Classifying Boolean Nets for Region-based Synthesis

Jetty Kleijn[1], Maciej Koutny[2],
Marta Pietkiewicz-Koutny[2], and Grzegorz Rozenberg[1,3]

[1] LIACS, Leiden University, The Netherlands
{kleijn,rozenber}@liacs.nl
[2] School of Computing Science, Newcastle University, UK
{maciej.koutny,marta.koutny}@ncl.ac.uk
[3] Department of Computer Science, University of Colorado at Boulder, U.S.A.

Abstract. A Petri net model is referred to as Boolean if the only possible markings are sets, i.e., places are marked or not without further quantification; moreover, also the enabling conditions and firing rule are based on this principle of set-based token arithmetic. Elementary Net systems are an example of a class of Boolean nets, and so are the recently introduced SET-nets. In our investigation of the synthesis problem for SET-nets, it would be useful to know how this new net model can be fitted into the general theory of net synthesis based on the generic concept of τ-nets. Here, we demonstrate how SET-nets and the idea of Boolean operations on tokens provide an opportunity to classify a wide variety of Boolean nets that are amenable to region-based net synthesis.
Keywords: Petri net, Boolean net, step semantics, concurrency, conflict, net synthesis, theory of regions, transition system, NET-type

1 Introduction

Recently, in [10], a new class of Petri nets, called SET-nets, has been introduced to provide a net based computational model matching very closely the computations exhibited by reaction systems [5, 7, 8], a framework for the investigation of processes carried out by biochemical reactions in living cells. The formalisation leading to reaction systems has been motivated by properties that are common to many biochemical reactions. This has resulted in a model based on principles that are different from most existing models of computation. Of particular importance for the net model inspired by reaction systems, are the non-counting features (motivated by the two main regulation mechanisms of facilitation and inhibition) implying that entities are either present or not present and enable reactions by their presence or absence. As a consequence, there is no conflict between reactions in the sense that the occurrence of one reaction might imply that another reaction which is also enabled at the current state, cannot occur.

The new class of Petri nets, SET-nets, provides a faithful computational model matching very closely that exhibited by reaction systems. The key difference between standard Petri nets like Place/Transition nets (PT-nets) and SET-nets is that the former support multiset-based token arithmetic, whereas the latter support set-based (or Boolean) operations on tokens.

Thus, the computational intuition embedded in bio-processes has led to a new class of nets with yet to discover properties. On the other hand, an important motivation behind the wish to establish the link with net theory is to establish whether Petri net based concepts (such as causal processes) and methods (such as synthesis of nets from a specification of their behaviour) could be used to provide analytical tools for reaction systems.

The research presented in this paper originates with the synthesis problem for SET-nets. We show that the new class of nets can be treated within the general theory of region-based net synthesis. More precisely, we show that SET-nets are an instance of τ-nets [1] which incorporate many Petri net classes, and for which the synthesis problem has been investigated and solved using regions of transition systems [6]. Moreover, τ-nets with maximally concurrent semantics (the semantics of SET-nets when used to model reaction systems) fall within the general framework of τ-nets with *policies* introduced in [2]. In this paper, we will actually concern ourselves with the wider task of dealing with a whole variety of net models similar to SET-nets and referred to as Boolean nets. It is our aim to classify such nets thus working towards automatic net synthesis algorithms. The key part of our investigation is a detailed study of connection monoids (CONN-monoids) for Boolean nets which allows us to capture not only the step semantics of nets but also structural conflicts between transitions in Boolean nets, thanks to a special 'blocking' connection which can be used to capture the essence of conflicts in (Boolean) τ-nets. In this way, CONN-monoids emerge as a single formalism which can be used to deal with conflicts, concurrency and net synthesis.

The paper is organised as follows. First, we present the basics of SET-nets and other types of nets. Section 3 recalls the general setup of [1, 2] in which Petri net classes are defined using CONN-monoids and τ-nets. Section 4 shows how to build CONN-monoids for EN-systems and SET-nets. The observations in this section are generalised in Section 5 to Boolean τ-nets.

2 SET-nets, EN-systems, and PT-nets

In this section we present SET-nets and relate them to elementary net systems (EN-systems) [13] and Place/Transition nets (PT-nets) [4].

The main idea underlying SET-nets is that there is no concept of token counting. Places are marked or not marked and arcs have no weights. In this way, SET-nets resemble EN-systems, a fundamental net model in the study of basic features of concurrent systems. However, the execution semantics of the two models differ significantly.

Both SET-nets and EN-systems have an (unweighted) net as their underlying structure. A *net* is a triple (P, T, F) such that P and T are disjoint finite sets of *places* and *transitions*, respectively, and $F \subseteq (T \times P) \cup (P \times T)$ is the *flow* of the net. We use the standard dot-notation: for any place or transition x, we let ${}^\bullet x = \{y \mid (y, x) \in F\}$ be its set of input elements and $x^\bullet = \{y \mid (x, y) \in F\}$ its output elements. This extends in the usual way to sets of places and/or transitions. For EN-systems, we have the additional structural assumption that the underlying net has no 'self-loops' i.e., ${}^\bullet t \cap t^\bullet = \varnothing$ for all $t \in T$.

A *marking* of a SET-net or EN-system is a subset of places of their underlying net. A place belonging to a given marking is said to be marked. In diagrams, places are drawn as circles and transitions as rectangles. If $(x, y) \in F$, then (x, y) is an *arc* leading from *node* x to *node* y. Markings are indicated by drawing in each place belonging to a given marking, a small black dot (a 'token').

A SET-*net* is a tuple $N = (P, T, F, M_0)$ such that (P, T, F) is a net and $M_0 \subseteq P$ is its *initial* marking. A EN-*system* is a tuple $N = (P, T, F, M_0)$ such that (P, T, F) is a net without self-loops and $M_0 \subseteq P$ is its *initial* marking.

The dynamics of SET-nets is defined as follows. Let $N = (P, T, F, M_0)$ be a SET-net and let $t \in T$. Then t is *enabled* at a marking M if ${}^\bullet t \subseteq M$. In such a case, t can occur (*fire*), leading to the marking $M' = (M \setminus {}^\bullet t) \cup t^\bullet$. A subset U of T, a *step*, is *enabled* at M if ${}^\bullet U \subseteq M$. If U is enabled, it can occur, leading to $M' = (M \setminus {}^\bullet U) \cup U^\bullet$.

Hence in a SET-net, a step U is enabled whenever each of its input places belongs to the current marking, in other words, each of its elements is enabled. When U occurs, its input places lose their token, while all output places will be marked. If a place is both input and output for U, it is marked before and after the occurrence of U. Furthermore, output places of U that were marked before its occurrence will remain marked. It is also worthwhile to observe that there may be distinct transitions $t, u \in U$ for which ${}^\bullet t \cap {}^\bullet u \neq \varnothing$ or $t^\bullet \cap u^\bullet \neq \varnothing$. This has no effect on their participation in the occurrence of U.

The dynamics of EN-systems is defined in a similar way, except that the enabling conditions are crucially different. Let $N = (P, T, F, M_0)$ be an EN-system and let $t \in T$. Then t is *enabled* at a marking M if ${}^\bullet t \subseteq M$ and $t^\bullet \cap M = \varnothing$. If t is enabled at M, it can fire which results in the marking $M' = (M \setminus {}^\bullet t) \cup t^\bullet$. A step U of T is *enabled* at M if each $t \in U$ is enabled at M and $({}^\bullet t \cup t^\bullet) \cap ({}^\bullet u \cap u^\bullet) = \varnothing$ for any two distinct transitions $t, u \in U$. Then U can occur leading to $M' = (M \setminus {}^\bullet U) \cup U^\bullet$.

Hence, in an EN-system, if a step U is enabled at marking M then each of its input places is marked and none of its output places is marked. Actually, a step can only ever be enabled if the input/output neighbourhood of the transitions in U do not overlap (i.e., if there is no *structural conflict* in U).

In SET-nets and EN-systems markings are sets and tokens are manipulated using set-based rather than multiset-based arithmetic. We will refer to such Petri net models as being Boolean. In contrast to both SET-nets and EN-systems, and

Boolean nets in general, Place/Transition nets (PT-nets) have a multiset-based arithmetic. [4]

A (*weighted*) PT-*net* is specified as a tuple $N = (P, T, W, M_0)$, where P and T are, as before, finite, disjoint sets of places and transitions; $W : (P \times T) \cup (T \times P) \to \mathbb{N}$ specifies the arcs of N by their weights; and $M_0 : P \to \mathbb{N}$ is the initial marking. In general, for PT-nets, markings are multisets, rather than sets. In diagrams, whenever $W(x, y) \geq 1$ for some $(x, y) \in (T \times P) \cup (P \times T)$, then (x, y) is an arc from x to y; it is annotated with $W(x, y)$ if this is 2 or more. Given a marking M of N and a place $p \in P$, we say that $M(p)$ is the number of tokens in p. Note that the dot-notations are now multisets, indicating the multiplicity of each input/output element.

A transition t of N is *enabled* at a marking M of N, if ${}^\bullet t \leq M$. If t is enabled at M, it can fire which leads to the new marking $M' = M - {}^\bullet t + t^\bullet$. Thus M' is obtained from M by deleting $W(p, t)$ tokens from each place p and adding $W(t, p)$ tokens to each place p. A step of a PT-net N is a multiset of transitions. Step U is *enabled* at a marking M of N if $\sum_{t \in T} U(t) \cdot {}^\bullet t \leq M$. Thus, in order for U to be enabled at M, for each place p, the number of tokens in p under M should at least be equal to the accumulated number of tokens needed as input to each of the transitions in U, respecting their multiplicities in U. If U is enabled at M, it may occur which leads to $M' = M - \sum_{t \in T} U(t) \cdot {}^\bullet t + \sum_{t \in T} U(t) \cdot t^\bullet$. Thus the effect of executing U is the accumulated effect of executing each of its transitions (taking into account their multiplicities in U). Note that there is no concept of structural conflict in the class of PT-nets: transitions may occur together in a step whenever a marking supplies enough tokens.

Inhibitor and activator arcs. To each of the above net models we can add *inhibitor arcs* and *activator arcs* connecting places to transitions, by adding relations *Inh* and *Act* to their specification. Given the set of places P and set of transitions T of a SET-net, or EN-system, or PT-net, $Inh, Act \subseteq P \times T$ define its set of *inhibitor* or *activator* arcs, respectively. For each transition $t \in T$, we denote ${}^\circ t = \{p \mid (p, t) \in Inh\}$ for the set of inhibitor places of t and ${}^\blacklozenge t = \{p \mid (p, t) \in Act\}$ for its activator places. (Both notions are extended to sets of transitions, and to multisets, disregarding multiplicities.)

The intuition behind these *context arcs* is that in order for a transition to be enabled at a marking, its activator places should be marked (have at least one token) and its inhibitor places should not be marked (have no token). Thus the dynamics of these extended net classes is adapted in the following way: a step U is *enabled* at a marking when it is enabled in the underlying SET-net, or EN-system, or PT-net, <u>and</u> ${}^\blacklozenge U \subseteq M$ <u>and</u> ${}^\circ U \cap M = \varnothing$. When U is enabled at M and it occurs, then the resulting marking is defined as before (here the activator and inhibitor arcs have no effect). Note that this semantics is an *a priori semantics* (see, e.g., [12])

[4] A multiset μ over a set X is a function $\mu : X \to \mathbb{N} = \{0, 1, 2, \ldots\}$; such multiset may be represented by listing its elements with repetitions. Sets can be considered as multisets without repetitions.

3 Connections and connection monoids

Now we are ready to recall the general setup of [1, 2] in which Petri net classes are defined on the basis of individual connections between places and transitions. Moreover, the effect of the simultaneous execution of a *step* (a set or multiset of transitions) on a given place is calculated using a dedicated commutative monoid which returns the composite connection between that place and the step. For Boolean nets, we will assume that each step is a set of transitions rather than a multiset as in [1, 2]. This simplifies the presentation and is harmless as Boolean Petri nets as we consider them here, would not allow true multiset steps anyway.

Connection monoids describe the relation between a place and a step. A *connection monoid* (or CONN-monoid) is a set $\mathbb{S}$ of *connections* with a commutative and associative binary composition operation $\oplus$, and a neutral element (identity) $\mathbf{0}$. We will use the same symbol $\mathbb{S}$ for a CONN-monoid and for its underlying set of connections. Moreover, for each $s \in \mathbb{S}$ we let $\bigoplus^0 s = \mathbf{0}$ and $\bigoplus^{n+1} s = (\bigoplus^n s) \bigoplus s$ for all $n \in \mathbb{N}$.

Let $\mathbb{S}$ be a CONN-monoid. Then, a NET-*type over* $\mathbb{S}$ is a transition system $\tau = (Q, \mathbb{S}, \Delta)$ where Q is a set of *states*, and $\Delta : Q \times \mathbb{S} \to Q$ is a partial function such that $\Delta(q, \mathbf{0}) = q$, for all $q \in Q$. For every state q, the set $enbld_\tau(q) = \{s \mid \Delta(q, s) \text{ is defined}\}$ consists of all connections from $\mathbb{S}$ that are *enabled* at q.

As an example let us consider the CONN-monoid $\mathbb{S}_{PT} = (\mathbb{N}\times\mathbb{N}, \oplus, \mathbf{0})$ with $\mathbf{0} = (0, 0)$ and point-wise arithmetic addition $\oplus$. Using this monoid, the connections between places and multisets of transitions in PT-nets can be expressed through the NET-type $\tau_{PT} = (\mathbb{N}, \mathbb{S}_{PT}, \Delta_{PT})$ over $\mathbb{S}_{PT}$, where $\Delta_{PT} = \{(n, (m, k)) \mapsto n - m + k \mid n \geq m \geq 0\}$. Intuitively, this states that a place containing n tokens enables steps which take no more than n tokens, and that the resulting number of tokens is $n-m+k$ where m and k are the numbers of tokens taken and produced, respectively, by all occurrences of transitions in that step together. A fragment of τ_{PT}, interpreted as a labelled directed graph, is shown in Figure 1(d).

In general, each NET-type $\tau = (Q, \mathbb{S}, \Delta)$ defines a class of nets, the so-called τ-*nets*. The NET-type specifies through Q the values that can be assigned to places; through the connections in $\mathbb{S}$ the effect of combining connections; and through Δ, the enabling conditions and newly generated values.

A τ-*net* is a tuple $N = (P, T, F, M_0)$, where P and T are, respectively, disjoint finite sets of places and transitions, $F : (P \times T) \to \mathbb{S}$ is a *connection mapping*, and M_0 is the *initial marking* of N (in general, a marking is a mapping from P to Q). For a place p of N and a step U of transitions, we define the composite connection between U and p by $F(p, U) = \bigoplus_{t\in T}(\bigoplus^{U(t)} (F(p, t))$. Thus, if U is a set, then $F(p, U) = \bigoplus_{t\in U} F(p, t)$ and $F(p, \varnothing) = \bigoplus_{t\in\varnothing} F(p, t) = \mathbf{0}$.

A step U is *(resource) enabled* at a marking M if $F(p, U) \in enbld_\tau(M(p))$ for every place $p \in P$. The *firing* of such a step produces the marking M' such that $M'(p) = \Delta(M(p), F(p, U))$, for every place $p \in P$. The *concurrent reachability graph* $CRG(N)$ of N is formed by firing inductively from M_0 all possible (resource) enabled steps of N.

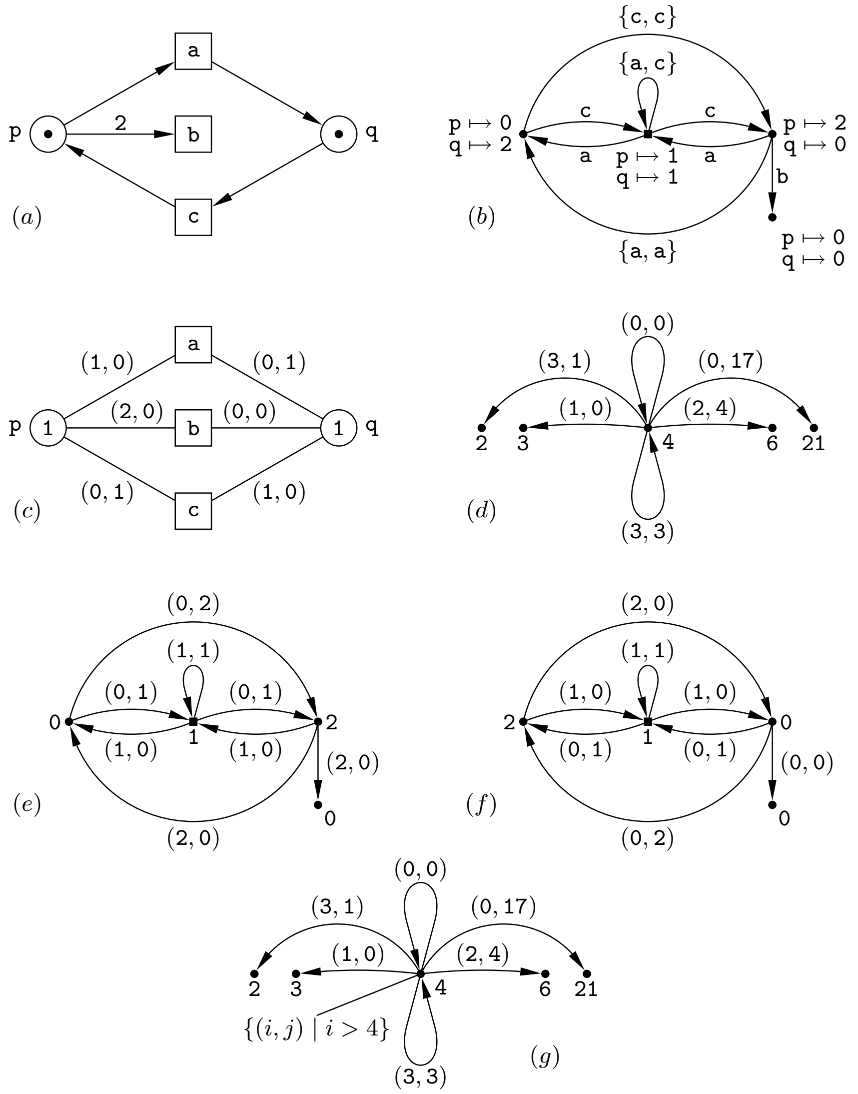


Fig. 1. A PT-net (a); its concurrent reachability graph (b) with the initial state represented by a small square; and its rendering as a τ_{PT}-net system (c). A fragment of the NET-type τ_{PT} is shown in (d). In (e) and (f) we re-trace in (b) the behaviour of places p and q, respectively, in terms of the NET-type τ_{PT}. An alternative graphical representation of the NET-type τ_{PT} is shown in (g).

The NET-type τ_{PT} defines τ_{PT}-nets. In order to view a PT-net as a τ_{PT}-net, all one needs to do is to associate integers, representing the number of tokens, with each place, and set $F(p,t) = (W(p,t), W(t,p))$, for all places p and transitions t. The CONN-monoid $\mathbb{S}_{PT}$ together with the transition system τ_{PT}

provides accurate information about the enabling and firing of steps U. Indeed, all one needs to do is to calculate $F(p, U) = (inwgt, outwgt)$ using the monoid operation of point-wise addition (for all input and output weights of all multiple occurrences of transitions in U).

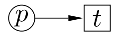
As an illustration, let us consider the PT-net depicted in Figure $1(a)$. This PT-net is represented by the τ_{PT}-net in Figure $1(c)$. Notice that, in particular, $F(\mathtt{q}, \mathtt{b}) = (\mathtt{0}, \mathtt{0})$ means that q and b in Figure $1(a)$ are disconnected. Also the markings are represented (by indicating the number of tokens by an appropriate integer 0, 1, 2, etc.). Figure $1(b)$ gives the concurrent reachability graph. We furthermore obtain $F(\mathtt{p}, \{\mathtt{a}, \mathtt{c}\}) = (\mathtt{1}, \mathtt{0}) \oplus (\mathtt{0}, \mathtt{1}) = (\mathtt{1}, \mathtt{1})$ and $F(\mathtt{q}, \{\mathtt{a}, \mathtt{c}\}) = (\mathtt{0}, \mathtt{1}) \oplus (\mathtt{1}, \mathtt{0}) = (\mathtt{1}, \mathtt{1})$ which, together with $\Delta_{PT}(\mathtt{1}, (\mathtt{1}, \mathtt{1})) = \mathtt{1}$, means that: (i) the net in Figure $1(a)$ enables the step $\{\mathtt{a}, \mathtt{c}\}$ at the initial marking at which both p and q have one token; and (ii) its firing results in the same marking. On the other hand the step $\{c, c\}$ is not enabled at the initial marking because $F(\mathtt{q}, \{\mathtt{c}, \mathtt{c}\}) = (\mathtt{1}, \mathtt{0}) \oplus (\mathtt{1}, \mathtt{0}) = (\mathtt{2}, \mathtt{0})$ and $(2, \mathtt{0})$ is not enabled at 1. However, it is enabled at the marking M with $M(q) = 2$ and $M(p) = 0$ and then its firing results in the marking M' with $M'(q) = 0$ and $M'(p) = 2$. Now focus in the concurrent reachability graph in Figure $1(b)$ on the local markings of the place p in combination with the connections that lead to changes of those local markings. We can do this by labelling each state with the corresponding marking of p, and each arc with the cumulative arc weight w.r.t. p of the step U labelling that arc i.e., $F(\mathtt{p}, U)$. The result is shown in Figure $1(e)$. Repeating the same procedure for the place q, yields Figure $1(f)$. Note that both graphs in Figure $1(e)$ and (f) can also be seen in the graph of the NET-type τ_{PT}, see Figure $1(d)$.

4 Connection monoids for en-systems and set-nets

Starting from EN-systems, we will now present a number of specific classes of Boolean nets defined on basis of their place-transition connections. In what follows we describe the structure of the connection monoids by a Cayley table displaying the outcome of all possible combinations of connections.

EN-systems
In EN-systems, there are three basic connections between places and transitions:

- $F(p, t) = \top$ p and t are disconnected (independent) 

- $F(p, t) = \mathtt{in}$ there is an arc from p to t 

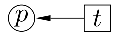
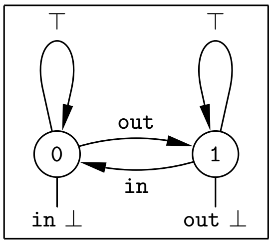
- $F(p, t) = \mathtt{out}$ there is an arc from t to p 

Figure $2(a)$ depicts τ_{EN}, the NET-type showing how the connections between a place and a transition in an EN-system determine the enabledness of the transition w.r.t. that place and the resulting marking if it fires. In the diagram, 0 and 1 mean that the place is respectively *empty* (i.e., not marked) and *full* (marked). Thus, if the place p is marked and there is an arc from p to the transition t $(F(p, t) = \mathtt{in})$, then t may fire as far as p is concerned and the effect will be that p is empty after the occurrence of t. If p and t are not connected, t may always occur from the point of view of p and its occurrence has no effect on the marking of p. There is also an explicit reference to $F(p,t) = \texttt{in}$ for the case that p is empty and one to $F(p,t) = \texttt{out}$ when p is full. In these cases the marking of p prohibits the enabledness of t. In addition to the three standard types of connections, τ_{EN} has a special 'blocking' connection $\bot$ which does not label any arc (is never enabled), hence $\bot \notin \mathit{enbld}_{\tau_{EN}}(\texttt{0}) \cup \mathit{enbld}_{\tau_{EN}}(\texttt{1})$. The connection $\bot$ is also used to capture *structural conflict* between transitions. As such it is a convenient device to capture precisely those steps which are not allowed, because of the internal conflicting relations between their transitions w.r.t. the place.

The CONN-monoid $\mathbb{S}_{EN} = (\{\top, \texttt{out}, \texttt{in}, \bot\}, \oplus_{EN}, \top)$. is defined through its Cayley table in Figure 2(b). Here $\texttt{out} \oplus_{EN} \texttt{out} = \texttt{out} \oplus_{EN} \texttt{in} = \texttt{in} \oplus_{EN} \texttt{out} = \texttt{in} \oplus_{EN} \texttt{in} = \bot$ corresponds directly to the requirement that the neighbourhoods of transitions in a step must be disjoint for it to be enabled to occur.

For example, if we have two transitions, t and u, both removing tokens from place p, $\boxed{t} \leftarrow (p) \rightarrow \boxed{u}$, thus both have p as an input place, then the connection of the step $\{t,u\}$ w.r.t. p is calculated as $F(p,\{t,u\}) = F(p,t) \oplus_{EN} F(p,u) = \texttt{in} \oplus_{EN} \texttt{in} = \bot$ implying that $\{t,u\}$ can never occur together (on account of p).



(a)

$\oplus_{EN}$	$\top$	out	in	$\bot$
$\top$	$\top$			
out	out	$\bot$		
in	in	$\bot$	$\bot$	
$\bot$	$\bot$	$\bot$	$\bot$	$\bot$

(b)

$\oplus_{EN}$	out	in
out	$\bot$	
in	$\bot$	$\bot$

(c)

Fig. 2. NET-type τ_{EN}, and the Cayley table of $\mathbb{S}_{EN}$.

In all CONN-monoids , $\top$ will be the identity element and — if present — $\bot$ is the absorbing element. The monoid $\mathbb{S}_{EN}$ is the most restrictive monoid over $\top$, $\texttt{in}$, $\texttt{out}$, and $\bot$, because its operation does not yield any non-$\bot$ results except when $\top$ is involved. This is clearly seen in Figure 2(c) which depicts the non-trivial part of the Cayley table from Figure 2(b), while omitting the values implicitly due to commutativity. In what follows we will present CONN-monoids using a minimal presentation of their Cayley table as in Figure 2(c).

A *basis* of a CONN-monoid is any irreducible subset of its non-$\bot$ connections such that any other non-$\bot$ connection can be derived from it.

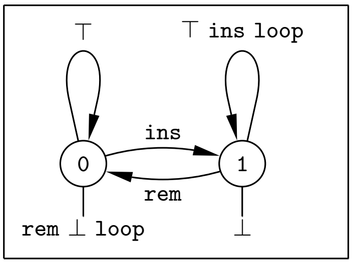
Proposition 1. *$\{\top, \texttt{in}, \texttt{out}\}$ is the only basis of $\mathbb{S}_{EN}$.*

Proof. Follows directly from the table in Figure 2(c). □

SET-nets
Now there are four basic connections between places and transitions:

- $F(p,t) = \top$ p and t are disconnected (independent)
- $F(p,t) = \texttt{rem}$ there is an arc from p to t
- $F(p,t) = \texttt{ins}$ there is an arc from t to p
- $F(p,t) = \texttt{loop}$ there is an arc from t to p, and from p to t

Figure 3(a) depicts τ_{SN}. Comparing Figure 3(a) and Figure 2(a) brings to light the important difference between the meaning of an arc from a transition to a place in EN-systems (connection `out`) and the meaning of an arc from a transition to a place in SET-nets (connection `ins`).



(a)

$\oplus_{SN}$	ins	rem	loop
ins	ins		
rem	loop	rem	
loop	loop	loop	loop

(b)

Fig. 3. NET-type τ_{SN}, and the simplified table of $\mathbb{S}_{SN} = (\{\top, \texttt{ins}, \texttt{rem}, \texttt{loop}\}, \oplus_{SN}, \top)$.

Figure 4 shows the `out`-labelled arc in τ_{EN} and the `ins`-labelled arc in τ_{SN}.

Fig. 4. Difference between arcs from transitions to places in EN-systems and SET-nets.

The simplified Cayley table of the CONN-monoid $\mathbb{S}_{SN}$ is shown in Figure 3(b). From the table we see, e.g., that if p is an output place of a transition t and input place to u, $\boxed{t} \dashrightarrow (p) \dashrightarrow \boxed{u}$, then the connection of the step $\{t,u\}$ w.r.t. p is given by $F(p,\{t,u\}) = F(p,t) \oplus_{SN} F(p,u) = \texttt{ins} \oplus_{SN} \texttt{rem} = \texttt{loop}$ and so, as far as p is concerned, $\{t,u\}$ can occur if p contains a token; moreover, p will also have a token after the occurrence of $\{t,u\}$.

Another important property of the $\mathbb{S}_{SN}$ monoid is the idempotence of its operation (the diagonal of the Cayley table of $\mathbb{S}_{SN}$). This reflects one of the main features of SET-nets, namely that since resources are not quantified, they can be used by many transitions with the same connectivity in tandem, as though they were just one such transition. Note furthermore, that since SET-nets know no structural conflict, $\perp$ is not introduced through $\oplus_{SN}$. Consequently, $\perp$ is not necessary in the case of τ_{SN} and $\mathbb{S}_{SN}$. Actually, also $\mathbb{S}_{PN}$ did not need $\perp$, as PT-nets know no structural conflicts either. However, if we consider the class of k-bounded PT-nets then the situation is rather different as the corresponding CONN-monoid is defined as $\mathbb{S}_{BPT} = (\{\perp\} \cup \mathbb{N}_k \times \mathbb{N}_k, +_k, (0,0))$, where $\mathbb{N}_k = \{0, 1, \ldots, k\}$, $\perp$ is the absorbing element, and, for all $n, m, n', m' \in \mathbb{N}_k$,

$$(n,m) +_k (n',m') = \begin{cases} (n+n', m+m') & \text{if } n+n' \leq k \ \wedge \ m+m' \leq k \\ \perp & \text{otherwise .} \end{cases}$$

Proposition 2. $\{\top, \mathtt{ins}, \mathtt{rem}\}$ *is the only basis of* $\mathbb{S}_{SN}$.

Proof. Follows directly from the table in Figure 3(b). □

This insight forms a formal justification of the way in which τ_{SN}-nets are drawn: direct dashed arrows are used for `ins` and `rem`, but `loop` as a 'compound' connection can be depicted by the 'compound' representation for `ins` and `rem`. Note that in the $\perp$-less version of $\mathbb{S}_{SN}$, `loop` is the absorbing element, but this will change when we add inhibitor arcs. First however, we add inhibitor arcs to EN-systems.

EN-systems with inhibitor arcs

In comparison with EN-systems, we now have one more connection to take into account:

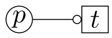
– $F(p,t) = \mathtt{inh}$ there is an inhibitor arc from p to t 

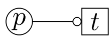
Figure 5 shows the NET-type τ_{ENI}, and the simplified Cayley table of the CONN-monoid $\mathbb{S}_{ENI} = (\{\top, \mathtt{out}, \mathtt{in}, \mathtt{inh}, \perp\}, \oplus_{ENI}, \top)$. From this we see that the monoid $\mathbb{S}_{ENI}$ captures an additional type of structural conflict: $\mathtt{in} \oplus_{ENI} \mathtt{inh} = \mathtt{inh} \oplus_{ENI} \mathtt{in} = \perp$. That $\mathtt{out} \oplus_{ENI} \mathtt{inh} = \mathtt{inh} \oplus_{ENI} \mathtt{out} = \mathtt{out}$ is a consequence of the a priori semantics.

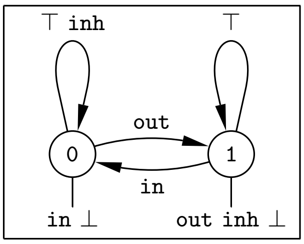
Proposition 3. $\{\top, \mathtt{in}, \mathtt{out}, \mathtt{inh}\}$ *is the only basis of* $\mathbb{S}_{ENI}$.

Proof. Follows directly from the table in Figure 5(b). □

SET-nets with inhibitor arcs

Again, we have to cater for one additional connection:

– $F(p,t) = \mathtt{inh}$ there is an inhibitor arc from p to t 



(a)

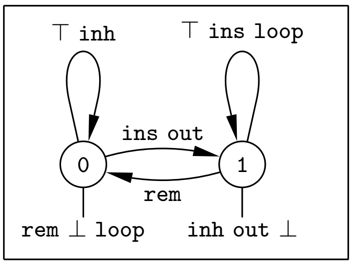
$\oplus_{ENI}$	out	in	inh
out	$\bot$		
in	$\bot$	$\bot$	
inh	out	$\bot$	inh

(b)

Fig. 5. NET-type τ_{ENI} (a), and the simplified Cayley table of $\mathbb{S}_{ENI}$ (b).

Figure 6 shows the NET-type τ_{SNI}, and the simplified Cayley table of the CONN-monoid $\mathbb{S}_{SNI} = (\{\top, \texttt{ins}, \texttt{rem}, \texttt{loop}, \texttt{inh}, \texttt{out}, \bot\}, \oplus_{SNI}, \top)$. In this case we do need $\bot$ as structural conflicts occur when inhibitors are combined with consumption (exactly as in EN-systems). Thus $\mathbb{S}_{ENI}$ captures a conflict: $\texttt{rem} \oplus_{SNI} \texttt{inh} = \texttt{inh} \oplus_{SNI} \texttt{rem} = \bot$.

Furthermore, the monoid must be closed w.r.t. its operation and so due to the a priori step semantics for SNI-nets, out had to be added as a new connection to describe $\texttt{ins} \oplus_{SNI} \texttt{inh}$. Notice that out, although on its own has the same meaning here as in EN-systems and ENI-systems, it is understood differently when combined with other connections. An example is $\texttt{out} \oplus_{SNI} \texttt{out} = \bot$ rather than $\texttt{out} \oplus_{ENI} \texttt{out} = \texttt{out}$ since the step semantics of SNI-nets is different from that of ENI-systems.



(a)

$\oplus_{SNI}$	ins	rem	inh	loop	out
ins	ins				
rem	loop	rem			
inh	out	$\bot$	inh		
loop	loop	loop	$\bot$	loop	
out	out	$\bot$	out	$\bot$	out

(b)

Fig. 6. NET-type τ_{SNI} (a), and the simplified Cayley table of $\mathbb{S}_{SNI}$ (b).

Proposition 4. $\{\top, \texttt{ins}, \texttt{rem}, \texttt{inh}\}$ *is the only basis of* $\mathbb{S}_{SNI}$.

Proof. Follows directly from the table in Figure 6(b). □

Like for $\mathbb{S}_{SNI}$, also the operation of $\mathbb{S}_{SNI}$ is idempotent. Even stronger:

Proposition 5. *Let $T \neq \varnothing$ be a set of transitions and $\mathbb{T} = \{F(p,t) \mid t \in T\}$.*

$$F(p,T) = \begin{cases} \top & \text{if } \ \mathbb{T} = \{\top\} \\ \texttt{ins} & \text{if } \ \mathbb{T} \subseteq \{\texttt{ins}, \top\} \qquad\qquad \wedge \ \texttt{ins} \in \mathbb{T} \\ \texttt{rem} & \text{if } \ \mathbb{T} \subseteq \{\texttt{rem}, \top\} \qquad\qquad \wedge \ \texttt{rem} \in \mathbb{T} \\ \texttt{inh} & \text{if } \ \mathbb{T} \subseteq \{\texttt{inh}, \top\} \qquad\qquad \wedge \ \texttt{inh} \in \mathbb{T} \\ \texttt{out} & \text{if } \ \mathbb{T} \subseteq \{\texttt{inh}, \texttt{ins}, \texttt{out}, \top\} \ \wedge (\texttt{out} \in \mathbb{T} \ \vee \{\texttt{inh}, \texttt{ins}\} \subseteq \mathbb{T}) \\ \texttt{loop} & \text{if } \ \mathbb{T} \subseteq \{\texttt{ins}, \texttt{rem}, \texttt{loop}, \top\} \wedge (\texttt{loop} \in \mathbb{T} \vee \{\texttt{rem}, \texttt{ins}\} \subseteq \mathbb{T}) \\ \bot & \textit{otherwise} \ . \end{cases}$$

Proof. Follows directly from the table in Figure 6(b). The table shows that $\oplus_{SNI}$ is idempotent and that $\{\texttt{ins}, \texttt{rem}, \texttt{inh}, \top\}$ is $\mathbb{S}_{SNI}$'s basis. □

EN-systems with inhibitor and activator arcs

The last connection we consider is:

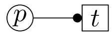
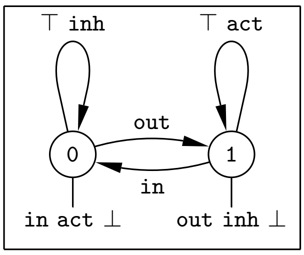
– $F(p,t) = \texttt{act}$ there is an activator arc from p to t 

Figure 7 shows the NET-type τ_{ENC}, and the simplified Cayley table of the CONN-monoid $\mathbb{S}_{ENC}$ for ENC-systems (with the a priori step semantics).



(a)

$\oplus_{ENC}$	out	in	inh	act
out	$\bot$			
in	$\bot$	$\bot$		
inh	out	$\bot$	inh	
act	$\bot$	in	$\bot$	act

(b)

Fig. 7. NET-type τ_{ENC} (a), and the simplified Cayley table of $\mathbb{S}_{ENC}$ (b).

In the simplified Cayley table of $\mathbb{S}_{ENC}$, we see that $\texttt{inh} \oplus_{ENC} \texttt{out} = \texttt{out}$ and $\texttt{act} \oplus_{ENC} \texttt{in} = \texttt{in}$. These pairs of connections reflect that while the transitions involved are enabled with respect to the given place (which should be empty for the inh and out connections, and marked for the act and in connections) they affect it in a different way. The connections that induce a state change (out and in) are 'stronger' , while the connections designated for testing (inh and act) are 'weaker'.

Proposition 6. $\{\top, \texttt{out}, \texttt{in}, \texttt{inh}, \texttt{act}\}$ *is the only basis of* $\mathbb{S}_{ENC}$.

Proof. Follows directly from the table in Figure 7(b). □

SET-nets with inhibitor and activator arcs
Again we add an activator connection:

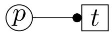
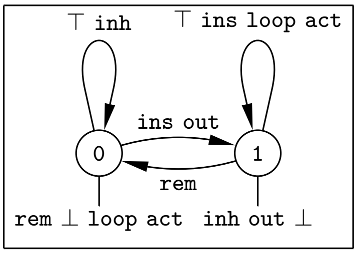
– $F(p,t) = \texttt{act}$ there is an activator arc from p to t 

Figure 8 shows the NET-type τ_{SNC}, and the simplified Cayley table of the CONN-monoid $\mathbb{S}_{SNC} = (\{\top, \texttt{ins}, \texttt{rem}, \texttt{inh}, \texttt{act}, \texttt{loop}, \texttt{out}, \bot\}, \oplus_{SNC}, \top)$ for SET-nets with inhibitor and activator arcs (under the a priori step semantics).



(a)

$\oplus_{SNC}$	ins	rem	inh	act	loop	out
ins	ins					
rem	loop	rem				
inh	out	$\bot$	inh			
act	loop	rem	$\bot$	act		
loop	loop	loop	$\bot$	loop	loop	
out	out	$\bot$	out	$\bot$	$\bot$	out

(b)

Fig. 8. NET-type τ_{SNC} (a), and the simplified Cayley table of $\mathbb{S}_{SNC}$ (b).

Proposition 7. $\{\top, \texttt{ins}, \texttt{rem}, \texttt{inh}, \texttt{act}\}$ *is the only basis of* $\mathbb{S}_{SNI}$.

Proof. Follows directly from the table in Figure 8(b). □

There are 9 different patterns for the various connections: from `0` (unmarked) and from `1` (marked), either to `0` or to `1`, or undefined; see Figure 9.

In each CONN-monoid considered before, different connections had different topological patterns in the associated NET-type. This changes now, as in τ_{SNC} both `loop` and `act` give rise to the same pattern. The effect of combining `act` and `loop` however is `loop` rather than `act`. This is because, according to the step semantics of SET-nets, adding tokens happens after removing or testing. So, in this combination, `loop` as a connection that induces a change of the state is 'stronger' than `act`. Another interesting pair in the table is formed by `act` and `rem`. The effect of composing them is `rem` which differs from `in` due to the different underlying step semantics even though the arc pattern of `rem` in τ_{SNC} and that of `in` in τ_{ENC} are the same.

Thus we arrive at the crucial point in our considerations where it becomes clear that the sophisticated (and sometimes surprising) nature of different connections necessarily involves algebraic properties in addition to topological ones.

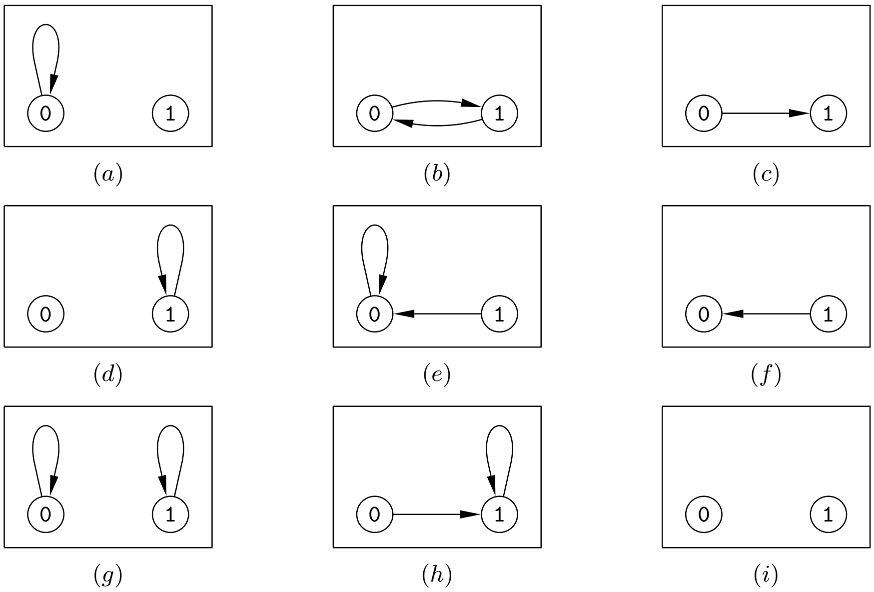


Fig. 9. Topological patterns of connections in Boolean τ-nets.

5 General Boolean τ-nets

We now propose a general classification of *all* possible connections in Boolean τ-nets. We take the general view that each connection defines enabling and effect, and has an associated strength (weak or strong).

Weak connections impose constraints only with respect to enabling, but unlike strong connections, they do not impose constraints on the state resulting from the transition firing. So, when combined with a transition with a stronger connection, it is the latter that dictates the final result. For example, `loop` is strong in $\mathbb{S}_{SN}$ as it 'finishes' by adding a token in operational sense, as this is supported by the algebraic property of absorbtion. `rem`, on the other hand, is weak in $\mathbb{S}_{SN}$ as with this connection the enabling conditions are important, but the state of the net place (connected in this way to some transition) after the transitions fired may be changed by another transition (removing tokens or testing precedes token insertion). This leads to 25 different connections ∂_{xy} where $x, y \in \{\mathtt{w}, \mathtt{s}, \overline{\mathtt{w}}, \overline{\mathtt{s}}, \mathtt{n}\}$. Here x refers to arrows outgoing from `0`, and y refers to arrows outgoing from `1`; `w` means a weak arrow, `s` a strong arrow, and `n` no arrow (non-enabledness); finally, $\overline{(.)}$ implies changing the state (from `0` to `1` or vice versa). In particular, we have the following encoding of the previously discussed connections:

$\bot$	$\top$	`in`	`out`	`ins`	`rem`	`loop`	`inh`	`act`
$\partial_{\mathtt{nn}}$	$\partial_{\mathtt{ww}}$	$\partial_{\mathtt{n}\overline{\mathtt{s}}}$	$\partial_{\overline{\mathtt{s}}\mathtt{n}}$	$\partial_{\overline{\mathtt{s}}\mathtt{s}}$	$\partial_{\mathtt{n}\overline{\mathtt{w}}}$	$\partial_{\mathtt{ns}}$	$\partial_{\mathtt{wn}}$	$\partial_{\mathtt{nw}}$

The corresponding topological patterns are shown in Figure 10.

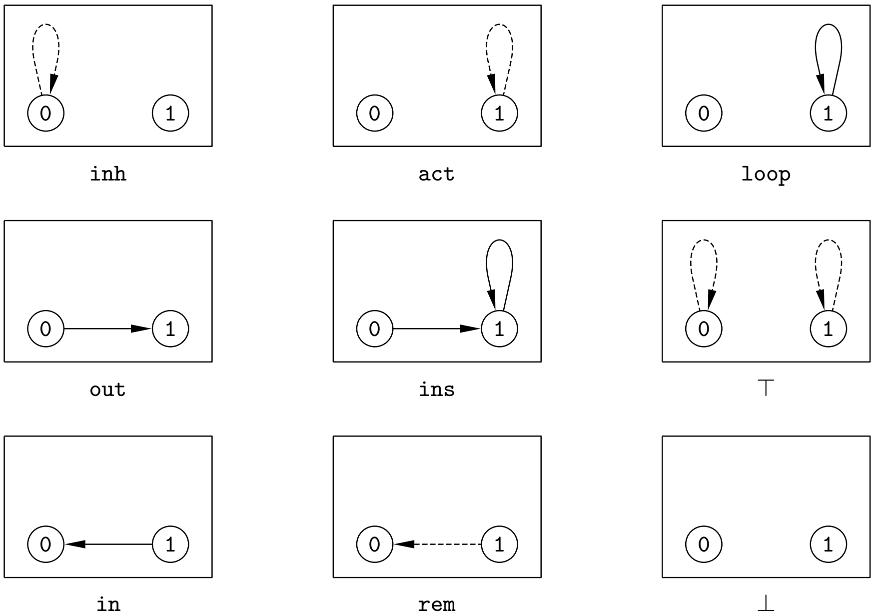


Fig. 10. Connections in Boolean τ-nets with weak arcs indicated by dashed lines.

We will now formalise in a general way algebraic operations on the 25 connections. Almost all will be motivated by enabledness, and the idea of weak and strong. In this way, associativity will be automatic in most cases. Let $\partial_{xy} \oplus \partial_{x'y'} = \partial_{x \odot x' \; y \odot y'}$. where $\odot$ is a commutative operation given by:

$\odot$	w	$\overline{\mathtt{w}}$	s	$\overline{\mathtt{s}}$	n
w	w				
$\overline{\mathtt{w}}$	$\overline{\mathtt{w}}$	$\overline{\mathtt{w}}$			
s	s	s	s		
$\overline{\mathtt{s}}$	$\overline{\mathtt{s}}$	$\overline{\mathtt{s}}$	n	$\overline{\mathtt{s}}$	
n	n	n	n	n	n

The intuition behind, for example, $\partial_{\mathtt{ss}}$ is that the transition is always enabled but its execution keeps the marking in the place unchanged.

Proposition 8. $\mathbb{S}_{conn} = (\{\mathtt{w}, \mathtt{s}, \overline{\mathtt{w}}, \overline{\mathtt{s}}, \mathtt{n}\}, \odot, \mathtt{w})$ *is a commutative monoid.*

Proof. We need to show that $(a \odot b) \odot c = a \odot (b \odot c)$ for all $a, b, c \in \{\mathtt{w}, \mathtt{s}, \overline{\mathtt{w}}, \overline{\mathtt{s}}, \mathtt{n}\}$. To start with, if $\mathtt{n} \in \{a, b, c\}$ then $(a \odot b) \odot c = \mathtt{n} = a \odot (b \odot c)$. Otherwise, we observe that the following hold:

- If $\mathtt{s} \in \{a, b, c\}$ and $\overline{\mathtt{s}} \notin \{a, b, c\}$ then $(a \odot b) \odot c = \mathtt{s} = a \odot (b \odot c)$.
- If $\overline{\mathtt{s}} \in \{a, b, c\}$ and $\mathtt{s} \notin \{a, b, c\}$ then $(a \odot b) \odot c = \overline{\mathtt{s}} = a \odot (b \odot c)$.

$\oplus_{ENC}$	out	in	inh	act
out	$\perp$			
in	$\perp$	$\perp$		
inh	out	$\perp$	inh	
act	$\perp$	in	$\perp$	act

$\oplus_{ENC}$	out	in	inh	act
out	$\perp$			
in	$\perp$	$\perp$		
inh	$\perp$	$\perp$	inh	
act	$\perp$	$\perp$	$\perp$	act

Fig. 11. a-priori (a) and a-posteriori (b) semantics for ENC-systems.

– If $\overline{\mathtt{s}} \in \{a, b, c\}$ and $\mathtt{s} \in \{a, b, c\}$ then $(a \odot b) \odot c = \mathtt{n} = a \odot (b \odot c)$.
– If $a, b, c \in \{\mathtt{w}, \overline{\mathtt{w}}\}$ and $\overline{\mathtt{w}} \in \{a, b, c\}$ then $(a \odot b) \odot c = \overline{\mathtt{w}} = a \odot (b \odot c)$.
– If $a = b = c = \mathtt{w}$ then $(a \odot b) \odot c = \mathtt{w} = a \odot (b \odot c)$. □

Theorem 1. $\mathbb{S}_{bool} = (\{\partial_{xy} \mid x, y \in \{\mathtt{w}, \mathtt{s}, \overline{\mathtt{w}}, \overline{\mathtt{s}}, \mathtt{n}\}\}, \oplus, \partial_{\mathtt{ww}})$ *is a* CONN*-monoid.*

Proof. Follows from Proposition 8 and $\partial_{xy} \oplus \partial_{x'y'} = \partial_{x \odot x'\ y \odot y'}$. □

$\mathbb{S}_{SN}$, $\mathbb{S}_{SNI}$ and $\mathbb{S}_{SNC}$ are all sub-monoids of $\mathbb{S}_{bool}$. The $\mathbb{S}_{SN}$ sub-monoid is a special one; it is *non-blocking* as composing connections never yields $\perp$.

We can now formally describe Boolean net models as those classes of nets that are defined by a net type over (a submonoid of) $\mathbb{S}_{bool}$. From [2], it follows that thanks to the interpretation of the step semantics in term of monoids, Boolean nets are instances of τ-nets for which there exists a region-based solution to the synthesis problem. Moreover, τ-nets with maximally concurrent semantics (the semantics of SET-nets when used to model reaction systems) fall within the general framework of τ-nets with *policies* introduced in [2].

Finally, we should point out that in order to arrive at this general classification of Boolean nets, we have had to make some (slightly arbitrary) assumptions when the intended operational meaning of a combination was not clear. In particular, $\mathtt{w} \odot \overline{\mathtt{w}}$ has been defined in such a way as to give more priority to the change of state. This and perhaps other assumptions are not cast in stone. With differently motivated models in mind, one may freely modify them, study, appreciate the differences. Also, we decided to define $\mathtt{s} \odot \overline{\mathtt{s}}$ as $\mathtt{n}$ because $\mathtt{s}$ and $\overline{\mathtt{s}}$ are both strong and so 'uncompromising': one changes the state whereas the other insists on preserving the state. This contradiction cannot be reconciled.

A posteriori vs. a priori execution semantics Interestingly, CONN-monoids can distinguish between the 'a priori' semantics defined at the end of Section 2, and the 'a posteriori' execution semantics. In the setting of EN-systems, 'a posteriori' is exactly the same as 'a priori' with one extra condition for an enabled set of transitions: ${}^{\bullet}U \cap {}^{\blacklozenge}U = U^{\bullet} \cap {}^{\circ}U = \varnothing$. Figure 11 exhibits this difference.

6 Conclusions

The reader might wonder why we included in our presentation PT-nets which are clearly non-Boolean nets. Apart from certain didactic motivations, we thought that PT-nets come with a 'calculus of connections' based on a simpler monoid of natural numbers. To our initial surprise, a similar effect can be achieved in our symbolic setting where the monoid of ∂_{xy} connections is completely determined by a simpler monoid with the $\odot$ operation. This could, perhaps, suggest a general approach for constructing practical implementations of synthesis algorithms for SET-nets.

Note that there are variations of Petri nets, such as Boolean Petri nets, where adding a token to an already marked place does not add another token [3, 9]. Also, behaviour of this kind was mentioned in [1] in the context of net synthesis. Having said that, the semantics considered in prior works known to us was based on single transition firings, rather than steps as is the case for SET-nets.

Acknowledgement We would like to thank the anonymous reviewers for their suggestions and comments. This research was supported by the Pascal Chair award from Leiden University and the EPSRC VERDAD project.

References

1. Badouel, E., Darondeau, Ph.: Theory of Regions. In: Part I of [14] (1998) 529–586
2. Darondeau, P., Koutny, M., Pietkiewicz-Koutny, M., Yakovlev, A.: Synthesis of Nets with Step Firing Policies. Fundamenta Informaticae **94** (2009) 275–303
3. De Bra, P., Houben, G.J., Kornatzky, Y.: A Formal Approach to Analyzing the Browsing Semantics of Hypertext. Proc. CSN-94 Conference (1994) 78–89
4. Desel, J., Reisig, W.: Place/Transition Petri Nets. In: Part I of [14] (1998) 122–173
5. Ehrenfeucht, A., Main, M., Rozenberg, G.: Combinatorics of Life and Death for Reaction Systems. International Journal of Foundations of Computer Science **22** (2009) 345–356
6. Ehrenfeucht, A., Rozenberg, G.: Partial 2-structures; Part I: Basic Notions and Representation Problem, and Part II: State Spaces of Concurrent Systems. Acta Informatica **27** (1990) 315–368
7. Ehrenfeucht, A., Rozenberg, G.: Reaction Systems. Fundamenta Informaticae **76** (2006) 1–18
8. Ehrenfeucht, A., Rozenberg, G.: Events and Modules in Reaction Systems. Theoretical Computer Science **376** (2007) 3–16
9. Heiner, M., Gilbert, D., Donaldson, R.: Petri Nets for Systems and Synthetic Biology. Lecture Notes in Computer Science **5016** (2008) 215–264
10. Kleijn, J., Koutny, M., Rozenberg, G.: Modelling Reaction Systems with Petri Nets. Technical Report CS-1244. Newcastle University (2011)
11. Koutny, M., Pietkiewicz-Koutny, M.: Synthesis of Elementary Net Systems with Context Arcs and Localities. Fundamenta Informaticae **88** (2008) 307–328
12. Pietkiewicz-Koutny, M.: The Synthesis Problem for Elementary Net Systems with Inhibitor Arcs. Fundamenta Informaticae **40** (1999) 251–283
13. Rozenberg, G., Engelfriet, J.: Elementary Net Systems. In: Part I of [14] (1998) 12–122
14. Reisig, W., Rozenberg, G. (eds.): Lectures on Petri Nets. Lecture Notes in Computer Science **1491, 1492** (1998)