A Normal Form for Stateful Connectors

Roberto Bruni[1](✉), Hernán Melgratti[2], and Ugo Montanari[1]

[1] Department of Computer Science, University of Pisa, Pisa, Italy
bruni@di.unipi.it
[2] Departamento de Computación, FCEyN, Universidad de Buenos Aires - Conicet,
Buenos Aires, Argentina

Abstract. In this paper we consider a calculus of connectors that allows for the most general combination of synchronisation, non-determinism and buffering. According to previous results, this calculus is tightly related to a flavour of Petri nets with interfaces for composition, called Petri nets with boundaries. The calculus and the net version are equipped with equivalent bisimilarity semantics. Also the buffers (the net places) can be one-place (C/E nets) or with unlimited capacity (P/T nets). In the paper we investigate the idea of finding normal form representations for terms of this calculus, in the sense that equivalent (bisimilar) terms should have the same (isomorphic) normal form. We show that this is possible for finite state terms. The result is obtained by computing the minimal marking graph (when finite) for the net with boundaries corresponding to the given term, and reconstructing from it a canonical net and a canonical term.

Keywords: Algebras of connectors · Petri nets with boundaries

1 Introduction

One of the foci of our long-standing collaboration with José Meseguer has been concerned with the algebraic properties of Petri nets and their computations, exploiting suitable symmetric (strict) monoidal categories [13,14,22,23]. In the context of the ASCENS project[1], we have recently investigated a flavour of composable Petri nets, called *Petri nets with boundaries*, originally proposed by Pawel Sobocinski in [28]. Petri nets with boundaries should not be confused with bounded nets: the former come equipped with left/right interfaces for composition, the latter require the existence of a bound on the number of tokens that can be present in the same place during the computation. Petri nets with boundaries allow to conveniently model stateful connectors in component-based systems and have been related to other widely adopted component-based frameworks, like BIP [4], in [10]. In particular we have shown in [12] that they are

This research was supported by the EU project IP 257414 (ASCENS), EU 7th FP under grant agreement no. 295261 (MEALS), by the Italian MIUR Project CINA (PRIN 2010/11), and by the UBACyT Project 20020130200092BA.

[1] http://www.ascens-ist.eu/.

© Springer International Publishing Switzerland 2015
N. Martí-Oliet et al. (Eds.): Meseguer Festschrift, LNCS 9200, pp. 205–227, 2015.
DOI: 10.1007/978-3-319-23165-5_9

equivalent to the algebra of stateless connectors from [8] extended with one-place buffers. In this paper we consider an algebra of connectors that allow for the most general combination of synchronisation, non-determinism and buffering and investigate the idea of finding a normal form representation for terms of this algebra, under some finiteness hypotheses.

Component-based design is a modular engineering practice that relies on the separation of concerns between coordination and computation. Component-based systems are built from loosely coupled computational entities, the *components*, whose interfaces comprise the number, kind and peculiarities of communication ports. The term *connector* denotes entities that glue the interaction of components [25], by imposing suitable constraints on the allowed communications. The evolution of a network of components and connectors is as if played in rounds: At each round, the components try to interact through their ports and the connectors allow/disallow some of the interactions selectively. A connector is called *stateless* when the interaction constraints it imposes are the same at each round; *stateful* otherwise.

In the case of the algebra of stateless connectors [8], terms are assigned input-output sorts, written $P : (n, m)$ or $P : n \to m$, where n is the arity (i.e., the number of ports) of the left-interface and m of the right-interface. Terms are constructed by composing in series and in parallel five kinds of basic connectors (and their duals, together with identities $\mathsf{I} : (1, 1)$) that express basic forms of (co) monoidal synchronisation and non-determinism: (self-dual) symmetry $\mathsf{X} : (2, 2)$, synchronisation $\nabla : (1, 2)$ and $\Delta : (2, 1)$, mutual exclusion $\wedge : (1, 2)$ and $\vee : (2, 1)$, hiding $\perp : (1, 0)$ and $\top : (0, 1)$, and inaction $\perp : (1, 0)$ and $\top : (0, 1)$. The parallel composition $P_1 \otimes P_2$ of two terms $P_1 : (n_1, m_1)$ and $P_2 : (n_2, m_2)$ has sort $(n_1 + n_2, m_1 + m_2)$ and corresponds to put the two connectors side by side, without interaction constraints between them. The sequential composition $P_1; P_2 : (n, m)$ is defined only if the right-interface k of $P_1 : (n, k)$ matches with the left-interface of $P_2 : (k, m)$ and corresponds to plug together such interfaces, enforcing port-wise synchronisation. It is immediate to see that each term $P : (n, m)$ has a corresponding dual $P^c : (m, n)$ (defined recursively by letting $(P_1 \otimes P_2)^c = P_1^c \otimes P_2^c$ and $(P_1; P_2)^c = P_2^c; P_1^c$) and a normal form axiomatisation is provided in [8] whose equivalence classes form a symmetric strict monoidal category (PROduct and Permutation category, PROP [16,21]) of so-called tick-tables. All such connectors are stateless.

The simplest extension to stateful connectors consists of adding one-place buffers as basic terms: $\bigcirc : (1, 1)$ denotes the empty buffer, willing to receive a "token" when an action is executed on its left port; and $\odot : (1, 1)$ denotes the full buffer, willing to give the "token" away when an action is executed on its right port. This way, certain interactions can be dynamically enabled or disabled depending on the presence or absence of "tokens" in the buffers. Such stateful connectors can be put in correspondence with Petri nets with boundaries up to bisimilarity [9,12,28]. In fact, the operational semantics of connectors and Petri nets with boundaries can be expressed in terms of labelled transition systems (LTS) whose labels are pairs (a, b) with a being a string that describes the actions observed on the ports of the left-interface and b those on the right-interface. In our case a basic action observed on a single port is a natural number, describing the number of firings on which that port is involved, or equivalently, the number of “tokens” travelling on that port; therefore a and b are strings of natural numbers. A transition with such an observation is written $P \xrightarrow[b]{a} P'$. In the case of connectors, states are terms of the algebra, while in the case of nets states are markings. In both cases the “sizes” of the interfaces are preserved by transitions, e.g., if $P \xrightarrow[b]{a} P'$ and $P : (n, m)$, then $|a| = n$, $|b| = m$ and $P' : (n, m)$. Interestingly, the abstract semantics induced by ordinary bisimilarity over such LTS is a congruence w.r.t. sequential and parallel composition. Regarding the correspondence, first, it is shown that any net $N : m \to n$ with initial marking X can be associated with a connector $\boldsymbol{T}_{N_{\mathcal{X}}} : (m, n)$ that preserves and reflects the semantics of N. Conversely, for any connector $P : (m, n)$ there exists a bisimilar net $\{[P]\} : m \to n$ defined by structural recursion on P. Roughly, in both cases, the one-place buffers of the connector correspond to the places of the Petri net.

The problem of finding an axiomatisation for stateful connectors such that normal forms can be found for bisimilarity classes is complicated by the fact that the number of buffers is not preserved by bisimilarity: the same “abstract state” can be described by a different combination of places. As a simple example, take a net with two transitions α and β and a place p whose pre-set is $\{\alpha\}$ and whose post-set is $\{\beta\}$. Clearly if p is substituted by any number of places connected in the same way to α and β, then the overall behaviour is not changed.

The solution provided here is to translate a term P to the corresponding net $\{[P]\}$. Then we build the marking graph of $\{[P]\}$. It must be finite because only a finite number of markings exist. Moreover we observe that marking graphs can be represented up to bisimilarity by a Petri net with boundaries that has one place for each reachable marking of $\{[P]\}$ (i.e., one place for each state of the marking graph). Finally, the translation of such net to the corresponding connector gives a canonical representation of P, in the sense that any other term P' bisimilar to P will yield the same term (up to suitable permutations).

The same procedure can be followed when Place/Transition (P/T) Petri nets with boundaries are considered. In this case, places capacity is unconstrained, i.e., a place can contain any number of tokens. Correspondingly, we start from terms of the P/T Petri calculus, where the basic constructors $\bigcirc$ and $\odot$ are replaced by a denumerable set of constructors $(\!| n |\!)$ for any natural number n, each representing a buffer with n tokens. Given the correspondence in [12], between P/T Petri calculus terms and P/T nets with boundaries, we can again translate a term P to the corresponding net $\{[P]\}$, but building a finite marking graph of $\{[P]\}$ requires the net to be bounded.[2] This is equivalent to require that only a finite set of terms is reachable from the term P via transitions. The marking graph can then be minimised (w.r.t. the number of states, up to bisimilarity) and translated to an equivalent P/T Petri calculus term.

[2] Formally, a net is *bounded* if $\exists k \in \mathbb{N}$ such that in any reachable marking the number of tokens in any place is less than or equal to k, i.e., k is a bound for the capacity of places. Note that the marking graph of a net is finite iff the net is bounded.

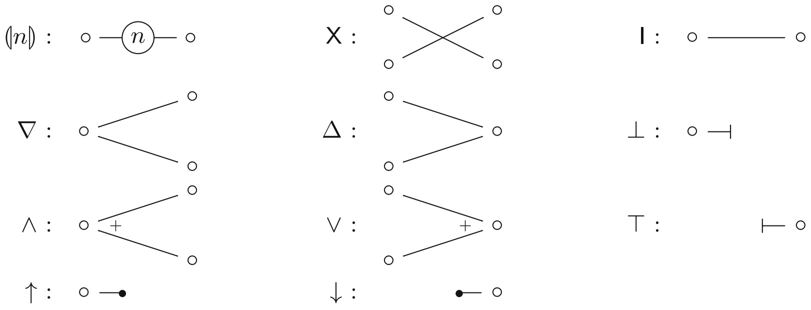


Fig. 1. Graphical representation of terms

Structure of the Paper. Section 2 introduces the P/T and the C/E Petri calculi, together with their bisimilarity semantics. Section 3 recalls Petri nets with boundaries and their tight correspondence with Petri calculi. Section 4 shows how to obtain a normal form for a P/T Petri calculus term P by computing the minimal marking graph for $\{[P]\}$ and from it a canonical P/T Petri net N. Finally, the canonical form of P is obtained by mapping N back into a term of the P/T calculus. A similar process is outlined in Sect. 5 for terms of the C/E Petri calculus. Section 6 concludes the paper.

2 Petri Calculi

As a matter of presentation, along the paper we find it convenient to present first the more general version (P/T case) of the definition and constructions, because it can be largely reused in the simpler case (C/E).

2.1 The P/T Petri Calculus

The P/T calculus is an algebra of connectors that mixes freely elementary synchronization constraints with mutual exclusion and (unbounded) memory. It is obtained by extending the algebra of stateless connectors with a denumerable set of constants $(\!|n|\!)$ (one for any $n \in \mathbb{N}$), each of them representing a buffer that currently contains n data items, aka tokens.

The syntax of terms of the P/T Calculus is below, where $n \in \mathbb{N}$.

$P \ ::= (\!|n|\!)$	buffer with n data items		
$\mid \mathsf{I}$	identity wire	$\mid \mathsf{X}$	twist
$\mid \nabla \mid \Delta$	duplicator and its dual	$\mid \perp \mid \top$	hiding and its dual
$\mid \wedge \mid \vee$	mutex and its dual	$\mid \downarrow \mid \uparrow$	inaction and its dual
$\mid P \otimes P$	parallel composition	$\mid P \,;\, P$	sequential composition

The diagrammatical representation of terms is shown in Fig. 1. Any term P has a unique associated *sort* (k, l) with $k, l \in \mathbb{N}$, that fixes the size k of the left interface and the size l of the right interface of P (see Fig. 2).

$$\llparenthesis n \rrparenthesis : (1,1) \qquad \mathsf{I} : (1,1) \qquad \mathsf{X} : (2,2) \qquad \nabla : (1,2) \qquad \Delta : (2,1)$$

$$\bot : (1,0) \qquad \top : (0,1) \qquad \wedge : (1,2) \qquad \vee : (2,1) \qquad \downarrow : (1,0) \qquad \uparrow : (0,1)$$

$$\frac{P_1 : (k,l) \quad P_2 : (m,n)}{P_1 \otimes P_2 : (k+m, l+n)} \qquad \frac{P_1 : (k,n) \quad P_2 : (n,l)}{P_1 \ ; P_2 : (k,l)}$$

Fig. 2. Sort inference rules

$$\frac{n,h,k \in \mathbb{N} \quad k \leq n}{\llparenthesis n \rrparenthesis \xrightarrow[k]{h} \llparenthesis n+h-k \rrparenthesis} \ (\textsc{TkIO}_{n,h,k}) \qquad \frac{k \in \mathbb{N}}{\mathsf{I} \xrightarrow[k]{k} \mathsf{I}} \ (\textsc{Id}_k) \qquad \frac{h,k \in \mathbb{N}}{\mathsf{X} \xrightarrow[kh]{hk} \mathsf{X}} \ (\textsc{Tw}_{h,k})$$

$$\frac{k \in \mathbb{N}}{\nabla \xrightarrow[kk]{k} \nabla} \ (\nabla_k) \qquad \frac{k \in \mathbb{N}}{\Delta \xrightarrow[k]{kk} \Delta} \ (\Delta_k) \qquad \frac{k \in \mathbb{N}}{\bot \xrightarrow{k} \bot} \ (\bot_k) \qquad \frac{k \in \mathbb{N}}{\top \xrightarrow[k]{} \top} \ (\top_k)$$

$$\frac{h,k \in \mathbb{N}}{\wedge \xrightarrow[hk]{h+k} \wedge} \ (\wedge_{h,k}) \qquad \frac{h,k \in \mathbb{N}}{\vee \xrightarrow[h+k]{hk} \vee} \ (\vee_{h,k}) \qquad \frac{}{\downarrow \xrightarrow{0} \downarrow} \ (\downarrow) \qquad \frac{}{\uparrow \xrightarrow[0]{} \uparrow} \ (\uparrow)$$

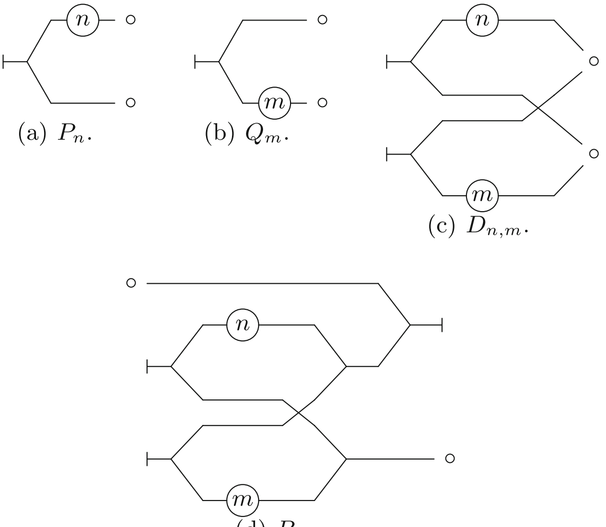
$$\frac{P \xrightarrow[b]{a} Q \quad R \xrightarrow[d]{c} S}{P \otimes R \xrightarrow[bd]{ac} Q \otimes S} \ (\textsc{Ten}) \qquad \frac{P \xrightarrow[c]{a} Q \quad R \xrightarrow[b]{c} S}{P \ ; R \xrightarrow[b]{a} Q \ ; S} \ (\textsc{Cut})$$

Fig. 3. Operational semantics of P/T calculus

The operational semantics is defined by means of the LTS in Fig. 3 whose states are terms P of the algebra and whose transitions are labelled by pairs $(a,b) \in \mathbb{N}^* \times \mathbb{N}^*$, written $P \xrightarrow[b]{a} P'$, where if $P : (k,l)$ then $|a| = k$, $|b| = l$ and $P' : (k,l)$. For each $i \in \{1 \dots k\}$, a_i is the number of actions executed on the i-th port of the left interface. Analogously, for each $j \in \{1 \dots l\}$, b_j is the number of actions executed on the j-th port of the right interface. Since data items can be created and deleted, but all connectors are maintained by the rules, the target P' preserves the overall structure of P (i.e., P and P' can differ only for sub-terms of the form $\llparenthesis n \rrparenthesis$).

We remark that some of the rules are more precisely schemes. For instance, there is one particular instance of rule $(\textsc{TkIO}_{n,h,k})$ for any possible choice of n, h and k. We think the rules are self-explanatory: Rule $(\textsc{TkIO}_{n,h,k})$ models the case where a buffer with n tokens releases $k \leq n$ tokens and receives h new tokens in the same step; at the end $n+h-k$ tokens are left in the buffer. Rule $(\textsc{Id}_k)$ and $(\textsc{Tw}_{h,k})$ just (re)wire the observation on the left interface to the one on the right. Rules (∇_k) and (Δ_k) enforce action synchronization on all ports. Rules $(\bot_k)$ and $(\top_k)$ hide any action on its interface. Rules $(\wedge_{h,k})$ and $(\vee_{h,k})$ mix the actions observed on the interface with two ports. Rules $(\downarrow)$ and $(\uparrow)$ enforce inaction on their (single) ports. Finally, rules $(\textsc{Ten})$ and $(\textsc{Cut})$ deal with parallel and sequential composition.

Notably, the induced bisimilarity is a congruence w.r.t. $\otimes$ and ; [12].



(a) P_n.

(b) Q_m.

(c) $D_{n,m}$.

(d) $B_{n,m}$.

Fig. 4. Petri calculus term for a buffer of capacity n

Example 1. As an example, we show one possible way to represent a buffer with capacity n. First, let $P_n = \top \,;\, \nabla \,;\, ((\!|n|\!) \otimes \mathsf{I}) : (0,2)$ and $Q_m = \top \,;\, \nabla \,;\, (\mathsf{I} \otimes (\!|m|\!)) : (0,2)$ shown in Fig. 4(a) and (b). It is immediate to check that, for any $h \leq n$ the only transitions for P_n are of the form $P_n \xrightarrow[hk]{} P_{n+k-h}$ and symmetrically, for Q_m and $k \leq m$, are of the form $Q_m \xrightarrow[hk]{} Q_{m+h-k}$. Let $C = (\mathsf{I} \otimes \mathsf{X} \otimes \mathsf{I}) \,;\, (\Delta \otimes \Delta) : (4,2)$. Again, it is immediate to check that the only transitions for C are of the form $C \xrightarrow[hk]{hkhk} C$. Then, let $D_{n,m} = (P_n \otimes Q_m) \,;\, C : (0,2)$ shown in Fig. 4(c). We have that $D_{n,m} \xrightarrow[hk]{} D_{n+k-h,m+h-k}$ with $h \leq n$ and $k \leq m$. Note that $(n-h+k) + (m-k+h) = n-m$, i.e., the numbers of tokens in the connector is invariant under transitions. Thus, the term $B_{n,m} = (\mathsf{I} \otimes D_{n,m}) \,;\, ((\Delta \,;\, \bot) \otimes \mathsf{I}) : (1,1)$ shown in Fig. 4(d) has transitions $B_{n,m} \xrightarrow[k]{h} B_{n+k-h,m+h-k}$ with $h \leq n$ and $k \leq m$ and $B_{n,0}$ is a buffer of capacity n (the sub-term P_n counts the free positions of the buffer, while Q_0 the busy ones).

2.2 The C/E Petri Calculus

It is quite common to impose some capacity over buffers. For example, we could think to consider only buffers of the form $(\!|c,n|\!)$ with $n \leq c$, where n is the number of tokens in the buffer and c is its maximal capacity. In this case, the transition $(\!|c,n|\!) \xrightarrow[k]{h} (\!|c,m|\!)$ would be possible only if $k \leq n$ and $h \leq c-n$ with $m = n+h-k$. ($(\!|c,n|\!)$ roughly corresponds to the process $B_{c-n,n}$ from Example 1).

In this section we focus on the simplest such case, where buffers have capacity one, also called one-place buffers. The corresponding calculus, originally introduced in [28], can be seen as the consequent restriction of the P/T Petri calculus

$$\frac{}{\bigcirc \xrightarrow[0]{1} \odot}\ (\textsc{TkI}) \qquad \frac{}{\odot \xrightarrow[1]{0} \bigcirc}\ (\textsc{TkO}) \qquad \frac{}{\bigcirc \xrightarrow[0]{0} \bigcirc}\ (\textsc{TkE}) \qquad \frac{}{\odot \xrightarrow[0]{0} \odot}\ (\textsc{TkF})$$

Fig. 5. Operational semantics for the one-place buffer (of the C/E Petri Calculus)

to operate over one-place buffers; in Petri net terminology, this restriction is called Condition/Event (C/E). Terms of the C/E Petri Calculus are defined by the grammar:

$$P ::= \bigcirc \mid \odot \mid \mathsf{I} \mid \mathsf{X} \mid \nabla \mid \Delta \mid \bot \mid \top \mid \wedge \mid \vee \mid \downarrow \mid \uparrow \mid P \otimes P \mid P \,;\, P$$

The constructors are the same as the ones of P/T calculus except for $\bigcirc$ and $\odot$ that respectively mimic the behaviour of $(\!|0,1|\!)$ and $(\!|1,1|\!)$. As before, any term P has a unique associated *sort*, with $\bigcirc : (1,1)$ and $\odot : (1,1)$ (remaining cases are defined as in Fig. 2).

The operational semantics is then defined by replacing Rule $(\textsc{TkIO}_{n,h,k})$ in Fig. 3 with the four rules in Fig. 5, representing respectively: the arrival of a token in the empty buffer (rule $(\textsc{TkI})$); the release of a token from the full buffer (rule $(\textsc{TkO})$); the inactivity of the empty/full buffer (rules $(\textsc{TkE})$, $(\textsc{TkF})$).

Remark 1. The semantics of the C/E Petri calculus presented here slightly differs from the original one in [28] and all its variants considered in [12]. If we restrict to consider stateless connectors, i.e., terms not involving $\bigcirc$ and $\odot$, then their semantics is the one called 'weak' in [12], whereas the 'strong' semantics would allow only one action at a time to take place in a port, e.g., only transitions $\wedge \xrightarrow[00]{0} \wedge$, $\wedge \xrightarrow[10]{1} \wedge$ and $\wedge \xrightarrow[01]{1} \wedge$ would be considered for the connector $\wedge$. Differently from the weak case, here we forbid tokens to traverse buffers during a step, in agreement with the classical C/E semantics where a loop cannot fire. However, other variants can be nicely accounted for by changing the rules for $\bigcirc$ and $\odot$. For example, consume/produce loops can be dealt with by adding the transition $\odot \xrightarrow[1]{1} \odot$. On the one hand, we think the semantics proposed here improves the correspondence between C/E Petri calculus and C/E Petri nets with boundaries (avoiding the use of the 'contention' relation from [12]) and, on the other hand, it yields a more uniform definition with the P/T case, preserving all good properties, like bisimilarity being a congruence w.r.t. $\otimes$ and ;.

Example 2. A buffer with capacity n can be represented by combining n buffers of capacity 1: we just let $B_1 = \bigcirc : (1,1)$ and $B_{n+1} = \wedge \,;\, (B_n \otimes \bigcirc) \,;\, \vee : (1,1)$.

3 Nets with Boundaries

Nets with boundaries extends ordinary Petri nets by equipping them with left and right interfaces made of ports. Ports are different from places in that places in the pre-set of a transition α impose a bound on the number of instances of α that can be fired concurrently, while ports do not. In fact ports can account for an unbounded number of instances of transitions attached to them to fire concurrently. This is desirable, not an anomaly, because we can account for any execution context in which the nets with boundaries are plugged in.

3.1 P/T Petri Nets with Boundaries

Petri nets [26] consist of *places*, which are repositories of *tokens*, and *transitions* that remove and produce tokens. Places of a *Place/Transition net* (P/T net) can hold zero, one or more tokens and arcs are weighted. The state of a P/T net is described in terms of *(P/T) markings*, i.e., (finite) multisets of tokens.

A multiset on a set X is a function $X \to \mathbb{N}$. The set of multisets on X is denoted $\mathcal{M}_X$. We let $\mathcal{U}, \mathcal{V}$ range over $\mathcal{M}_X$. For $\mathcal{U}, \mathcal{V} \in \mathcal{M}_X$, we write $\mathcal{U} \subseteq \mathcal{V}$ iff $\forall x \in X : \mathcal{U}(x) \leq \mathcal{V}(x)$ and we use the usual multiset operations for union ($\cup$), difference ($-$) and scalar multiplication ($\cdot$). We use $\varnothing \in \mathcal{M}_X$ for the empty multiset s.t. $\varnothing(x) = 0$ for all $x \in X$ and we write x for the singleton multiset $\mathcal{U}$ such that $\mathcal{U}(x) = 1$ and $\mathcal{U}(y) = 0$ for all $y \neq x$. Given a finite X, if $f : X \to \mathcal{M}_Y$ and $\mathcal{U} \in \mathcal{M}_X$ then we shall abuse notation and write $f(\mathcal{U}) = \bigcup_{x \in X} \mathcal{U}(x) \cdot f(x)$.

Definition 1 (P/T net). *A* P/T net *is a 4-tuple* $(P, T, {}^\circ-, -^\circ)$ *where:* P *is a set of* places*;* T *is a set of* transitions*; and* ${}^\circ-, -^\circ : T \to \mathcal{M}_P$ *are functions assigning* pre- *and* post-sets *to transitions.*

Let $\mathcal{X} \in \mathcal{M}_P$, we write $N_\mathcal{X}$ for the marked P/T net N with marking $\mathcal{X}$.

Definition 2 (P/T step semantics). *Let* $N = (P, T, {}^\circ-, -^\circ)$ *be a P/T net,* $\mathcal{X}, \mathcal{Y} \in \mathcal{M}_P$*. For* $\mathcal{U} \in \mathcal{M}_T$ *a multiset of transitions, we write:*

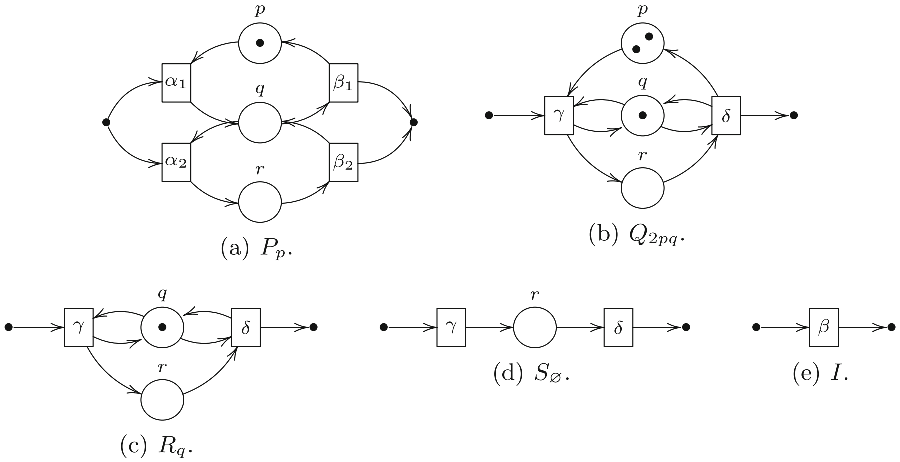
$$N_\mathcal{X} \to_\mathcal{U} N_\mathcal{Y} \quad \stackrel{\text{def}}{=} \quad {}^\circ\mathcal{U} \subseteq \mathcal{X},\ \mathcal{U}^\circ \subseteq \mathcal{Y}\ \&\ \mathcal{X} - {}^\circ\mathcal{U} = \mathcal{Y} - \mathcal{U}^\circ.$$

The remaining of this section recalls the composable nets proposed in [28]. Due to space limitation, we refer to [12] for a detailed presentation. In the following we let $\underline{n}$ range over finite ordinals, i.e., $\underline{n} \stackrel{\text{def}}{=} \{0, 1, \ldots, n-1\}$.

Definition 3 (P/T net with boundaries). *Let* $m, n \in \mathbb{N}$*. A (finite)* P/T net with boundaries $N : m \to n$ *is a tuple* $N = (P, T, {}^\circ-, -^\circ, {}^\bullet-, -^\bullet)$*, where:*

- $(P, T, {}^\circ-, -^\circ)$ *is a finite P/T net;*
- ${}^\bullet- : T \to \mathcal{M}_{\underline{m}}$ *and* $-^\bullet : T \to \mathcal{M}_{\underline{n}}$ *are functions that bind transitions to the left and right boundaries of* N*;*

Let $\mathcal{X} \in \mathcal{M}_P$, we write $N_\mathcal{X}$ for the P/T net N with boundaries whose current marking is $\mathcal{X}$. Note that, for any $k \in \mathbb{N}$, there is a bijection $\ulcorner - \urcorner : \mathcal{M}_{\underline{k}} \to \mathbb{N}^k$ between multisets on $\underline{k}$ and strings of natural numbers of length k, defined by $\ulcorner \mathcal{U} \urcorner_i \stackrel{\text{def}}{=} \mathcal{U}(i)$, namely, the i-th natural number in the string $\ulcorner \mathcal{U} \urcorner$ assigned to the multiset $\mathcal{U}$ is the multiplicity of the i-th port in $\mathcal{U}$. For example, given the multiset $\mathcal{U} = \{0, 0, 2\} \in \mathcal{M}_{\underline{4}}$ we have $\ulcorner \mathcal{U} \urcorner = 2\ 0\ 1\ 0$.



(a) P_p.

(b) Q_{2pq}.

(c) R_q.

(d) $S_\varnothing$.

(e) I.

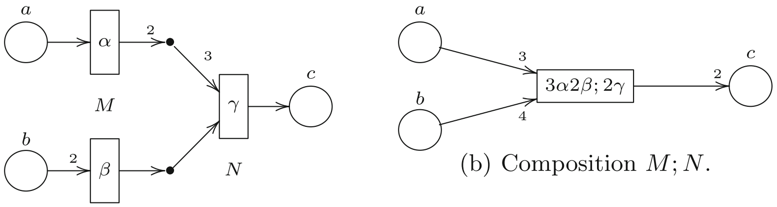
Fig. 6. Five marked P/T nets with boundaries

Definition 4 (P/T Labelled Semantics). *Let* $N = (P, T, {}^\circ-, -^\circ, {}^\bullet-, -^\bullet)$ *be a P/T net with boundaries and* $\mathcal{X}, \mathcal{Y} \in \mathcal{M}_P$*. We write*

$$N_{\mathcal{X}} \xrightarrow[b]{a} N_{\mathcal{Y}} \quad \overset{\text{def}}{=} \quad \exists \mathcal{U} \in \mathcal{M}_T \textit{ s.t. } N_{\mathcal{X}} \to_{\mathcal{U}} N_{\mathcal{Y}},\ a = \ulcorner {}^\bullet\mathcal{U} \urcorner \ \&\ b = \ulcorner \mathcal{U}^\bullet \urcorner. \tag{1}$$

Example 3. Figure 6 shows five different marked P/T nets with boundaries. Places are circles and a marking is represented by the presence or absence of tokens; rectangles are transitions and arcs stand for pre- and post-set relations. The left (respectively, right) interface is depicted by points situated on the left (respectively, on the right). Figure 6(a) shows the marked net $P_p : 1 \to 1$ containing three places, four transitions and initially marked with one token in place p. Figure 6(b) shows the marked net $Q_{2pq} : 1 \to 1$ containing three places, two transitions and initially marked with two tokens in p and one in q. These two nets are bisimilar: they both model a buffer with capacity two, in which messages are produced over the left interface and consumed over the right interface. Figure 6(c) and (d) show two different models for unbounded buffers. They are not bisimilar: while R_q serialises all operations on the buffer, $S_\varnothing$ allows for the concurrent production/consumption of messages. Note that transition γ in Fig. 6(d) has an empty pre-set and δ has an empty post-set. Figure 6(e) shows the net $I : 1 \to 1$ that contains no places. The sole transition β has empty pre and post-sets. This net can forward any quantity of tokens received on its left port to the right port and, hence, it is neither bisimilar to R_q nor to $S_\varnothing$.

While from the point of view of ordinary Petri nets having empty pre-/post-sets is quite a peculiar feature, which makes life harder when defining the operational semantics, we emphasize that in our context of decomposing nets into their minimal components this is a highly valuable property. In fact, the interfaces of



(a) Two P/T with boundaries M and N.

(b) Composition $M; N$.

Fig. 7. Composition of P/T with boundaries

nets with boundaries have the role of synchronizing the transitions of different components. In this perspective, it is natural to have nets without places as basic components.

Nets with boundaries can be composed in parallel and in series.

Given $N_{\mathcal{X}} : m \to n$ and $M_{\mathcal{Y}} : k \to l$, their tensor product is the net $N_{\mathcal{X}} \otimes M_{\mathcal{Y}} : m + k \to n + l$ whose sets of places and transitions are the disjoint union of the corresponding sets in N and M, whose maps ${}^{\circ}-$, $-^{\circ}$, ${}^{\bullet}-$, $-^{\bullet}$ are defined according to the maps in N and M and whose initial marking is $\mathcal{X} \cup \mathcal{Y}$. Intuitively, the tensor product corresponds to put the nets N and M side-by-side.

The sequential composition $N_{\mathcal{X}}; M_{\mathcal{Y}} : m \to n$ of $N_{\mathcal{X}} : m \to k$ and $M_{\mathcal{Y}} : k \to n$ is slightly more involved. Intuitively, transitions attached to the left or right boundaries can be seen as transition fragments, that can be completed by attaching other complementary fragments to that boundary. When two transition fragments in N share a boundary node, then they are two mutually exclusive options for completing a fragment of M attached to the same boundary node. Thus, the idea is to combine the transitions of N with those of M when they share a common boundary, as if their firings were synchronised. As in general (infinitely) many combinations are possible, the composed nets is defined by selecting a minimal (multi-)set of synchronisations that suffices to represent any other possible synchronisation as a linear combinations of the chosen ones (i.e., as the concurrent firing of several transitions). The initial marking is $\mathcal{X} \cup \mathcal{Y}$ (formal definition can be found at [12]). As an example, Fig. 7(b) shows the sequential composition of the nets $M : 0 \to 2$ and $N : 2 \to 0$ from Fig. 7(a). A firing of α produces two tokens on the port to which γ is also attached, while a firing of γ requires three tokens from the same port and one from the other port, to which β is attached to. Therefore the minimal multi-set of transitions that allows the synchronization between α, β and γ contains three instances of α and two instances of β and γ.

3.2 From P/T Nets with Boundaries to P/T Calculus and Back

The contribution in [12] enlightens a tight semantics correspondence between P/T calculus and P/T nets with boundaries. Concretely, two translations are defined. The first encoding $\boldsymbol{T}_{\_}$ shows that each net $N_{\mathcal{X}}$ can be mapped into a P/T calculus process $\boldsymbol{T}_{N_{\mathcal{X}}}$ that preserves and reflects operational semantics (and thus also bisimilarity). The second encoding $\{\![-]\!\}$ provides the converse translation, from a P/T Petri calculus process P to a P/T net with boundaries $\{\![P]\!\}$, defined by structural induction. We recall here the two main correspondence results and omit the details due to space constraints.

Theorem 1. *Let P be a term of P/T calculus.*

(i) if $P \xrightarrow[b]{a} P'$ then $\{\![P]\!\} \xrightarrow[b]{a} \{\![P']\!\}$.
(ii) if $\{\![P]\!\} \xrightarrow[b]{a} N_{\mathcal{X}}$ then $\exists P'$ such that $P \xrightarrow[b]{a} P'$ and $\{\![P']\!\} = N_{\mathcal{X}}$.

Theorem 2. *Let N be a finite P/T net with boundaries, then*

(i) if $N_{\mathcal{X}} \xrightarrow[b]{a} N_{\mathcal{Y}}$ then $\boldsymbol{T}_{N_{\mathcal{X}}} \xrightarrow[b]{a} \boldsymbol{T}_{N_{\mathcal{Y}}}$.
(ii) if $\boldsymbol{T}_{N_{\mathcal{X}}} \xrightarrow[b]{a} Q$ then $\exists \mathcal{Y}$ such that $N_{\mathcal{X}} \xrightarrow[b]{a} N_{\mathcal{Y}}$ and $Q = \boldsymbol{T}_{N_{\mathcal{Y}}}$.

3.3 C/E Nets with Boundaries

A well-known subclass of bounded P/T nets are C/E nets. In C/E nets, places have maximum capacity 1 and pre- and post-set of transitions are restricted to sets (instead of multisets). Formally,

Definition 5 (C/E net). *A* C/E net *is a P/T net $N = (P, T, {}^\circ-, -^\circ)$ where:*[3] *P is a set of* places*; T is a set of* transitions*; and ${}^\circ-, -^\circ : T \to 2^P$ are functions.*

In addition, a *C/E marking* is just a subset of places $X \subseteq P$ (not a multiset). We let N_X denote the net N with marking X.

Definition 6 (C/E step semantics). *Let $N = (P, T, {}^\circ-, -^\circ)$ be a C/E net, $X, Y \subseteq P$ and $\mathcal{U} \subseteq \mathcal{M}_T$ a multiset of transitions s.t. ${}^\circ\mathcal{U}$ and $\mathcal{U}^\circ$ are sets, write:*

$$N_X \to_{\mathcal{U}} N_Y \quad \stackrel{\text{def}}{=} \quad {}^\circ\mathcal{U} \subseteq X,\ \mathcal{U}^\circ \cap X = \varnothing \ \&\ Y = (X \setminus {}^\circ\mathcal{U}) \cup \mathcal{U}^\circ.$$

We remark that the constraint on ${}^\circ\mathcal{U}$ and $\mathcal{U}^\circ$ to be sets ensures that every pair of transitions in $\mathcal{U}$ has disjoint pre- and post-sets. This definition allows the concurrent firing of several instances of the same transition when its pre- and post-sets are both empty: As explained before, even if places are bounded this will allow for ports of unbounded capacity (w.r.t. the number of actions that can take place concurrently) in C/E nets with boundaries.

Definition 7 (C/E nets with boundaries). *A P/T net with boundaries $N = (P, T, {}^\circ-, -^\circ, {}^\bullet-, -^\bullet)$ is a C/E net with boundaries if $(P, T, {}^\circ-, -^\circ)$ is a C/E net.*

A marking of a C/E net with boundaries is just a set of places of the net, i.e., $X \subseteq P$. Note that while pre- and post-set of transitions are sets and not multisets, multiplicity are maintained by ${}^\bullet-$ and $-^\bullet$ w.r.t. left and right ports: many tokens can be exchanged concurrently over a single port in one step.

[3] In the context of C/E nets some authors call places *conditions* and transitions *events*.

Definition 8 (C/E Labelled Semantics). *Let* $N = (P, T, {}^\circ{-}, {-}^\circ, {}^\bullet{-}, {-}^\bullet)$ *be a C/E net with boundaries and* $X, Y \subseteq P$. *Write:*

$$N_X \xrightarrow[b]{a} N_Y \quad \overset{\text{def}}{=} \quad \exists \mathcal{U} \subseteq \mathcal{M}_T \ \textit{s.t.}\ N_X \to_{\mathcal{U}} N_Y,\ a = \ulcorner {}^\bullet\mathcal{U} \urcorner \ \&\ b = \ulcorner \mathcal{U}^\bullet \urcorner \qquad (2)$$

Remark 2. Following the presentation of the C/E Petri Calculus in Sect. 2.2 (see Remark 1), we have presented here a slightly different definition for C/E Petri nets with boundaries w.r.t. [12] by allowing richer observations over interfaces (strings of natural numbers instead of just 0/1).

Example 4. All nets in Fig. 6 except from Q_{2pq} (Fig. 6(b)) can be interpreted as C/E nets with boundaries. We remark that P_p has the same behaviour when considering both the P/T net and the C/E labelled semantics (because P_p is a 1-bounded P/T net). Similarly, the semantics of $I_\varnothing$ is also invariant under both views. Differently, the behaviour of R_q and $S_\varnothing$ changes when considering the C/E semantics. The former is deadlocked, because of the self-looping transitions, while the latter models a buffers of capacity one that alternates the production and consumption of tokens.

The correspondence results in Sect. 3.2 can be restated also for the case of the Petri Calculus and C/E nets with boundaries along the lines shown in [12].

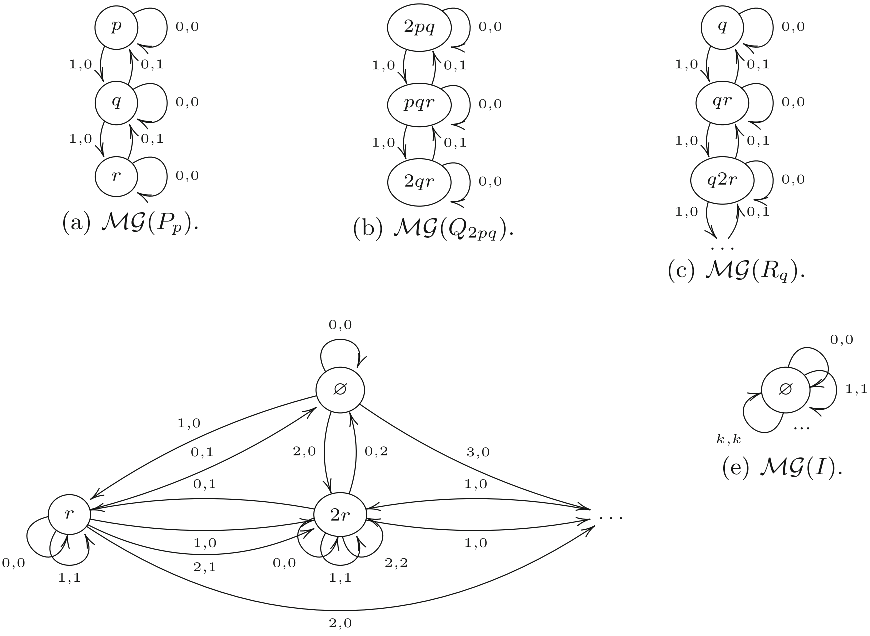
4 Normal Forms for Finite State P/T terms

This section shows how to obtain normal forms for finite state connectors. We will take advantage of the mutual encodings between P/T calculus terms and P/T nets with boundaries summarised in Sect. 3.2. In order to obtain the normal for a connector, we will proceed as follows: (i) we translate a P/T calculus term into an equivalent P/T net with boundaries by using the encoding $\{[\_]\}$, (ii) we compute a canonical representation (up to isomorphism) for the corresponding net with boundaries, (iii) we map back the canonical representation of the net into a term of the P/T calculus by using the encoding $\boldsymbol{T}_{\_}$. The canonical representation of the net is obtained by analysing its associated marking graph.

Definition 9 (Reachable marking). *Let* $N_{\mathcal{X}} : n \to m$ *be a* P/T net *with boundaries. Then,* $\mathcal{Y}$ *is a reachable marking of* $N_{\mathcal{X}}$ *if there exists a (possibly empty) finite sequence of transitions* $N_{\mathcal{X}} \xrightarrow[b_1]{a_1} N_{\mathcal{X}_1} \xrightarrow[b_2]{a_2} \dots \xrightarrow[b_k]{a_k} N_{\mathcal{Y}}$ *with* $a_i \in \mathbb{N}^n$ *and* $b_i \in \mathbb{N}^m$. *We write* $\mathcal{RM}(N_{\mathcal{X}})$ *for the set of all reachable markings of* $N_{\mathcal{X}}$.

Definition 10 (Marking graph of a net with boundaries). *Let* $N : n \to m$ *be a* P/T net *with boundaries with initial marking* $\mathcal{X}$. *The marking graph of* $N_{\mathcal{X}}$ *is the state transition graph* $\mathcal{MG}(N_{\mathcal{X}}) = (\mathcal{RM}(N_{\mathcal{X}}), \mathbb{T})$ *where* $\mathbb{T} \subseteq \mathcal{M}_P \times \mathbb{N}^n \times \mathbb{N}^m \times \mathcal{M}_P$ *is as follows:* $\mathbb{T} = \{(\mathcal{Y}, a, b, \mathcal{Z}) \mid \mathcal{Y}, \mathcal{Z} \in \mathcal{RM}(N_{\mathcal{X}}) \wedge N_{\mathcal{Y}} \xrightarrow[b]{a} N_{\mathcal{Z}}\}$.

We say $\mathcal{MG}(N_{\mathcal{X}})$ is *(in)finite state* when $\mathcal{RM}(N_{\mathcal{X}})$ is (in)finite. We say $\mathcal{MG}(N_{\mathcal{X}})$ is *finite* when it is finite state and $\mathbb{T}$ is also finite, we say it is *infinite* otherwise.



(a) $\mathcal{MG}(P_p)$.

(b) $\mathcal{MG}(Q_{2pq})$.

(c) $\mathcal{MG}(R_q)$.

(d) $\mathcal{MG}(S_\varnothing)$.

(e) $\mathcal{MG}(I)$.

Fig. 8. Marking graphs

$\mathcal{MG}(N_\mathcal{X})$ is finitely branching if for any $\mathcal{Y} \in \mathcal{RM}(N_\mathcal{X})$ it holds that $\mathbb{T}_\mathcal{Y} = \{(\mathcal{V}, a, b, \mathcal{Z}) \mid (\mathcal{V}, a, b, \mathcal{Z}) \in \mathbb{T} \wedge \mathcal{V} = \mathcal{Y}\}$ is finite.

Note that for any $N_\mathcal{X} : n \to m$, it holds that $N_\mathcal{X} \xrightarrow[0^m]{0^n} N_\mathcal{X}$. Therefore, every node in a marking graph of the net has a self-loop with label $(0^n, 0^m)$.

Example 5. Figure 8 shows the marking graphs for the nets in Fig. 6. We remark that the marking graphs for P_p and Q_{2pq} are finite and isomorphic. On the contrary, the remaining three are infinite. The marking graph for R_q and $S_\varnothing$ are infinite state (because the corresponding nets are unbounded). Nevertheless, while $\mathcal{MG}(R_q)$ is finitely branching, $\mathcal{MG}(S_\varnothing)$ is not (e.g., any state in $\mathcal{MG}(S_\varnothing)$ has a transition labelled $(k, 0)$ for any $k \in \mathbb{N}$). Although $\mathcal{MG}(I)$ is finite state, it is infinitely branching.

Remark 3. The marking graph of a net with boundaries is finite state only if the underlying net is bounded. Note that the marking graph of a net containing a transition with empty pre-set and non-empty post-set is unbounded (for instance, the net $S_\varnothing$ in Fig. 8(d)).

Remark 4. The marking graph of a P/T net with boundaries containing a transition with an empty preset is infinitely branching (e.g., the nets $I_\varnothing$ in Fig. 6(e) and $S_\varnothing$ in Fig. 8(d)). On the contrary, when every transitions in the net has a non-empty preset, the marking graph is finitely branching because each marking constraints the number of concurrently fireable instances of each transition.

The remaining of the section is devoted to the definition of the normal form of (finite state) connectors. We deal with the general case by using a *divide et impera* approach. We solve two sub-problems first: (i) the encoding of nets with finite marking graphs (Sect. 4.1) and (ii) the encoding of infinitely branching stateless nets (Sect. 4.2).

4.1 Finite (State and Transition) Marking Graphs

In this section we show how to obtain the normal form for P/T nets with boundaries whose marking graph is finite, i.e., when it is bounded and every transition has a non-empty preset. We leave this as an implicit assumption for all the nets considered in this section.

We first note that for a finite graph we can apply, e.g., a partition refinement algorithm [18,24] to obtain the smallest (up-to iso) (in terms of states and transitions) automaton amongst all those bisimilar to the given graph. We write $\mathit{min}(MG)$ for the minimal graph in the equivalence class of MG.

We note that any finite marking graph can be represented by a P/T net with boundaries as follows:

Definition 11 (Marking graph as a net with boundaries). *Let $MG = (S, T)$ with $T \in S \times \mathbb{N}^n \times \mathbb{N}^m \times S$ be a marking graph. The corresponding P/T net with boundaries is $\mathcal{NB}(MG) = (S, T', {}^\circ-, -^\circ, {}^\bullet-, -^\bullet) : n \to m$ s.t.*

- *$T' = T \setminus \{(s, 0^n, 0^m, s) \mid s \in S\}$ (we can safely omit self-looping transitions that are not attached to ports);*
- *${}^\circ(s, a, b, t) = s$ and $(s, a, b, t)^\circ = t$;*
- *${}^\bullet(s, a, b, t) = \mathcal{U}$ where $\mathcal{U} \in \mathcal{M}_{\underline{n}}$ and $\ulcorner \mathcal{U} \urcorner = a$;*
- *$(s, a, b, t)^\bullet = \mathcal{V}$ where $\mathcal{V} \in \mathcal{M}_{\underline{m}}$ and $\ulcorner \mathcal{V} \urcorner = b$.*

We let $\mathit{can}(N_{\mathcal{X}}) \overset{\text{def}}{=} \mathcal{NB}(\mathit{min}(\mathcal{MG}(N_{\mathcal{X}})))_{\{\mathcal{X}\}}$.

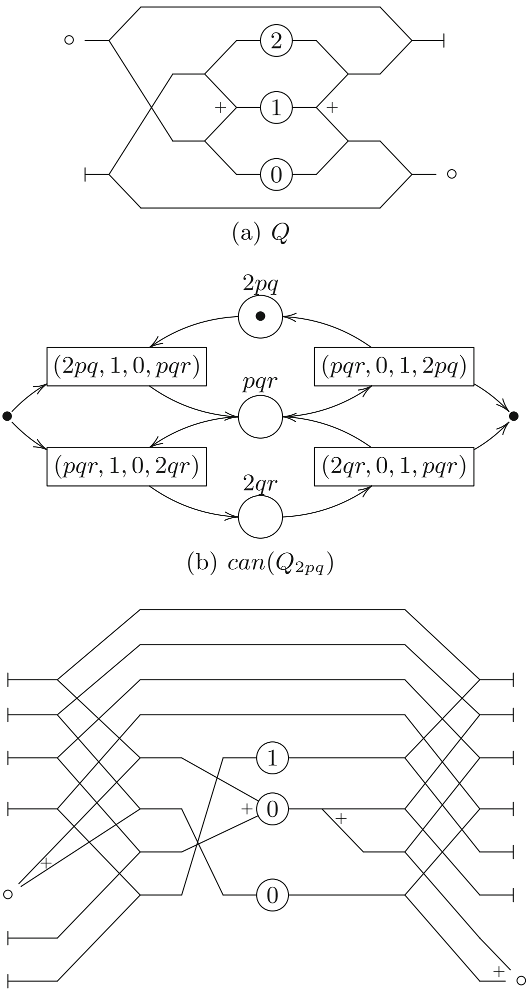
Lemma 1 (Minimal net with boundaries). *Let $N : n \to m$ be a net with boundaries, then we have that $N_{\mathcal{X}}$ and $\mathit{can}(N_{\mathcal{X}})$ are bisimilar.*

Proof. It follows by noting that $N_{\mathcal{X}}$ and $\mathcal{MG}(N_{\mathcal{X}})$ are bisimilar by construction; $\mathcal{MG}(N_{\mathcal{X}})$ and $\mathit{min}(\mathcal{MG}(N_{\mathcal{X}}))$ are bisimilar by definition; and $\mathit{min}(\mathcal{MG}(N_{\mathcal{X}}))$ and $\mathcal{NB}(\mathit{min}(\mathcal{MG}(N_{\mathcal{X}})))_{\{\mathcal{X}\}}$ are bisimilar by construction.

Corollary 1. *$\mathit{can}(N)$ is unique (up-to iso) because $\mathcal{NB}(-)$ and $\mathcal{MG}(-)$ are functions and the minimal automaton is also unique (up-to iso).*

Corollary 2. *Given two bisimilar nets with boundaries $N_{\mathcal{X}}$ and $M_{\mathcal{X}}$, the nets $\mathit{can}(N_{\mathcal{X}})$ and $\mathit{can}(M_{\mathcal{X}})$ are isomorphic.*

Example 6. Consider the P/T term $Q = (\nabla \otimes (\top ; \nabla)) ; (\mathsf{I} \otimes T \otimes \mathsf{I}) ; ((\Delta ; \bot) \otimes \Delta)$ with $T = \mathsf{X} ; (\nabla \otimes \nabla) ; ((\!|2|\!) \otimes (\vee ; (\!|1|\!) ; \wedge) \otimes (\!|0|\!)) ; (\Delta \otimes \Delta)$ depicted in Fig. 9(a). The equivalent P/T net with boundaries $\{\![Q]\!\}$ is the net Q_{2pq} shown in Fig. 6(b). The corresponding marking graphs is in Fig. 8(b). This graph is minimal, i.e., there does not exist a bisimilar graph with a smaller number of states and/or transitions. Therefore, $min(\mathcal{MG}(Q_{2pq})) = \mathcal{MG}(Q_{2pq})$ and $can(Q_{2pr}) = \mathcal{NB}(\mathcal{MG}(Q_{2pr}))_{2pr}$, which is shown in Fig. 9(b). Then, the normal form $nf(Q)$ is obtained by encoding back $can(Q_{2pr})$ as a P/T term (shown in Fig. 9(c)).



(a) Q

(b) $can(Q_{2pq})$

(c) $nf(Q)$

Fig. 9. Normal form of a term with finite marking graph

We remark that the marking graph $\mathcal{MG}(P_p))$ (Fig. 8(a)), corresponding to P_p in Fig. 6(a), is isomorphic to the marking graph of Q_{2pq}. This implies that both P_p and Q_{2pq} have the same normal form.

4.2 Stateless, Infinitely Branching Marking Graphs

The simplest case of finite state, but infinite branching marking graph, is a net without places, like the net I in Fig. 6(e), whose marking graph is (partially) depicted in Fig. 8(e).

We introduce a minimization procedure for stateless nets that removes redundant transitions, i.e., transitions that can be mimicked by a combination of other transitions in the net.

Definition 12 (Redundant transition and minimal net). *Let $N : m \to n$ be the stateless P/T net with boundaries $N = (\varnothing, T, {}^{\circ}-, -^{\circ}, {}^{\bullet}-, -^{\bullet})$. A transition $t \in T$ is redundant if there exists $\mathcal{U} \in \mathcal{M}_{T-\{t\}}$ s.t. ${}^{\bullet}t = {}^{\bullet}\mathcal{U}$ and $t^{\bullet} = \mathcal{U}^{\bullet}$. We say that a stateless net is minimal if every transition is not redundant.*

Lemma 2. *Let $N : m \to n$ be the stateless P/T net with boundaries $N = (\varnothing, T, {}^{\circ}-, -^{\circ}, {}^{\bullet}-, -^{\bullet})$ with $t \in T$ redundant. Define $T' = T - \{t\}$ and*

$$N' = (\varnothing, T', {}^{\circ}-|_{T'}, -^{\circ}|_{T'}, {}^{\bullet}-|_{T'}, -^{\bullet}|_{T'}).$$

Then, $N_{\varnothing}$ and $N'_{\varnothing}$ are bisimilar.

The above result provides a minimization procedure by iteratively removing redundant transitions. The procedure is effective: it takes each transition $t \in T$ and compares pre- and post-sets with each possible multisets $\mathcal{U}$ of T. Since ${}^{\bullet}t$ and $t^{\bullet}$ are finite, there is just a finite number of multisets $\mathcal{U}$ of T to consider. We note $\tilde{N}$ the result of the minimization algorithm over N.

The above procedure converges in a finite number of steps, because T is finite. The procedure is non-deterministic (w.r.t. the choice of the redundant transition t to eliminate) but it always converges to the same result.

Lemma 3. *Let N be a stateless net with boundaries, then $\tilde{N}$ is uniquely defined (up-to iso).*

Proof. We proceed by contradiction. Suppose that different orders in which redundant transitions are eliminated can lead to two different outcomes

$$N' = (\varnothing, T', {}^{\circ}-, -^{\circ}, {}^{\bullet}-, -^{\bullet}) \text{ and } N'' = (\varnothing, T'', {}^{\circ}-, -^{\circ}, {}^{\bullet}-, -^{\bullet}).$$

Clearly it cannot be the case that $T' \subset T''$ or $T'' \subset T'$ (otherwise T' or T'' would contain redundant transitions). Hence $T'' \setminus T' \neq \varnothing$ and $T' \setminus T'' \neq \varnothing$.

Let $t' \in T' \setminus T''$. Since $t' \in T' \subseteq T$, it must be redundant w.r.t. the transitions in T'', i.e., there must exist $\mathcal{U}' \in \mathcal{M}_{T''}$ s.t. ${}^{\bullet}t' = {}^{\bullet}\mathcal{U}'$ and $t'^{\bullet} = \mathcal{U}'^{\bullet}$. Following a similar reasoning, any transition t'' in $T'' \setminus T'$ must be redundant w.r.t. the transitions in T' and expressible as a suitable $\mathcal{U}'' \in \mathcal{M}_{T'}$.

Moreover, there must be at least one transition $t'' \in \mathcal{U}'$, non isomorphic to t', such that $t'' \in T'' \setminus T'$ (otherwise t' would be redundant w.r.t. transitions in T'). Then, since any such t'' can be expressed in terms of $\mathcal{U}'' \in \mathcal{M}_{T'}$, it follows that t' can be expressed as a multiset $\mathcal{U} \in \mathcal{M}_{T'}$. Now there are two cases:

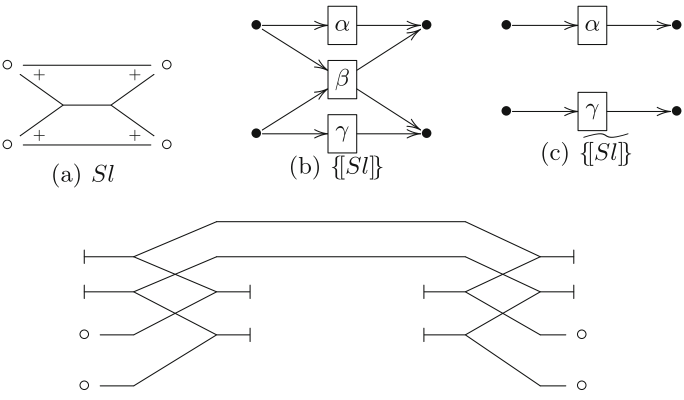
- $t' \notin \mathcal{U}$, but this is absurd, because t' would be redundant;
- $t' \in \mathcal{U}$, but this is absurd, because we would have $\mathcal{U} = t'$ isomorphic to t''.

Lemma 4. *Let N be a stateless net, then $\tilde{N}_\varnothing$ and $N_\varnothing$ are bisimilar.*

Lemma 5. *Let N and M be two stateless bisimilar nets, then $\tilde{N}_\varnothing = \tilde{M}_\varnothing$ (up-to iso).*

Proof. The proof follows by contradiction. Assume that there is a transition t in $\tilde{N}$ that is not matched by a transition in $\tilde{M}$. Let $\tilde{N}_\varnothing \to_t \tilde{N}_\varnothing$. Then, $N_\varnothing \to_t N_\varnothing$. Since $N_\varnothing$ and $M_\varnothing$ are bisimilar, $M_\varnothing \to_\mathcal{U} M_\varnothing$ with ${}^\bullet t = {}^\bullet\mathcal{U}$ and $t^\bullet = \mathcal{U}^\bullet$. Consequently, $M_\varnothing \to_\mathcal{U} M_\varnothing$. If $|\mathcal{U}| = 1$, we are done. Otherwise, $\mathcal{U} = k_1 \cdot t_1 \cup \ldots \cup k_n \cdot t_n$ with $n > 1$. Then, for any transition t_i we conclude that $\tilde{N}_\varnothing \to_{t_i} \tilde{N}_\varnothing$. Hence, t is redundant in $\tilde{N}$, which contradicts the assumption that $\tilde{N}$ is minimal.

Example 7. Consider the stateless term $Sl = (\wedge \otimes \wedge);(\mathsf{I} \otimes (\Delta;\nabla) \otimes \mathsf{I});(\vee \otimes \vee)$ depicted in Fig. 10(a). The corresponding net with boundaries $\{[Sl]\}$ is shown in Fig. 10(b). Note that the transition β is redundant because it can be expressed as the concurrent firing of α and β; consequently, it is removed by the minimization algorithm, which produces the minimal net $\widetilde{\{[Sl]\}}$ shown in Fig. 10(c). Finally, the normal form $nf(Sl)$ for the net Sl is obtained by encoding back $\widetilde{\{[Sl]\}}$ as the Petri calculus term shown in Fig. 10(d).



(a) Sl (b) $\{[Sl]\}$ (c) $\widetilde{\{[Sl]\}}$

(d) $nf(Sl)$

Fig. 10. A stateless term of the P/T calculus Sl

4.3 Finite State and Infinitely Branching Marking Graph

When the marking graph is finite state but infinitely branching, the associated net has both transitions with empty pre- and post-set and transitions with non-empty post-set (by Remark 3, the net cannot contain transitions with empty pre-set and non-empty post-set). We show first that the behaviour of a net can be described by combining the behaviour of two subnets containing respectively the stateless and stateful behaviours.

Definition 13 (Stateless and Stateful subnets). *Let $N : m \to n$ be the P/T net with boundaries $N = (P, T, {}^\circ-, -^\circ, {}^\bullet-, -^\bullet)$. A transition $t \in T$ is stateless if ${}^\circ t = t^\circ = \varnothing$. We write T^{sl} for the set of all stateless transitions and $T^{\mathsf{sf}} = T \backslash T^{\mathsf{sl}}$ denote the set of stateful transitions. The stateless subnet of N is*

$$N^{\mathsf{sl}} = (\varnothing, T^{\mathsf{sl}}, {}^\circ - |_{T^{\mathsf{sl}}}, -^\circ|_{T^{\mathsf{sl}}}, {}^\bullet - |_{T^{\mathsf{sl}}}, -^\bullet|_{T^{\mathsf{sl}}})$$

Similarly, the stateful subnet is

$$N^{\mathsf{sf}} = (P, T^{\mathsf{sf}}, {}^\circ - |_{T^{\mathsf{sf}}}, -^\circ|_{T^{\mathsf{sf}}}, {}^\bullet - |_{T^{\mathsf{sf}}}, -^\bullet|_{T^{\mathsf{sf}}})$$

We can now tightly relate the behaviour of N with those of N^{sl} and N^{sf}.

Lemma 6. *Let $N_{\mathcal{X}}$ be a marked P/T net with boundaries. Then,*

- *If $N_{\mathcal{X}} \xrightarrow[b]{a} N_{\mathcal{Y}}$, then there exist a_1, a_2, b_1 and b_2 such that $a = a_1 + a_2$, $b = b_1 + b_2$, ${N^{\mathsf{sf}}}_{\mathcal{X}} \xrightarrow[b_1]{a_1} {N^{\mathsf{sf}}}_{\mathcal{Y}}$ and ${N^{\mathsf{sl}}}_{\varnothing} \xrightarrow[b_2]{a_2} {N^{\mathsf{sl}}}_{\varnothing}$.*
- *If ${N^{\mathsf{sf}}}_{\mathcal{X}} \xrightarrow[b_1]{a_1} {N^{\mathsf{sf}}}_{\mathcal{Y}}$ and ${N^{\mathsf{sl}}}_{\varnothing} \xrightarrow[b_2]{a_2} {N^{\mathsf{sl}}}_{\varnothing}$, then $N_{\mathcal{X}} \xrightarrow[b_1+b_2]{a_1+a_2} N_{\mathcal{Y}}$.*

Proof. The proof follows by definition of the subnets and the operational semantics of P/T nets, as transitions of N are just partitioned into N^{sl} and N^{sf}.

In the following we let $\mathsf{I}_n \stackrel{\text{def}}{=} \bigotimes_n \mathsf{I} : (n, n)$ and define the following terms of the P/T calculus, $\forall n \in \mathbb{N}$:

$$\begin{array}{ll}
\mathsf{X}_0 \stackrel{\text{def}}{=} \mathsf{I} : (1,1) & \mathsf{\Lambda}_0 = \mathsf{V}_0 \stackrel{\text{def}}{=} \uparrow ; \downarrow : (0,0) \\
\mathsf{X}_1 \stackrel{\text{def}}{=} \mathsf{X} : (2,2) & \quad \mathsf{X}_{n+1} \stackrel{\text{def}}{=} (\mathsf{X}_n \otimes \mathsf{I}) \; ; \; (\mathsf{I}_n \otimes \mathsf{X}) : (n+2, n+2) \\
\mathsf{\Lambda}_1 \stackrel{\text{def}}{=} \mathsf{\Lambda} : (1,2) & \quad \mathsf{\Lambda}_{n+1} \stackrel{\text{def}}{=} (\mathsf{\Lambda} \otimes \mathsf{\Lambda}_n) \; ; \; (\mathsf{I} \otimes \mathsf{X}_n \otimes \mathsf{I}_n) : (n+1, 2n+2) \\
\mathsf{V}_1 \stackrel{\text{def}}{=} \mathsf{V} : (2,1) & \quad \mathsf{V}_{n+1} \stackrel{\text{def}}{=} (\mathsf{V} \otimes \mathsf{V}_n) \; ; \; (\mathsf{I} \otimes \mathsf{X}_n \otimes \mathsf{I}_n) : (2n+2, n+1)
\end{array}$$

It can be proved by induction that the only transitions for $\mathsf{\Lambda}_n$ and V_n are $\mathsf{\Lambda}_n \xrightarrow[b]{a} \mathsf{\Lambda}_n$ and $\mathsf{V}_n \xrightarrow[a]{b} \mathsf{V}_n$ with $|a| = n$, $|b| = 2n$, and $a_i = b_i + b_{n+i}$ for all $i < n$.

Definition 14. *Let P be a P/T calculus term s.t. $\{\![P]\!\} : m \to n$ and $\mathcal{MG}(\{\![P]\!\})$ is finite state. The normal form of P, written $nf(P)$, is as follows*

$$nf(P) = \mathsf{\Lambda}_m ; (\boldsymbol{T}_{can(\{\![P]\!\}^{\mathsf{sf}})} \otimes \boldsymbol{T}_{\widetilde{\{\![P]\!\}^{\mathsf{sl}}}}) ; \mathsf{V}_n$$

Lemma 7. *Let P be a P/T calculus term s.t. $\mathcal{MG}(\{\![P]\!\})$ is finite state. Then, P and $nf(P)$ are bisimilar.*

Proof. It follows from the behaviour of $\mathsf{\Lambda}_n$ and V_n, Lemmata 1, 4 and 6 and the correspondence Theorems 1 and 2.

Lemma 8. *Let P and Q be two bisimilar P/T calculus terms s.t. $\mathcal{MG}(\{\![P]\!\})$ and $\mathcal{MG}(\{\![Q]\!\})$ are finite state. Then, $\mathit{nf}(P) = \mathit{nf}(Q)$ (up-to iso).*

Proof. It follows by contradiction. Assume that $\mathit{min}(\{\![P]\!\}^{\mathsf{sf}}) = \mathit{min}(\{\![Q]\!\}^{\mathsf{sf}})$ and $\widetilde{\{\![P]\!\}^{\mathsf{sl}}} = \widetilde{\{\![Q]\!\}^{\mathsf{sl}}}$ does not hold. Therefore, it should be the case that either i) $\mathit{min}(\{\![P]\!\}^{\mathsf{sf}})$ and $\mathit{min}(\{\![Q]\!\}^{\mathsf{sf}})$ are not bisimilar; or ii) $\widetilde{\{\![P]\!\}^{\mathsf{sl}}} = \widetilde{\{\![Q]\!\}^{\mathsf{sl}}}$ are not bisimilar. In both cases we conclude that $\mathit{min}(\{\![P]\!\})$ and $\mathit{min}(\{\![Q]\!\})$ (and therefore P and Q) are not bisimilar. For (i), we note that the marking graphs differ in a transition connecting two different states (and this cannot be mimicked by stateless transitions); for (ii) every state will miss at least a self-loop transition (since $\mathcal{MG}(\{\![P]\!\})$ and $\mathcal{MG}(\{\![Q]\!\})$ are finite state, all infinite self-loops in the marking graphs are originated by stateless transitions).

Corollary 3 (Idempotency). *$\mathit{nf}(P) = \mathit{nf}(\mathit{nf}(P))$ (up-to iso).*

5 Normal Forms for the C/E Petri Calculus

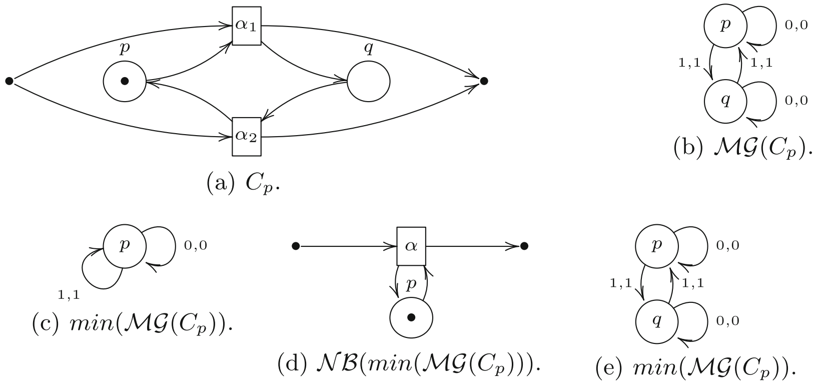
The case of C/E Petri calculus is quite interesting, because now any term P models a finite state connector, so that we can reduce to normal form any term.

Lemma 9. *Let P be a Petri calculus term. Then $\mathcal{MG}(\{\![P]\!\})$ is finite state.*

Proof. The C/E net with boundaries $\mathcal{MG}(P)$ has as many places as the number of subterms $\bigcirc$ and $\odot$ in P and the reachable states of $\mathcal{MG}(P)$ are just subsets of the places in $\mathcal{MG}(P)$, thus they are finitely many.

Now by using the approach for P/T nets we can obtain the normal form for every Petri calculus term. The only subtlety to deal with is when mapping a marking graph into a C/E net, because marking graphs can contain self-loops, as illustrated by the following example.

Example 8. Consider the C/E net C_p in Fig. 11(a). The corresponding marking graph is in Fig. 11(b) and the corresponding minimal automaton is in Fig. 11(c). If we apply $\mathcal{NB}$ we obtain the net in Fig. 11(d). Note that transition α cannot be fired under the C/E semantics because it inhibits consume/produce loops. Hence, the obtained net is not bisimilar to C_p. In order to translate back the minimal marking graph to a C/E net, we need to handle self-loops differently. While $\mathcal{NB}$ already removes any trivial self-loop (i.e., with empty observation) from the minimal marking graph, non-trivial self-loops are handled by duplicating states, as illustrated in Fig. 11(e). Then, the normal form is obtained by using the C/E corresponding to the minimal marking graph without non-trivial self-loops.



(a) C_p.

(b) $\mathcal{MG}(C_p)$.

(c) $min(\mathcal{MG}(C_p))$.

(d) $\mathcal{NB}(min(\mathcal{MG}(C_p)))$.

(e) $min(\mathcal{MG}(C_p))$.

Fig. 11. Minimisation of C/E nets

6 Concluding Remarks

In this paper we have considered a calculus of connectors that allows for the most general combination of synchronisation, non-determinism and buffering. The touchstone of its generality is its ability of modeling a variety of Petri nets compositionally, up to bisimilarity. Often bisimilarity implies the existence of a minimal representative, but such a construction has not been exhibited yet for Petri nets, at least directly. Thus in the paper we interpret the case graph of a net as a transition system labelled with the synchronizations observable on its boundaries. Then we can minimize such a LTS and reinterpret it univocally as a net and as a term of the calculus. Thus minimality is restricted to a case graph (step) semantics, which we might say observes parallelism but not concurrency.

Related Work. An algebra consisting of five kinds of basic stateless connectors (plus their duals) is presented in [8], together with the operational, observational and denotational semantics and a complete normal-form axiomatisation. The behaviour of connectors $\wedge$ and $\vee$ is slightly different from the one considered here, because in [8] only one action can take place at the time, e.g., only transitions $\wedge \xrightarrow[10]{1} \wedge$ and $\wedge \xrightarrow[01]{1} \wedge$ are considered instead of $\wedge \xrightarrow[nm]{n+m} \wedge$.

The Tile Model [15] offers a semantic framework for concurrent systems, of which the algebra of stateless connectors is just a particular instance. Roughly, the semantics of component-based systems can be expressed via tiles when configurations and observations form two monoidal categories with the same objects. Tiles define LTSs whose labels are pairs ⟨trigger, effect⟩. In this context, the usual notion of equivalence is called *tile bisimilarity*, which is a congruence (w.r.t. sequential and parallel composition) when a suitable rule format is met [15].

Reo [1] is an exogenous coordination model based on channel-like connectors that mediate the flow of data among components. Notably, a small set of point-to-point primitive connectors is sufficient to express a large variety of interesting interaction patterns, including several forms of mutual exclusion, synchronisation, alternation, and context-dependency. Components and primitive connectors can be composed into larger Reo circuits by disjoint union up-to the merging of shared nodes. The semantics of Reo has been formalised in many ways, tile model included [2]. See [17] for a recent survey.

BIP [4] is a component framework for constructing systems by superposing three layers of modelling: (1) Behaviour, representing the sequential computation of individual components; (2) Interaction, defining the handshaking mechanisms between these components; and (3) Priority, assigning a partial order of privileges to interactions. In absence of priorities, the interaction layer admits the algebraic presentation given in [5] and has been related to connectors in [10].

The wire calculus [27] takes inspiration from [19,20] but shares similarities with the tile model. It is presented as a process algebra where each process comes with a sort, written $P : (n, m)$ for a process P with n ports on the left and m on the right. The usual action prefixes $a.P$ of process algebras are extended by allowing the simultaneous input of a trigger a and output of an effect b, written $\frac{a}{b}.P$, where a (resp. b) is a string of actions, one for each port of the process. The Petri calculus [9,28] can be regarded as a dialect of the wire calculus.

Nets with boundaries [28] take inspiration from the open nets of [3], whose interfaces consist of places instead of ports.

Future Work. Some recent work [6,7] exploits an algebra of connectors similar to ours to define a relational denotational semantics and a structural operational semantics for *signal flow graphs*, a classical structure in control theory and signal processing. We plan to investigate connections between Petri nets with boundaries and signal flow graphs. We might also consider extending the results of this paper to other more expressive semantics, observing e.g. causality. Another direction in which our results could be extended is dealing with systems with a higher degree of dynamism, that adapt their behavior to evolving environments: e.g., systems whose structure and interaction capabilities can change at runtime. Some recent progresses in this direction are discussed in [11].

Acknowledgements. We thank the anonymous reviewers for their careful reading and helpful suggestions for improving the presentation. We would like to express infinite gratitude to José, for his guidance, support and friendship during our long-standing collaboration.

References

1. Arbab, F.: Reo: a channel-based coordination model for component composition. Math. Struct. Comp. Sci. **14**(3), 329–366 (2004)
2. Arbab, F., Bruni, R., Clarke, D., Lanese, I., Montanari, U.: Tiles for Reo. In: Corradini, A., Montanari, U. (eds.) WADT 2008. LNCS, vol. 5486, pp. 37–55. Springer, Heidelberg (2009)
3. Baldan, P., Corradini, A., Ehrig, H., Heckel, R.: Compositional semantics for open Petri nets based on deterministic processes. Math. Struct. Comp. Sci. **15**(1), 1–35 (2005)

4. Basu, A., Bozga, M., Sifakis, J.: Modeling heterogeneous real-time components in BIP. In: SEFM 2006, pp. 3–12. IEEE Computer Society (2006)
5. Bliudze, S., Sifakis, J.: The algebra of connectors - structuring interaction in BIP. IEEE Trans. Comput. **57**(10), 1315–1330 (2008)
6. Bonchi, F., Sobociński, P., Zanasi, F.: A categorical semantics of signal flow graphs. In: Baldan, P., Gorla, D. (eds.) CONCUR 2014. LNCS, vol. 8704, pp. 435–450. Springer, Heidelberg (2014)
7. Bonchi, F., Sobocinski, P., Zanasi, F.: Full abstraction for signal flow graphs. In: POPL 2015, pp. 515–526. ACM (2015)
8. Bruni, R., Lanese, I., Montanari, U.: A basic algebra of stateless connectors. Theor. Comput. Sci. **366**(1–2), 98–120 (2006)
9. Bruni, R., Melgratti, H., Montanari, U.: A connector algebra for P/T nets interactions. In: Katoen, J.-P., König, B. (eds.) CONCUR 2011. LNCS, vol. 6901, pp. 312–326. Springer, Heidelberg (2011)
10. Bruni, R., Melgratti, H., Montanari, U.: Connector algebras, Petri nets, and BIP. In: Clarke, E., Virbitskaite, I., Voronkov, A. (eds.) PSI 2011. LNCS, vol. 7162, pp. 19–38. Springer, Heidelberg (2012)
11. Bruni, R., Melgratti, H., Montanari, U.: Behaviour, interaction and dynamics. In: Iida, S., Meseguer, J., Ogata, K. (eds.) Specification, Algebra, and Software. LNCS, vol. 8373, pp. 382–401. Springer, Heidelberg (2014)
12. Bruni, R., Melgratti, H.C., Montanari, U., Sobocinski, P.: Connector algebras for C/E and P/T nets' interactions. Log. Methods Comput. Sci. **9**(3), 1–65 (2013)
13. Bruni, R., Meseguer, J., Montanari, U., Sassone, V.: Functorial models for Petri nets. Inf. Comput. **170**(2), 207–236 (2001)
14. Degano, P., Meseguer, J., Montanari, U.: Axiomatizing the algebra of net computations and processes. Acta Inf. **33**(7), 641–667 (1996)
15. Gadducci, F., Montanari, U.: The tile model. In: Proof, Language, and Interaction, pp. 133–166. The MIT Press (2000)
16. Hackney, P., Robertson, M.: On the category of props (2012). arXiv:1207.2773
17. Jongmans, S.S.T., Arbab, F.: Overview of thirty semantic formalisms for Reo. Sci. Ann. Comput. Sci. **22**(1), 201–251 (2012)
18. Kanellakis, P.C., Smolka, S.A.: CCS expressions, finite state processes, and three problems of equivalence. In: PODC 1983, pp. 228–240. ACM (1983)
19. Katis, P., Sabadini, N., Walters, R.F.C.: Representing place/transition nets in Span(Graph). In: Johnson, M. (ed.) AMAST 1997. LNCS, vol. 1349. Springer, Heidelberg (1997)
20. Katis, P., Sabadini, N., Walters, R.F.C.: Representing place/transition nets in Span(Graph). In: Johnson, M. (ed.) AMAST 1997. LNCS, vol. 1349, pp. 322–336. Springer, Heidelberg (1997)
21. MacLane, S.: Categorical algebra. Bull. AMS **71**(1), 40–106 (1965)
22. Meseguer, J., Montanari, U.: Petri nets are monoids. Inf. Comp. **88**(2), 105–155 (1990)
23. Meseguer, J., Montanari, U., Sassone, V.: On the semantics of place/transition Petri nets. Math. Struct. Comp. Sci. **7**(4), 359–397 (1997)
24. Paige, R., Tarjan, R.E.: Three partition refinement algorithms. SIAM J. Comput. **16**(6), 973–989 (1987)
25. Perry, D.E., Wolf, E.L.: Foundations for the study of software architecture. ACM SIGSOFT Soft. Eng. Notes **17**, 40–52 (1992)
26. Petri, C.: Kommunikation mit Automaten. Ph.D. thesis, Institut für Instrumentelle Mathematik, Bonn (1962)

27. Sobocinski, P.: A non-interleaving process calculus for multi-party synchronisation. In: ICE 2009, EPTCS, vol. 12, pp. 87–98 (2009)
28. Sobociński, P.: Representations of Petri net interactions. In: Gastin, P., Laroussinie, F. (eds.) CONCUR 2010. LNCS, vol. 6269, pp. 554–568. Springer, Heidelberg (2010)