN*-transition systems are deterministic.

Conditions (iii) and (iv) pertain to the independence relation. Condition (iii) says that $e_1 I e_2$ and $q \xRightarrow{e_1} q_1$ and $q \xRightarrow{e_2} q_2$ implies $\exists q'.\ q_1 \xRightarrow{e_2} q'$ and $q_2 \xRightarrow{e_1} q'$. Since $e_1 I e_2$, we know that $q'' \xrightarrow{\{e_1,e_2\}} q'''$ somewhere in TS. By the previous proposition, the pre and postregions of e_1 and e_2 are disjoint, so if both e_1 and e_2 are enabled at a state q, then (by axiom (A5')) the step $\{e_1, e_2\}$ must be enabled as well. The result then follows from Proposition 3.6, which asserts that all steps in a *PN*-transition system can be broken up into substeps in a consistent way.

Condition (iv) follows by a similar argument.

To verify that $\mathsf{SA}(TS)$ satisfies axioms ATS1' and ATS2', notice that by the previous proposition, any region $r \in \mathcal{R}_{TS}^{0/1}$ would correspond to a region in $\mathsf{SA}(TS)$. Since TS satisfies axioms (A4') and (A5') with respect to regions in $\mathcal{R}_{TS}^{0/1}$, it follows that the corresponding regions in $\mathsf{SA}(TS)$ are sufficient to satisfy axioms ATS1' and ATS2'.

We then have to verify that for any morphism $f : TS_1 \rightarrow TS_2, i = 1, 2$, $\mathsf{SA}(f) = \hat{f}$ is a morphism from $\mathsf{SA}(TS_1)$ to $\mathsf{SA}(TS_2)$, where $\mathsf{SA}(TS_i) = (Q_i, E_i, \Rightarrow_i, q_{in}^i, I_i), i = 1, 2$. We basically have to verify that if $e_1 I_1 e_2$ and both $\hat{f}(e_1)$ and $\hat{f}(e_2)$ are defined, then $\hat{f}(e_1) I_2 \hat{f}(e_2)$, where $\hat{f}(e_1) = f(e_1)$ and $\hat{f}(e_2) = f(e_2)$. If $e_1 I_1 e_2$, then, by the definition of I_1, we know that $q \xrightarrow{\{e_1,e_2\}}_1 q'$ somewhere in TS_1. This implies that $f(q) \xrightarrow{\{f(e_1),f(e_2)\}}_2 f(q')$ in TS_2. So, if $f(e_1)$ and $f(e_2)$ are both defined, then, by the definition of I_2, we have

$f(e_1)I_2f(e_2)$ and we are done. $\square$

Theorem 8.10 *The functor* SA *is left adjoint to the functor* AS*. The unit of the adjunction is a natural isomorphism.*

Proof Let $TS \in \mathcal{SPN}ts$ and $ATS \in \mathcal{A}ts_0$. Suppose that $f : TS \to \mathsf{AS}(ATS)$ is a morphism. Then, since TS and $\mathsf{SA}(TS)$ have the same underlying sets of states and events and $\mathsf{AS}(ATS)$ and ATS have the same underlying sets of states and events, it is fairly straightforward to see that $\hat{f} : \mathsf{SA}(TS) \to ATS$ is also a morphism, where $\hat{f}_Q = f_Q$ and $\hat{f}_E = f_E$. Conversely, if $g : \mathsf{SA}(TS) \to ATS$ is a morphism, we can show that $\overline{g} : TS \to \mathsf{AS}(ATS)$ is also a morphism, where $\overline{g}_Q = g_Q$ and $\overline{g}_E = g_E$. Further, $\overline{\hat{f}} = f$ and $\hat{\overline{g}} = g$ for all morphisms $f \in \mathcal{SPN}ts$ and $g \in \mathcal{A}ts_0$. This establishes a bijection between $\mathbf{Hom}(TS, \mathsf{AS}(ATS))$ and $\mathbf{Hom}(\mathsf{SA}(TS), ATS)$. It is not difficult to show that this bijection is natural in both $\mathcal{SPN}ts$ and $\mathcal{A}ts_0$, thereby establishing the adjunction.

It is also not difficult to show that the unit $\eta_{TS} : TS \to \mathsf{AS} \circ \mathsf{SA}(TS)$ is an isomorphism for all $TS \in \mathcal{SPN}ts$. $\square$

So we have established a coreflection between our category of safe *PN*-transition systems and the category of asynchronous transition systems $\mathcal{A}ts_0$ defined by Winskel and Nielsen in [19].

The reason that this correspondence is a coreflection and not a categorical equivalence has to do with the nature of the independence relation. In an asynchronous transition system two events can be independent without ever being enabled simultaneously to give rise to a concurent step. When representing an asynchronous transition system as a safe *PN*-transition system, we lose information about these “unused” independences. These “unused” independences can be regarded as providing some “structural” information about the system which may not be directly detectable in its concurrent behaviour. For example, two events being independent of each other could denote the fact that they occur at different locations and do not interfere with each other. Under such an interpretation, one comes across very natural examples of asynchronous transition systems in which independent events are never simultaneously enabled (see, for instance, [10]).

It is not difficult to show that our category of safe *PN*-transition systems is equivalent to a subcategory of $\mathcal{A}ts_0$ whose objects satisfy the additional constraint that for every pair $(e_1, e_2) \in I$, there is a state q where both e_1 and e_2 are enabled.

We also pointed out a mismatch between the definition of safe *PN*-transition systems we use in this section and the one we proposed in the Section 6. The additional assumption we have made here is that every event in E have an occurrence. This is required because asynchronous transition systems in the category $\mathcal{A}ts_0$ satisfy this restriction.

However, this restriction on asynchronous transition systems is a consequence of how conditions are defined. For an asynchronous transtition system $ATS = (Q, E, \Rightarrow, q_{in}, I)$, it is easy to see that one cannot define a condition $b \in {}^\bullet e$ for any $e \in E$ which does not occur in ATS. This is because conditions are defined as subsets of transitions which are present in the system. Since we cannot find any $b \in {}^\bullet e$ for an event e which does not occur, Axiom ATS2 would then require e to be enabled at every state in the system, which is a contradiction. So, for asynchronous transition systems in $\mathcal{A}ts_0$, condition (i) in Definition 8.1 is implied by Axiom ATS2.

However, in the generalized set up of regions, it is possible to define "disabling regions" which take the value 0 at all states but which are the preregion of some event e, thereby ensuring that e is never enabled. The regional version of the second axiom, Axiom ATS2', would clearly permit such permanently disabled events to be part of the specification of the system.

It is not difficult to generalize Definition 8.1 by dropping condition (i) and building a slightly larger category $\mathcal{A}ts'_0$ satisfying Axioms ATS1' and ATS2'. It then turns out that the coreflection we have described here goes through between the category $\mathcal{SPN}ts$ as originally defined in the previous section and the more generous category $\mathcal{A}ts'_0$.

9 Safe nets revisited

We showed in Section 6 that there is a coreflection between $\mathcal{SPN}ts$, the category of safe PN-transition systems and $\mathcal{SN}et$, our category of safe nets. Then, we showed that there is also a coreflection between $\mathcal{SPN}ts$ and $\mathcal{A}ts_0$, where $\mathcal{A}ts_0$ is a subcategory of the $\mathcal{A}ts$, the category of asynchronous transition systems. In [19], Winskel and Nielsen have established a coreflection between $\mathcal{A}ts_0$ and a category of safe nets which we shall call $\mathcal{WN}et$.

Unfortunately, the category $\mathcal{WN}et$ is not the same as the category $\mathcal{SN}et$ we have defined here. However, we show now that there is an adjunction between these two categories. Further, we can establish a coreflection between the subcategories of $\mathcal{SN}et$ and $\mathcal{WN}et$ consisting of only *saturated* nets, where saturated nets are those nets which arise out of the regional construction in going from transition systems to nets.

The only difference between $\mathcal{WN}et$ and $\mathcal{SN}et$ is that the morphisms of $\mathcal{WN}et$ are slightly stricter than those of $\mathcal{SN}et$.

Let us briefly recall the definition of the category $\mathcal{SN}et$. The objects of $\mathcal{SN}et$ are safe nets, as given by Definition 6.1. Morphisms between safe nets are the same as those between general nets, as given in Definition 2.2. However, since the definition of a morphism becomes slightly simpler when restricted to safe nets, we pause to spell it out in detail.

Let $PN_i = (S_i, T_i, W_i, M^i_{in})$, $i = 1, 2$, be a pair of safe nets. An $\mathcal{SN}et$-morphism $\phi : PN_1 \to PN_2$ is a pair $\phi = (\phi_S, \phi_T)$ where:

(i) $\phi_S : S_2 \rightharpoonup S_1$ is a partial function.

(ii) $\phi_T : T_1 \rightharpoonup T_2$ is a partial function.

(iii) $\forall s_1 \in S_1.\ \forall s_2 \in S_2.$ If $s_1 = \phi_S(s_2)$ then $M^1_{in}(s_1) = M^2_{in}(s_2)$.

(iv) $\forall t_1 \in T_1.$ If $\phi_T(t_1)$ is undefined then $\phi_S^{-1}({}^\bullet t_1) = \phi_S^{-1}(t_1{}^\bullet) = \emptyset$.

(v) $\forall t_1 \in T_1.$ If $\phi_T(t_1) = t_2$ then $\phi_S^{-1}({}^\bullet t_1) = {}^\bullet t_2$ and $\phi_S^{-1}(t_1{}^\bullet) = t_2{}^\bullet$.

The category $\mathcal{WN}et$ also has as its objects safe nets, like $\mathcal{SN}et$. However, the morphisms are slightly stricter than those of $\mathcal{SN}et$. $\phi : PN_1 \to PN_2$ is a morphism in $\mathcal{WN}et$ if ϕ is an $\mathcal{SN}et$-morphism, and, in addition, $\phi_S^{-1}(M^1_{in}) = M^2_{in}$ (where, abusing notation, $M^i_{in}, i = 1, 2$ denote the subsets of S_1 and S_2 which are marked initially in PN_1 and PN_2 respectively).

So, the essential difference between a $\mathcal{WN}et$-morphism and an $\mathcal{SN}et$-morphism is that in a $\mathcal{WN}et$-morphism ϕ_S is a *total* function when restricted to those places marked initially in the second net.

Clearly, every $\mathcal{WN}et$-morphism is also an $\mathcal{SN}et$-morphism. So $\mathcal{WN}et$ is a subcategory of $\mathcal{SN}et$, though not a full subcategory.

It turns out that we can construct a left adjoint to the inclusion functor from $\mathcal{WN}et$ to $\mathcal{SN}et$ (though this will not constitute a reflection because $\mathcal{WN}et$ is not a full subcategory of $\mathcal{SN}et$ [6]).

In going from $\mathcal{SN}et$ to $\mathcal{WN}et$, we have, in general, to make an $\mathcal{SN}et$-morphism into a $\mathcal{WN}et$-morphism. In other words, we have to convert the map on the initial marking from a partial function to a total function. A standard way to convert a partial function to a total function is to augment the range of the function with a special "undefined" value. Similarly, here we augment the net that is the source of the morphism with an isolated marked place.

Formally, define a functor $\mathsf{SNWN} : \mathcal{SN}et \to \mathcal{WN}et$ as follows.

- For $PN = (S, T, W, M_{in}) \in \mathcal{SN}et$, $\mathsf{SNWN}(PN) = (S \uplus \{\breve{s}\}, T, W', M_{in}')$, where:
  - $\breve{s} \notin S$ (we use $\uplus$ to denote disjoint union).
  - $\forall s \in S \uplus \{\breve{s}\}.\ \forall t \in T.\ W'(s,t) = \begin{cases} W(s,t) & \textit{if}\ s \in S \\ 0 & \textit{otherwise} \end{cases}$
  - $\forall s \in S \uplus \{\breve{s}\}.\ \forall t \in T.\ W'(t,s) = \begin{cases} W(t,s) & \textit{if}\ s \in S \\ 0 & \textit{otherwise} \end{cases}$
  - ${M_{in}}'(\breve{s}) = 1$ and $\forall s \in S.\ {M_{in}}'(s) = M_{in}(s)$.
- Let $\phi : PN_1 \to PN_2$ be a morphism between PN_1 and PN_2, where $PN_i = (S_i, T_i, W_i, M_{in}^i)$, $i = 1, 2$. Then, $\mathsf{SNWN}(\phi) = \phi' : \mathsf{SNWN}(PN_1) \to \mathsf{SNWN}(PN_2)$ is given as follows, where $\mathsf{SNWN}(PN_i) = (S_i \uplus \{\breve{s}_i\}, T_i, W_i', {M_{in}^i}')$, $i = 1, 2$.
  - $\forall t \in T_1.\ \phi'(t) = \phi(t)$.
  - $\forall s \in S_2 \uplus \{\breve{s}_2\}.\ \phi'(s) = \begin{cases} \breve{s}_1 & \textit{if}\ (s = \breve{s}_2)\ \textit{or} \\ (s \in S_2, {M_{in}^2}'(s) = 1\ \textit{and} & \\ \phi(s)\ \textit{is undefined}) & \\ \phi(s) & \textit{otherwise} \end{cases}$

Theorem 9.1 *$\mathsf{SNWN} : \mathcal{SN}et \to \mathcal{WN}et$ is left adjoint to the inclusion functor.*

Proof Let $PN_1 = (S_1, T_1, W_1, M_{in}^1)$ and $PN_2 = (S_2, T_2, W_2, M_{in}^2)$ be two safe nets. We shall establish a bijection between $\mathbf{Hom}(PN_1, PN_2)$ and $\mathbf{Hom}(\mathsf{SNWN}(PN_1), PN_2)$, where $\mathsf{SNWN}(PN_1) = (S_1 \uplus \breve{s}_1, T_1, W_1', {M_{in}^1}')$.

We first define a map $\lambda : \mathbf{Hom}(PN_1, PN_2) \to \mathbf{Hom}(\mathsf{SNWN}(PN_1), PN_2)$. Suppose that $\phi : PN_1 \to PN_2 \in \mathbf{Hom}(PN_1, PN_2)$. Define $\lambda(\phi) : \mathsf{SNWN}(PN_1) \to PN_2$ as follows.

- $\forall t \in T_1.\ \lambda(\phi)(t) = \phi(t)$.
- $\forall s \in S_2.\ \lambda(\phi)(s) = \begin{cases} \breve{s}_1 & \textit{if}\ M_{in}^2(s) = 1\ \textit{and}\ \phi(s)\ \textit{undefined} \\ \phi(s) & \textit{otherwise} \end{cases}$

Next we define a map $\mu : \mathbf{Hom}(\mathsf{SNWN}(PN_1), PN_2) \to \mathbf{Hom}(PN_1, PN_2)$. Let $\psi : \mathsf{SNWN}(PN_1) \to PN_2$. Then $\mu(\psi) : PN_1 \to PN_2$ is given as follows.

- $\forall t \in T_1.\ \mu(\psi)(t) = \psi(t)$.
- $\forall s \in S_2.\ \mu(\psi)(s) = \begin{cases} \mathit{undefined} & \mathit{if}\ \psi(s) = \breve{s}_1 \\ \psi(s) & \mathit{otherwise} \end{cases}$

It is straightforward to show that $\mu(\lambda(\phi)) = \phi$ and $\lambda(\mu(\psi))$ for all $\phi \in \mathbf{Hom}(PN_1, PN_2)$ and $\psi \in \mathbf{Hom}(\mathsf{SNWN}(PN_1), PN_2)$. It is not difficult to show that this bijection is natural in $\mathcal{SN}et$ and $\mathcal{WN}et$, and we are done.

$\square$

As we mentioned at the beginning of this section, we can establish a slightly stronger result when we look at the safe nets actually arising out of the regional construction from transition systems.

In Section 6, we have described a functor STN which associates a net $\mathsf{STN}(TS)$ with each safe *PN*-transition system TS. Following [12], we can call such a net *saturated*, because it contains all possible places which are consistent with the behaviour described by TS. A crucial feature of the construction is that these saturated nets have *no* isolated places because we only use non-trivial regions in the construction of the saturated net.

In [19], Winskel and Nielsen describe a similar functor, which we can call AWN, going from $\mathcal{A}ts_0$ to $\mathcal{WN}et$. Once again, given an asynchronous transition system ATS, $\mathsf{AWN}(ATS)$ will be a saturated net. Here, saturation is with respect to the underlying sequential behaviour of ATS as well as the independence relation I specified by ATS. An important difference between the construction described in [19] and the construction we describe in Section 6 is that the construction in [19] adds trivial regions as well.

The reason why the construction in [19] also includes trivial regions is to do with the stricter notion of a net morphism in the category $\mathcal{WN}et$. Notice that it is always possible to define a trivial morphism between two transition systems in which the map on events is empty. Corresponding to this, in the category $\mathcal{SN}et$ it is always possible to define a trivial morphism between two nets where the map on places and the map on transitions are both empty. However, in the category $\mathcal{WN}et$, such trivial maps do not always exist, because of the strong condition on how the initial markings have to be related. If the net that is the source of a morphism has an isolated marked place, however, such a trivial map can also be defined in $\mathcal{WN}et$. Hence, to transport the trivial maps between asynchronous transition systems in $\mathcal{A}ts_0$ faithfully to trivial maps between the associated nets in $\mathcal{WN}et$, it is essential that the functor AWN create isolated places.

Let $\mathcal{S}at\mathcal{SN}et$ be the subcategory of $\mathcal{SN}et$ where for every net $PN \in \mathcal{S}at\mathcal{SN}et$, there is a safe *PN*-transition system $TS \in \mathcal{SPN}ts$ such that PN is isomorphic to $\mathsf{STN}(TS)$. Similarly, let $\mathcal{S}at\mathcal{WN}et$ be the subcategory of $\mathcal{WN}et$ such that for every net $PN \in \mathcal{S}at\mathcal{WN}et$, there is an asynchronous transition system $ATS \in \mathcal{A}ts_0$ such that PN is isomorphic to $\mathsf{AWN}(ATS)$.

The functor $\mathsf{SNWN} : \mathcal{SPN}ts \to \mathcal{WN}et$ restricts to a functor from $\mathcal{S}at\mathcal{SN}et$ to $\mathcal{S}at\mathcal{WN}et$, which we shall again call SNWN, for convenience.

Going in the opposite direction, starting with a net $PN \in \mathcal{S}at\mathcal{WN}et$, we can first apply the functor WNA, which is the right adjoint of AWN, to obtain an asynchronous

$$q_{in} \xrightarrow{e_1} q_1 \xrightarrow{e_2} q_2 \xrightarrow{e_3} q_3$$

Figure 3:

transition system corresponding to PN. Then by applying AS and STN we obtain a net in $\mathcal{S}at\mathcal{SN}et$.

Theorem 9.2 *The functor* $\mathsf{SNWN} : \mathcal{S}at\mathcal{SN}et \to \mathcal{S}at\mathcal{WN}et$ *is left adjoint to the functor* $\mathsf{STN} \circ \mathsf{AS} \circ \mathsf{WNA} : \mathcal{S}at\mathcal{WN}et \to \mathcal{S}at\mathcal{SN}et$*. The unit of the adjunction is a natural isomorphism.*

Proof The proof is tedious but straightforward, based on several results we have proved already, so we omit the details. □

So, even at the level of saturated nets, we only get a coreflection and not a categorical equivalence between $\mathcal{S}at\mathcal{SN}et$ and $\mathcal{S}at\mathcal{WN}et$. This is because a safe net that is saturated with respect its description as an asynchronous transition system need not be saturated with respect to its description as a safe PN-transition system. So, the "obvious" functor from $\mathcal{S}at\mathcal{WN}et$ to $\mathcal{S}at\mathcal{SN}et$ which just removes the isolated places will not, in general, yield a net in $\mathcal{S}at\mathcal{SN}et$ at all.

Consider, for example, the simple transition system TS in Figure 3. If we view this as a safe PN-transition system, the corresponding saturated net $\mathsf{STN}(TS)$ would have a place s such that $s \in e_1{}^{\bullet}$ and $s \in {}^{\bullet}e_3$.

However, we can make TS into an asynchronous transition system in more than one way. The obvious asynchronous transition system version of TS has the empty independence relation. But, we can also specify that e_1 and e_3 are independent. This would mean that in the net $\mathsf{AWN}(TS)$, the neighbourhoods of e_1 and e_3 would be disjoint, hence ruling out the place s connecting e_1 to e_3 which is present in $\mathsf{STN}(TS)$.

Another way of comparing the categories $\mathcal{SN}et$ and $\mathcal{WN}et$ is to examine subcategories of $\mathcal{SN}et$ and $\mathcal{WN}et$ where we saturate the nets in both subcategories with respect to the *same* class of transition systems.

First, we can relate $\mathcal{SPN}ts$ and $\mathcal{WN}et$ by functors $\mathsf{STWN} : \mathcal{SPN}ts \to \mathcal{WN}et$ and $\mathsf{SWNT} : \mathcal{WN}et \to \mathcal{SPN}ts$ in much the same way as we related $\mathcal{SPN}ts$ and $\mathcal{SN}et$ by STN and SNT, except that STWN constructs places corresponding to both trivial and non-trivial regions. It is then easy to establish a coreflection between STWN and SWNT. We can then look at the category $\mathcal{S}at\mathcal{WN}et'$, consisting of safe nets which are isomorphic to $\mathsf{STWN}(TS)$ for some $TS \in \mathcal{SPN}ts$. It is not difficult to show that $\mathcal{S}at\mathcal{SN}et$ and $\mathcal{S}at\mathcal{WN}et'$ are categorically equivalent, where the functor from $\mathcal{S}at\mathcal{SN}et$ to $\mathcal{S}at\mathcal{WN}et'$ is SNWN as before and the functor in the opposite direction is the one which strips off isolated places from a net.

In a similar way, we can define a coreflection between $\mathcal{A}ts_0$ and $\mathcal{SN}et$ in terms of functors $\mathsf{AN} : \mathcal{A}ts_0 \to \mathcal{SN}et$ and $\mathsf{NA} : \mathcal{SN}et \to \mathcal{A}ts_0$, where AN constructs places corresponding to only non-trivial regions. We can then look at the category $\mathcal{S}at\mathcal{SN}et'$ consisting of safe nets which are isomorphic to $\mathsf{AN}(ATS)$ for some $ATS \in \mathcal{A}ts_0$. It turns out that $\mathcal{S}at\mathcal{SN}et'$ and $\mathcal{S}at\mathcal{WN}et$ are categorically equivalent.

So, provided we use the same notion of saturation in both $\mathcal{SN}et$ and $\mathcal{WN}et$, we end up with equivalent subcategories of saturated nets.

10 Discussion

In this paper we have shown how to define subcategories of *PN*-transition systems which describe the behaviour of safe nets and elementary net systems. This is achieved by "tuning" the notion of a region appropriately. It then turns out that the coreflection established between the categories $\mathcal{PN}ts$ and $\mathcal{PN}et$ in [9] can be restricted to coreflections between the corresponding subcategories of these two categories.

We have examined the relationship between sequential and step transition systems in the setting of *PN*-transition systems. In general, there is a coreflection between sequential *PN*-transition systems and "normal" *PN*-transition systems with steps. However, when we restrict our attention to transition systems describing the behaviour of elementary net systems, the subcategories of sequential and step transition systems are equivalent. This shows that for elementary net systems, all information about concurrency can be recovered by examining the sequential behaviour of the system.

We have also established a coreflection between safe *PN*-transition system and asynchronous transition systems. This result shows that asynchronous transition systems are, in a sense, a more concrete model of behaviour than step transition systems because the independence relation can provide "structural" information about a system which cannot be inferred directly from an examination of its behaviour.

A brief remark is in order about the way we have described the correspondence between step transition systems and Petri nets. We have chosen to present the relationship between step transition systems and different classes of Petri nets directly in terms of coreflections, by identifying special categories of step transition systems corresponding to each class of nets. Instead, we could have followed the approach adopted by Winskel and Nielsen in [19] and first established the existence of left adjoints for the natural functors from nets to step transition systems and then "cut down" the adjunctions to coreflections by restricting the class of step transition systems under consideration.

In a sense, it would have been more uniform to follow the approach of [19], because the right adjoints in all the coreflections we establish between transition systems and nets correspond to the *same* functor—the one taking a net to its "step" marking diagram. We have chosen to directly present the results in terms of coreflections because these coreflections denote, in our opinion, stronger and more relevant relationships between the two classes of models than those represented by simple adjunctions. An adjunction between step transition systems and a particular class of nets describes the minimal way of "massaging" a given step transition system so that it represents the behaviour of some net from the class of nets under consideration. On the other hand, if we have a coreflection between a class of step transition systems and a class of nets, we are guaranteed that the class of transition systems we are considering captures *precisely* the behaviours describable by the class of nets we are interested in.

It is natural to ask what we achieve by establishing these formal relationships between different models of concurrency. One motivation for establishing such relationships is that they provide a basis for translating results from one model to another. This gives us the freedom to work within whichever framework is most convenient and "automatically" obtain connections to other approaches.

For instance, to obtain a non-interleaved model for a process calculus such as CCS [8], it is intuitively easier to enrich the standard interleaved transition system semantics to obtain a more faithful representation of concurrency, rather than providing a semantics directly in terms of nets [2, 13] or event structures [18]. Thus, using a very simple extension of the standard operational semantics for CCS, we can provide a non-interleaved semantics for a rich subclass of the language in terms of asynchronous transtition systems from the subcategory $\mathcal{A}ts_0$ [10]. This implies, by the results connecting $\mathcal{A}ts_0$ and $\mathcal{WN}et$, that we automatically obtain a net semantics for this language.

The other natural question that one may ask is why we work within the framework of category theory. One reason is that it provides a convenient mathematical language to phrase the kinds of correspondences we would like to describe. For instance, coreflections succintly capture the idea of one model being embedded in another.

The other advantage of working with categories is that many interesting operations that one defines on these models can be captured as universal categorical constructions. For instance, parallel composition corresponds to a notion of categorical product, while nondeterministic choice can be described in terms of coproducts. Thus, by relating categories of models, we can also compare how these constructions behave in different models. This issue is discussed in some detail in [19], where a number of relationships between models for concurrency are established in a categorical setting, spanning the spectrum of linear-time, branching-time and partial-order approaches to modelling the behaviour of concurrent systems.

We conclude by pointing out a major issue which we have ignored in our study—that of labelling. In the theory of Petri nets, abstraction is achieved by adding a set of labels which can be associated with the underlying events of the system. This is crucial for using nets to provide, say, a semantics for CCS-like langages. In [19], labelling is introduced into the categorical treatment of different models of concurrency by means of fibrations and cofibrations. Though they point out some problems in defining these constructions over categories of nets, it does not seem to prevent the coreflection between unlabelled transition systems and unlabelled nets from being extended to the corresponding labelled categories. So, while we have not explicitly handled labelling in our framework, we are confident that we can follow the route set out in [19] without too much difficulty.

Acknowledgment We thank Mogens Nielsen for helpful comments.

A Appendix

We fix some terminology and notation regarding multisets.

Definition A.1 *Let A be a set.*

- *A* multiset *u over A is a function $u : A \to \mathbf{N}_0$, where $\mathbf{N}_0$ is the set of natural numbers $\{0, 1, 2, \ldots\}$. The set of all multisets over A is denoted by $MS(A)$.*
- *For $u \in MS(A)$, let $|u|$, the* size of *u, be given by $\sum_{a \in A} u(a)$. u is* finite *iff $|u|$ is finite. The set of all finite multisets over A is denoted by $MS_{fin}(A)$.*

- *The* empty multiset *over A is the unique function $0_A : A \to \mathbf{N}_0$ such that $\forall a \in A.\ 0_A(a) = 0$.*

- *Let $u, v \in MS(A)$. Then u is a* submultiset *of v, written $u \subseteq_{MS} v$, in case $u(a) \leq v(a)$ for all $a \in A$.*

Thus, if u is a multiset over A, for each $a \in A$, $u(a)$ is the number of occurrences of a in u. Abusing notation, we shall write $a \in u$ to signify that $u(a) \geq 1$. For simplicity, we shall usually write out multisets as sets with multiplicities — for instance, if $a, b \in A$, then $\{a, a, b\}$ denotes the multiset u over A which assigns $u(a) = 2$, $u(b) = 1$ and $u(c) = 0$ for all $c \in A$ such that $c \neq a$ and $c \neq b$.

Multisets can be added and subtracted pointwise — if u and v are two multisets over A, then $u + v$ and $u - v$ are defined as follows:

- $\forall a \in A.\ (u + v)(a) = u(a) + v(a)$.

- If $v \subseteq_{MS} u$ then $\forall a \in A.\ (u - v)(a) = u(a) - v(a)$.

Given a partial function $f : A \rightharpoonup B$ between sets, f can be extended in a natural way to a (total) function $\hat{f} : MS_{fin}(A) \to MS_{fin}(B)$ as follows:

$$\forall u \in MS_{fin}(A).\ \forall b \in B.\ \hat{f}(u)(b) = \sum_{\{a \in A \mid f(a) = b\}} u(a)$$

By convention, $\hat{f}(u) = 0_B$ in case $f(a)$ is undefined for all $a \in u$.

For convenience, we shall denote both f and its extension $\hat{f}$ to multisets by f.

References

[1] M.A. Bednarczyk: Categories of asynchronous systems, PhD Thesis, Report 1/88, Computer Science, University of Sussex (1988).

[2] P. Degano, R. de Nicola, U. Montanari: A distributed operational semantics for CCS based on condition/event systems, *Acta Informatica*, **26**, 59–91 (1988).

[3] A. Ehrenfeucht, G. Rozenberg: Partial 2-structures; Part II: State spaces of concurrent systems, *Acta Informatica*, **27**, 348–368 (1990).

[4] H.J. Hoogeboom, G. Rozenberg: Diamond properties of elementary net systems, *Fundamenta Informaticae*, **XIV**, 287–300 (1991).

[5] K. Lodaya, R. Ramanujam, P.S. Thiagarajan: A logic for distributed transition systems, *Springer Lecture Notes in Computer Science*, **354**, 508–522 (1989).

[6] S. Mac Lane: *Categories for the working mathematician*, Springer-Verlag, New York/Berlin (1971).

[7] A. Mazurkiewicz: Basic notions of trace theory, *Springer Lecture Notes in Computer Science*, **354**, 285–363 (1989)

[8] R. Milner: *Communication and Concurrency*, Prentice-Hall, London (1989).

[9] M. Mukund: A transition system characterization of Petri nets, *Report TCS-91-2*, School of Mathematics, SPIC Science Foundation, Madras, India (1991).

[10] M. Mukund, M. Nielsen: CCS, locations and asynchronous transition systems, *Report DAIMI-PB-395*, Computer Science Department, Aarhus Univeristy, Aarhus, Denmark (1992).

[11] M. Nielsen, G. Plotkin, G. Winskel: Petri nets, event structures and domains, part I, *Theoretical Computer Science*, **13**, 45–57 (1981).

[12] M. Nielsen, G. Rozenberg, P.S. Thiagarajan: Elementary transition systems, *Theoretical Computer Science*, **96**, 1, 3-33 (1992).

[13] E.-R. Olderog: *Nets, Terms and Formulas*, Cambridge University Press, Cambridge (1991).

[14] W. Reisig: *Petri nets*, Springer-Verlag, Berlin (1985).

[15] M.W. Shields: Concurrent machines, *Computer Journal*, **28**, 449–465 (1985).

[16] P.S. Thiagarajan: Elementary nets systems, *Springer Lecture Notes in Computer Science*, **254**, 60–94 (1987).

[17] G. Winskel: Event structures, *Springer Lecture Notes in Computer Science*, **255**, 325–392 (1987).

[18] G. Winskel: Event structure semantics of CCS and related languages, *Springer Lecture Notes in Computer Science*, **140**, 561–577 (1982).

[19] G. Winskel, M. Nielsen: Models for concurrency, (to appear in S. Abramsky, D.M. Gabbay, T.S.E. Maibaum eds. *Handbook of Logic in Computer Science*).