Non-Pure Nets and Their Transition Systems

Marta Pietkiewicz-Koutny and Alex Yakovlev*
Department of Computing Science
University of Newcastle upon Tyne
Newcastle upon Tyne NE1 7RU England

July 11, 1995

Abstract

This paper extends the theory of regions developed by Nielsen, Rozenberg and Thiagarajan within a set-theoretic framework, to accommodate the class of non-pure nets and their transition systems. Those are called semi-elementary nets and semi-elementary transition systems, respectively. The main motivation of such an extension is practical, the need to model asynchronous hardware structures, where certain events happen only when some conditions (these are called co-conditions) are true but without changing the state of these conditions. One of the applications of this theory is synthesis of Petri net models from state-based specifications. As an example we present a Petri net model of control of a counterflow pipeline for Sproull's asynchronous processor. This control was originally specified as a transition system which did not satisfy elementarity axioms of Nielsen et al.

Keywords: *asynchronous systems, concurrency, elementary nets, Petri nets, regions, transition systems.*

1 Introduction

Petri nets are known to be a powerful language for modelling digital systems with concurrent behaviour. They are usually employed as a syntax-level or system-level model. The user of such a model, or better say the system designer, can specify a system expressing causality, concurrency, choice and conflicts in terms of events and local conditions between events. Expressing all those and other paradigms locally cannot however guarantee that some global properties of the behaviour intended by the designer will be automatically fulfilled in the system. Such properties, both safety and progress ones, are typically characterised at a semantical or behavioural level, the level of states and transitions. Transition systems present a language used for explaining the operational semantics of systems behaviour. It is often the case that the designer prefers to use a transition system to capture the intended behaviour, as for example was in [1, 2]. Here,

*Supported by EPSRC grant GR/J52327.

the key aspects of synchronisation between two pipelines with data flowing in two opposite directions were much easier to define in the form of a state graph. Existing methods and tools for asynchronous hardware design are based on Petri nets [3, 4, 5, 6]. An important task would therefore be to synthesise a Petri net model from a state-based description. This and other examples of circuit synthesis, which involve transformations between concurrency models and Petri nets, have been presented in [7]. Most of them require passing through the intermediate semantic level, the level of transition systems.

It has been stressed in [7] that the class of Petri nets normally used for hardware synthesis is restricted to 1-safe nets. Such nets are closely related to *elementary nets*, whose transition systems have been studied in pioneering work of Nielsen et al. [8]. This work presented an elegant idea of a *region*, that is a subset of transition system states which can be associated with a place in the synthesised Petri net. Further studies of Mukund [9], Badouel et al. [11] and Bernardinello et al. [10] have developed other versions of regions to deal with more powerful types of nets. These region definitions are however less intuitive than that of Nielsen et al., which is defined in a simple set-theoretical framework. With a certain re-formulation the results of Nielsen et al. have been found more amenable to rendering symbolic manipulation (based on binary decision diagrams) algorithms and software[1] for hardware design [7]. It would be fair to note that promising algorithmisation ideas, based on linear algebraic structures, come from [11]. They, however, appear to be quite complex in practice, minding the fact that they are applicable for the class of k-bounded nets. As we have already noted, asynchronous hardware design essentially bears on the binary nature of markings in 1-safe nets.

Although relying strongly on the set-theoretical approach of Nielsen et al., work started in [7] aims at 1-safe nets and at filling the gap between them and elementary nets. We have analysed limitations of elementary transition systems and elementary nets from the perspective of using them for asynchronous hardware modelling and design. This analysis has revealed that the major limitation is concerned with the impossibility to conveniently represent the following causal paradigm. Imagine that an event is caused by (pre-)conditions some of which when the event fires should remain true. In other words, the firing of an event in some operational case of an elementary net should not necessarily make one of its pre-conditions false. A simple example of such an effect is shown in Figure 1. Here, the state of the output of each gate is represented by a pair of conditions, one for the True value and the other for False. Thus, the events labelled with rising (e.g., $a+$) and falling (e.g., $a-$) transitions have the above conditions as preconditions, which is depicted by single arcs directed from the conditions to events. On the other hand, each gate has inputs which are the outputs of other gates and the state of these inputs determines the circumstances under which the gate's output changes it state. In terms of the net model this means that each event must also include as its preconditions the conditions modelling the state of its inputs. However, since the change of the output of the gates does not affect the change of its inputs, it would be irrational to assume that such a precondition has to become False once the event has fired. A consistent way to adequately model this situation would be to declare such a precondition also as a postcondition of the event. Note that in our net model of the circuit such relationship between conditions and events is depicted by two single-headed

[1]This software package is called **petrify**; its first version has been developed by J. Cortadella at the Politechic University of Catalonya. The tool is capable of synthesising 1-safe Petri nets from state graphs using the concept of minimal regions. It also has extra functionalities such as optimisation of Petri nets on the number of transitions and places, synthesis of nets within the class of free-choice nets etc. Those are based on the idea of splitting event labels. Currently the tool is developed further in the direction of supporting asynchronous circuit synthesis.

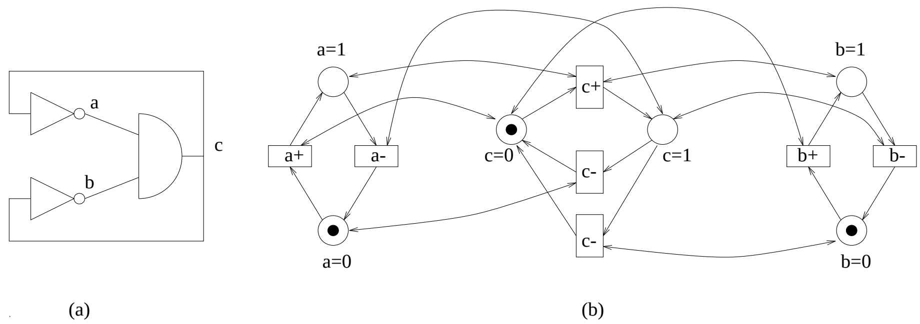

Figure 1: Example of a net with self-loops

arcs – an obvious situation of a *self-loop* in the net[2].

The above example, requiring the use of self-loops in nets (historically, Petri nets with self-loops have been called non-pure, hence the title of our work), can be easily extended to modelling situations where a circuit interacts with its environment through input and output signals. The transitions of input signals are associated with events which set to True the preconditions of events modelling the internal or output signal transitions. In some cases, these conditions have to remain True until some other event in the environment resets them to False. While self-loops are intuitively obvious at the syntactic level, it often happens that a behaviour of that sort is easier to capture at the transition system level [1, 2]. As a result, certain transition systems do not satisfy the elementarity axioms of Nielsen et al. [8] precisely for the reasons of being associated with non-pure nets.

In the view of the above practical motivation, this paper is aimed at theoretical extension of the class of elementary nets to semi-elementary nets [3]. Similarly, we formulate axioms for transition systems so as to check their semi-elementarity (Section 2). With these modifications we present a certain type of semi-elementary transition systems which are typical for asynchronous hardware behaviours with conflicts, such as arbitration and latches with independent clocks. These transition systems include specific fragments called *ladders*. We prove a condition (Theorem 3.8) under which one can delete "rungs" from ladders so that the transition system remains semi-elementary. Finally, Section 4 provides an illustration of the use of our theoretical results in deriving a semi-elementary net for a transition system defining the behaviour of a counterflow pipeline synchronisation from [1].

2 Extending the theory of elementary transition systems to accommodate self loops

Mukund [9] approaches the notion of a region using functions. This allows for a theoretical framework sufficiently powerful to characterise transition systems produced by k-bounded Petri nets. We prefer to follow the approach of Nielsen et al. [8] and try to benefit from its intuitively simpler idea of a region as a subset of states. This section provides the minimum set of extensions

[2] For simplicity, bidirectional arcs are often used to represent self-loops in Petri nets.

[3] In fact, this is almost an extension to 1-safe Petri nets since for each non-self-loop place in a semi-elementary net a unique complement place can be added without changing the net's behaviour; such new net will effectively acts as a 1-safe Petri net except for the cases when an event is only linked to the rest of the net by self-loop places.

to the definitions and results of Nielsen et al. [8] relating transition systems and non-pure nets. Although it would be desirable if the reader were familiar with the theory of elementary transition systems [8], we do not generally assume this to be a pre-requisite.

Definition 2.1 [8] A *transition system* is a quadruple $TS = (S, E, T, s_{in})$, where

- S is a non-empty set of states,
- E is a set of events,
- $T \subseteq S \times E \times S$ is the transition relation,
- $s_{in} \in S$ is the initial state.

We assume TS satisfies the following conditions (axioms):

(A1) For every $(s, e, s') \in T$, $s \neq s'$.

(A2) For every $e \in E$ there are s and s' such that $(s, e, s') \in T$.

(A3) For every $s \in S - \{s_{in}\}$ there are $e_0, e_1, \ldots, e_{n-1} \in E$ and $s_0, \ldots, s_n \in S$ $(n \geq 1)$ such that $s_0 = s_{in}, s_n = s$ and $(s_i, e_i, s_{i+1}) \in T$, for $0 \leq i < n$. □

We will often write $s \stackrel{e}{\longrightarrow} s'$ if $(s, e, s') \in T$, and $s \stackrel{e}{\longrightarrow}$ if $(s, e, s') \in T$, for some s'. Note that, due to a practical motivation of this work as outlined in the introduction, we do not formulate our results in terms of categories and $G-$ and $N-$morphisms defined in [8]. This allows us to eliminate some constraints imposed on a transition system, namely axiom (A2) from [8]. Recall that (A2) did not allow multiple arcs between a pair of states. This restriction was introduced to avoid the so-called non-simple nets, nets where different events may have the same sets of pre- and post-conditions.

Definition 2.2 [8] Let $TS = (S, E, T, s_{in})$ be a transition system. Then $r \subseteq S$ is a *region* of TS if the following two conditions are satisfied:

- $(s, e, s') \in T \wedge s \in r \wedge s' \notin r \Rightarrow (\forall (s_1, e, s_1') \in T)[s_1 \in r \wedge s_1' \notin r]$
- $(s, e, s') \in T \wedge s \notin r \wedge s' \in r \Rightarrow (\forall (s_1, e, s_1') \in T)[s_1 \notin r \wedge s_1' \in r]$ □

If we call by R_{TS} the set of *non-trivial* regions (without S and $\emptyset$) then by R_s we will mean the set of non-trivial regions containing $s \in S$,

$$R_s = \{r \in R_{TS} \mid s \in r\}.$$

The sets of *pre-regions* and *post-regions* of an event $e \in E$ are defined as follows:

- ${}^\circ e = \{r \in R_{TS} \mid (\exists (s, e, s') \in T)[s \in r \wedge s' \notin r]\}$
- $e^\circ = \{r \in R_{TS} \mid (\exists (s, e, s') \in T)[s \notin r \wedge s' \in r]\}$

Proposition 2.3 *[8] Let $TS = (S, E, T, s_{in})$ be a transition system.*

1. *$r \subseteq S$ is a region iff $\overline{r} = S - r$ is a region.*
2. *If $e \in E$ then $e^\circ = \{\overline{r} \mid r \in {}^\circ e\}$ and ${}^\circ e = \{\overline{r} \mid r \in e^\circ\}$.*
3. *If $s \xrightarrow{e} s'$ then $R_s - R_{s'} = {}^\circ e$ and $R_{s'} - R_s = e^\circ$.*
   Consequently ${}^\circ e \subseteq R_s$ and $e^\circ \cap R_s = \emptyset$ and $R_{s'} = (R_s - {}^\circ e) \cup e^\circ$. □

The following definition is new; it characterises the relationship between a region and an event in which a transition labelled by the event is (i.e., both its source and destination states are) completely inside the region.

Definition 2.4 Let $TS = (S, E, T, s_{in})$ be a transition system and $r \in R_{TS}$ be a non-trivial region. By $B_r^e = \{(s, e, s') \in T \mid s \in r \wedge s' \in r\}$ we will denote the set of all the arcs labelled by e which are 'buried' in r.
The set of *co-regions* of an event $e \in E$ is defined as follows

$$\overset{\circ}{e} = \{r \in R_{TS} \mid B_r^e \neq \emptyset \wedge B_{\overline{r}}^e = \emptyset\}.$$

□

The following proposition adds to the properties of regions associated with an event observed in Proposition 2.3.

Proposition 2.5 *Let $TS = (S, E, T, s_{in})$ be a transition system.*

1. *If $e \in E$ and $r \in \overset{\circ}{e}$ then $r \notin {}^\circ e \cup e^\circ$ and $\overline{r} \notin {}^\circ e \cup \overset{\circ}{e} \cup e^\circ$.*
2. *If $s \xrightarrow{e} s'$ then $\overset{\circ}{e} \subseteq R_s \cap R_{s'}$.* □

With the aid of the notion of co-regions, we modify one of the remaining two properties formulated in [8].

Definition 2.6 A transition system $TS = (S, E, T, s_{in})$ is said to be *semi-elementary* if it satisfies, in addition to (A1)-(A3), the two regional axioms:

(A4) For all $s, s' \in S$, $[R_s = R_{s'} \Rightarrow s = s']$
(State Separation Property)

(A5) For all $s \in S$ and $e \in E$, $[({}^\circ e \subseteq R_s \wedge \overset{\circ}{e} \subseteq R_s) \Rightarrow s \xrightarrow{e}]$.
(Forward Closure Property). □

Now, with a very slight (but important) change wrt [8], we define a net.

Definition 2.7 A *net* is a triple $N = (B, E, F)$ where:

1. B is a set of conditions,
2. E is a set of events,
3. $B \cap E = \emptyset$,
4. $F \subseteq (B \times E) \cup (E \times B)$ is a flow relation,
5. $(\forall x \in E)(\exists y_1, y_2 \in B)[(x, y_2) \in F \wedge (y_1, x) \in F \wedge (x, y_1) \notin F \wedge (y_2, x) \notin F]$,
6. $(\forall x \in B)(\exists y \in E)[(x, y) \in F \vee (y, x) \in F]$.

Observe that the additional constraint $(x, y_1) \notin F \wedge (y_2, x) \notin F$ placed in item 5, forbids a net with events having its connections with conditions **only** via self-loops. This means that every event must have at least one 'pure' predecessor and one 'pure' successor condition. Such a precaution is obviously caused by axiom (A1) of a transition system.

Definition 2.8 Let $N = (B, E, F)$ be a net and $x \in B \cup E$.

$^\bullet x = \{y \mid (y, x) \in F \wedge (x, y) \notin F\}$ (the pre-elements of x),

$x^\bullet = \{y \mid (x, y) \in F \wedge (y, x) \notin F\}$ (the post-elements of x),

$\overset{\bullet}{x} = \{y \mid (x, y) \in F \wedge (y, x) \in F\}$ (the co-elements of x). □

The next definition introduces the class of Petri nets which will be dealt with in this paper.

Definition 2.9 A *semi-elementary net system* is a quadruple $N = (B, E, F, c_{in})$ where (B, E, F) is called the *underlying net* of N (denoted $\mathcal{N}_N$) and $c_{in} \subseteq B$ is the *initial case* (a *case* is a subset of B). □

The semantics of a semi-elementary net system is given through the transition relation defined below.

Definition 2.10 Let $N = (B, E, F)$ be a net. Then $\rightarrow_N \subseteq 2^B \times E \times 2^B$ is the transition relation of N and is given by

$$\rightarrow_N = \{(c, e, c') \mid c - c' = {}^\bullet e \wedge c' - c = e^\bullet \wedge \overset{\bullet}{e} \subseteq c \cap c'\}.$$

□

We will often use $c \xrightarrow{e}$ whenever $(c, e, c') \in \rightarrow_N$, for some c', and say that an event e is *enabled* at case c.
The difference between elementary [8] and semi-elementary net systems is that the latter allow an event to fire only if its co-conditions are true. The state of these conditions cannot be changed by the event. Such co-conditions may however be pre- or post-conditions for some other event(s), which can change their state. In the rest of the paper we will only consider semi-elementary net systems. For brevity, we will call them simply net systems.
The following two definitions are modified versions of corresponding definitions in [8].

Definition 2.11 Let $N = (B, E, F, c_{in})$ be a net system.

1. C_N is the *state space of* N which is defined as the least subset of 2^B containing c_{in} which satisfies:

$$(c, e, c') \in \rightarrow_{\mathcal{N}_N} \wedge c \in C_N \Rightarrow c' \in C_N.$$

2. $\rightarrow_N$ is the *transition relation* of N which is defined as $\rightarrow_{\mathcal{N}_N}$ restricted to $C_N \times E \times C_N$.

3. E_N is the set of *active events* of N which is defined as the subset of E given by

$$E_N = \{e \mid \exists (c, e, c') \in \rightarrow_N\}.$$

□

It is possible to associate a transition system with any net system.

Definition 2.12 Let $N = (B, E, F, c_{in})$ be a net system. Then the transition system

$$TS_N = (C_N, E_N, \rightarrow_N, c_{in})$$

is called the transition system *associated with* N. □

The following proposition is basically the same as the one of [8] except for its first statement, which requires co-conditions to be taken into account.

Proposition 2.13 *Let* $N = (B, E, F, c_{in})$ *be a net system.*

1. $(\forall c \in C_N)(\forall e \in E)[c \xrightarrow{e} \Leftrightarrow ({}^\bullet e \cup \overset{\bullet}{e} \subseteq c \wedge e^\bullet \cap c = \emptyset)]$.
2. $(\forall (c, e, c') \in \rightarrow_N)[c' = (c - {}^\bullet e) \cup e^\bullet]$.
3. $(\forall (c_1, e, c_2), (c_3, e, c_4) \in \rightarrow_N)[c_1 - c_2 = c_3 - c_4 \wedge c_2 - c_1 = c_4 - c_3]$.
4. $(c, e, c_1), (c, e, c_2) \in \rightarrow_N \Rightarrow c_1 = c_2$. □

In the following theorems we 'adjust' the relationship of [8] between net systems and transition systems to the semi-elementarity case.

Theorem 2.14 *Let* $N = (B, E, F, c_{in})$ *be a net system. Then* $TS_N = (C_N, E_N, \rightarrow_N, c_{in})$ *satisfies (A1)-(A5)(is a semi-elementary transition system).* □

The definition of a net system associated with a transition system from [8] has to be modified as follows.

Definition 2.15 Let $TS = (S, E, T, s_{in})$ be a transition system. Then $N_{TS} = (R_{TS}, E, F_{TS}, R_{s_{in}})$ where

$$F_{TS} = \{(r, e) \mid e \in E \wedge r \in {}^\circ e \cup \overset{\circ}{e}\} \cup \{(e, r) \mid e \in E \wedge r \in e^\circ \cup \overset{\circ}{e}\}$$

is a net system *associated with* TS. □

Theorem 2.16 *Let $TS = (S, E, T, s_{in})$ be a semi-elementary transition system. Then $N_{TS} = (R_{TS}, E, F_{TS}, R_{s_{in}})$ is a semi-elementary net system.* □

Lemma 2.17
Let $TS = (S, E, T, s_{in})$ be a semi-elementary transition system and $N = (R_{TS}, E, F_{TS}, R_{s_{in}})$ be a semi-elementary net system associated with it.

1. $C_N = \{R_s \mid s \in S\}$,
2. $\rightarrow_N = \{(R_s, e, R_{s'}) \mid (s, e, s') \in T\}$,
3. $E_N = E$. □

Here is the main result of this section, which finalises our 'adjustments' of Nielsen et al. [8] to the class of semi-elementary nets and their transition system counterparts.

Theorem 2.18 *Let $TS = (S, E, T, s_{in})$ be a semi-elementary transition system and $N = (R_{TS}, E, F_{TS}, R_{s_{in}})$ be a semi-elementary net system associated with it.*
Then $TS_N = (C_N, E_N, \rightarrow_N, c_{in})$ is isomorphic to TS. □

3 Properties of semi-elementary transition systems

In this section we first investigate how adding a self-loop to a net system changes the associated transition system.

Definition 3.1 Let $TS = (S, E, T, s_{in})$ be a transition system and $T_{RM} \subseteq T$.
By $TS(T_{RM})$ we will denote the maximal transition system with the initial state s_{in} included in TS after removing the arcs T_{RM}. □

Theorem 3.2 *Let $TS = (S, E, T, s_{in})$ be a semi-elementary transition system and $N = (R_{TS}, E, F_{TS}, R_{s_{in}})$ be the semi-elementary net system associated with it. Moreover, let $a \in E$ and $x \in R_{TS}$ be such that $(a, x) \notin F_{TS}$ and $(x, a) \notin F_{TS}$.*
Define $N' = (R_{TS}, E, F_{TS} \cup \{(a, x), (x, a)\}, R_{s_{in}})$.
Then $TS_{N'}$ is (isomorphic to) $TS(T_{RM})$ where $T_{RM} = \{(s, a, s') \in T \mid (s, a, s') \notin B_x^a\}$.

Proof. ¿From $(x, a), (a, x) \notin F_{TS}$ it follows that $\rightarrow_{N'} \subseteq \rightarrow_N$. This and Lemma 2.17(2) mean that it suffices to prove that for every $(s, e, s') \in T$,

$$(R_s, e, R_{s'}) \notin \rightarrow_{\mathcal{N}_{N'}} \Leftrightarrow e = a \wedge (s, e, s') \notin B_x^a.$$

Note that $(s, e, s') \in T$ implies $(R_s, e, R_{s'}) \in \rightarrow_N$ and consider three cases.
Case 1
$(s, e, s') \in T$ and $e \neq a$.
Then, $(R_s, e, R_{s'}) \in \rightarrow_{\mathcal{N}_{N'}}$ follows from $(R_s, e, R_{s'}) \in \rightarrow_N$ and the fact that the flow relation for $e \neq a$ is unchanged.

Case 2
$(s,a,s') \in T - B_x^a$.
Then, by $(x,a),(a,x) \notin F_{TS}$, $(s,a,s') \in B_{S-x}^a$. Hence $s \notin x$ and so $x \notin R_s$, but (x,a) is an arc in N'. As a result, $(R_s,a,R_{s'}) \notin \rightarrow_{\mathcal{N}_{N'}}$.
Case 3
$(s,a,s') \in B_x^a$.
Then $x \in R_s \cap R_{s'}$ which together with $(R_s,a,R_{s'}) \in \rightarrow_N$ yields $(R_s,a,R_{s'}) \in \rightarrow_{\mathcal{N}_{N'}}$. ■ 3.2

The above result can be used to characterise the relationship between net systems associated with transition systems comparable w.r.t. the inclusion relation.

Theorem 3.3 *Let $TS = (S,E,T,s_{in})$ be a transition system, $a \in E$ and $TS' = (S,E,T',s_{in})$ be a semi-elementary transition system obtained from TS by adding at least one arc labelled by a.*
If there is $x \in R_{TS'}$ such that $B_x^a = \{(s,a,s') \in T' \mid (s,a,s') \in T\}$ then TS is generated by N which can be obtained from $N_{TS'}$ by adding two arcs: (a,x) and (x,a).

Proof. Since $a \in E$ and TS is a transition system (see (A2)), $B_x^a \neq \emptyset$ in TS'. Moreover, since we added at least one arc, $B_{S-x}^a \neq \emptyset$ in TS'. Hence there is no arc (in either direction) between x and a in $N_{TS'}$. Thus TS' satisfies the assumptions of Theorem 3.2 (but playing the role of TS!). Hence after adding (a,x) and (x,a) to $N_{TS'}$ we obtain $\overline{N}$ which generates transition system $\overline{TS} = TS'(T'_{RM})$ where

$$T'_{RM} = \{(s,a,s') \in T' \mid (s,a,s') \notin B_x^a\}.$$

One can now see that $\overline{TS} = TS$, because TS was a transition system and from axiom (A3) we know that every state was reachable from the initial state, so the arcs removed from TS' are those which were added to TS. ■ 3.3

We now turn our attention to the specific sub-structures of transition systems called ladders, which, as we mentioned in the introduction, play an important role in dealing with certain types of asynchronous circuits.

Definition 3.4 Let $TS = (S,E,T,s_{in})$ be a transition system.

1. A *path* in TS is a sequence $\sigma = s_1e_1s_2\ldots s_{n-1}e_{n-1}s_n$ such that $n \geq 1$ and $(s_i,e_i,s_{i+1}) \in T$ for $1 \leq i < n$. We will denote $states(\sigma) = \{s_1,\ldots,s_{n-1},s_n\}$.

2. Let $\sigma = s_1e_1s_2\ldots s_{n-1}e_{n-1}s_n$ and $\sigma' = s'_1e_1s'_2\ldots s'_{n-1}e_{n-1}s'_n$ be two paths in TS and $a \in E$ be such that $|states(\sigma)| = |states(\sigma')| = n$, $states(\sigma) \cap states(\sigma') = \emptyset$ and $(s_k,a,s'_k) \in T$, for some $1 \leq k \leq n$. Then the triple:

$$ldd = (\sigma,a,\sigma')$$

is a *ladder* in TS. We will also denote $I(ldd) = \{1 \leq i \leq n \mid (s_i,a,s'_i) \in T\}$ and $rungs(ldd) = \{(s_i,a,s'_i) \mid i \in I(ldd)\}$. □

Our first result states that if (σ,a,σ') is a ladder then each pre-region of a contains all the states of path σ and, similarly, each post-region of a contains all the states of path σ'.

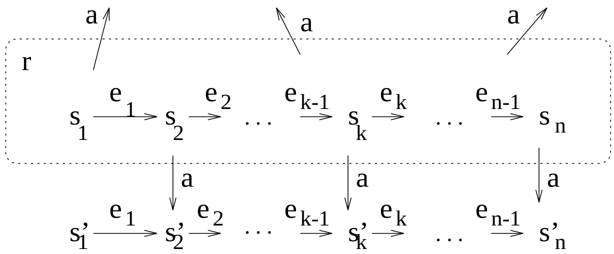

Figure 2: Illustration for Lemma 3.5

Lemma 3.5
Let $TS = (S, E, T, s_{in})$ be a transition system and ldd be a ladder as in Definition 3.4. Moreover, let $r \in R_{TS}$.

1. *If $r \in {}^\circ a$ then $states(\sigma) \subseteq r$.*
2. *If $r \in a^\circ$ then $states(\sigma') \subseteq r$.*

Proof. (see Figure 2 for illustration) ¿From $r \in {}^\circ a$ we have $s_k \in r$ and $s'_k \notin r$. Suppose $k - 1 \geq 1$. ¿From $\xrightarrow{e_{k-1}} s_k$ and $\xrightarrow{e_{k-1}} s'_k$ and $s_k \in r$ and $s'_k \notin r$ we have $s_{k-1} \in r$ and $s'_{k-1} \notin r$. We can continue the same procedure: If $k - 2 \geq 1$ then from $\xrightarrow{e_{k-2}} s_{k-1}$ and $\xrightarrow{e_{k-2}} s'_{k-1}$ and $s_{k-1} \in r$ and $s'_{k-1} \notin r$ we have $s_{k-2} \in r$ and $s'_{k-2} \notin r$ etc. Hence $\{s_1, \dots, s_{k-1}\} \subseteq r$. Similarly, one can show that $\{s_{k+1}, \dots, s_n\} \subseteq r$. ■ 3.5

The next result shows that deleting rungs from a ladder does not change the set of regions of the resulting transition system.

Lemma 3.6
Let $TS = (S, E, T, s_{in})$ be a transition system and ldd be a ladder as in Definition 3.4. Moreover, let $TS' = (S, E, T', s_{in})$ be a transition system obtained from TS by deleting some (but not all) arcs from $rungs(ldd)$. Then $R_{TS} = R_{TS'}$.

Proof. It suffices to show that the result holds after deleting a single arc $(s_k, a, s'_k) \in rungs(ldd)$. Note that $T' = T - \{(s_k, a, s'_k)\}$ in such a case.
Showing that $R_{TS} \subseteq R_{TS'}$ is straightforward.

We prove that $R_{TS'} \subseteq R_{TS}$ by assuming that there is $r \in R_{TS'}$ such that $r \notin R_{TS}$. ¿From the definition of region we know that there are arcs $(s, e, s') \in T$ and $(\hat{s}, e, \hat{s}') \in T$ which have different 'crossing relationship' with r. We consider two cases.

1. $e \neq a$.
   The arcs (s, e, s') and $(\hat{s}, e, \hat{s}')$ belong to TS', so r cannot be a region in TS', a contradiction.
2. $e = a$.
   Since r is a region in TS' we can assume, without loss of generality, that (s_k, a, s'_k) is $(\hat{s}, a, \hat{s}')$. According to the assumptions, not all arcs in *rungs(ldd)* were deleted. Suppose (s_m, a, s'_m), where $1 \leq m \leq n$ and $m \neq k$, is still in TS'. If (s_m, a, s'_m) has different crossing

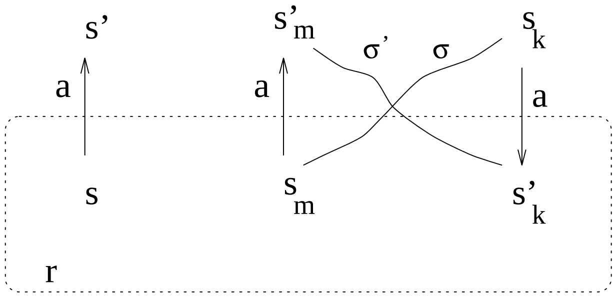

Figure 3: Illustration for Lemma 3.6

relationship with r than (s, a, s') we have a contradiction, because they both belong to TS', so r cannot be a region in TS'.
Let us assume (s_m, a, s'_m) has the same crossing relationship with r as (s, a, s').
Suppose $s, s_m \in r$ and $s', s'_m \notin r$. We have three cases to consider:

- $s_k \notin r$ and $s'_k \in r$ (see Figure 3 for illustration).
- $s_k \in r$ and $s'_k \in r$.
- $s_k \notin r$ and $s'_k \notin r$.

In all three cases it is easy to show that there exist $(s_i, e_i, s_{i+1}) \in T$ and $(s'_i, e_i, s'_{i+1}) \in T$ on the paths σ and σ' respectively which has different crossing relationships with r (since $states(\sigma) \cap states(\sigma') = \emptyset$) and they are not removed from TS'. Hence r cannot be a region in TS'.
All other cases, $s, s_m \notin r$ and $s', s'_m \in r$ etc. are similar.

Hence $R_{TS} = R_{TS'}$. ■ 3.6

It turns out every two rungs in a ladder are separated by a region.

Lemma 3.7
Let $TS = (S, E, T, s_{in})$ be a semi-elementary transition system and ldd be a ladder as in Definition 3.4. Then for every two distinct arcs $\tau, \tau' \in rungs(ldd)$ there is $r \in R_{TS}$ such that $\tau \in B^a_r$ and $\tau' \notin B^a_r$.

Proof. Suppose $i \neq j \in I(ldd)$ are such that for all $r \in R_{TS}$, $(s_i, a, s'_i) \in B^a_r \Leftrightarrow (s_j, a, s'_j) \in B^a_r$.
¿From Lemma 3.5 it follows that:
$(\forall r \in {}^\circ a)\ s_i, s_j \in r \ \wedge\ (\forall r \in a^\circ)\ s_i, s_j \notin r$.
As a result every region $r \in R_{TS}$ either contains both s_i and s_j or none of them. Hence $R_{s_i} = R_{s_j}$ and, by (A4), we obtain $s_i = s_j$. This, however, contradicts $i \neq j$ and $|states(\sigma)| = n$. ■ 3.7

We end this section formulating and proving a major result that characterises certain situations under which deleting rungs from a ladder has no effect on being a semi-elementary transition system.

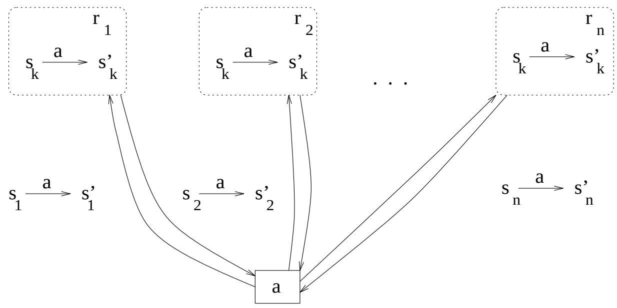

Figure 4: Illustration for Theorem 3.8 ($\forall j (s_j, a, s'_j) \notin B^a_{r_j}$)

Theorem 3.8 *Let $TS = (S, E, T, s_{in})$ be a semi-elementary transition system and ldd be a ladder as in Definition 3.4. Moreover, let rungs(ldd) be the only a-labelled arcs in TS.*
Let $TS' = (S, E, T', s_{in})$ be a transition system obtained from TS by deleting all but one arc from rungs(ldd). Then TS' is semi-elementary.

Proof. (see Figure 4 for illustration) TS' satisfies the assumptions of Lemma 3.6, so $R_{TS} = R_{TS'}$. Hence, since (A4) was true for TS, it is true for TS' as the sets of places and non-trivial regions are the same. We now prove that (A5) is satisfied. Suppose $\tau = (s_k, a, s'_k)$ is the only arc from *rungs(ldd)* which belongs to TS'. ¿From Lemma 3.7 it follows that:
$(\forall j \in I(ldd) - \{k\})\, (\exists r_j \in R_{TS})\, \tau \in B^a_{r_j} \;\wedge\; (s_j, a, s'_j) \notin B^a_{r_j}$
which implies
$(*)\; (\forall j \in I(ldd) - \{k\})\, (\exists r_j \in R_{TS})\, \tau \in B^a_{r_j} \;\wedge\; r_j \notin R_{s_j}$.
Axiom (A5) is satisfied for all $e \neq a$, because it was satisfied for TS, and the set of states is unchanged, the set of regions is unchanged, and the set of e-labelled arcs is unchanged. What we need to show is that in TS':
$(\forall s \in S)\; {}^\circ a \subseteq R_s \wedge \mathring{a} \subseteq R_s \Rightarrow s \xrightarrow{a}$.
We first observe that ${}^\circ a$ (in TS) is the same as ${}^\circ a$ (in TS') since $R_{TS} = R_{TS'}$. Moreover, $\mathring{a}$ (in TS) is a subset of $\mathring{a}$ (in TS'). Thus, the only property we need to check is:
$(**)\; (\forall j \in I(ldd) - \{k\})\; \mathring{a} \not\subseteq R_{s_j}$ (in TS').
But we know that in TS':
$\mathring{a} = \{r \mid r \in R_{TS'} \wedge (s_k, a, s'_k) \in B^a_r\}$.
This and (*) yields:
$(\forall j \in I(ldd) - \{k\})\, (\exists r_j \in R_{TS})\; r_j \in \mathring{a} \;\wedge\; r_j \notin R_{s_j}$.
Hence (**) holds. ■ 3.8

4 Application

In this section we briefly illustrate how the theory developed in the previous section can be applied in synthesising a net model from an initial specification of a system by means of a transition system. Our example originates from [1], where a counterflow pipeline processor, now called the Sproull's processor, is described. The key part in the distributed control structure of the processor is played by a device which provides mutual synchronisation between two pipelines, an instruction pipeline and a results pipeline. This device is supposed to be placed into each

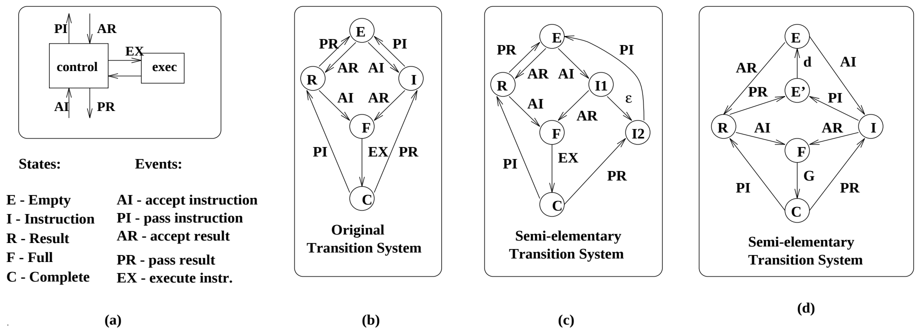

Figure 5: Counterflow pipeline example: constructing semi-elementary transition systems

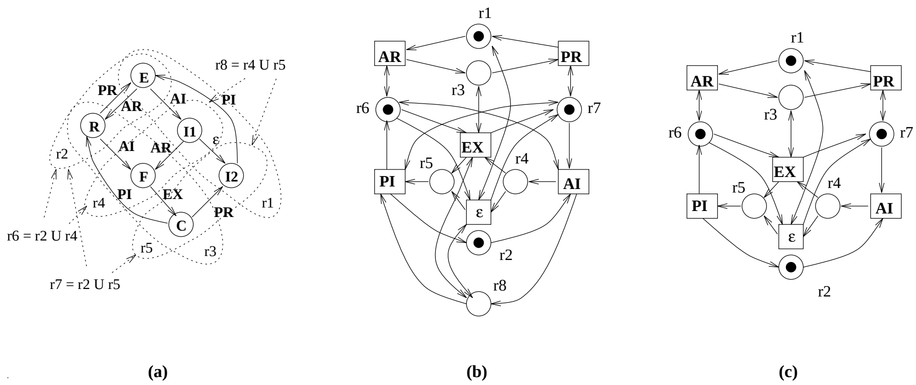

Figure 6: Counterflow pipeline example: constructing semi-elementary nets from ladder transition system

stage of the counterflow pipeline. The original idea of synchronisation belongs to Charles Molnar, who described it in the form of a transition system, which is reproduced in Figure 5,a,b. This figure also assigns the meaning to the signals and the corresponding events and the states of the stage control. We shall not discuss here any details and ideas behind this specification. since the purpose of this section is to illustrate out theory. Similarly, we avoid discussing the reasons why we have been able to transform this initial transition system either to the "asymmetric" one, shown in Figure 5,c, or the "symmetric" one, shown in Figure 5,d. Both transformations involve inserting a "dummy" event and yield semi-elementary transition systems. Note that in the symmetric case we have been able to preserve the trace equivalence (up to hiding the dummy event). In the asymmetric case however, the system cannot execute event **AR** after executing **PR** from state **C** – the original model allowed that possibility. Both solutions were studied in [2]. For our illustration purposes, let us further look at the asymmetric case.

The transition system shown in Figure 5,c can be converted into a semi-elementary net. This is shown in Figure 6. In the (a) part of this figure we showed all the regions which gave rise to the conditions in the net depicted in the (b) part. This transition system has a ladder structure, for example with respect to events **AR** or **PR**. It is easy to notice that the ladder satisfies the conditions of Theorem 3.8. We could remove some of the ladder rungs labelled with **AR** and **PR** up to the case when only a single rung of each of them remains. For example, if we consider the case shown in Figure 6,a, the rungs labelled with **PR** will be buried in region $r7$ which is the union of regions $r2$ and $r5$. Now, removing the **PR** rung between **C** and **I2** makes the remaining

PR rung between **R** and **E** be buried in region $r7$, $r2$ and $r6$. In terms of the associated net this would correspond to adding the connections with double-headed arcs between event **PR** and, first, condition $r2$, and then condition $r6$. All such nets remain semi-elementary, with the same structure except that they differ only in the self-loop conditions, which constrain concurrency of some events. With circuit design techniques available from 1-safe nets, one can adjust the specification at the semantic level by changing the structure of ladders. Indeed, adding some ordering constraints (by rung removal) often helps to satisfy some timing or mutual exclusion requirements.

Note also that events **EX** and ϵ (dummy) are also constrained by self-loops with conditions $r3$ and $r1$ respectively. This is concerned with their transitions being buried in correponding regions in the transition system. Similar self-loop constructions were used in [2] for the symmetric case shown in Figure 5,c.

5 Concluding remarks

It should be noted that our extension of the elementarity to accommodate self-loops in net models is not semantically concerned with having self-loops in the transition systems. Self-loops (an arc labelled with some event leads to the same state) in transition systems have a different interpretation from the one we intended to capture. With our extension we would like stay within the limits of modelling systems where each event is significant in the sense that it changes the state of the system. Such an assumption is perfectly acceptable for modelling asynchronous systems. As could be noted from our examples, the effect of our co-regions and self-loop conditions on events is always made in conjunction with pre- and post-conditions. In some sense the role of such co-conditions is purely logical or predicative, very much similar (up to inversion) to that of inhibitor arcs in inhibitor nets. The reader may easily imagine how straightforward would be to extend the results further to accomodate inhibitor nets. It should also be obvious that certain gap still remains between 1-safe nets and semi-elementary nets, though such a gap has been narrowed to the effect of axiom (A1). We further conjecture that changing (A1) to (A1') $\neg[(\exists e \in E) S \times \{e\} \times S \subseteq T]$ will lead to a class of transition systems equivalent to 1-safe Petri nets without isolated transitions. We have recently found out some useful practical considerations (e.g., the modelling of systems with synchronous components) in favour of this extension.

Acknowledgements

We would like to thank Jordi Cortadella, Luciano Lavagno and Michael Kishinevsky for cooperation in applying the ideas of regions for Petri net synthesis and for numerous past and on-going discussions. We are grateful also to Maciej Koutny for useful comments about our work, and in particular for pointing to the possibility of extending the region technique for inhibitor nets.

References

[1] R.F. Sproull,I. Sutherland and C.E. Molnar. The Counterflow Pipeline Processor Architecture. *IEEE Design and Test of Computers*, Fall 1994, pp. 48 – 59.

[2] A. Yakovlev. Designing Control Logic for Counterflow Pipeline Processor Using Petri Nets. Technical Report Series No. 522, Department of Computing Science, University of Newcastle upon Tyne, May 1995.

[3] L. Y. Rosenblum and A. V. Yakovlev. Signal Graphs: from Self-timed to Timed Ones. International Workshop on Timed Petri Nets, Torino, Italy, July 1985, pp. 199 – 207.

[4] T.-A. Chu. Synthesis of Self-timed VLSI Circuits from Graph-theoretic Specifications. PhD Thesis, MIT, June 1987.

[5] L. Lavagno and A. Sangiovanni-Vincentelli. Algorithms for synthesis and testing of asynchronous circuits. Kluwer Academic Publishers, Boston, MA, 1993.

[6] M. Kishinevsky, A. Kondratyev, A. Taubin, V. Varshavsky. Concurrent Hardware: The Theory and Practice of Self-Timed Design. John Wiley and Sons, London, 1993.

[7] J. Cortadella, M. Kishinevsky, L. Lavagno and A. Yakovlev. Synthesizing Petri Nets from State-Based Models. accepted to ICCAD'95, San Jose, November 1995 (also available as Technical Report Universitat Politecnica de Catalunya, RR 95/09 UPC/DAC, April, 1995).

[8] M. Nielsen, G. Rozenberg and P.S. Thiagarajan. Elementary transition systems. *Theoretical Computer Science*, Vol. 96, pp. 3 - 33, 1992.

[9] M. Mukund. Petri Nets and Step Transition Systems. Int. Journal of Foundations of Computer Science, Vol. 3 (4), pp. 443-478, 1992.

[10] L. Bernardinello, G. De Michelis, K. Petruni and S. Vigna. On Synchronic Structure of Transition Systems. Universita di Milano, 1994.

[11] E. Badouel, L. Bernardinello and Ph. Darondeau. Polynomial Algorithms for the Synthesis of Bounded Nets. Technical Report No. 2316, INRIA, RENNES Cedex, France, 1994.