



# A Congruence for Petri Nets

Vladimiro Sassone

*University of Sussex*

Paweł Sobociński

*Università di Pisa*

---

## Abstract

We introduce a way of viewing Petri nets as open systems. This is done by considering a bicategory of cospans over a category of p/t nets and embeddings. We derive a labelled transition system (LTS) semantics for such nets using GIPOs and characterise the resulting congruence. Technically, our results are similar to the recent work by Milner on applying the theory of bigraphs to Petri Nets. The two main differences are that we treat p/t nets instead of c/e nets and we deal directly with a category of nets instead of encoding them into bigraphs.

*Keywords:* Petri Nets, contexts, congruences

---

## 1 Introduction

The theory of Petri nets is an attractive graphical formalism which captures several interesting issues associated with concurrency. More recently, the field of process calculus has been concerned with exploring syntactic formalisms for the description of concurrent and mobile systems and with associated proof techniques. This paper is an attempt to relate the two perspectives by treating Petri nets as a sort of algebra, together with a labelled transition system semantics and an associated bisimulation congruence.

---

\* This work has been partially supported by EU FET-GC ‘MyTHS: Models and Types for Security in Mobile Distributed Systems’, IST-2001-32617 and ‘AGILE: Architectures for Mobility’, IST-2001-32747; and EC RTN 2-2001-00346 SEGRAVIS.

We introduce the notion of *Petri context* which is a marked Petri net enriched with an *inner* and an *outer* interface; for us an interface is simply an arbitrary subset of the places of the net. A net with inner interface of cardinality  $m$  and an outer interface of cardinality  $n$  is an arrow of a bicategory.

The composition of two such nets is simply the glueing together of nets at their common interface. The usual “token-game” semantics of nets is implemented as a reduction relation on the *ground* nets, which are nets with empty inner interface. Indeed, each reaction rule models the firing of a net transition, in the sense that the left hand side of the rule is the transition with its input places marked and the output places unmarked, while the right hand side is the transition with its input places unmarked and its output places marked. The fact that we are concerned with terms (ground nets), contexts (Petri contexts) and a reduction semantics clarifies our earlier remark about viewing Petri nets as a kind of process calculus.

The advantage of such an approach is that we are now able to use intuitions and techniques from the field of process-calculi to reason about a net’s behaviour. Indeed, instead of studying a (ground) net’s internal behaviour as is usual in Petri net theory, we restrict our view of the net to the outer interface and study how the net interacts with its environment by substituting it into an arbitrary Petri context and studying the behaviour of the resulting larger system. We believe that this is a valid and interesting way of treating nets as open systems. In this paper, we begin the study of such nets and characterise the canonical strong bisimilarity that arises by synthesising a labelled transition system using the technique introduced by Leifer and Milner [6] and expanded by the authors in [9].

Leifer and Milner introduced the notion of reactive systems in [6] as a generalisation of several situations that occur in the field of process calculus. Indeed, since the publication of Berry and Boudol’s influential work on the chemical abstract machine [2], it has become commonplace to define the operational semantics of calculi via a reduction semantics – often generated from basic rules and closed by substituting for parameters and into arbitrary reactive/evaluation contexts. A labelled transition system is then normally given later; these are vital because the notion of bisimulation and the associated coinductive reasoning is a powerful tool for reasoning about contextual equivalence, provided that bisimilarity is sound for such an equivalence.

Leifer and Milner were particularly interested in using their theory on examples where the contexts were not term algebras, but rather were of a graphical nature. This work, developed to some extent in Leifer’s thesis [5], has led to the definition of *bigraphs* by Milner and Jensen [3]. Bigraphs are a powerful formalism intended to act as a sort of unifying language for con-

current formalisms – in the sense that the study the relationship between two languages, one may first encode them into bigraphs and study their theory as *bigraphical reactive systems*. Unfortunately, relative pushouts (RPOs) typically do not exist in such cases, essentially because the algebraic structure of contexts induces non trivial context automorphisms. Leifer and Milner solved this problem by adding intensional information to the objects or the arrows of the categories involved, taking away the possibility of automorphisms. As a side-effect arrow composition becomes a partial operation, giving rise to Milner’s notion of *precategories*.

The present authors, in [9], showed an alternative approach. They proceeded not by forgetting the automorphisms, rather by keeping them as a first-class member of the category – this meant working in a 2-category or bicategory instead of an ordinary category. The resulting generalisation of relative pushouts, *groupoidal relative pushouts* or GRPOs satisfy similar properties to RPOs. Moreover, the solutions proposed by Milner and Leifer can be recast naturally in this more abstract approach, as illustrated in [8].

Recently, the authors have shown that GRPOs exist within a wide class of cospan bicategories over *adhesive categories*, the latter were introduced by the second author and S. Lack [4]. The technical details of this paper can be largely seen as an application of that general theorem to the adhesive category of Petri nets and embeddings **MNet** (cf. section 2).

Recently, Milner [7] has applied the theory of bigraphs to *c/e* Petri nets, by encoding the nets into bigraphs and studying their behavioural theory. Unfortunately, in order to make this work, Milner had to introduce a further complication into the theory of bigraphs – namely a type system. This was necessary in order to restrict the allowable contexts – since it is possible to construct bigraphs which are not in the image of the encoding. We avoid such encoding problems by working directly with a category of Petri nets.

Our work is related to the work on open Petri nets by Baldan, Corradini, Ehrig and Heckel [1]. While we leave it as future work to determine the precise relationship, we mention here the main structural difference. Firstly, an open place in their terminology is a place in our outer interface. Baldan et al. make a distinction between *input places* and *output places*, where an input place is able to receive tokens from the environment and an output place is able to contribute tokens to the environment. All the places in our interfaces are both input and output in their sense, since they are able to both receive tokens from and contribute tokens to the environment.

In section 2 we introduce the category of nets which shall be relevant throughout the paper. We turn such nets into open systems by providing them with inner and outer interfaces in section 3. We provide the semantics of the

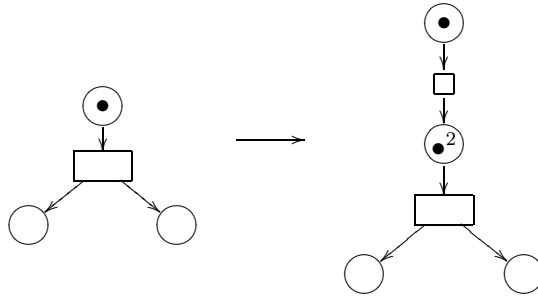


Fig. 1. Example of an embedding.

model is section 4 and proceed to derive the labelled transition semantics using GRPOs. In section 5 we characterise the resulting bisimulation congruence using a simpler LTS. We conclude and offer directions for future work in section 6. In order to increase the readability of this introductory exposition we have left out most of the proofs.

## 2 Petri nets and embeddings

A *marked net*  $p$  is a place-transition net together with a marking; more formally it is a quadruple  $\langle P, T, K, s, t, k \rangle$  where  $P, T$  and  $K$  are respectively finite sets of places, transitions and tokens,  $s, t : T \rightarrow S^\oplus$  are, respectively, the sources and targets of a transition, and  $k : K \rightarrow P$  is the positions of the tokens on the places. For the purpose of this paper, we rule out nets with *self-loops*, e.g., we assume that transitions have disjoint pre- and post-sets. We shall define a structural notion of Petri net morphism, as opposed to the usual behavioural notions; this is because we shall use the morphisms to speak about the topology of a net instead of its behaviour. Thus a net *morphism*  $f : p \rightarrow p'$  consists of maps  $f_0 : P \rightarrow P', f_1 : T \rightarrow T'$  and  $f_2 : K \rightarrow K'$  such that  $s'f_1 = f_0^\oplus s$  and  $t'f_1 = f_0^\oplus t$  and  $k'f_2 = f_0k$ . As we are interested exclusively in embeddings, we shall assume that  $f_0, f_1$  and  $f_2$  are injective. Let **MNet** be the category of marked nets and embeddings.

**Example 2.1** Consider the two marked nets illustrated in Figure 1. There is an obvious embedding of the left net into the right net, as indicated by the arrow.

**Proposition 2.2** **MNet** is adhesive.

**Proof.** The category **MNet** is actually the subcategory of monos of a presheaf topos. We leave the details to the reader.  $\square$

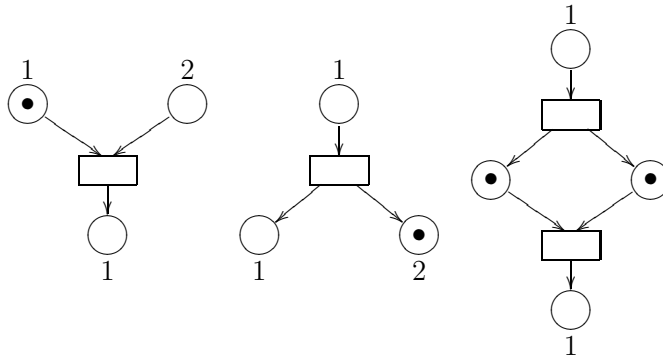


Fig. 2. Nets  $p_1$ ,  $p_2$  and their composition  $p_2p_1 : 1 \rightarrow 1$ .

### 3 Nets with interfaces

We are now ready to endow our nets with interfaces which shall allow us to compose them.

**Definition 3.1 (Net with interfaces)** Given finite ordinals  $m$  and  $n$ , a (marked) net with inner interface  $m$  and outer interface  $n$ , denoted  $p : m \rightarrow n$ , is a marked net  $p = \langle P, T, K, s, t, k \rangle$  together with injective functions  $\iota : m \rightarrow P$  and  $o : n \rightarrow P$ . Graphically, we represent such a net by labelling a subset of places with 1 through to  $m$ , the inner-face, and a subset with 1 through to  $n$ , the outer-face.

We shall follow the convention of drawing the numbers corresponding to the inner interface under a place and the numbers corresponding to the outer interface over a place.

We shall sometimes refer to a net  $p : m \rightarrow n$  with interfaces as a *Petri context*. When  $m = 0$  we shall refer to such net as a *Petri term*.

**Example 3.1** Consider the three nets illustrated in Figure 2. The leftmost net has inner interface 1 and outer interface 2, the second net has inner interface 2 and outer interface 1 while the rightmost net has both inner and outer interface 1.

**Definition 3.2 (Bicategory of nets with interfaces)** The bicategory of nets with interfaces **INet** has:

- objects: the finite ordinals  $0, 1, \dots$ ;
- arrows: the arrows from  $m$  to  $n$  are the marked nets  $p : m \rightarrow n$  with inner interface  $m$  and outer interface  $n$ ;
- 2-cells: marked net isomorphisms which preserve the inner and the outer interface – in other words, a 2-cell  $\alpha : p \rightarrow p' : m \rightarrow n$  is an embedding

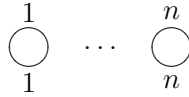


Fig. 3. The identity  $\text{id} : n \rightarrow n$ .

$\alpha : p \rightarrow p'$  which is surjective on places, transitions and tokens and moreover  $\alpha\iota = \iota'$  (inner interfaces are preserved) and  $\alpha o = o'$  (outer interfaces are preserved).

We illustrate the identity arrows of **INet** in Figure 3.

Composition in **INet** can be described intuitively as the glueing of marked nets along their common interface. More formally, given nets  $p_1 : k \rightarrow l$  and  $p_2 : l \rightarrow m$ , the composite net  $p_2 p_1 : k \rightarrow m$  is obtained by taking the pushout in **MNet** of  $o_1 : l \rightarrow p_1$  and  $\iota_2 : l \rightarrow p_2$ . Thus the result is obtained by putting the two nets side by side, equating the places with shared interface and adding together the tokens on these equated places.

**Example 3.2** The rightmost net in Figure 2 is the composition of the other two nets.

We shall say that a marked net is *discrete* when it contains no transitions and no tokens – in other words it is a set of places. The bicategory **INet** is actually (biequivalent to) the full subcategory of the bicategory  $\text{Cospan}(\mathbf{INet})$  with discrete nets as objects. This fact is useful for us because we are able to apply the central result of [10] which provides a construction of GRPOs in certain cospan bicategories over adhesive categories.

**Lemma 3.3** **INet** has GRPOs.

**Proof.** Because **INet** can be seen as a full subcategory of a cospan bicategory over an adhesive category (see Proposition 2.2), the main theorem of [10] applies to the larger cospan bicategory. Thus, it remains only to verify that, starting with a diagram in **INet**, the construction of GRPOs within the larger bicategory  $\text{Cospan}(\mathbf{INet})$  results in a diagram in **INet**. We leave out this routine calculation. □

## 4 The token game as reaction

In this section, we formalise the token-game semantics of nets via reaction rules. This allows us to use the framework of reactive systems (cf. [6, 8, 11]) and derive a canonical labelled transition system on which bisimilarity is a congruence.

**Definition 4.1** ( $\mathbb{P}$ ) We obtain a reactive system  $\mathbb{P}$  from **INet** by letting the

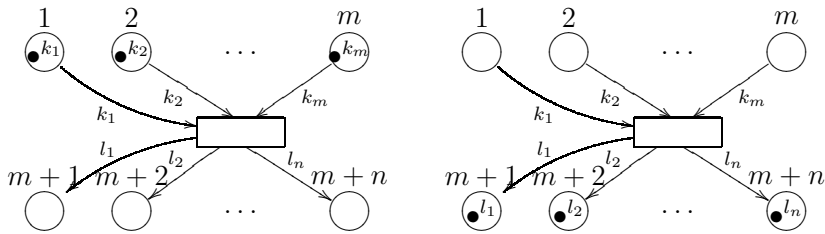


Fig. 4. A reaction rule  $\langle l : 0 \rightarrow m + n, r : 0 \rightarrow m + n \rangle \in \mathcal{R}$ .

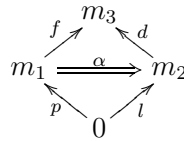


Fig. 5. GIPO in **INet** giving rise to a label in  $\text{LTS}(\mathbb{P})$ .

set of reactions  $\mathcal{R}$  include, for every  $m, n \in \mathbb{N}$  and  $k_1, \dots, k_m, l_1, \dots, l_n \in \mathbb{N}^+$  a reaction rule as illustrated in Figure 4. One also needs to provide reaction rules for transitions which have places both in the pre- and post-set; we leave the details to the reader.

Let  $\longrightarrow$  denote the reaction relation obtained by closing the reaction rules detailed in Definition 4.1 under composition with arbitrary Petri contexts. Intuitively, this relation corresponds to the ordinary Petri net semantics in the sense that we have  $p \longrightarrow^* p'$  iff  $p'$  can be obtained from  $p$  by firing a number of transitions.

Using GIPOs, we can generate a labelled transition system  $\text{LTS}(\mathbb{P})$  to reason about marked nets with interfaces. The states are Petri terms modulo isomorphism. The labels are the smallest contexts which allow a reaction to occur, in the sense that the resulting redex diagram is a GIPO. We direct the reader to [9, 8, 11] for background on such construction and for further details.

**Definition 4.2** ( $\text{LTS}(\mathbb{P})$ ) The labelled transition system  $\text{LTS}(\mathbb{P})$  has:

- states: Petri terms modulo isomorphism;
- transitions:  $p \xrightarrow{f} dr$  if there exists a reaction rule  $\langle l, r \rangle \in \mathcal{R}$ , a Petri context  $d$  and a net isomorphism  $\alpha : fp \rightarrow dl$  so that the diagram in Figure 5 is a GIPO in **INet**.

**Example 4.1** We illustrate an example of a transition in Figure 6. The corresponding GIPO is illustrated in Figure 7, where the four Petri nets depict the arrows on the respective edges of the  $\alpha$  redex-square.

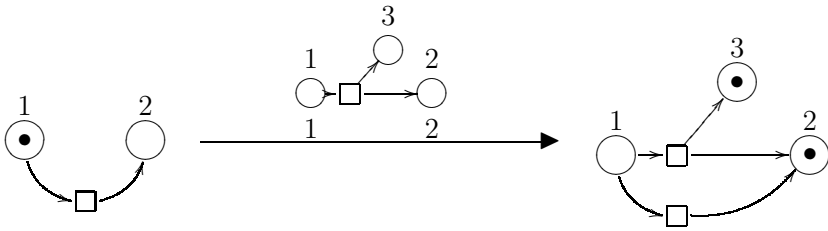


Fig. 6. Typical transition.

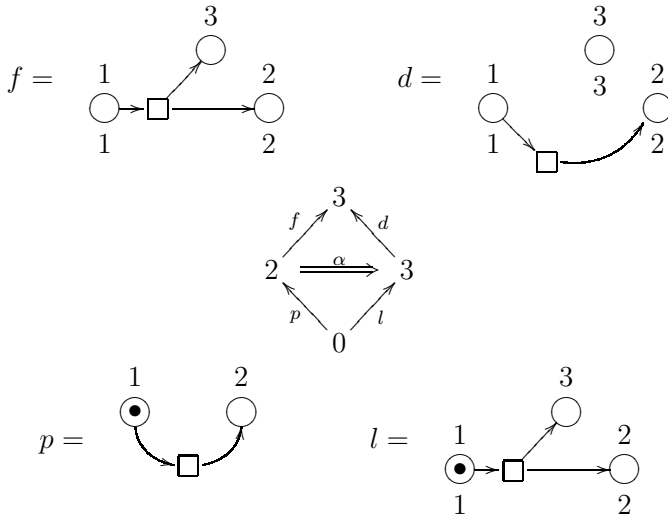


Fig. 7. GIPO corresponding to the transition in Figure 6.

As a consequence of a general result [9] (which is a generalisation of the original result [6]) by Leifer and Milner, strong bisimilarity on  $LTS(\mathbb{P})$  is a congruence. In this paper, we concentrate solely on strong bisimilarity; we leave the treatment of weak bisimilarity as future work. Weak bisimilarity promises to be a more interesting equivalence because it does not distinguish terms based only on internal behaviour. However, strong bisimilarity already makes several interesting identifications on Petri terms, some of which we illustrate in Figure 8.

## 5 Characterising bisimilarity

The labelled transition system  $LTS(\mathbb{P})$  derived using GIPOs is canonical and bisimilarity on it is a congruence. However, because of the nature of the underlying reactive system,  $LTS(\mathbb{P})$  is easily seen to be infinitely branching and may seem rather complex at first. Here we characterise the bisimulation



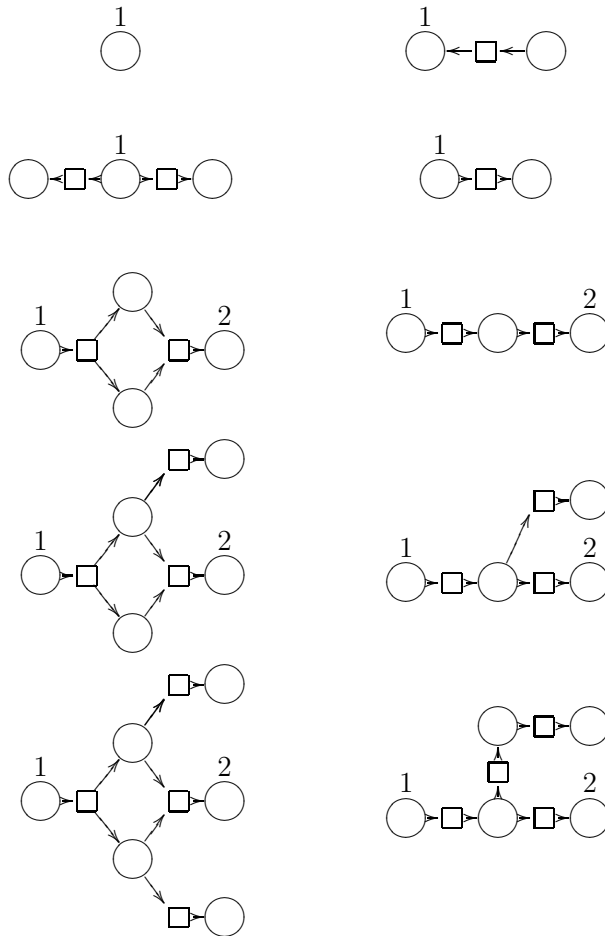


Fig. 8. Pairs of bisimilar Petri terms.

congruence using a much simpler setting; indeed, we shall define an alternative simple labelled transition system on Petri terms and prove that the resulting bisimilarity coincides with bisimilarity on  $LTS(\mathbb{P})$ .

**Definition 5.1** We define an lts  $T_{\pm}$  as follows:

- states are Petri terms modulo isomorphism;
- there are three types of transitions:
  - $p \xrightarrow{+i} p'$  if  $p'$  is the net resulting from  $p$  by adding a token at its  $i$ th outer place;
  - $p \xrightarrow{-i} p'$  if  $p$  has a token at its  $i$ th outer node and  $p'$  is the net resulting from  $p$  after removing a token from its  $i$ th outer place;
  - $p \xrightarrow{\tau} p'$  if  $p'$  results from  $p$  through the firing of one transition (without

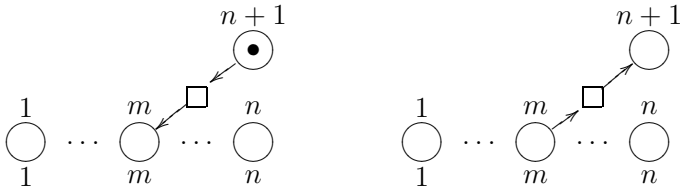


Fig. 9.  $in_{m,n} : n \rightarrow n + 1$  and  $out_{m,n} : n \rightarrow n + 1$

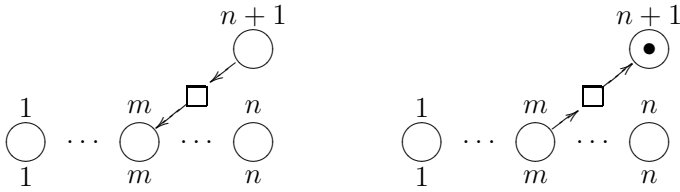


Fig. 10.  $\overline{in}_{m,n} : n \rightarrow n + 1$  and  $\overline{out}_{m,n} : n \rightarrow n + 1$ .

any interaction with the environment).

Let  $\sim_{\pm}$  denote (strong) bisimilarity on such lts.

The following lemma presents some of the properties of  $\sim_{\pm}$ .

**Lemma 5.2**

- (i) if  $p \sim_{\pm} q$  then  $p$  and  $q$  have equal interfaces, with the same number of tokens on each outer place;
- (ii) if  $p \sim_{\pm} q$  and  $p$  contains a transition which consumes  $k_{i_1}, \dots, k_{i_h}$  tokens from respectively  $p$ 's  $i_1$ th,  $\dots$ ,  $i_h$ th outer place, then  $q$  contains a transition which also consumes  $k_{i_1}, \dots, k_{i_h}$  tokens from  $q$ 's  $i_1$ th,  $\dots$ ,  $i_h$ th outer place.

Returning to  $LTS(\mathbb{P})$ , it is easy to see that the labels generated via GIPOs contain at most one Petri net transition, since every individual reaction corresponds to the the firing of a single transition in the token game.

**Proposition 5.1** If  $p \xrightarrow{f} p'$  then  $f$  contains at most one transition.

It is clear that if  $p \xrightarrow{f} p'$  and  $f$  is discrete then the transition corresponds to an internal firing of a transition without any interaction from the environment.

The following lemma relates such internal transitions in both the transition systems.

**Lemma 5.3**  $p \xrightarrow{\tau} p'$  in  $T_{\pm}$  iff  $p \xrightarrow{id_n} p'$  in  $LTS(\mathbb{P})$ .

**Proof.** Omitted. □

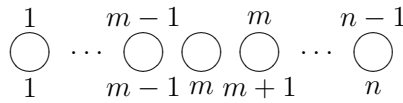


Fig. 11.  $\setminus_{m,n} : n \rightarrow n - 1$ .

It is also possible that  $f$  contains no transitions but instead adds tokens to one or more outer places of  $p$  which allows a transition of within  $p$  to fire. The properties of GRPOs guarantee that the tokens added are *precisely* the tokens required for the transition to fire, that is, no unnecessary tokens are added.

The following lemma relates such transitions with traces in  $T_{\pm}$ .

**Lemma 5.4** Suppose that  $p \xrightarrow{f} p'$  and  $f$  contains no transitions and is not discrete. Then there exist  $m > 0$ ,  $1 \leq i_1, \dots, i_m \leq n$  and  $k_1, \dots, k_m > 0$  such that  $p \xrightarrow{+i_1}_{k_1} \dots \xrightarrow{+i_m}_{k_m} \xrightarrow{\tau} p'$ , where  $\xrightarrow{i}$  denotes a sequence of  $i$  transitions.

We now turn our attention to the only other possible case for  $f$ , by relating in the following lemma relates labelled transitions in  $LTS(\mathbb{P})$  which contain a Petri transition, and certain traces in  $T_{\pm}$ .

**Lemma 5.5** Consider a diagram as in Figure 5, where  $f$  is a Petri context which contains exactly one transition  $t$  with  $m$  input places,  $n$  output places, with  $k_i$  tokens being consumed at the  $i$ th input place and  $l_i$  tokens being produced at the  $i$ th outer place, as in Figure 4.

Let  $i_1, \dots, i_{m'}$  be the places of  $p$  which are identified with the input places of  $t$ , and let  $k'_1, \dots, k'_{m'}$  denote the number of tokens consumed from these places as a result of firing the transition.

Similarly, let  $i'_1, \dots, i'_{n'}$  be the places of  $p$  which are identified with the output places of  $t$ , and let  $l'_1, \dots, l'_{n'}$  be the number of tokens which result from firing the transition.

Then, the following facts hold, where  $\xrightarrow{i}$  denotes a sequence of  $i$  transitions, and  $\bar{f}$  represent the residue of  $f$  after the firing.

- (i) if  $p \xrightarrow{f} p''$  is a transition in  $LTS(\mathbb{P})$  then there exists  $p'$  such that  $p'' = \bar{f}p'$  and  $p \xrightarrow{-i_1}_{k'_1} \dots \xrightarrow{-i_{m'}}_{k'_{m'}} \xrightarrow{+i'_1}_{l'_1} \dots \xrightarrow{+i'_{n'}}_{l'_{n'}} p'$  is a trace in  $T_{\pm}$ ;
- (ii) if  $p \xrightarrow{-i_1}_{k'_1} \dots \xrightarrow{-i_{m'}}_{k'_{m'}} \xrightarrow{+i'_1}_{l'_1} \dots \xrightarrow{+i'_{n'}}_{l'_{n'}} p'$  is a trace in  $T_{\pm}$  then  $p \xrightarrow{f} \bar{f}p'$  is a transition in  $LTS(\mathbb{P})$ .

**Proof.** Omitted. □

Two particular labels of transitions in  $LTS(\mathbb{P})$  shall be useful,  $in_{m,n}$  and  $out_{m,n} : n \rightarrow n + 1$ , illustrated in Figure 9. A transition  $in_{m,n}$  results in

the introduction of a token at the  $m$ th outer place of a net. Similarly, a transition  $out_{m,n}$  results in the taking away one token from the  $m$ th outer place. The transitions  $in_{m,n}$  and  $out_{m,n}$  are not exactly the same as  $+m$  and  $-m$  of  $T_{\pm}$  because they leave behind a residue  $\overline{in_{m,n}}$  and  $\overline{out_{m,n}}$  as illustrated in Figure 10.

In the case of  $in_{m,n}$ , the residue can be forgotten as we now demonstrate. In the following let  $\setminus_{m,n}$  be the Petri context illustrated in Figure 11. We use  $\setminus_n$  as a shorthand for  $\setminus_{n,n}$ .

**Lemma 5.6** Given an arbitrary Petri term  $p$  with outer interface  $n$ , we have that  $\setminus_{n+1}\overline{in_{m,n}}p \sim p$ .

**Corollary 5.7** For any two Petri terms  $p$  and  $q$ , if  $\overline{in_{m,n}}p \sim \overline{in_{m,n}}q$  then  $p \sim q$ .

**Proof.** Using the conclusion of Lemma 5.6 and the fact that  $\sim$  is a congruence we have  $p \sim \setminus_{n+1}\overline{in_{m,n}}p \sim \setminus_{n+1}\overline{in_{m,n}}q \sim q$ . □

The situation for  $out$  is slightly trickier in that the analogue of Lemma 5.6 is actually false – the transition introduced into a net by  $out$  can keep firing if more tokens appear at its input place. A simple counterexample is the Petri term  $p = \overset{1}{\odot}$ . Although  $p$  clearly cannot perform any internal behaviour, we have that  $\setminus_2\overline{out_{1,1}}p$  can perform a reaction. Fortunately, the analogue of Corollary 5.7 holds.

**Lemma 5.8** For any two Petri terms  $p$  and  $q$ , if  $\overline{out_{m,n}}p \sim \overline{out_{m,n}}q$  then  $p \sim q$ .

**Proof.** Omitted. □

We are now ready to prove the main result of this section which is that the observational power of the two labelled transition systems coincides.

**Theorem 5.9**  $\sim = \sim_{\pm}$ .

**Proof.** (1)  $\sim \subseteq \sim_{\pm}$ . First we shall show that the labels of  $T_{\pm}$  are at least as powerful as the labels generated via GIPOs.

It is enough to show that  $\{(p, q) \mid p \sim q\}$  is a  $T_{\pm}$  bisimulation. Indeed, suppose that  $p \xrightarrow{+i} p'$ . Then  $p \xrightarrow{in_{i,n}} \overline{in_{i,n}}p'$  using part 2 of Lemma 5.5. Using the fact that  $p \sim q$ , we have  $q \xrightarrow{in_{i,n}} q''$  and  $\overline{in_{i,n}}p' \sim q''$ ; using part 1 of Lemma 5.5, there exists a  $q'$  such that  $q'' = \overline{in_{i,n}}q'$  and  $q \xrightarrow{+i} q'$ . Moreover, we have  $\overline{in_{i,n}}p' \sim \overline{in_{i,n}}q'$ , which using Corollary 5.7 yields  $p' \sim q'$ .

The case of  $p \xrightarrow{-i} p'$  is similar, with Lemma 5.8 playing a key role.

Finally, using Lemma 5.3, if  $p \xrightarrow{\tau} p'$  then  $p \xrightarrow{id} p'$ . Then  $q \xrightarrow{id} q'$  for some  $q'$  such that  $p' \sim q'$ . Using Lemma 5.3 again yields  $q \xrightarrow{\tau} q'$ .

(2)  $\sim_{\pm} \subseteq \sim$ . We shall show that the contextual closure of  $\sim_{\pm}$ ,  $A = \{ (cp, cq) \mid p \sim_{\pm} q \}$  is a LTS( $\mathbb{P}$ )-bisimulation.

Indeed, suppose that for some  $p \sim_{\pm} q$  we have  $cp \xrightarrow{f} p'$ . Using standard theory [9], we can decompose the GIPO corresponding to the label and obtain a transition  $p \xrightarrow{f'} p''$  and context  $d'$  so that  $d'p'' = p'$ .

We now use the fact that  $f'$  is either discrete, consists only of tokens, or consists of exactly one transition. If  $f'$  is discrete, one uses Lemma 5.3 to obtain that also  $q \xrightarrow{f'} q''$ , with  $p'' \sim_{\pm} q''$ . Because GIPOs compose, we obtain  $cq \xrightarrow{f'} d'q''$ , and clearly  $(p', d'q'') \in A$ .

If  $f'$  is not discrete but contains only tokens, we use Lemma 5.4 to obtain a trace  $p \xrightarrow{+i_1}_{k_1} \dots \xrightarrow{+i_m}_{k_m} \xrightarrow{t} p''$ ; yielding a corresponding trace  $q \xrightarrow{+i_1}_{k_1} \dots \xrightarrow{+i_m}_{k_m} \xrightarrow{\tau} q''$  with  $p'' \sim_{\pm} q''$ . Using the second part of Lemma 5.2, we are able to conclude that  $q$  has a transition which requires precisely the added tokens to fire; meaning that we are able to derive  $q \xrightarrow{f'} q''$ . We are now able to use the fact that GIPOs compose as in the previous case.

Finally, if  $f$  contains a single transition  $t$ , then using part 1 of Lemma 5.5, there exist  $i_1, \dots, i_m$  and  $j_1, \dots, j_n$  such that  $p \xrightarrow{-i_1}_{k_1} \dots \xrightarrow{-i_m}_{k_m} \xrightarrow{+j_1}_{l_1} \dots \xrightarrow{+j_n}_{l_n} p'''$  such that  $p'' = \overline{f'}p'''$ . Then we have that  $q$  can also perform the above trace and arrive at  $q'$  such that  $p''' \sim q'$ . Using part 2 of Lemma 5.5,  $q \xrightarrow{f'} \overline{f'}q'$ . Using the fact that GIPOs compose, we have  $cq \xrightarrow{f} d'\overline{f'}q'$ .  $\square$

## 6 Conclusion and future work

We have described a way of using Petri nets in order to model open systems. The approach is based on previous theoretical work and inherits a canonical labelled transition system and the resulting strong bisimulation congruence. We have characterised the congruence using a simpler transition system. Our work is technically very similar to Milner’s treatment of Petri nets inside bigraphs.

As future work, we plan to analyse the exact relationship of our approach with open nets [1]. We also plan to study weak bisimilarity which promises to be more interesting than strong bisimilarity from an operational perspective.

## Acknowledgement

The authors thank the anonymous referee for several useful comments which have allowed us to improve the article.

## References

- [1] Baldan, P., A. Corradini, H. Ehrig and R. Heckel, *Compositional modeling of reactive systems using open nets*, in: *CONCUR '01*, Lecture Notes in Computer Science **2154** (2001), pp. 502–518.
- [2] Berry, G. and G. Boudol, *The chemical abstract machine*, Theoretical Computer Science **96** (1992), pp. 217–248.
- [3] Jensen, O. H. and R. Milner, *Bigraphs and mobile processes*, Technical Report 570, Computer Laboratory, University of Cambridge (2003).
- [4] Lack, S. and P. Sobociński, *Adhesive categories*, in: *Foundations of Software Science and Computation Structures FoSSaCS '04*, LNCS (Lecture Notes in Computer Science) **2987** (2004), pp. 273–288.
- [5] Leifer, J., “Operational congruences for reactive systems,” Phd thesis, University of Cambridge (2001).
- [6] Leifer, J. and R. Milner, *Deriving bisimulation congruences for reactive systems*, in: *International Conference on Concurrency Theory Concur '00*, LNCS (Lecture Notes in Computer Science) **1877** (2000), pp. 243–258.
- [7] Milner, R., *Bigraphs for petri nets*, in: *Lectures on Concurrency and Petri Nets 2003*, Lecture Notes in Computer Science **3098** (2004), pp. 686–701.
- [8] Sassone, V. and P. Sobociński, *Deriving bisimulation congruences: 2-categories vs. precategories*, in: *Foundations of Software Science and Computation Structures FoSSaCS '03*, LNCS (Lecture Notes in Computer Science) **2620** (2003).
- [9] Sassone, V. and P. Sobociński, *Deriving bisimulation congruences using 2-categories*, Nordic Journal of Computing **10** (2003), pp. 163–183.
- [10] Sassone, V. and P. Sobociński, *Congruences for contextual graph-rewriting*, Technical Report RS-04-11, BRICS, University of Aarhus (2004).
- [11] Sobociński, P., “Deriving process congruences from reaction rules,” Ph.D. thesis, BRICS, University of Aarhus (2004).