Logical Methods in Computer Science
Vol. 4 (4:3) 2008, pp. 1–41
www.lmcs-online.org

Submitted Dec. 23, 2007
Published Oct. 11, 2008

BISIMILARITY AND BEHAVIOUR-PRESERVING RECONFIGURATIONS OF OPEN PETRI NETS

PAOLO BALDAN [a], ANDREA CORRADINI [b], HARTMUT EHRIG [c], REIKO HECKEL [d], AND BARBARA KÖNIG [e]

[a] Dipartimento di Matematica Pura e Applicata, Università di Padova, Italy
e-mail address: baldan@math.unipd.it

[b] Dipartimento di Informatica, Università di Pisa, Italy
e-mail address: andrea@di.unipi.it

[c] Institut für Softwaretechnik und Theoretische Informatik, Technische Universität Berlin, Germany
e-mail address: ehrig@cs.tu-berlin.de

[d] Department of Computer Science, University of Leicester, UK
e-mail address: reiko@mcs.le.ac.uk

[e] Abteilung für Informatik und Angewandte Kognitionswissenschaft, Universität Duisburg-Essen, Germany
e-mail address: barbara_koenig@uni-due.de

Abstract. We propose a framework for the specification of behaviour-preserving reconfigurations of systems modelled as Petri nets. The framework is based on open nets, a mild generalisation of ordinary Place/Transition nets suited to model open systems which might interact with the surrounding environment and endowed with a colimit-based composition operation. We show that natural notions of bisimilarity over open nets are congruences with respect to the composition operation. The considered behavioural equivalences differ for the choice of the observations, which can be single firings or parallel steps. Additionally, we consider weak forms of such equivalences, arising in the presence of unobservable actions. We also provide an up-to technique for facilitating bisimilarity proofs. The theory is used to identify suitable classes of reconfiguration rules (in the double-pushout approach to rewriting) whose application preserves the observational semantics of the net.

1998 ACM Subject Classification: F.3.1, F.4.2.

Key words and phrases: Open systems, Petri nets, bisimilarity, compositionality, reconfiguration, behaviour preserving transformations.

Research partially supported by the EU IST-2004-16004 SENSORIA, the MIUR Project ART, the DFG project SANDS, the DFG project Behaviour-GT and CRUI/DAAD VIGONI "Models based on Graph Transformation Systems: Analysis and Verification".

LOGICAL METHODS IN COMPUTER SCIENCE

DOI:10.2168/LMCS-4 (4:3) 2008

Introduction

Petri nets are a well-known model of concurrent and distributed systems, widely used both in theoretical and applicative areas. In classical approaches, such as [34], nets are intended to represent closed, completely specified systems evolving autonomously through the firing of transitions. In order to represent *open* systems, namely systems which can interact with the surrounding environment or, from a different perspective, systems which are only partially specified, several extensions of the basic model of Petri nets have been considered in the literature. Conceptually, this effort dates back to the early works on net composition and refinement and to the studies concerning the development of compositional semantics for Petri nets (a discussion of the related literature can be found in the concluding section).

Generally speaking, important issues that must be faced when modelling open systems can be summarised as follows. Firstly, a large (possibly still open) system is typically built out of smaller open components. Syntactically, an open system is equipped with suitable interfaces, over which the interaction with the external environment can take place. Semantically, openness can be represented by defining the behaviour of a component as if it were embedded in general environments, determining any possible interaction over the interfaces.

Secondly, often the building components of an open system are not statically determined, but they can change during the evolution of the system, according to predefined reconfiguration rules triggered by internal or external solicitations.

The work in this paper outlines a framework where open systems can be modelled as Petri nets, capturing both the requirements mentioned above. Observational semantics based on (weak) bisimulation are shown to be congruences with respect to the composition operation defined over Petri nets. Building on this, suitable reconfigurations of such systems can be specified as net rewritings, which preserve the behaviour of the system. The relation with other approaches in the literature addressing similar issues will be discusses in Section 7.

The framework presented here is based on so-called *open nets*, a mild generalisation of ordinary Petri nets introduced in [3, 4] to answer the first of the requirements above, i.e., the possibility of interacting with the environment and of composing a larger net out of smaller open components. An open net is an ordinary net with a distinguished set of places, designated as open, through which the net can interact with the surrounding environment. As a consequence of such interaction, tokens can be freely generated and removed in open places. In the mentioned papers open nets are endowed with a composition operation, characterised as a pushout in the corresponding category, suitable to model both interaction through open places and synchronisation of transitions.

In the first part of the paper, after having extended the existing theory for open nets to deal with *marked* nets, we introduce bisimulation-based observational equivalences for open nets. Following a common intuition about reactive systems (see, e.g., [43, 29] or the recent [20]) such equivalences are based on the observation of the interactions between the given net and the surrounding environment. The framework treats uniformly *strong bisimilarity*, where every transition firing is observed, and *weak bisimilarity*, where a subset of unobservable transition labels is fixed (corresponding to τ-actions in process calculi) and the firings of transitions carrying such labels are considered invisible. We also consider *step bisimilarity* (see, e.g., [44, 30]), obtained by taking as observations possibly parallel steps

rather than single firings of transitions, thus capturing, to some extent, the concurrency properties of the system.

The considered notions of bisimilarity are shown to be congruences with respect to the composition operation over open nets. Interestingly enough, this holds also when the set of non-observable labels is not empty, i.e., for weak bisimilarities: some natural questions regarding the relation with weak bisimilarity in CCS are addressed. In addition, we propose an up-to technique for facilitating bisimilarity proofs.

Exploiting the results in the first part of the paper we next introduce a framework for open net reconfigurations. The fact that open net components are combined by means of pushouts naturally suggests a setting for specifying net reconfigurations, based on double-pushout (DPO) rewriting [14]. Using the congruence result for bisimilarity we identify classes of transformation rules which ensure that reconfigurations of the system do not affect its observational behaviour.

In order to understand this paper some basic knowledge of category theory (see for instance [32]) is required.

1. Marked Open Nets

An *open net*, as introduced in [3, 4], is an ordinary P/T Petri net with a distinguished set of *open places*, which represent the interface through which the environment can interact with the net. An open place can be an *input place*, meaning that the environment can put tokens into it, or an *output place*, from which the environment can remove tokens, or both. In this section we introduce the basic notions for open nets as presented in [4], generalising them to nets with initial marking: this will be needed in the treatment of bisimilarity in Section 4.

Given a set X we write $\mathbf{2}^X$ for the powerset of X and $X^\oplus$ for the free commutative monoid over X, with monoid operation $\oplus$, whose elements will be referred as *multisets* over X. Moreover, given a function $h : X \to Y$ we denote by the same symbol $h : \mathbf{2}^X \to \mathbf{2}^Y$ its extension to sets, and by $h^\oplus : X^\oplus \to Y^\oplus$ its monoidal extension. Given a multiset $u \in X^\oplus$, with $u = \bigoplus_{x\in X} u_x \cdot x$, for $x \in X$ we will write $u(x)$ to denote the coefficient u_x. With little abuse of notation, we will write $x \in u$ iff $u(x) \geq 1$. Given $u, v \in X^\oplus$ we write $u \leq v$ when $u(x) \leq v(x)$ for any $x \in X$. In this case the *multiset difference* $v \ominus u$ is the multiset w such that $u \oplus w = v$. The symbol 0 denotes the empty multiset.

Definition 1.1 (multiset projection)**.** Given a function $f : X \to Y$ and a multiset $u \in Y^\oplus$ we denote by $(u\downarrow f) \in X^\oplus$ the *projection of u along f*, which is the multiset over X defined as $(u\downarrow f) = \bigoplus_{x\in X} u_{f(x)} \cdot x$.

In other words, $(\_\downarrow f) : Y^\oplus \to X^\oplus$ is the monoidal extension of the function $(\_\downarrow f) : Y \to X^\oplus$ defined by $(y \downarrow f) = x_1 \oplus \ldots \oplus x_n$ when $f^{-1}(y) = \{x_1, \ldots, x_n\}$. For instance, given $f : \{s_0, s_1, s_2\} \to \{s'_1, s'_2, s'_3\}$ such that $f(s_0) = f(s_1) = s'_1$ and $f(s_2) = s'_2$, we have $(2s'_1 \oplus s'_2 \oplus s'_3 \downarrow f) = 2s_0 \oplus 2s_1 \oplus s_2$. In the following we will mainly work with injective functions, for which the projection operation satisfies some expected properties, such as $f^\oplus((u\downarrow f)) \leq u$ and $(f^\oplus((u\downarrow f))\downarrow f) = (u\downarrow f)$.

We consider nets where transitions are labelled over a fixed set of labels Λ.

Definition 1.2 (P/T Petri net)**.** A *P/T Petri net* is a tuple $N = (S, T, \sigma, \tau, \lambda)$ where S is the set of places, T is the set of transitions (with $S \cap T = \emptyset$), $\sigma, \tau : T \to S^\oplus$ are functions

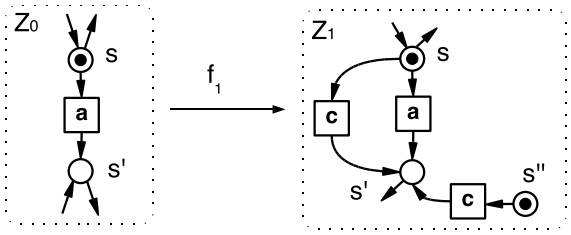


Figure 1: Two open nets and an open net morphism.

mapping each transition to its pre- and post-set and $\lambda : T \to \Lambda$ is a labelling function for transitions.

In the sequel we will denote by ${}^\bullet(\cdot)$ and $(\cdot)^\bullet$ the monoidal extensions of the functions σ and τ to functions from $T^\oplus$ to $S^\oplus$. Moreover, given $s \in S$, the pre- and post-set of s are defined by ${}^\bullet s = \{t \in T : s \in t^\bullet\}$ and $s^\bullet = \{t \in T : s \in {}^\bullet t\}$.

Definition 1.3 (Petri net category)**.** Let N_0 and N_1 be Petri nets. A *Petri net morphism* $f : N_0 \to N_1$ is a pair of total functions $f = \langle f_T, f_S \rangle$ with $f_T : T_0 \to T_1$ and $f_S : S_0 \to S_1$, such that for all $t_0 \in T_0$, ${}^\bullet f_T(t_0) = f_S^\oplus({}^\bullet t_0)$, $f_T(t_0)^\bullet = f_S^\oplus(t_0{}^\bullet)$ and $\lambda_1(f_T(t_0)) = \lambda_0(t_0)$. The category of P/T Petri nets and Petri net morphisms is denoted by **Net**.

It is worth recalling that category **Net** is a subcategory of the category **Petri** of [24], which has the same objects, but more general morphisms which can map a place to a multiset of places.

We next introduce the notion of open net. As anticipated above, differently from [3, 4], we work here with marked nets.

Definition 1.4 (open net)**.** An *open net* is a pair $Z = (N_Z, O_Z)$, consisting of a P/T Petri net $N_Z = (S_Z, T_Z, \sigma_Z, \tau_Z, \lambda_Z)$ and a pair $O_Z = (O_Z^+, O_Z^-) \in \mathbf{2}^{S_Z} \times \mathbf{2}^{S_Z}$, the sets of *input open*, respectively, *output open places* of the net. A *marked* open net is a pair $(Z, \hat{u})$ where Z is an open net and $\hat{u} \in S_Z^\oplus$ is the initial marking.

Hereafter, unless stated otherwise, all open nets will be assumed implicitly to be marked. An open net will be denoted simply by Z and the corresponding initial marking by $\hat{u}$. Subscripts carry over to the net components. The graphical representation for open nets is similar to that for standard nets. In addition, the fact that a place is input or output open is represented by an ingoing or outgoing dangling arc, respectively. For instance, in net Z_1 of Fig. 1, place s is both input and output open, while s' is only output open.

The notion of enabledness for transitions is the usual one, but besides the changes produced by the firing of the transitions of the net, we consider also the interaction with the environment which is modelled by events, denoted by $+_s$ or $-_s$, which produce or consume a token in an open place s. Such events corresponds to the pseudo-transitions of [43] and to the transition in the universal context of [29].

Definition 1.5 (set of extended events)**.** Let Z be an open net. The *set of extended events* of Z, denoted by $\bar{T}_Z$ and ranged over by ϵ is defined as

$$\bar{T}_Z = T_Z \cup \{+_s : s \in O_Z^+\} \cup \{-_s : s \in O_Z^-\}.$$

Defining ${}^\bullet +_s = 0$ and $+_s{}^\bullet = s$, and symmetrically, ${}^\bullet -_s = s$ and $-_s{}^\bullet = 0$, the notion of pre- and post-set extends to multisets of extended events.

Given a marking $u \in O_Z^{+\oplus}$, we denote by $+_u$ the multiset $\bigoplus_{s \in O_Z^+} u(s) \cdot +_s$. Similarly, $-_u = \bigoplus_{s \in O_Z^-} u(s) \cdot -_s$ for $u \in O_Z^{-\oplus}$.

Definition 1.6 (firings and steps)**.** Let Z be an open net. A *step* in Z consists of the execution of a multiset of (extended) events $A \in \bar{T}_Z^{\oplus}$, i.e.,

$$u \oplus {}^\bullet A \; [A\rangle \; u \oplus A^\bullet.$$

A step is called a *firing* when A consists of a single event, i.e., $A = \epsilon \in \bar{T}_Z$.

A firing can be (i) the execution of a transition $u \oplus {}^\bullet t \; [t\rangle \; u \oplus t^\bullet$, with $u \in S_Z^{\oplus}$, $t \in T_Z$; (ii) the creation of a token by the environment $u \; [+_s\rangle \; u \oplus s$, with $u \in S_Z^{\oplus}$, $s \in O_Z^+$; (iii) the deletion of a token by the environment $u \oplus s \; [-_s\rangle \; u$, with $u \in S_Z^{\oplus}$, $s \in O_Z^-$. A step is the execution of a multiset of transitions and interactions with the environment, of the kind $A \oplus -_w \oplus +_v$ for $A \in T_Z^{\oplus}, w \in O_Z^{-\oplus}$ and $v \in O_Z^{+\oplus}$.

We now introduce suitable morphisms relating open nets, which are morphisms between the underlying P/T nets, satisfying certain conditions on the open places and on the initial marking. In particular, given an injective morphism $f : Z_1 \to Z_2$, we can think of N_{Z_1} as a subnet of N_{Z_2}. In this case, we require that a place of Z_1 is input/output open in Z_2 only if it is so in Z_1, and that a transition in $T_{Z_2} - T_{Z_1}$ can put/remove a token on/from a place of Z_1 only if that place is input/output open in Z_1. Furthermore, any place of Z_1 must have the same number of tokens of its image in Z_2. This is formalized by the following definition, which introduces general morphisms, possibly non-injective.

Definition 1.7 (open net category)**.** An *open net morphism* $f : Z_1 \to Z_2$ is a Petri net morphism $f : N_{Z_1} \to N_{Z_2}$ such that, if we define $\mathsf{in}(f) = \{s \in S_{Z_1} : {}^\bullet f_S(s) - f_T({}^\bullet s) \neq \emptyset\}$ and $\mathsf{out}(f) = \{s \in S_{Z_1} : f_S(s)^\bullet - f_T(s^\bullet) \neq \emptyset\}$, then

(1) (i) $f_S^{-1}(O_{Z_2}^+) \cup \mathsf{in}(f) \subseteq O_{Z_1}^+$ and (ii) $f_S^{-1}(O_{Z_2}^-) \cup \mathsf{out}(f) \subseteq O_{Z_1}^-$.
(2) $\hat{u}_1 = (\hat{u}_2 \downarrow f_S)$ (reflection of initial marking).

The morphism f is called an *open net embedding* if both f_T and f_S are injective. We will denote by **ONet** the category of open nets and open net morphisms.

Conceptually, condition 1 formalizes the intuition that each open net can interact with the environment only through open places. In fact, given an embedding $f : Z_1 \to Z_2$, if s is a place of Z_1 which is open in Z_2, then an interaction of the environment with Z_2 through s would also affect Z_1: therefore s must be open in Z_1 as well. That is, input/output open places must be reflected by the embedding, as stated by the first part of conditions 1.(i) and 1.(ii). Furthermore, if a transition in $T_{Z_2} - T_{Z_1}$ can put a token in a place s of Z_1, this is seen from Z_1 as an interaction with the environment, and therefore s must be (input) open in Z_1: this is formalized by the second part of conditions 1.(i) and 1.(ii). Finally, condition 2 requires the marking of Z_1 to be the projection of the marking of Z_2: any place $s_1 \in S_{Z_1}$ must carry the same number of tokens as its image $f(s_1) \in S_{Z_2}$, i.e., $\hat{u}_1(s_1) = \hat{u}_2(f(s_1))$ for any $s_1 \in S_{Z_1}$.

Consider, for instance, morphism $f_1 : Z_0 \to Z_1$ in Fig. 1: the mapping of places and transitions is suggested by the shape and labelling of the nets. Note that in Z_1 a "new" c-labelled transition is attached to the places s and s'. This is legal since the corresponding places in Z_0 are output open and input open, respectively. Note also that the number of tokens in places in Z_0 and in their image through f_1 is the same. Instead, the number of tokens in the place s'' in Z_1 is not constrained since it is not in the image of f_1: the place is marked, but f_1 would have been a legal morphism also if s'' were not marked.

It is worth observing that most of the constructions in the paper will be defined for open net embeddings, hence readers can limit their attention to embeddings if this helps the intuition. Still, on the formal side, working in a larger host category with more general morphisms is essential to obtain a characterisation of the composition operation in terms of pushouts. Specifically, non-injective open net morphisms are needed as mediating morphisms (recall, for example, that the category of sets with injective functions does not have all pushouts).

Observe that the constraints characterising open nets morphisms have an intuitive graphical interpretation:

- The connections of transitions to their pre-set and post-set have to be preserved. New connections cannot be added.
- In the larger net, a new arc may be attached to a place only if the corresponding place of the subnet has a dangling arc in the same direction. Dangling arcs may be removed, but cannot be added in the larger net.
- The number of tokens in each place in the source net must be preserved in the target. Instead, there are no restrictions on the marking of places of the target net which are not in the image of the source net.

In the sequel, given an open net morphism $f = \langle f_S, f_T \rangle : Z_1 \to Z_2$, to lighten the notation we will omit the subscripts "S" and "T" in its place and transition components, writing $f(s)$ for $f_S(s)$ and $f(t)$ for $f_T(t)$. Moreover we will write $f^{\oplus} : \bar{T}^{\oplus}_{Z_1} \to \bar{T}^{\oplus}_{Z_2}$ to denote the monoidal function defined on the generators by $f^{\oplus}(t) = f(t)$ for $t \in T_{Z_1}$ and, for $x \in \{+, -\}$, $f^{\oplus}(x_s) = x_{f(s)}$, if $f(s) \in O^x_{Z_2}$ and $f^{\oplus}(x_s)$ undefined, otherwise. Note that $f^{\oplus}$ can be partial since open places can be mapped to closed places.

The next proposition explicitly shows that category **ONet**, as introduced in Definition 1.7, is well defined. To prove this fact we will use the well-definedness of the category of unmarked open nets, introduced in [4]. This category, denoted here by $\mathbf{ONet}^u$, has (unmarked) open nets as objects and mappings satisfying only condition 1 in Definition 1.7 as morphisms. These will be referred to as *unmarked open net morphisms*.

Proposition 1.8. *Open net morphisms are closed under composition.*

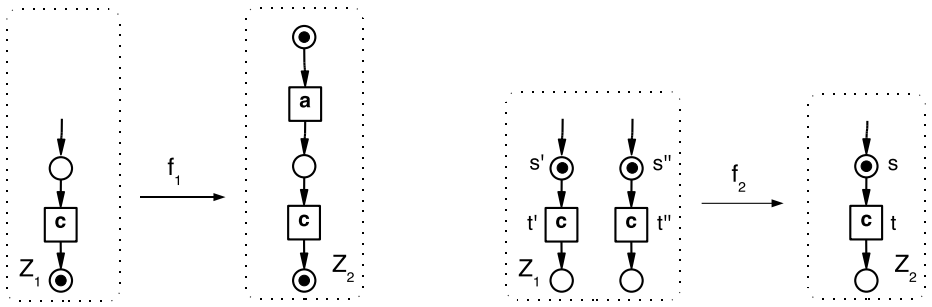
Proof. Let $f_1 : Z_1 \to Z_2$ and $f_2 : Z_2 \to Z_3$ be open net morphisms. Then f_1 and f_2 are unmarked open net morphisms and thus, since $\mathbf{ONet}^u$ is a well-defined category, also $f_2 \circ f_1$ is an unmarked open net morphism. In order to prove that $f_2 \circ f_1$ is a well defined open net morphism it remains to show that it satisfies also condition 2 in Definition 1.7, i.e., that it reflects the initial marking. But this fact follows easily from the definition. In fact, for any $s_1 \in S_{Z_1}$,

$$\begin{aligned} &\hat{u}_3(f_2(f_1(s_1))) = \\ &\quad = \hat{u}_2(f_1(s_1)) \quad \text{[since } f_2 \text{ is an open net morphism]} \\ &\quad = \hat{u}_1(s_1) \quad \text{[since } f_1 \text{ is an open net morphism]} \end{aligned}$$

□

Unlike most of the morphisms considered over Petri nets in the literature, open net morphisms are *not* simulations. As an example, consider the open net embedding in Fig. 2(a). While the transition labelled c in the net Z_1 can fire infinitely many times, its image in the second net Z_2 can fire only once.

Instead, since open net embeddings are designed to capture the idea of inserting a net into a larger one, they are expected to reflect the behaviour, in the sense that given an embedding $f : Z_0 \to Z_1$, the behaviour of Z_1 can be projected along f to the behaviour



(a) (b)

Figure 2: (a) Open net morphisms are not simulations and (b) an example of non-injective open net morphism.

of Z_0. The target net of a morphism is in general more "instantiated" and thus more constrained than the source net (e.g., a place which is open in the source net can be closed in the target). We will come back to this fact in the conclusions.

Although the paper will mainly use open net embeddings, a remark about non-injective morphisms is in order. Consider the open net morphism f_2 in Fig. 2(b), where $f_2(t') = f_2(t'') = t$ and $f_2(s') = f_2(s'') = s$. As, intuitively, the two transitions of Z_1 become the same transition in Z_2, in this case by reflection of behaviour we mean that the firing of t in Z_2 must be reflected to the parallel firing of t' and t'' in Z_1. Note that this is the case, e.g., for the initial markings: s enables t and its projection $(s \downarrow f_2) = s' \oplus s''$ enables $t' \oplus t''$.

In the rest of this section we formalize the intuition that an open net embedding $f : Z \to Z'$ reflects the behaviour by showing that each step of Z' can be projected along f to a step of Z. It could be shown that the behaviour of an open net is reflected along non-injective morphisms as well, but this would require some technical complications which we prefer to avoid, as it will not be used in the rest of the paper.

We start by defining the projection of multisets of extended events along open net embeddings.

Definition 1.9 (projecting extended events)**.** Given an open net embedding $f : Z \to Z'$, the *projection of extended events along* f, denoted $(\_ \Downarrow f) : \bar{T}_{Z'} \to \bar{T}_Z^{\oplus}$, is defined as follows. For each $\epsilon' \in \bar{T}_{Z'}$,

- if $\epsilon' = t' \in T_{Z'}$ is a transition, then

$$(t' \Downarrow f) = \begin{cases} t & \text{if } t \in T_Z \text{ and } f(t) = t' \\ -_{(\ {}^\bullet t' \downarrow f)} \oplus +_{(t' {}^\bullet \downarrow f)} & \text{if } t' \notin f(T_Z) \end{cases}$$

- if $\epsilon' = x_{s'}$, with $x \in \{+, -\}$, then $(x_{s'} \Downarrow f) = x_{(s' \downarrow f)}$.

The monoidal extension of $(\_ \Downarrow f)$ to *multisets* of extended events will be denoted by the same symbol $(\_ \Downarrow f) : \bar{T}_{Z'}^{\oplus} \to \bar{T}_Z^{\oplus}$.

In words, if we think of the embedding $f : Z \to Z'$ as an inclusion, then given a transition t', the projection $(t' \Downarrow f)$ is the transition itself if t' is in Z. Otherwise, if t' is not in Z but it consumes or produces tokens in places of Z, the projection of t' contains

the corresponding extended events, expressing the interactions over open places. Similarly, the projection of an extended event $+_{s'}$ is the event itself if s' is in Z, and it is the empty multiset otherwise: in fact, in this case $(s' \downarrow f) = 0$.

It is easily checked that the projection operation is well-defined, in the sense that, e.g., if $+_s \in (\epsilon \Downarrow f)$ then $s \in O_Z^+$. In fact, if $+_s \in (t' \Downarrow f)$ then $s \in \mathsf{in}(f)$, while if $+_s \in (+_{s'} \Downarrow f)$, then $s' \in O_{Z'}^+$ and $f(s) = s'$. In both cases $s \in O_Z^+$ by condition 1.(i) of Definition 1.7.

The projections of multisets of places and extended events enjoy nice properties which are summarized by the next lemma.

Lemma 1.10 (properties of projection)**.** *Let $f : Z \to Z'$ be an open net embedding. Then*

(1) *for $u_1, u_2 \in S^{\oplus}_{Z'}$ we have*

$$((u_1 \oplus u_2) \downarrow f) = (u_1 \downarrow f) \oplus (u_2 \downarrow f) \quad \textit{and} \quad (0 \downarrow f) = 0$$

and for $u \in S^{\oplus}_{Z}$

$$(f^{\oplus}(u) \downarrow f) = u$$

(2) *for $x_1, x_2 \in \bar{T}^{\oplus}_{Z'}$ we have*

$$((x_1 \oplus x_2) \Downarrow f) = (x_1 \Downarrow f) \oplus (x_2 \Downarrow f) \quad \textit{and} \quad (0 \Downarrow f) = 0$$

and for $x \in \bar{T}^{\oplus}_{Z}$, if $f^{\oplus}(x)$ is defined we have

$$(f^{\oplus}(x) \Downarrow f) = x$$

(3) *given $A' \in \bar{T}^{\oplus}_{Z'}$*

$$({}^{\bullet}A' \downarrow f) = {}^{\bullet}(A' \Downarrow f) \quad \textit{and} \quad (A'^{\bullet} \downarrow f) = (A' \Downarrow f)^{\bullet}$$

(4) *for $u \in S^{\oplus}_{Z'}$ we have*

$$f^{\oplus}((u \downarrow f)) \leq u$$

Proof. Proofs are routine. We prove explicitly only the third point. Since ${}^{\bullet}(\cdot)$ and $(\cdot)^{\bullet}$ are monoidal functions it is sufficient to prove the result only on the generators. We concentrate on ${}^{\bullet}(\cdot)$, since the proof for $(\cdot)^{\bullet}$ is completely analogous.

We distinguish various cases:

- $A' = t' \in T_{Z'}$
  If there exists $t \in T_Z$ such that $f(t) = t'$, then $(t' \Downarrow f) = t$. Since f is an open net morphism $f^{\oplus}({}^{\bullet}t) = {}^{\bullet}t'$ and thus, as desired
  $${}^{\bullet}(t' \Downarrow f) = {}^{\bullet}t = (f^{\oplus}({}^{\bullet}t) \downarrow f) = ({}^{\bullet}t' \downarrow f)$$
  where the second equality is justified by point (1).
  If, instead, $t' \notin f(T_Z)$ we have that $(t' \Downarrow f) = -_{({}^{\bullet}t' \downarrow f)} \oplus +_{(t'^{\bullet} \downarrow f)}$. Hence, in this case the result is obvious since
  $${}^{\bullet}(t' \Downarrow f) = {}^{\bullet}(-_{({}^{\bullet}t' \downarrow f)} \oplus +_{(t'^{\bullet} \downarrow f)}) = ({}^{\bullet}t' \downarrow f)$$
- $A' = +_{s'}$ or $A' = -_{s'}$
  Suppose, e.g., that $A' = -_{s'}$. In this case $(A' \Downarrow f) = -_{(s' \downarrow f)}$ and the result trivially holds.

□

We are now ready to present the main result of this section.

Lemma 1.11 (reflection of behaviour)**.** *Let $f : Z \to Z'$ be an open net embedding. For every step $u'\ [A'\rangle\ v'$ in Z' there is a step $(u' \downarrow f)\ [(A' \Downarrow f)\rangle\ (v' \downarrow f)$ in Z, called the* projection of $u'\ [A'\rangle\ v'$ along f*.*

Proof. Let $f : Z \to Z'$ be an open net embedding and assume that $u'\ [A'\rangle\ v'$ is a step in Z'. Therefore

$$u' = u'' \oplus {}^\bullet A' \quad \text{and} \quad v' = u'' \oplus A^\bullet$$

Now, we have

$$\begin{aligned}(u' \downarrow f) &= \\ &= (u'' \downarrow f) \oplus ({}^\bullet A' \downarrow f) && \text{[by Lemma 1.10.(1)]} \\ &= (u'' \downarrow f) \oplus {}^\bullet(A' \Downarrow f) && \text{[by Lemma 1.10.(3)]}\end{aligned}$$

and similarly

$$(v' \downarrow f) = (u'' \downarrow f) \oplus (A' \Downarrow f)^\bullet$$

Therefore, as desired, there is the step

$$(u' \downarrow f) = (u'' \downarrow f) \oplus {}^\bullet(A' \Downarrow f)\ [(A' \Downarrow f)\rangle\ (u'' \downarrow f) \oplus (A' \Downarrow f)^\bullet = (v' \downarrow f).$$

□

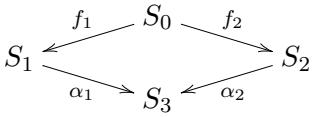
Observe that there is an obvious forgetful functor $\mathcal{F} : \mathbf{ONet} \to \mathbf{Net}$, defined by $\mathcal{F}(Z) = N_Z$ and $\mathcal{F}(f : Z_0 \to Z_1) = f : N_{Z_0} \to N_{Z_1}$. Since functor $\mathcal{F}$ acts on arrows as the identity, with abuse of notation, given an open net morphism $f : Z_0 \to Z_1$ we will often write $f : \mathcal{F}(Z_1) \to \mathcal{F}(Z_2)$ instead of $\mathcal{F}(f) : \mathcal{F}(Z_1) \to \mathcal{F}(Z_2)$.

2. Composing Open Nets

We introduce next a basic mechanism for composing open nets which is characterised as a pushout construction in category **ONet**. A pushout is a canonical way of describing a gluing construction. The case of unmarked nets was already discussed in [4]. Here we extend the theory to deal with marked open nets. This will allow later to define reconfigurations of open nets, where the applicability of a reconfiguration rule can depend on the marking. Intuitively, two open nets Z_1 and Z_2 are composed by specifying a common subnet Z_0, and then by joining the two nets along Z_0.

Let us start with a technical definition which will be useful below.

Proposition 2.1 (composition of multisets)**.** *Consider a pushout diagram in the category of sets as below, where all morphisms are injective.*
Given $u_1 \in S_1^\oplus$ and $u_2 \in S_2^\oplus$ such that $(u_1 \downarrow f_1) = (u_2 \downarrow f_2) = u_0$, there is a (unique) multiset $u_3 \in S_3^\oplus$ such that $(u_3 \downarrow \alpha_i) = u_i$, for $i \in \{1, 2\}$. Such a multiset u_3 will be denoted by $u_3 = u_1 \uplus_{u_0} u_2$ or simply by $u_1 \uplus u_2$ when making u_0 explicit is not needed.



Additionally, if $u_3 = u_1 \uplus_{u_0} u_2$ and $u_3' = u_1' \uplus_{u_0'} u_2'$, then $u_3 \oplus u_3' = (u_1 \oplus u_1') \uplus_{(u_0 \oplus u_0')} (u_2 \oplus u_2')$.

Proof. Define $u_3 \in S_3^\oplus$ as follows: for each $s \in S_3$,

$$u_3(s) = \begin{cases} u_1(s_1) & \text{if } \exists s_1 \in S_1 \text{ such that } \alpha_1(s_1) = s \\ u_2(s_2) & \text{if } \exists s_2 \in S_2 \text{ such that } \alpha_2(s_2) = s \end{cases}$$

Let us start checking that u_3 is well-defined. In fact, firstly, the definition assigns a coefficient to every $s \in S_3$ because α_1 and α_2 are jointly surjective. Secondly, if there are $s_1 \in S_1$ and $s_2 \in S_2$ such that $\alpha_1(s_1) = \alpha_2(s_2)$, since the square is a pushout and all functions are injective we have ${f_1}^{-1}(s_1) = \{s_0\}$ and ${f_2}^{-1}(s_2) = \{s_0\}$ for some $s_0 \in S_0$: thus, since $(u_1 \downarrow f_1) = (u_2 \downarrow f_2)$ by hypothesis, we obtain $u_1(s_1) = u_1(f_1(s_0)) = (u_1 \downarrow f_1)(s_0) = (u_2 \downarrow f_2)(s_0) = u_2(f_2(s_0)) = u_2(s_2)$.

Now, in order to prove (for $i \in \{1,2\}$) that $(u_3 \downarrow \alpha_i) = u_i$, notice that, since α_i is injective, this amounts to show that for any $s \in S_i$ we have $u_i(s) = u_3(\alpha_i(s))$, which is immediate by the definition of u_3.

Concerning the second part of the statement, let $u_3 = u_1 \uplus_{u_0} u_2$ and $u'_3 = u'_1 \uplus_{u'_0} u'_2$. Then just observe that by Lemma 1.10.(1), we have for $i \in \{1,2\}$

$$((u_3 \oplus u'_3)\downarrow \alpha_i) = (u_3 \downarrow \alpha_i) \oplus (u'_3 \downarrow \alpha_i) = u_i \oplus u'_i$$

hence the result $u_3 \oplus u'_3 = (u_1 \oplus u'_1) \uplus_{(u_0 \oplus u'_0)} (u_2 \oplus u'_2)$ follows by the defining property of the composition of markings. □

Intuitively, the multiset $u_1 \uplus_{u_0} u_2$ can be seen as the "least upper bound" of the images of the two multisets in $S_3^{\oplus}$.

As in [3, 4], two embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ are called *composable* if the places which are used as interface by f_1, i.e., the places $\mathsf{in}(f_1)$ and $\mathsf{out}(f_1)$, are mapped by f_2 to input and output open places of Z_2, respectively, and also the symmetric condition holds.

Definition 2.2 (composability of embeddings)**.** Let $f_1 : Z_0 \to Z_1$, $f_2 : Z_0 \to Z_2$ be embeddings in **ONet** (see Fig. 3).We say that f_1 and f_2 are *composable* if

(1) $f_2(\mathsf{in}(f_1)) \subseteq O^+_{Z_2}$ and $f_2(\mathsf{out}(f_1)) \subseteq O^-_{Z_2}$;
(2) $f_1(\mathsf{in}(f_2)) \subseteq O^+_{Z_1}$ and $f_1(\mathsf{out}(f_2)) \subseteq O^-_{Z_1}$.

Composability is necessary and sufficient to ensure that the pushout of f_1 and f_2 can be computed in **Net** and then lifted to **ONet**.

Proposition 2.3 (pushouts in **ONet**)**.** *Let $f_1 : Z_0 \to Z_1$, $f_2 : Z_0 \to Z_2$ be embeddings in* **ONet** *(see Fig. 3). Compute the pushout of the corresponding diagram in category* **Net** *obtaining net N_{Z_3} and morphisms α_1 and α_2,*[1] *and then take as open places, for $x \in \{+,-\}$,*

$$O^x_{Z_3} = \{s_3 \in S_{Z_3} : \alpha_1^{-1}(s_3) \subseteq O^x_{Z_1} \ \wedge \ \alpha_2^{-1}(s_3) \subseteq O^x_{Z_2}\}$$

and as initial marking $\hat{u}_3 = \hat{u}_1 \uplus_{\hat{u}_0} \hat{u}_2$, defined according to Proposition 2.1. Then $(\alpha_1, Z_3, \alpha_2)$ is the pushout in **ONet** *of f_1 and f_2 if and only if f_1 and f_2 are composable. In this case we write $Z_3 = Z_1 +_{f_1,f_2} Z_2$.*

Proof. We know by [4] (Proposition 6) that the above result holds for unmarked nets, i.e., in the category $\mathbf{ONet}^u$. Here we must additionally show that (i) the α_i are marked morphisms and that (ii) if we take any other net Z'_3, with $\alpha'_i : Z_i \to Z'_3$ making the diagram commute, then the mediating morphism $\gamma : Z_3 \to Z'_3$ (which exists uniquely as an unmarked net morphism by the result in [4]) respects the condition on the marking.

Now, (i) is immediate since Proposition 2.1 tells us that $(\hat{u}_3 \downarrow \alpha_i) = \hat{u}_i$ for $i \in \{1,2\}$. Property (ii) can be proved along the same lines. □

[1]The pushout in **Net** is computed componentwise on places and transitions, by defining the pre- and post-set functions, for any $t_i \in T_{Z_i}$, $i \in \{1,2\}$, as $\sigma_{Z_3}(\alpha_i(t_i)) = {\alpha_i}^{\oplus}(\sigma_{Z_i}(t_i))$ and $\tau_{Z_3}(\alpha_i(t_i)) = {\alpha_i}^{\oplus}(\tau_{Z_i}(t_i))$. It is routine to show that this definition is well given.

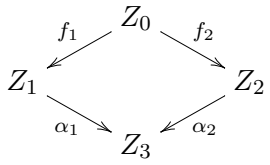


Figure 3: Pushout in **ONet**.

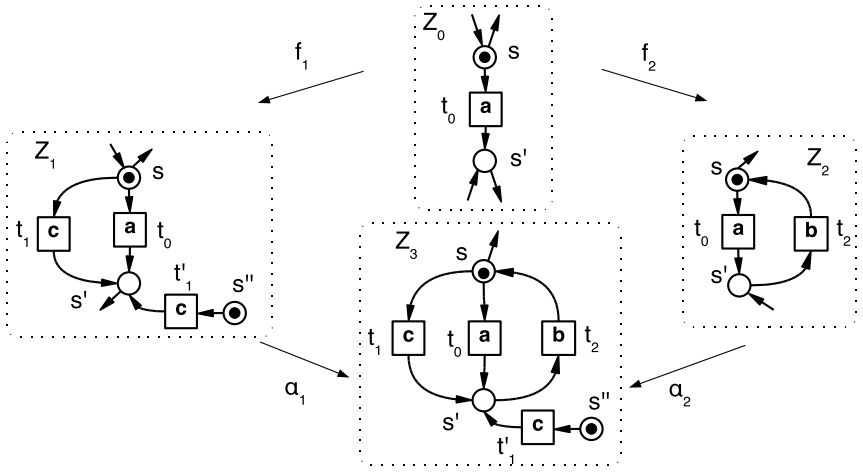


Figure 4: An example of a pushout in **ONet**.

As an example, the open net embeddings f_1 and f_2 in Fig. 4 are composable. In fact, $\mathsf{in}(f_1) = \{s'\}$, $\mathsf{out}(f_1) = \{s\}$ and $\mathsf{in}(f_2) = \{s\}$, $\mathsf{out}(f_2) = \{s'\}$, and thus it is easy to see that the conditions of Definition 2.2 are satisfied. The net Z_3 is the resulting pushout object.

3. Composing Steps

In this section we analyse the behaviour of an open net Z_3 arising as the composition of two nets Z_1 and Z_2 along an interface Z_0. More specifically, we show that steps of the component nets Z_1 and Z_2 can be "composed" to give a step of Z_3 when they agree on the interface and satisfy suitable compatibility conditions.

For instance, concerning the example pushout in Fig. 4, note that net Z_1 can fire the transition labelled a and the lower transition labelled c. If this is "mimicked" in Z_2 by firing a and putting a token into the lower place s' (via an interaction $+_{s'}$ with the environment), then such steps are compatible in a sense made precise below and can be combined into a step of the composed net Z_3.

We start with a technical lemma which will be pivotal in the paper. Assume that the first component makes a step and the second component imitates this step, acting only on the places of the common interface, without firing any internal transition. Then the two local steps can be combined to a step of the composed net.

Lemma 3.1. *Let $Z_3 = Z_1 +_{f_1,f_2} Z_2$ be the pushout of two composable embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ in* **ONet** *(see Fig. 3). Let $u_1\ [A_1\rangle\ v_1$ and $u_2\ [A_2\rangle\ v_2$ be steps in Z_1 and Z_2, respectively, such that $(u_1 \downarrow f_1) = (u_2 \downarrow f_2)$ and $A_2 = f_2^{\oplus}((A_1 \Downarrow f_1))$.*

Then $(v_1 \downarrow f_1) = (v_2 \downarrow f_2)$ and, if we define $A_3 = \alpha_1^{\oplus}(A_1)$,

$$u_1 \uplus u_2\ [A_3\rangle\ v_1 \uplus v_2.$$

Proof. Let us start showing that $A_3 = \alpha_1^{\oplus}(A_1)$ is defined, i.e., that for $x \in \{+, -\}$ if $x_s \in A_1$ then $\alpha_1(s) \in O^x_{Z_3}$. In fact $x_s \in A_1$ implies that $s \in O^x_{Z_1}$; now either $s \notin f_1(S_{Z_0})$ and then $\alpha_1(s) \in O^x_{Z_3}$ by Proposition 2.3. Otherwise, since f_1 is an embedding, there is exactly one place in S_{Z_0} which is mapped to s. With a little abuse of notation let such place be denoted $f_1^{-1}(s)$. Then clearly $f_1^{-1}(s) \in O^x_{Z_0}$ because f_1 is a morphism, and $f_2(f_1^{-1}(s)) \in O^x_{Z_2}$ because $f_2^{\oplus}((A_1 \Downarrow f_1)) = A_2$ is defined by hypothesis; thus again $\alpha_1(s) \in O^x_{Z_3}$ by Proposition 2.3.

Next observe that, since $A_2 = f_2^{\oplus}((A_1 \Downarrow f_1))$ is defined, by Lemma 1.10.(2),

$$(A_2 \Downarrow f_2) = (A_1 \Downarrow f_1).$$

Let $A_0 = (A_i \Downarrow f_i)$, for $i \in \{1, 2\}$, be the common projection. As a consequence, we have ${}^\bullet(A_2 \Downarrow f_2) = {}^\bullet(A_1 \Downarrow f_1)$ and thus, by Lemma 1.10.(3)

$$({}^\bullet A_1 \downarrow f_1) = ({}^\bullet A_2 \downarrow f_2)$$

so that we can consider the composition of markings ${}^\bullet A_1 \uplus_{{}^\bullet A_0} {}^\bullet A_2$. We claim that

$${}^\bullet A_3 = {}^\bullet A_1 \uplus_{{}^\bullet A_0} {}^\bullet A_2 \tag{3.1}$$

and symmetrically, since $(A_1{}^\bullet \downarrow f_1) = (A_2{}^\bullet \downarrow f_2)$, that

$$A_3{}^\bullet = A_1{}^\bullet \uplus_{A_0{}^\bullet} A_2{}^\bullet$$

Let us concentrate on ${}^\bullet(\cdot)$, as the other case is analogous. To prove (3.1), by Proposition 2.1 we can show that $({}^\bullet A_3 \downarrow \alpha_1) = {}^\bullet A_1$ and $({}^\bullet A_3 \downarrow \alpha_2) = {}^\bullet A_2$. In fact we have

$$\begin{array}{lll} ({}^\bullet A_3 \downarrow \alpha_1) = & & \\ \quad = {}^\bullet(A_3 \Downarrow \alpha_1) & \text{[by Lemma 1.10.(3)]} \\ \quad = {}^\bullet(\alpha_1^{\oplus}(A_1) \Downarrow \alpha_1) & \text{[by definition of } A_3\text{]} \\ \quad = {}^\bullet A_1 & \text{[by Lemma 1.10.(2)]} \end{array}$$

and

$$\begin{array}{lll} ({}^\bullet A_3 \downarrow \alpha_2) = & & \\ \quad = {}^\bullet(A_3 \Downarrow \alpha_2) & \text{[by Lemma 1.10.(3)]} \\ \quad = {}^\bullet(\alpha_1^{\oplus}(A_1) \Downarrow \alpha_2) & \text{[by definition of } A_3\text{]} \end{array}$$

Thus to conclude we must show that ${}^\bullet(\alpha_1^{\oplus}(A_1) \Downarrow \alpha_2) = {}^\bullet A_2$, and this is proved by showing

$$(\alpha_1^{\oplus}(A_1) \Downarrow \alpha_2) = f_2^{\oplus}((A_1 \Downarrow f_1))[= A_2] \tag{3.2}$$

Since $(\_ \Downarrow \_)$ is monoidal in the first argument by Lemma 1.10.(1), it is sufficient to show (3.2) on generators:

- $A_1 = t_1$

  We distinguish two subcases. If $(t_1 \Downarrow f_1) = t_0 \in T_{Z_0}$ then $A_2 = f_2(t_0) = (\alpha_1(t_1) \Downarrow \alpha_2)$, as desired, by construction of the pushout.

  If, instead, $(t_1 \Downarrow f_1) = -_{({}^\bullet t_1 \downarrow f_1)} \oplus +_{(t_1{}^\bullet \downarrow f_1)}$, then

  $A_2 = -_{f_2^{\oplus}(({}^\bullet t_1 \downarrow f_1))} \oplus +_{f_2^{\oplus}((t_1{}^\bullet \downarrow f_1))}$

  On the other hand, we have

$(\alpha_1^{\oplus}(A_1) \Downarrow \alpha_2) = (\alpha_1(t_1) \Downarrow \alpha_2) = -_{(\ {}^\bullet\alpha_1(t_1) \downarrow \alpha_2)} \oplus +_{(\alpha_1(t_1)^\bullet \downarrow \alpha_2)}$

Now, by exploiting the fact that Z_3 is a pushout, it is easy to see that $f_2^{\oplus}((\ {}^\bullet t_1 \downarrow f_1)) = (\ {}^\bullet\alpha_1(t_1) \downarrow \alpha_2)$ and similarly $f_2^{\oplus}((t_1{}^\bullet \downarrow f_1)) = (\alpha_1(t_1)^\bullet \downarrow \alpha_2)$. Hence we conclude that $A_2 = (\alpha_1^{\oplus}(A_1) \Downarrow \alpha_2)$, as desired.

- $A_1 = +_{s_1}$ or $A_1 = -_{s_1}$

  Assume, for instance, that $A_1 = +_{s_1}$ (the other case is completely analogous). Therefore

  $A_2 = f_2^{\oplus}((A_1 \Downarrow f_1)) = +_{f_2^{\oplus}((s_1 \downarrow f_1))}$

  On the other hand

  $(\alpha_1^{\oplus}(A_1) \Downarrow \alpha_2) = (+_{\alpha_1(s_1)} \Downarrow \alpha_2) = +_{(\alpha_1(s_1) \downarrow \alpha_2)}$

  and, again, by the fact that Z_3 is a pushout, we deduce easily that $f_2^{\oplus}((s_1 \downarrow f_1)) = (\alpha_1(s_1) \downarrow \alpha_2)$, hence the desired equality.

This concludes the proof of (3.2), from which (3.1) follows.

Now, by exploiting (3.1) we can easily conclude. In fact, the steps in Z_1 and Z_2 are of the kind

$u_i = u_i' \oplus {}^\bullet A_i \ [A_i\rangle \ u_i' \oplus A_i{}^\bullet = v_i$

for $i \in \{1, 2\}$. First observe that, since $(u_1 \downarrow f_1) = (u_2 \downarrow f_2)$ and $(\ {}^\bullet A_1 \downarrow f_1) = (\ {}^\bullet A_2 \downarrow f_2)$, we immediately get:

$(u_1' \downarrow f_1) = (u_2' \downarrow f_2)$

Let $u_0' = (u_i' \downarrow f_i)$, for $i \in \{1, 2\}$, be the common projection. Since $v_i = u_i' \oplus A_i{}^\bullet$, for $i \in \{1, 2\}$, by the fact that $(A_1{}^\bullet \downarrow f_1) = (A_2{}^\bullet \downarrow f_2)$, we deduce that, as desired

$(v_1 \downarrow f_1) = (v_2 \downarrow f_2)$

Hence, if $v_0 = (v_i \downarrow f_i)$ is the common projection, we can define $v_3 = v_1 \uplus_{v_0} v_2$.

Now, if we set $u_3' = u_1' \uplus_{u_0'} u_2'$ we have

$$\begin{aligned} u_3 &= u_1 \uplus_{u_0} u_2 = \\ &= (u_1' \oplus {}^\bullet A_1) \uplus_{u_0' \oplus {}^\bullet A_0} (u_2' \oplus {}^\bullet A_2) = \\ &= (u_1' \uplus_{u_0'} u_2') \oplus (\ {}^\bullet A_1 \uplus {}^\bullet A_0 \ {}^\bullet A_2) = && \text{[by Proposition 2.1]} \\ &= u_3' \oplus {}^\bullet A_3 && \text{[by (3.1)]} \end{aligned}$$

Therefore we have the step

$u_3 \ [A_3\rangle \ u_3' \oplus A_3{}^\bullet$.

By a sequence of passages analogous to those used above, we can show that $u_3' \oplus A_3{}^\bullet = v_3$ and thus, as desired, $u_3 \ [A_3\rangle \ v_3$.

The fact that such step projects to $u_i \ [A_i\rangle \ v_i$ for $i \in \{1, 2\}$ immediately follows by construction. □

We are now able to show how steps of the component nets can be "joined" to a step of their composition, provided that the steps satisfy a suitable compatibility condition, that we are going to introduce. Roughly, we must be able to split each of the two steps A_1, A_2 into an internal part A_i^I and an external part A_i^E, with the intuition that the external part can include only firings of transitions in the interface and interactions with the environment induced by the internal part of the other step.

Put more precisely, from the point of view of Z_1 the events can be of four different kinds: (1) transitions that are local to Z_1 (2) transitions that occur also in Z_0 (3) interactions with Z_2 (of the form $+_s, -_s$) (4) interactions with the environment of both nets (also of the form $+_s, -_s$). Now if one splits the set A_1 into A_1^I and A_1^E, it is necessary to put all events of

type (1) into A_1^I and all events of type (3) into A_1^E. For the remaining two types we have a choice, but whenever we put an event of Z_1 into A_1^E, we have to put the corresponding event in Z_2 into A_2^I (and vice versa).

For reasons of simplicity we have chosen to work with a split into only two sets instead of four, even if this split is non-unique.

Definition 3.2 (compatible steps)**.** Let $Z_3 = Z_1 +_{f_1,f_2} Z_2$ be a pushout in **ONet**. We say that two steps $u_i\ [A_i\rangle\ v_i$ $(i \in \{1,2\})$ are *compatible* if $(u_1 \downarrow f_1) = (u_2 \downarrow f_2)$ and we can decompose the steps as $A_i = A_i^I \oplus A_i^E$ $(i \in \{1,2\})$ such that

$$A_2^E = f_2^{\oplus}((A_1^I \Downarrow f_1)) \quad \text{and} \quad A_1^E = f_1^{\oplus}((A_2^I \Downarrow f_2))$$

It is immediate to see that if A_1 and A_2 are compatible, then $(A_1 \Downarrow f_1) = (A_2 \Downarrow f_2)$.

For instance, let us consider again the pushout in Fig. 4. Two compatible steps can be $A_1 = t_0 \oplus t_1'$ and $A_2 = t_0 \oplus +_{s'}$. The compatibility is witnessed by the decomposition $A_1^I = A_1$, $A_1^E = 0$ and $A_2^I = 0$, $A_2^E = A_2$. As mentioned above such decompositions are not uniquely determined: alternative ones are given by $A_1^I = t_1'$, $A_1^E = t_0$ and $A_2^I = t_0$, $A_2^E = +_{s'}$. Note that since transition t_0 also belongs to the interface, it can be considered either internal to Z_1 or internal to Z_2, while t_1' has to be considered internal to Z_1, and the interaction $+_{s'}$ on the open place s' has to be considered external to Z_2.

Another simple example of compatible steps is given by $A_1 = -_s$ and $A_2 = -_s$. In this case, we have the choice to consider the only event $-_s$ internal to Z_1 and external to Z_2 or vice versa.

Lemma 3.3 (composing steps)**.** *Let $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ be composable embeddings in* **ONet** *and let $Z_3 = Z_1 +_{f_1,f_2} Z_2$. Let $u_1\ [A_1\rangle\ v_1$ and $u_2\ [A_2\rangle\ v_2$ be compatible steps and let $A_i = A_i^I \oplus A_i^E$, for $i \in \{1,2\}$, be a corresponding decomposition (see Definition 3.2). Then there exists a unique step $u_3\ [A_3\rangle\ v_3$, with $A_3 = \alpha_1^{\oplus}(A_1^I) \oplus \alpha_2^{\oplus}(A_2^I)$, which is projected to $u_i\ [A_i\rangle\ v_i$ along α_i for $i \in \{1,2\}$.*

Vice versa, any step $u_3\ [A_3\rangle\ v_3$ projects over two compatible steps $u_1\ [(A_3 \Downarrow \alpha_1)\rangle\ v_1$ of Z_1 and $u_2\ [(A_3 \Downarrow \alpha_2)\rangle\ v_2$ of Z_2, whose composition gives back the original step.

Proof. Concerning the first part, by definition of compatibility, we know that A_1 and A_2 can be decomposed as $A_i = A_i^I \oplus A_i^E$ $(i \in \{1,2\})$ such that

$$A_2^E = f_2^{\oplus}((A_1^I \Downarrow f_1)) \quad \text{and} \quad A_1^E = f_1^{\oplus}((A_2^I \Downarrow f_2)).$$

Moreover, $(u_1 \downarrow f_1) = (u_2 \downarrow f_2)$.

Now, since $u_i\ [A_i^I \oplus A_i^E\rangle\ v_i$, we can find markings u_i^I, u_i^E, v_i^I, v_i^E such that

$$\begin{array}{llll} u_1^I\ [A_1^I\rangle\ v_1^I & u_2^E\ [A_2^E\rangle\ v_2^E, & (u_1^I \downarrow f_1) = (u_2^E \downarrow f_2) \\ u_1^E\ [A_1^E\rangle\ v_1^E, & u_2^I\ [A_2^I\rangle\ v_2^I, & (u_1^E \downarrow f_1) = (u_2^I \downarrow f_2) \end{array}$$

In fact, just observe that, since $u_i\ [A_i\rangle\ v_i$, the marking u_i must be of the kind $w_i \oplus {}^\bullet A_i^I \oplus {}^\bullet A_i^E$ and similarly $v_i = w_i \oplus A_i^{I\bullet} \oplus A_i^{E\bullet}$. Thus we could choose

$$u_1^I = {}^\bullet A_1^I, \quad v_1^I = A_1^{I\bullet}, \quad u_2^E = {}^\bullet A_2^E, \quad v_2^E = A_2^{E\bullet},$$

and dually

$$u_1^E = {}^\bullet A_1^E \oplus w_1, \quad v_1^E = A_1^{E\bullet} \oplus w_1 \quad u_2^I = {}^\bullet A_2^I \oplus w_2, \quad v_2^I = A_2^{I\bullet} \oplus w_2$$

Therefore, we can use Lemma 3.1 and, defining $u_3' = u_1^I \uplus u_2^E$, $u_3'' = u_1^E \uplus u_2^I$, $v_3' = v_1^I \uplus v_2^I$, $v_3'' = v_1^E \uplus v_2^E$, we conclude

$$u_3'\ [\alpha_1^{\oplus}(A_1^I)\rangle\ v_3' \text{ and } u_3''\ [\alpha_2^{\oplus}(A_2^I)\rangle\ v_3''$$

Therefore

$$u'_3 \oplus u''_3 \ [\alpha_1^{\oplus}(A_1^I) \oplus \alpha_2^{\oplus}(A_2^I)\rangle \ v'_3 \oplus v''_3$$

By exploiting Proposition 2.1, we easily see that $u'_3 \oplus u''_3 = (u_1^I \oplus u_1^E) \uplus (u_2^E \oplus u_2^I) = u_1 \uplus u_2$, where u_0 denotes the common projection of u_1 and u_2 over Z_0. Similarly, $v'_3 \oplus v''_3 = v_1 \uplus v_2$ and thus

$$u_1 \uplus u_2 \ [\alpha_1^{\oplus}(A_1^I) \oplus \alpha_2^{\oplus}(A_1^E)\rangle \ v_1 \uplus v_2$$

is the desired step. The fact that it projects over the steps we started from in Z_1 and Z_2 follows by construction.

For the second part, consider any step $u_3 \ [A_3\rangle \ v_3$. Let $A_1 = (A_3 \Downarrow \alpha_1)$ and $A_2 = (A_3 \Downarrow \alpha_2)$. Decompose A_3 as

$$A_3 = A_3^1 \oplus A_3^2 \oplus A_3^0 \oplus A_3^{open}$$

where A_3^j, for $j \in \{1, 2\}$ includes only transitions in $\alpha_i(T_{Z_i} - f_i(T_{Z_0}))$, A_3^0 includes only transitions in $\alpha_i(f_i(T_{Z_0}))$ and finally A_3^{open} includes only interactions with the environment.

Then, if we define

$$A_1^I = (A_3^1 \Downarrow \alpha_1) \qquad A_1^E = A_1 \ominus A_1^I$$
$$A_2^I = ((A_3^2 \oplus A_3^0 \oplus A_3^{open}) \Downarrow \alpha_2) \qquad A_2^E = A_2 \ominus A_2^I$$

it is easy to show that the decomposition satisfies the requirements in Definition 3.2, hence the two steps are compatible, and their composition is immediately seen to give back the original step. □

Note that, in the decomposition of steps A_1 and A_2 considered in the proof above, all firings of transitions in the interface Z_0 are included in the internal part of A_2, i.e., no such transition is included in A_1^I. The possibility of having a decomposition with these properties will be useful later, in the proof of the congruence results.

4. Bisimilarity of Open Nets

In this section we study various notions of bisimilarity for open nets, proving that they are congruences with respect to the colimit-based composition operation. The considered behavioural equivalences will differ for the choice of the observations, which can be single firings or parallel steps. Additionally, we will consider weak forms of such equivalences, arising in the presence of unobservable actions.

4.1. **A High Level View on the Congruence Results.**

A first step consists of defining suitable *labelled transition systems* (LTSs) associated with an open net. Generally speaking, net transitions carry a label which is observed when they fire. Additionally, in the labelled transition systems we also observe what happens at the open places. This corresponds to observing the potential interactions with the surrounding environment, as open places act as gluing points in the composition operation, and it is pivotal for the mentioned congruence results.

Given an open net Z, the labeled transition systems we shall consider will have all markings of the net, $S_Z^{\oplus}$, as states, but they will differ concerning the transitions and their labels. For example, in the *firing* LTS the transitions are generated by the firings of Z, and correspondingly they are labelled over the set

$$\Lambda_Z = \Lambda \cup \{+_s : s \in O_Z^+\} \cup \{-_s : s \in O_Z^-\}.$$

As discussed in the conclusions, the firing LTS resembles the labelled transition system arising from the view of Petri nets as reactive systems in [26, 35]. Analogous LTSs are also obtained in [43] with the use of pseudo-transitions and in [29] by inserting a net in a universal context.

Instead, in the *step* LTS the transitions are generated by the steps of Z, and they are labeled over $\Lambda_Z^\oplus$. The corresponding notion of bisimilarity will capture, to some extent, the concurrency properties of the system (see, e.g., [44, 30]).

For notational convenience we extend the labelling function λ_Z to the set of extended events $\bar{T}_Z$, by defining $\lambda_Z(x) = x$ for $x \in \bar{T}_Z - T_Z$ (i.e., for $x = +_s$ or $x = -_s$ with $s \in S_Z$).

Definition 4.1 (step and firing lts for an open net)**.** The *step* LTS associated to an open net Z is the pair $\langle S_Z^\oplus, \rightarrow_{\mathsf{S},Z}\rangle$, where states are markings $u_Z \in S_Z^\oplus$ and the transition relation $\rightarrow_{\mathsf{S},Z} \subseteq S_Z^\oplus \times \Lambda_Z^\oplus \times S_Z^\oplus$ includes all transitions

$$u_Z \xrightarrow{\lambda_Z^\oplus(A)}_{\mathsf{S},Z} u'_Z$$

for all markings $u_Z, u'_Z \in S_Z^\oplus$ and $A \in \bar{T}_Z^\oplus$ such that there is a step $u_Z\ [A\rangle\ u'_Z$ in Z. The *firing* LTS $\langle S_Z^\oplus, \rightarrow_{\mathsf{F},Z}\rangle$ is defined similarly: the transition relation $\rightarrow_{\mathsf{F},Z} \subseteq S_Z^\oplus \times \Lambda_Z \times S_Z^\oplus$ includes all transitions

$$u_Z \xrightarrow{\lambda_Z(\epsilon)}_{\mathsf{F},Z} u'_Z$$

such that there is a firing $u_Z\ [\epsilon\rangle\ u'_Z$ in Z, with $\epsilon \in \bar{T}_Z$.

As we have done above for the transition relations, in the sequel the subscripts "S" and "F" will be used for distinguishing notions based on the step and on the firing behaviour, respectively, of a net.

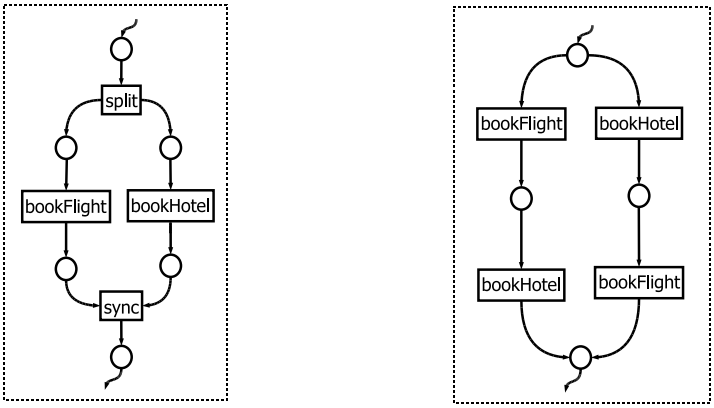
When observing the behaviour of a system, usually only a subset of events is considered visible. Here this is formalised by selecting a subset of labels representing internal firings, playing a role similar to τ-actions in process calculi, and then considering a corresponding notion of weak bisimilarity. Let $\Lambda_\tau \subseteq \Lambda$ be a subset of *unobservable* labels, fixed for the rest of the paper.

Definition 4.2 (weak transition systems)**.** For $\mathsf{x} \in \{\mathsf{S}, \mathsf{F}\}$ we write $v \overset{\ell}{\rightsquigarrow}_{\mathsf{x},Z} v'$ if $v, v' \in S_Z^\oplus$ are markings such that $v \xrightarrow{\ell'}_{\mathsf{x},Z} v'$ with $\ell = (\ell' \downarrow (\Lambda - \Lambda_\tau))$. Then the *weak* (step or firing) LTS is defined by letting

- $v \overset{0}{\Longrightarrow}_{\mathsf{x},Z} v'$ whenever $v \overset{0}{\rightsquigarrow}{}^*_{\mathsf{x},Z} v'$.
- $v \overset{\ell}{\Longrightarrow}_{\mathsf{x},Z} v'$ whenever $v \overset{0}{\rightsquigarrow}{}^*_{\mathsf{x},Z} \overset{\ell}{\rightsquigarrow}_{\mathsf{x},Z} \overset{0}{\rightsquigarrow}{}^*_{\mathsf{x},Z} v' \qquad \ell \neq 0.$

Transitions labelled with 0 will be often referred to as τ-transitions or silent transitions.

Weak step and firing bisimilarity is now defined in a standard way, but note that when the set of unobservable labels is empty, this actually corresponds to strong bisimilarity. Only, in order to be able to relate the extended events of the two nets, we need to specify for each open place of one net which is the corresponding open place in the other net; therefore bisimulations between two nets are parametrised by a bijection between their open places. Given two open nets Z_1 and Z_2 a *correspondence* $\eta = \langle \eta^+, \eta^-\rangle$ between Z_1 and Z_2 is a pair of bijections $\eta^+ : O_{Z_1}^+ \to O_{Z_2}^+$ and $\eta^- : O_{Z_1}^- \to O_{Z_2}^-$. In order to simplify the notation, in the following, given an open place $s_1 \in O_{Z_1}^+ \cup O_{Z_1}^-$ we will write simply



(a) Travel agency A. (b) Travel agency B.

Figure 5: Two open nets which are firing bisimilar but not step bisimilar.

$\eta(s_1)$ to denote its image through the appropriate component of η, i.e., a correspondence $\eta = \langle \eta^+, \eta^- \rangle$ will be identified with the function $\eta^+ \cup \eta^- : O^+_{Z_1} \cup O^-_{Z_1} \to O^+_{Z_2} \cup O^-_{Z_2}$.

Definition 4.3 ((weak) step and firing bisimilarity)**.** Let Z_1, Z_2 be open nets and $\eta : O_{Z_1} \leftrightarrow O_{Z_2}$ be a correspondence between Z_1 and Z_2. A (weak) η-x-bisimulation (with $\mathsf{x} \in \{\mathsf{S}, \mathsf{F}\}$ - S for *step* and F for *firing*) between Z_1 and Z_2 is a relation over markings $\mathcal{R} \subseteq S_1^{\oplus} \times S_2^{\oplus}$ such that if $(u_1, u_2) \in \mathcal{R}$ then

- if $u_1 \overset{\ell}{\rightsquigarrow}_{\mathsf{x},Z_1} u_1'$ in Z_1, then there exists u_2' such that $u_2 \overset{\eta^{\oplus}(\ell)}{\Longrightarrow}_{\mathsf{x},Z_2} u_2'$ in Z_2 and $(u_1', u_2') \in \mathcal{R}$;
- the symmetric condition holds;

where $\eta(+_s) = +_{\eta(s)}$, $\eta(-_s) = -_{\eta(s)}$, and $\eta(\ell) = \ell$ for any $\ell \in \Lambda$.

Two open nets Z_1 and Z_2 are *(weakly) η-x-bisimilar*, denoted $Z_1 \approx^{\mathsf{x}}_{\eta} Z_2$, if $\eta : O_{Z_1} \leftrightarrow O_{Z_2}$ is a correspondence and there exists a (weak) η-bisimulation $\mathcal{R}$ over Z_1 and Z_2 such that $(\hat{u}_1, \hat{u}_2) \in \mathcal{R}$. We will say that Z_1 and Z_2 are (weakly) x-bisimilar, written $Z_1 \approx^{\mathsf{x}} Z_2$, if $Z_1 \approx^{\mathsf{x}}_{\eta} Z_2$ for some correspondence η.

Clearly, step bisimilarity is finer than firing bisimilarity, i.e., if $Z_1 \approx^{\mathsf{S}} Z_2$ then $Z_1 \approx^{\mathsf{F}} Z_2$.

Observe that in the definition of step bisimilarity, whenever $v \overset{\ell}{\Longrightarrow}_{\mathsf{S},Z} v'$ and thus $v \overset{0}{\rightsquigarrow}^*_{\mathsf{S},Z} \overset{\ell}{\rightsquigarrow}_{\mathsf{S},Z} \overset{0}{\rightsquigarrow}^*_{\mathsf{S},Z} v'$, one can assume that the step inducing $\overset{\ell}{\rightsquigarrow}_{\mathsf{S},Z}$ does not include any τ-transition (since, if this is not the case, the τ-transitions can be anticipated or postponed).

As an example, consider the open nets in Fig. 5, which can be seen as the representation of (part of) the booking process in a travel agency. The bookings of the flight (bookFlight) and of the hotel (bookHotel) are independent and could be performed in parallel. However, this is possible only for agency A (Fig. 5(a)), while in agency B (Fig. 5(b)), where a single person takes care of all bookings, the two actions will be executed sequentially. Now, it is easy to check that, assuming that only the actions bookFlight and bookHotel are visible, the two nets are firing bisimilar, but they are not step bisimilar. Hence, as already mentioned, step bisimilarity discriminates also according to the degree of parallelism that is possible in a computation.

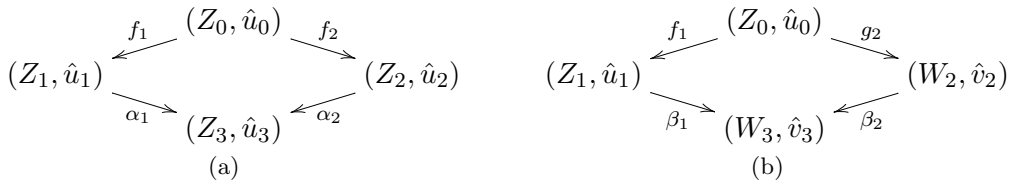


Figure 6: Pushouts in **ONet**.

As already mentioned, weak bisimilarity boils down to the notion of strong bisimilarity when all labels are observable, i.e., when $\Lambda_\tau = \emptyset$. For convenience of the reader we make explicit the notion of strong bisimilarity.

Definition 4.4 (strong bisimilarity)**.** When Z_1 and Z_2 are weakly η-x-bisimilar open nets, with $\Lambda_\tau = \emptyset$ we say that Z_1 and Z_2 are *strongly η-x-bisimilar* and write $Z_1 \sim^{\mathsf{x}}_{\eta} Z_2$ or simply $Z_1 \sim^{\mathsf{x}} Z_2$. Explicitly, a *strong η-x-bisimulation* over Z_1 and Z_2 is a relation over their markings $\mathcal{R} \subseteq S_1^{\oplus} \times S_2^{\oplus}$ such that if $(u_1, u_2) \in \mathcal{R}$ then

- if $u_1 \xrightarrow{\ell}_{\mathsf{x},Z_1} u_1'$ in Z_1, then there exists u_2' such that $u_2 \xrightarrow{\eta(\ell)}_{\mathsf{x},Z_2} u_2'$ in Z_2 and $(u_1', u_2') \in \mathcal{R}$;
- the symmetric condition holds.

We can finally state the congruence property for the considered behavioural equivalences with respect to the composition operation on open nets. The result will be proved separately for the various cases in the next subsection.

Theorem 4.5 (bisimilarity is a congruence)**.** *Let Z_0, Z_1, Z_2, W_2 be open nets. Let $Z_2 \approx^{\mathsf{x}}_{\eta} W_2$, for some correspondence η and $\mathsf{x} \in \{\mathsf{S}, \mathsf{F}\}$. Consider the nets $Z_3 = Z_1 +_{f_1,f_2} Z_2$ and $W_3 = Z_1 +_{f_1,g_2} W_2$, as in Fig. 6 where f_1, f_2 and g_2 are embeddings, f_1 and f_2 are composable, and f_1 and g_2 are composable as well.*

If $g_2|_{O_0} = \eta \circ (f_2|_{O_0})$ (i.e., f_2 and g_2 are consistent with η on open places) then $Z_3 \approx^{\mathsf{x}}_{\eta'} W_3$, where $\eta' : O_{Z_3} \leftrightarrow O_{W_3}$ is the correspondence defined as follows: for all $s \in O_{Z_3}$, $\eta'(s) = \beta_1(s')$ if $s = \alpha_1(s')$, and $\eta'(s) = \beta_2(\eta(s'))$ if $s = \alpha_2(s')$.

4.2. **Proofs of the Congruence Results.**

In order to prove the congruence results it is convenient to proceed as follows: we first consider strong step bisimilarity which can be more easily handled than its weak variant. Next the proof of the congruence result for the weak variant can adapted from the strong case. Finally, as firing bisimulation can (almost) be considered as a special case of step bisimulation, the proof of the corresponding congruence result easily follows from that of step bisimilarity. It is worth stressing that the complexity of the proof is mainly due to the fact that we consider steps instead of single firings.

We start with a technical lemma which will play a central role later. It states that for given *composable* embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$, any step in Z_2 where interactions with the environment only occur on places which are open also in $Z_1 +_{Z_0} Z_2$, can be projected along f_2 to Z_0 and then simulated in Z_1.

Lemma 4.6. *Let $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ be composable embeddings in* **ONet***, let $Z_3 = Z_1 +_{f_1,f_2} Z_2$ and let $u_i \in {S_{Z_i}}^{\oplus}$ ($i \in \{1, 2\}$) be markings such that $(u_1 \downarrow f_1) = (u_2 \downarrow f_2)$.*

Let $u_2\ [A_2\rangle\ v_2$ be a step such that if $x_s \in A_2$, for $x \in \{+,-\}$ then $\alpha_2(s) \in O^x_{Z_3}$. Then $u_1\ [f_1^{\oplus}((A_2 \Downarrow f_2))\rangle\ v_1$ and $u_1 \uplus u_2\ [\alpha_2^{\oplus}(A_2)\rangle\ v_1 \uplus v_2$.

Proof. Let $A_1 = f_1^{\oplus}((A_2 \Downarrow f_2))$. First note that A_1 is well-defined, i.e., $A_1 \in \bar{T}^{\oplus}_{Z_1}$. For instance, let us show that if $+_{s_1} \in A_1$ then s_1 is input open, i.e., $s_1 \in O^+_{Z_1}$. By definition of A_1 we deduce that there is $+_{s_0} \in (A_2 \Downarrow f_2)$ with $f_1(s_0) = s_1$. Now, by the assumptions on A_2, there are two possibilities:

- $+_{s_0} \in (t_2 \Downarrow f_2)$ with $t_2 \in T_{Z_2}$
  By the definition of projection for steps, this implies that $f_2(s_0) \in {}^\bullet t_2$, with $t_2 \notin f_2(T_{Z_0})$ and thus $s_0 \in \mathsf{in}(f_2)$. Since f_1 and f_2 are composable, we have that $s_1 = f_1(s_0) \in f_1(\mathsf{in}(f_2)) \subseteq O^+_{Z_1}$, as desired.
- $+_{s_0} \in (+_{s_2} \Downarrow f_2)$ with $\alpha_2(s_2) \in O^+_{Z_3}$
  Since the diagram in Fig. 3 commutes, we have that $\alpha_1(s_1) = \alpha_2(s_2)$. Since $\alpha_2(s_2) \in O^+_{Z_3}$, by condition (1) in the definition of open net morphism (Definition 1.7), $s_1 \in O^+_{Z_1}$, as desired.

Now observe that

$$\begin{aligned} {}^\bullet A_1 &= {}^\bullet(f_1^{\oplus}((A_2 \Downarrow f_2))) \\ &= f_1^{\oplus}({}^\bullet(A_2 \Downarrow f_2)) && \text{[by def. of open net morphism]} \\ &= f_1^{\oplus}(({}^\bullet A_2 \downarrow f_2)) && \text{[by Lemma 1.10.(3)]} \end{aligned}$$

Since the step $u_2\ [A_2\rangle\ v_2$ is enabled, we know that ${}^\bullet A_2 \leq u_2$, and thus

$$\begin{aligned} {}^\bullet A_1 &= f_1^{\oplus}(({}^\bullet A_2 \downarrow f_2)) \\ &\leq f_1^{\oplus}((u_2 \downarrow f_2)) \\ &= f_1^{\oplus}((u_1 \downarrow f_1)) && \text{[since } (u_2 \downarrow f_2) = (u_1 \downarrow f_1)\text{]} \\ &\leq u_1 && \text{[by Lemma 1.10.(4)]} \end{aligned}$$

Hence, the step $u_1\ [A_1\rangle\ v_1$ can be performed. Clearly, the two steps in Z_1 and Z_2 are compatible, and thus we conclude with Lemma 3.3. □

4.2.1. *Strong Step Bisimilarity.*

Theorem 4.7. *Strong step bisimilarity is a congruence.*

Proof. Let Z_0, Z_1, Z_2, W_2 be open nets, with $Z_2 \sim^{\mathsf{S}}_{\eta} W_2$, for some correspondence η. Let $Z_3 = Z_1 +_{f_1,f_2} Z_2$ and $W_3 = Z_1 +_{f_1,g_2} W_2$, as in Fig. 6, where f_1, f_2 and g_2 are embeddings, with f_1, f_2 and f_1, g_2 composable and $g_2|_{O_{Z_0}} = \eta \circ (f_2|_{O_{Z_0}})$.

To simplify the notation, assume, without loss of generality, that all the morphisms in the diagrams of Fig. 6 are inclusions and $\eta = id$. Hence $f_2|_{O_{Z_0}} = g_2|_{O_{Z_0}}$.

Now let $\mathcal{R}$ be a η-S-bisimulation over Z_2 and W_2 such that $(\hat{u}_2, \hat{v}_2) \in \mathcal{R}$, which exists by hypothesis. Consider the relation $\mathcal{R}'$ over Z_3 and W_3 defined as

$$\mathcal{R}' = \{(u_1 \uplus_{u_0} u_2, v_1 \uplus_{v_0} v_2) : (u_2, v_2) \in \mathcal{R} \ \wedge\ u_1 \ominus u_0 = v_1 \ominus v_0\}$$

The condition above on u_1 and v_1 means that the markings can differ, but only for the number of tokens in places of the interface net Z_0 (notice that the marking of Z_0 is completely determined by the marking of components Z_2 and W_2).

We claim that $\mathcal{R}'$ is a η'-S-bisimulation over Z_3 and W_3, where η' is again the identity on open places. Since, by the construction of the pushout, $(\hat{u}_3, \hat{v}_3) = (\hat{u}_1 \uplus_{\hat{u}_0} \hat{u}_2, \hat{u}_1 \uplus_{\hat{u}_0} \hat{v}_2) \in \mathcal{R}'$, this provides the desired result.

In order to prove that $\mathcal{R}'$ is a η'-S-bisimulation, assume that $u_3 \xrightarrow{\ell}_{\mathsf{S},Z_3} u_3'$. Therefore

$$u_3 \ [A_3\rangle \ u_3' \quad \text{with } \ell = \lambda^{\oplus}_{Z_3}(A_3)$$

and by Lemma 3.3 we can project the step A_3 over the components Z_1 and Z_2 thus getting for $i \in \{1,2\}$ the following steps in Z_i:

$$u_i \ [A_i\rangle \ u_i' \tag{4.1}$$

Since by the same lemma such steps are compatible, according to Definition 3.2, we can find partitions

$$A_i = A_i^I \oplus A_i^E \text{ with } i \in \{1,2\}$$

such that

$$A_1^E = f_1^{\oplus}((A_2^I \Downarrow f_2)) \qquad A_2^E = f_2^{\oplus}((A_1^I \Downarrow f_1)) \tag{4.2}$$

and

$$A_3 = \alpha_1^{\oplus}(A_1^I) \oplus \alpha_2^{\oplus}(A_2^I) \tag{4.3}$$

Additionally, as shown in the proof of Lemma 3.3, we can assume, w.l.o.g., that A_2^E consists only of interactions with the environment, i.e., $A_2^E \in \{x_s \mid x \in \{+,-\},\ s \in O^x_{Z_2}\}^{\oplus}$, or, equivalently, that A_1^I does not contain firings of transitions of Z_0.

Now, since $(u_2, v_2) \in \mathcal{R}$, the step (4.1) of Z_2 can be simulated by W_2, i.e., there is

$$v_2 \ [B_2\rangle \ v_2' \tag{4.4}$$

with $\lambda^{\oplus}(B_2) = \lambda^{\oplus}(A_2)$ and $(u_2', v_2') \in \mathcal{R}$.

We can now split B_2 in an "internal" and an "external" part, according to the splitting of A_1, i.e., we define

$$B_2^E = A_2^E \qquad B_2^I = B_2 \ominus B_2^E \tag{4.5}$$

Notice that we can legally define $B_2^E = A_2^E$ since A_2^E consists only of interactions with the environment, which are necessarily also in B_2 since $\lambda^{\oplus}(B_2) = \lambda^{\oplus}(A_2)$ (and recall that places in the interface have the same name in Z_2 and W_2).

Now, define

$$v_2^I = {}^{\bullet}B_2^I \qquad v_2^{I\prime} = B_2^{I\bullet} \tag{4.6}$$

$$v_2^E = v_2 \ominus v_2^I \qquad v_2^{E\prime} = v_2' \ominus v_2^{I\prime} \tag{4.7}$$

and thus we have

$$v_2^I \ [B_2^I\rangle \ v_2^{I\prime} \tag{4.8}$$

$$v_2^E \ [B_2^E\rangle \ v_2^{E\prime} \tag{4.9}$$

Now, the idea is to construct a step in W_3 by using separately the internal part of the step in W_2 and the internal part of the step in Z_1 (which plays the role of a context).

In order to apply Lemma 4.6 to the step in (4.8), we note that if $+_s \in B_2^I$ then $s \in O^+_{W_3}$ (and the same holds for $-_s$). In fact, if $+_s \in B_2^I$, then by construction of B_2^I and since $\lambda^{\oplus}(A_2) = \lambda^{\oplus}(B_2)$, we must have $+_s \in A_2^I$. Now, if $s \notin Z_0$ then, given that $s \in O^+_{Z_2}$ we have that $s \in O^+_{Z_3} = O^+_{W_3}$. Otherwise, if $s \in Z_0$ then, by (4.2), we have that $f_1^{\oplus}((+_s \downarrow f_2)) = +_s \in A_1^E$, thus $s \in O^+_{Z_1}$, and hence $s \in O^+_{Z_3} = O^+_{W_3}$.

Therefore if we define:

$$v_1^E = f_1^{\oplus}((v_2^I \downarrow g_2)) \qquad B_1^E = f_1^{\oplus}((B_2^I \Downarrow g_2)) \tag{4.10}$$

since clearly $(v_1^E \downarrow f_1) = (v_2^I \downarrow g_2)$, we can apply Lemma 4.6 to deduce that

$$v_1^E \ [B_1^E\rangle \ v_1^{E\prime}$$

and

$$v_1^E \uplus v_2^I \ [\beta_2{}^{\oplus}(B_2^I)\rangle \ v_1^{E\prime} \uplus v_2^{I\prime} \tag{4.11}$$

Note that $v_1^E \leq v_1$. In fact $v_2^I \leq v_2$. Therefore $(v_2^I \downarrow g_2) \leq (v_2 \downarrow g_2)$ and thus

$$v_1^E = f_1^{\oplus}((v_2^I \downarrow g_2)) \leq f_1^{\oplus}((v_2 \downarrow g_2)) \leq v_1$$

Let us now construct the other part of the step in W_3, arising as the composition of an internal step in Z_1 and the external part of the step in W_2. As mentioned before, since the component Z_1 plays the role of a context (it is the same in both composed nets) we can simply define:

$$B_1^I = A_1^I \tag{4.12}$$

If we let

$$v_1^I = v_1 \ominus v_1^E \tag{4.13}$$

then we can see that

$$v_1^I \ [B_1^I\rangle \ v_1^{I\prime} \tag{4.14}$$

We can show that indeed ${}^\bullet B_1^I \leq v_1^I$, with a long, but easy calculation. In fact, since $A_1^I = B_1^I$ by (4.12)

$${}^\bullet B_1^I = {}^\bullet A_1^I = ({}^\bullet A_1^I \downarrow (S_{Z_1} - S_{Z_0})) \oplus ({}^\bullet A_1^I \downarrow S_{Z_0}) \tag{4.15}$$

In the last expression, $({}^\bullet A_1^I \downarrow (S_{Z_1} - S_{Z_0}))$ and $({}^\bullet A_1^I \downarrow S_{Z_0})$ stands for the projections along the inclusions of $S_{Z_1} - S_{Z_0}$ and S_{Z_0}, respectively, into S_{Z_1}. Now, let us consider the two summands separately. Concerning the first one:

$$\begin{array}{ll}
({}^\bullet A_1^I \downarrow (S_{Z_1} - S_{Z_0})) \leq u_1 \ominus u_0 = & [\text{since } A_1^I \text{ enabled in } u_1 \text{ by (4.1)}] \\
= v_1 \ominus v_0 = & [\text{by construction of } \mathcal{R}'] \\
= v_1 \ominus f_1^{\oplus}((v_2 \downarrow g_2)) &
\end{array}$$

Let us consider the second one:

$$\begin{array}{ll}
({}^\bullet A_1^I \downarrow S_{Z_0}) = f_1^{\oplus}(({}^\bullet A_1^I \downarrow f_1)) = & \\
= f_1^{\oplus}(({}^\bullet A_2^E \downarrow f_2)) = & [\text{since } (A_1^I \Downarrow f_1) = (A_2^E \Downarrow f_2) \text{ by (4.2)}] \\
= f_1^{\oplus}(({}^\bullet B_2^E \downarrow g_2)) \leq & [\text{by (4.5) and the fact that } g_2, f_2 \text{ agree on } O_{Z_0}] \\
\leq f_1^{\oplus}((v_2^E \downarrow g_2)) & [\text{since by (4.9) } {}^\bullet B_2^E \leq v_2^E]
\end{array}$$

Putting together the two summands, from (4.15) we have

$$\begin{array}{ll}
{}^\bullet B_1^I \leq v_1 \ominus f_1^{\oplus}((v_2 \downarrow g_2)) \oplus f_1^{\oplus}((v_2^E \downarrow g_2)) = & \\
= v_1 \ominus f_1^{\oplus}((v_2 \ominus v_2^E \downarrow g_2)) = & [\text{since } v_2^E \leq v_2 \text{ and } f \text{ injective}] \\
= v_1 \ominus f_1^{\oplus}((v_2^I \downarrow g_2)) = & [\text{since } v_2^I = v_2 \ominus v_2^E \text{ by (4.7)}] \\
= v_1 \ominus v_1^E = & [\text{by (4.10)}] \\
= v_1^I & [\text{by (4.13)}]
\end{array}$$

In order to apply Lemma 4.6 to the step (4.14), we can prove that if $+_s \in B_1^I$ then $s \in O_{W_3}^+$ (and the same for $-_s$) as in the previous case. Additionally, we have

$$\begin{array}{ll}
(v_1^I \downarrow f_1) = ((v_1 \ominus v_1^E) \downarrow f_1) = & \text{[by def. of } v_1^I \text{ in (4.13)]} \\
= (v_1 \downarrow f_1) \ominus (v_1^E \downarrow f_1) = & \\
= (v_2 \downarrow g_2) \ominus (v_1^E \downarrow f_1) = & \text{[since } (v_1 \downarrow f_1) = (v_2 \downarrow g_2) \text{ by hypothesis]} \\
= (v_2 \downarrow g_2) \ominus (v_2^I \downarrow g_2) = & \text{[since } (v_1^E \downarrow f_1) = (v_2^I \downarrow g_2) \text{ by (4.10)]} \\
= ((v_2 \ominus v_2^I) \downarrow g_2) = & \\
= (v_2^E \downarrow g_2) & \text{[by def. of } v_2^E \text{ in (4.7)]}
\end{array}$$

and moreover

$$B_2^E = {g_2}^{\oplus}((B_1^I \Downarrow f_1)).$$

In fact

$$\begin{array}{ll}
B_2^E = A_2^E & \text{[by (4.5)]} \\
= f_2^{\oplus}((A_1^I \Downarrow f_1)) = & \text{[by (4.2)]} \\
= f_2^{\oplus}((B_1^I \Downarrow f_1)) = & \text{[by (4.12)]} \\
= {g_2}^{\oplus}((B_1^I \Downarrow f_1)) = & \text{[since } g_2 \text{ and } f_2 \text{ “agree” on } O_{Z_0} \text{]}
\end{array}$$

Therefore, by Lemma 4.6, we have that

$$v_2^E \ [B_2^E\rangle \ v_2^{E\prime} \tag{4.16}$$

and

$$v_1^I \uplus v_2^E \ [{\beta_1}^{\oplus}(B_1^I)\rangle \ v_1^{I\prime} \uplus v_2^{E\prime} \tag{4.17}$$

Now, by Proposition 2.1, we can join the steps (4.11) and (4.17) and obtain

$$(v_1^E \uplus v_2^I) \oplus (v_1^I \uplus v_2^E) \ [{\beta_1}^{\oplus}(B_1^I) \oplus {\beta_2}^{\oplus}(B_2^I)\rangle \ (v_1^{I\prime} \uplus v_2^{E\prime}) \oplus (v_1^{E\prime} \uplus v_2^{I\prime})$$

i.e., the desired step which can be used to simulate $u_3 \xrightarrow{\ell}_{\mathsf{S},Z_3} u_3'$. In fact the label is

$$\begin{array}{ll}
\lambda_{W_3}^{\oplus}({\beta_1}^{\oplus}(B_1^I) \oplus {\beta_2}^{\oplus}(B_2^I)) = & \\
= \lambda_{W_3}^{\oplus}({\beta_1}^{\oplus}(B_1^I)) \oplus \lambda_{W_3}^{\oplus}({\beta_2}^{\oplus}(B_2^I)) & \text{[since the diagram in Fig. 6(b) commutes]} \\
= \lambda_{Z_1}^{\oplus}(B_1^I) \oplus \lambda_{W_2}^{\oplus}(B_2^I) & \text{[since } A_1^I = B_1^I \text{ by (4.12) and} \\
& \ \lambda_{W_2}^{\oplus}(B_2^I) = \lambda_{Z_2}^{\oplus}(A_2^I) \text{ by construction (4.4)]} \\
= \lambda_{Z_1}^{\oplus}(A_1^I) \oplus \lambda_{Z_2}^{\oplus}(A_2^I) & \text{[since the diagram in Fig. 6(a) commutes]} \\
 & \\
= \lambda_{Z_3}^{\oplus}(\alpha_1^{\oplus}(A_1^I)) \oplus \lambda_{Z_3}^{\oplus}(\alpha_2^{\oplus}(A_2^I)) & \\
= \lambda_{Z_3}^{\oplus}(\alpha_1^{\oplus}(A_1^I) \oplus \alpha_2^{\oplus}(A_2^I)) & \text{[by (4.3)]} \\
= \lambda_{Z_3}^{\oplus}(A_3) & \\
= \ell &
\end{array}$$

Moreover, using (4.7), we have

$$(v_1^E \uplus v_2^I) \oplus (v_1^I \uplus v_2^E) = (v_1^I \oplus v_1^E) \uplus (v_2^I \oplus v_2^E) = v_1 \uplus v_2 = v_3.$$

And, if we define

$$v_1' = v_1^{I\prime} \oplus v_1^{E\prime}$$

recalling that, by (4.7), $v_2' = v_2^{I\prime} \oplus v_2^{E\prime}$, we have that the target state of the step is

$$v_3' = v_1' \uplus v_2'$$

Now, $(u'_2, v'_2) \in \mathcal{R}$ by construction. Moreover, the fact that $u'_1 \ominus u'_0 = v'_1 \ominus v'_0$ immediately follows from the fact that this property holds of the starting markings and we executed the same internal step in Z_1.

Hence $(u'_3, v'_3) \in \mathcal{R}'$ as desired. □

4.2.2. *Weak Step Bisimilarity.*

Theorem 4.8. *Weak step bisimilarity is a congruence.*

Proof. In order to show the desired result, we build on the proof of the strong case (Theorem 4.7). Let us use the same notation and define the relation $\mathcal{R}'$ in the same way. In order to prove that $\mathcal{R}'$ is an S-weak bisimulation we proceed as follows.

Let $u_3 \overset{\ell_3}{\rightsquigarrow}_{\mathsf{S},Z_3} u'_3$ and let us focus on the case $\ell_3 \neq 0$ (the case in which $\ell_3 = 0$ is completely analogous). This transition is induced by a step $u_3\ [A_3\rangle\ u'_3$, which can be projected over Z_1 and Z_2, thus getting, for $i \in \{1, 2\}$

$$u_i\ [A_i\rangle\ u'_i$$

Now, since $(u_2, v_2) \in \mathcal{R}$, the transition $u_2 \overset{\ell_2}{\rightsquigarrow}_{\mathsf{S},Z_2} u'_2$, induced by $u_2\ [A_2\rangle\ u'_2$ can be simulated in W_2, by $v \overset{\ell_2}{\Longrightarrow}_{\mathsf{S},W_2} v'$. Let the weak transition in W_2 arise from the sequence of steps

$$v_2 = v_2^0\ [B_2^1\rangle\ v_2^1 \ldots v_2^h\ [B_2^h\rangle\ v_2^{h+1} \ldots v_2^k\ [B_2^k\rangle\ v_2^{k+1} = v'_2$$

where $\lambda^{\oplus}_{W_2}(B_2^i) = 0$ for $i \neq h$ and $\lambda^{\oplus}_{W_2}(B_2^h) = \ell$ (and as remarked after Definition 4.3 we can assume that no transition in B_2^h has an unobservable label).

Now, any τ-step $v_2^i\ [B_2^i\rangle\ v_2^{i+1}$ $(i < h)$ consists only of firings of transitions of W_2. Hence, as in the strong case, by using Lemma 4.6 we can conclude that there is a "corresponding" step $v_1^i\ [B_1^i\rangle\ v_1^{i+1}$, consisting only of interactions with the environment, and their composition is a τ-step in W_3 of the kind $v_1^i\ [\alpha_2^{\oplus}(B_2^i)\rangle\ v_3^{i+1}$, with $\lambda^{\oplus}_{Z_1}(B_1^i) = 0$.

Note that since $v_1^i\ [B_1^i\rangle\ v_1^{i+1}$ consists only of interactions with the environment, $u_1 \ominus u_0 = v_1^{i+1} \ominus v_0^{i+1}$ for $i < h$.

For the "visible" step $v_2^h\ [B_2^h\rangle\ v_2^{h+1}$, we can apply the same argument as in the strong case, to get steps $v_1^h\ [B_1^h\rangle\ v_1^{h+1}$ and $v_3^h\ [B_3^h\rangle\ v_3^{h+1}$, with $\lambda^{\oplus}_{W_3}(B_3^h) = \ell$. Additionally, $u'_1 \ominus u'_0 = v_1^{h+1} \ominus v_0^{h+1}$.

Repeating the same argument for the remaining τ-steps, $v_2^i\ [B_2^i\rangle\ v_2^{i+1}$ $(i > h)$, i.e., using again Lemma 4.6, we can prove that there are steps $v_1^i\ [B_1^i\rangle\ v_1^{i+1}$, consisting only of interactions with the environment, correspondingly τ-steps in W_3 of the kind $v_1^i\ [\alpha_2^{\oplus}(B_2^i)\rangle\ v_3^{i+1}$, with $\lambda^{\oplus}_{Z_1}(B_1^i) = 0$, for $i > h$. Such sequence of further τ-steps in W_3 leads to a marking $v'_3 = v_1^{k+1} \uplus v_2^{k+1}$, where $v_1^{k+1} \ominus v_0^{k+1} = u'_1 \ominus u'_0$ and $v_2^{k+1} = v'_2$ with $(u'_2, v'_2) \in \mathcal{R}$. Hence $(u'_3, v'_3) \in \mathcal{R}'$.

In other words $v_3 \overset{\ell}{\Longrightarrow}_{\mathsf{S},W_3} v'_3$ and $(u'_3, v'_3) \in \mathcal{R}'$, as desired. □

4.2.3. *Weak (and Strong) Firing Bisimilarity.*

Theorem 4.9. *Strong and weak firing bisimilarity are congruences.*

Proof. The proof remains essentially the same as for step bisimulation (Theorem 4.7 and Theorem 4.8). Only some minor adaptations are required.

Let us focus on weak bisimulation, which is the more general case. We use the same notation as in Theorem 4.8 and define $\mathcal{R}'$ in the same way. In order to prove that $\mathcal{R}'$ is an S-weak bisimulation we proceed as follows.

Let $(u_3, v_3) \in \mathcal{R}'$ and let $u_3 \overset{\ell}{\rightsquigarrow}_{\mathsf{F},Z_3} u_3'$. Then there must be a step

$$u_3 \ [\epsilon_3\rangle \ u_3'$$

such that $\epsilon_3 \in \bar{T}_{Z_3}$ and $\lambda_{Z_3}(\epsilon_3) = \ell$. We can project the step over Z_2, thus getting

$$u_2 \ [A_2\rangle \ u_2' \tag{4.18}$$

The delicate case is the one in which $\epsilon_3 = t_3 \in T_{Z_3} - \alpha_2(T_{Z_2})$. In fact, in this case, A_2 is in general a proper multiset (of interactions with the environment) and thus we cannot argue, as in the case of step bisimulation, that the transition $u_2 \overset{\lambda^{\oplus}_{Z_2}(A_2)}{\longrightarrow}_{\mathsf{F},Z_2} u_2'$ must be simulated by W_2, since only single firings are simulated.

In order to proceed, we have first to linearise the step in (4.18) as

$$u_2 \overset{-_{s_1}}{\longrightarrow}_{\mathsf{F},Z_2} \cdots \overset{-_{s_k}}{\longrightarrow}_{\mathsf{F},Z_2} \overset{+_{s_{k+1}}}{\longrightarrow}_{\mathsf{F},Z_2} \cdots \overset{+_{s_{k+h}}}{\longrightarrow}_{\mathsf{F},Z_2} u_2' \tag{4.19}$$

Interestingly, the joint effect of the projection and of the linearization corresponds to the function ψ used in [43, page 96] to project a firing in the combined net to a firing sequence in the host net. Now we can say that this is simulated in W_2 by

$$v_2 \overset{-_{s_1}}{\Longrightarrow}_{\mathsf{F},W_2} \cdots \overset{-_{s_k}}{\Longrightarrow}_{\mathsf{F},W_2} \overset{+_{s_{k+1}}}{\Longrightarrow}_{\mathsf{F},W_2} \cdots \overset{+_{s_{k+h}}}{\Longrightarrow}_{\mathsf{F},W_2} v_2'$$

namely

$$\begin{aligned} v_2 &\overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \overset{-_{s_1}}{\longrightarrow}_{\mathsf{F},W_2} \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \cdots \\ &\cdots \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \overset{-_{s_k}}{\longrightarrow}_{\mathsf{F},W_2} \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \overset{+_{s_{k+1}}}{\longrightarrow}_{\mathsf{F},W_2} \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \cdots \\ &\cdots \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \overset{+_{s_{k+h}}}{\longrightarrow}_{\mathsf{F},W_2} \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} v_2' \end{aligned}$$

which in turn (since $-_{s_i}$ and $+_{s_j}$ firings can be clearly postponed and anticipated, respectively) can be reorganised as

$$v_2 \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \overset{-_{s_1}}{\longrightarrow}_{\mathsf{F},W_2} \cdots \overset{-_{s_k}}{\longrightarrow}_{\mathsf{F},W_2} \overset{+_{s_{k+1}}}{\longrightarrow}_{\mathsf{F},W_2} \cdots \overset{+_{s_{k+h}}}{\longrightarrow}_{\mathsf{F},W_2} \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} v_2'$$

and thus finally to

$$v_2 \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} \overset{\ell_2}{\Longrightarrow}_{\mathsf{F},W_2} \overset{0}{\Longrightarrow}_{\mathsf{F},W_2} v_2'$$

where $\ell_2 = (\bigoplus_{i=1}^{k} +_{s_1}) \oplus (\bigoplus_{i=1}^{h} -_{s_i})$. Then we can proceed exactly as in the proof for step bisimilarity. □

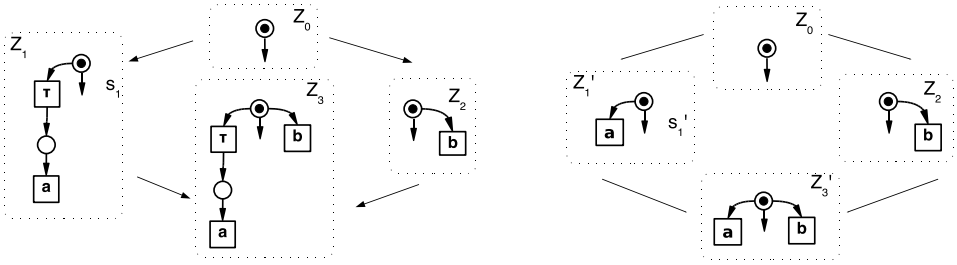


Figure 7: Two pushouts of open nets for the comparison to CCS.

4.3. **Comparison to CCS.**

We now give some hints as to why weak (firing) bisimilarity is a congruence in the case of open nets, but not in CCS [25]. Remember that a classical counterexample for CCS is as follows: $p_1 = \tau.a.0 \approx a.0 = p_2$, but $q_1 = \tau.a.0 + b.0 \not\approx a.0 + b.0 = q_2$. The reason for the latter inequality is that q_1 can do a τ and become $a.0$, while q_2 cannot mimic this step.

Fig. 7 shows a similar situation of nondeterministic choice for open nets, where τ is the only unobservable label. However, note that here the two nets Z_1 (corresponding to $\tau.a.0$) and Z_1' (corresponding to $a.0$) are *not* weakly firing bisimilar. Whenever the τ-transition is fired in Z_1, resulting in the marking m_1, this can not be mimicked in Z_1' by staying idle, since then in Z_1' a transition with label $-_{s_1'}$ is possible, while a transition labelled $-_{s_1}$ is not possible for the net Z_1 with marking m_1. Also note that the places s_1 respectively s_1' must be output open in order to allow composition with the net Z_2.

Roughly, this means that for open nets we are always able to observe the first invisible action in an open component, which is reminiscent of the definition of observation congruence in CCS: two processes p, q are called observation congruent if they are weakly bisimilar, with the additional constraint that whenever the first step of p is a τ-action, then it has to be answered by at least one τ-action of q (and vice versa). In both settings it is only the first τ-action that can be observed but not the subsequent ones.

5. Some Proof Techniques for Bisimilarity

We next present some properties of (strong and weak) bisimilarity, which can help in bisimilarity proofs. We first show that the set of open places can be uniformly reduced without altering the equivalence of open nets. Then we provide an up-to technique for firing bisimilarity.

We start by showing that given two bisimilar nets, if we "close" corresponding open places in both nets we still get two bisimilar nets. Given an open net Z and an open place $s \in O_Z^x$, let us denote by $Z - (s, x)$ the open net obtained from Z by closing place s, i.e., $Z' = (N, O_{Z'})$, where $O_{Z'}^x = O_Z^x - \{s\}$. The initial marking remains the same.

Proposition 5.1 ("closing" open places)**.** *Let $Z_1 \approx_\eta^{\mathsf{x}} Z_2$, with $\mathsf{x} \in \{\mathsf{F}, \mathsf{S}\}$. Let $s \in O_{Z_1}^x$ ($x \in \{-, +\}$) be an open place in Z_1. Then the nets $Z_1 - (s, x)$ and $Z_2 - (\eta(s), x)$ are η-x-bisimilar.*

Proof. Let $Z_1' = Z_1 - (s, x)$ and $Z_2' = Z_2 - (\eta(s), x)$. Let $\mathcal{R} \subseteq S_1^{\oplus} \times S_2^{\oplus}$ be an η-x-bisimulation such that $(\hat{u}_1, \hat{u}_2) \in \mathcal{R}$. Then $\mathcal{R}$ is a bisimulation between Z_1' and Z_2'. In fact, if $(u_1, u_2) \in \mathcal{R}$ and $u_1 \overset{\ell}{\rightsquigarrow}_{\mathsf{x},Z_1'} u_1'$ then clearly $u_1 \overset{\ell}{\rightsquigarrow}_{\mathsf{x},Z_1} u_1'$. Since $\mathcal{R}$ is a bisimulation for Z_1 and Z_2 this implies that $u_2 \overset{\eta(\ell)}{\Longrightarrow}_{\mathsf{x},Z_2} u_2'$ with $(u_1', u_2') \in \mathcal{R}$. Since ℓ is a label in Z_1' where place s has been closed, we are sure that $x_s \notin \ell$, and thus $u_2 \overset{\eta(\ell)}{\Longrightarrow}_{\mathsf{x},Z_2} u_2'$ implies $u_2 \overset{\eta(\ell)}{\Longrightarrow}_{\mathsf{x},Z_2'} u_2'$. Hence we get the desired result. □

We next provide a kind of *up-to technique* for firing bisimilarity. Given an open net Z, let us define the *out-degree* of a place $s \in S$ as the maximum number of tokens that the firing of an extended event can remove from s, formally:

$$deg(s) = \max\big(\{(\,{}^\bullet t)(s) : t \in T_Z\} \cup \{1 : s \in O_Z^-\}\big)$$

The idea, formalised by the notion of up-to bisimulation, is to allow tokens to be removed from input open places, when they exceed the out-degree of the place. More precisely, given a net Z and a marking $u \in S^{\oplus}$, let us say that a marking $v \in {O_Z^+}^{\oplus}$ is *subtractable* from u if $\forall s \in O_Z^+.\ v(s) \leq \max\{u(s) - deg(s), 0\}$. Note that when the number of tokens in a place s does not exceed its out-degree, i.e., $u(s) \leq deg(s)$, then $v(s) = 0$, i.e., no token is subtractable from s. If instead, $u(s) > deg(s)$, then the tokens in s which exceeds the out-degree of s can be safely subtracted from s. It is clear that when v is subtractable from u, all transitions enabled in u are also enabled in $u \ominus v$. Note that the empty marking is subtractable from any other marking.

Definition 5.2 (up-to firing bisimulation)**.** Let Z_1 and Z_2 be open nets, and let $\eta : O_{Z_1} \leftrightarrow O_{Z_2}$ be a correspondence between Z_1 and Z_2. A relation $\mathcal{R} \subseteq S_1^{\oplus} \times S_2^{\oplus}$ between markings is called an *up-to η-F-bisimulation* if whenever $(u_1, u_2) \in \mathcal{R}$ then

- if $u_1 \overset{\ell}{\rightsquigarrow}_{\mathsf{F},Z_1} u_1'$, then there exist markings u_2' such that $u_2 \overset{\eta(\ell)}{\Longrightarrow}_{\mathsf{F},Z_2} u_2'$, and $v_1 \in {O_{Z_1}^+}^{\oplus}$ subtractable from u_1', with $(u_1' \ominus v_1, u_2' \ominus \eta^{\oplus}(v_1)) \in \mathcal{R}$;
- the symmetric condition holds.

That is, the intuition behind up-to bisimulations is that some tokens might be superfluous since they are not necessary to fire a transition. Hence in the bisimulation game they can be removed in the two successor markings.

A first technical lemma shows an invariance property of up-to F-bisimulations, with respect to adding tokens in open places.

Lemma 5.3. *Let Z_1 and Z_2 be open nets, let $\eta : O_{Z_1} \leftrightarrow O_{Z_2}$ be a correspondence between Z_1 and Z_2, and let $\mathcal{R}$ be an up-to η-F-bisimulation between Z_1 and Z_2. Then*

(1) given any $s \in O_{Z_1}^+$, the relation $\mathcal{R}^s = \mathcal{R} \cup \{(u_1 \oplus s, u_2 \oplus \eta(s)) : (u_1, u_2) \in \mathcal{R}\}$ is an up-to η-F-bisimulation.

(2) $\mathcal{R}' = \mathcal{R} \cup \{(u_1 \oplus v_1, u_2 \oplus \eta^{\oplus}(v_1)) : (u_1, u_2) \in \mathcal{R} \wedge v_1 \in {O_{Z_1}^+}^{\oplus}\}$ is an up-to η-F-bisimulation.

Proof. 1. In order to simplify the notation, let us assume, without loss of generality, that η is the identity (i.e., $O_{Z_1}^+ = O_{Z_2}^+$ and $O_{Z_1}^- = O_{Z_2}^-$).

Let $(u_1 \oplus s, u_2 \oplus s) \in \mathcal{R}^s$. Let us show that if $u_1 \oplus s \overset{\ell}{\rightsquigarrow}_{\mathsf{F},Z_1} u_1'$ then there exists $u_2 \oplus s \overset{\ell}{\Longrightarrow}_{\mathsf{F},Z_2} u_2'$ and $v \in O_{Z_1}^+$ subtractable from u_1' with $(u_1' \ominus v, u_2' \ominus v) \in \mathcal{R}^s$. The other cases are completely analogous.

Observe that, since $s \in O_{Z_1}^+$, we have

$u_1 \overset{+_s}{\rightsquigarrow}_{\mathsf{F},Z_1} u_1 \oplus s$.

By definition of $\mathcal{R}^s$, we have $(u_1, u_2) \in \mathcal{R}$ and thus

$$u_2 \overset{+_s}{\Longrightarrow}_{\mathsf{F},Z_2} u_2'' \quad \text{and} \quad (u_1 \oplus s \ominus v', u_2'' \ominus v') \in \mathcal{R}^s \tag{5.1}$$

for a suitable $v' \in O^+_{Z_1}$ subtractable from $u_1' \oplus s$. Also notice that, since a $+_s$ can always be performed, we can assume that the firing sequence (5.1) is of the kind

$$u_2 \xrightarrow{+_s}_{\mathsf{F},Z_2} u_2 \oplus s \overset{0}{\Longrightarrow}_{\mathsf{F},Z_2} u_2'' \tag{5.2}$$

Now, if $u_1 \oplus s \overset{\ell}{\rightsquigarrow}_{\mathsf{F},Z_1} u_1'$, then, since v' is subtractable from $u_1 \oplus s$, also $u_1 \oplus s \ominus v' \overset{\ell}{\rightsquigarrow}_{\mathsf{F},Z_1} u_1' \ominus v'$. Thus, by (5.1)

$$u_2'' \ominus v' \overset{\ell}{\Longrightarrow}_{\mathsf{F},Z_2} u_2''' \quad \text{and} \quad (u_1' \ominus v' \ominus v'', u_2''' \ominus v'') \in \mathcal{R}^s \tag{5.3}$$

for a suitable $v'' \in {O^+_{Z_1}}^{\oplus}$, subtractable from $u_1 \ominus v'$.

Putting the above together with (5.2), we have that

$$u_2 \oplus s \overset{0}{\Longrightarrow}_{\mathsf{F},Z_2} u_2'' \overset{\ell}{\Longrightarrow}_{\mathsf{F},Z_2} u_2''' \oplus v'$$

i.e., $u_2 \oplus s \overset{\ell}{\Longrightarrow}_{\mathsf{F},Z_2} u_2''' \oplus v'$ and, if we denote $u_2' = u_2''' \oplus v'$, $(u_1' \ominus v' \ominus v'', u_2' \ominus v' \ominus v'') \in \mathcal{R}^s$. It is immediate to see that $v' \oplus v''$ is subtractable from u_1', and thus we conclude.

2. By an inductive reasoning, exploiting point 1, we can show that the relation $\mathcal{R}_n = \mathcal{R} \cup \{(u_1 \oplus v_1, u_2 \oplus \eta^{\oplus}(v_1)) : (u_1, u_2) \in \mathcal{R} \wedge v_1 \in {O^+_{Z_1}}^{\oplus} \wedge |v_1| \leq n\}$ is a η-F-weak bisimulation up-to for any n. Then we exploit the fact that the union of weak bisimulations up-to is again a weak-bisimulation up-to. □

We can finally prove the soundness of the up-to technique.

Proposition 5.4. *Let Z_1 and Z_2 be open nets, and let $\eta : O_{Z_1} \leftrightarrow O_{Z_2}$ be a correspondence between Z_1 and Z_2. Let $\mathcal{R}$ be an up-to η-F-bisimulation. Then for any $(u_1, u_2) \in \mathcal{R}$ we have that $(Z_1, u_1) \approx^{\mathsf{F}}_{\eta} (Z_2, u_2)$.*

Proof. In order to simplify the notation, let us assume, without loss of generality, that η is the identity (i.e., $O^+_{Z_1} = O^+_{Z_2}$ and $O^-_{Z_1} = O^-_{Z_2}$).

Let us show that

$$\mathcal{R}' = \{(u_1 \oplus v, u_2 \oplus v) : (u_1, u_2) \in \mathcal{R} \ \wedge \ v \in (O^+_{Z_1})^{\oplus}\}$$

is an η-F-bisimulation. Let $(u_1 \oplus v, u_2 \oplus v) \in \mathcal{R}'$, with $(u_1, u_2) \in \mathcal{R}$ and $v \in O^+_{Z_1}$, and assume that

$$u_1 \oplus v \overset{\ell}{\rightsquigarrow}_{\mathsf{F},Z_1} u_1'.$$

By Lemma 5.3 we know that $\mathcal{R}'$ is an up-to bisimulation, and thus there exists a transition

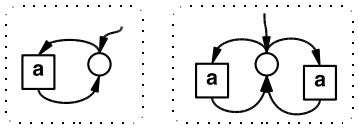
$$u_2 \oplus v \overset{\ell}{\Longrightarrow}_{\mathsf{F},Z_2} u_2'$$

and $v' \in {O^+_{Z_1}}^{\oplus}$, subtractable from u_1', such that $(u_1' \ominus v', u_2' \ominus v') \in \mathcal{R}'$. However, by construction of $\mathcal{R}'$, this implies that

$$(u_1', u_2') \in \mathcal{R}'$$

as desired. □

As it often happens with up-to techniques, the above result might allow to show that two nets are firing bisimilar by exhibiting finite relations (while bisimulations are typically infinite). E.g., consider the open nets on the right, where label a is observable. Then any firing bisimulation would include at least the pairs $\{(k \cdot s, k \cdot s) : k \in \mathbb{N}\}$, where s is the only place. Instead, according to the definition above $\{(0,0),(s,s)\}$ is an up-to bisimulation.



Note that, instead, the up-to technique does not extend to step bisimilarity: since an unbounded number of tokens can be needed to fire a parallel step there is no obvious generalisation of the notion of subtractable marking.

6. Reconfigurations of Open Nets

The results in the previous sections are used here to design a framework where a system specified as a (possibly open) Petri net can be reconfigured dynamically by transformation rules, triggered by the state/shape of the system. The congruence results allows one to characterise classes of reconfigurations which preserve the observational behaviour of the system.

6.1. **Behaviour Preserving Reconfigurations of Open Nets.**

The fact that the composition operation over open nets is defined in terms of a pushout construction suggests naturally a way of reconfiguring open nets by using the double-pushout approach to rewriting [14].

A *rewriting rule* over open nets consists of a pair of morphisms in **ONet**:

$$p = L_p \stackrel{l_p}{\leftarrow} K_p \stackrel{r_p}{\rightarrow} R_p$$

where L_p, K_p, R_p are open nets, called *left-hand side*, *interface* and *right-hand side* of the rule p, and l_p, r_p are open net embeddings. Intuitively, the rule specifies that, given a net Z, if the left-hand side L_p matches a subnet of Z then this can be reconfigured into Z' by replacing the occurrence of L_p with the right-hand side R_p, preserving the subnet K_p.

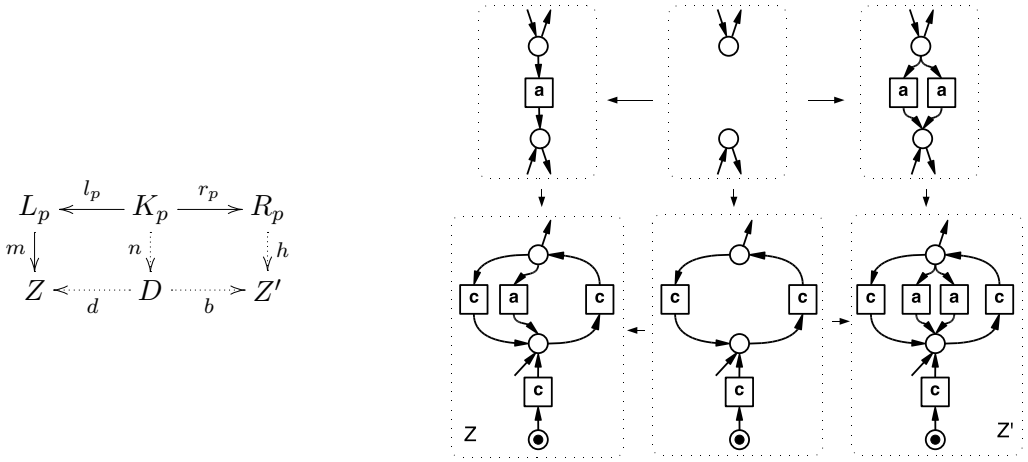
The notion of transformation is formally defined below.

Definition 6.1 (open net transformation)**.** Let p be a rewriting rule over open nets, let Z be an open net and let $m : L_p \to Z$ be a match, i.e., an open net embedding. We say that Z rewrites to Z' using p at match m, denoted $Z \Rightarrow^{p,m} Z'$ or simply $Z \Rightarrow^{p} Z'$, if the diagram of Fig. 8(a) can be constructed in **ONet**, where both squares are pushouts, and morphism n is composable with both l_p and r_p.

We stress that we are interested in transformations where the two pushout squares are built from composable arrows (technically, this ensures that the transformation can be performed in **Net** and then "lifted" to **ONet**).

We can now characterise the rules which do not alter the observational behaviour of an open Petri net as the rules with bisimilar left and right-hand side.

Definition 6.2 (behaviour preserving rules)**.** A x-behaviour preserving rule ($\mathsf{x} \in \{\mathsf{F}, \mathsf{S}\}$) is an open net rewriting rule p such that $L_p \approx^{\mathsf{x}}_{\eta} R_p$, where $\eta = (r_p \circ l_p^{-1})_{|O_{L_p}}$.



(a) (b)

Figure 8: Transforming open nets through DPO rewriting.

Then the next result is an easy consequence of Theorem 4.5.

Theorem 6.3 (behaviour-preserving reconfigurations)**.** *Let p be a x-behaviour preserving rule ($\mathsf{x} \in \{\mathsf{F}, \mathsf{S}\}$). Given an open net Z, if $Z \Rightarrow^{p,m} Z'$ via a match $m : L_p \to Z$, then $Z \approx^{\mathsf{x}} Z'$.*

Proof. Just observe that, in the DPO diagram of Figure 8(a), since the arrows l_p, n and r_p, n are composable, we can apply Theorem 4.5 to conclude that $Z \approx^{\mathsf{x}} Z'$. □

For instance, consider the double-pushout diagram in Fig. 8(b). It can be easily seen that the left- and right-hand sides of the applied rule are strongly (step) bisimilar. Hence we can conclude that Z and Z' are strongly (step) bisimilar as well.

6.2. **Applying Rules to Open Nets.**

As it is common in the categorical approaches to (graph) rewriting, the notion of open net transformation proposed in Definition 6.1 is rather "declarative" in style, because it requires the existence of two pushouts in category **ONet**, without stating how they can be constructed, and under which conditions. A more explicit description of the conditions under which a rule can be applied to an open net and of the way the resulting net can be constructed, is clearly necessary for practical purposes. Looking at Fig. 8(a), given a rule p and a match $m : L_p \to Z$, in order to build the open net transformation:

- The *pushout complement* of l_p and m must exist. The resulting arrows n and d must be such that l_p and n are composable. A necessary condition for the existence of the pushout complement is a sort of *dangling condition*: a place can be deleted only if all the transitions connected to this place are removed as well, otherwise the flow arcs of this transition would remain dangling. This ensures that the pushout complement exists and is unique in the underlying category **Net**, but, as discussed below, it is not sufficient, in general, to conclude the existence of the pushout complement in **ONet**.

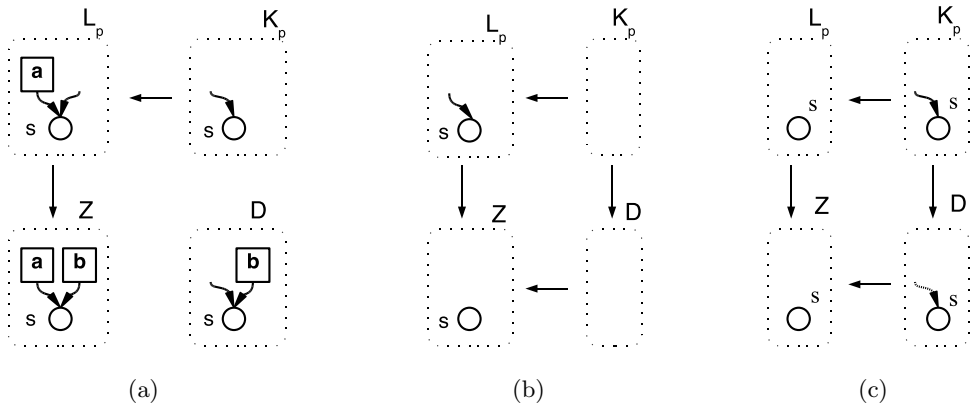


Figure 9: (a),(b) A pushout complement in **Net** which cannot be lifted to **ONet** and (c) A situation in which the pushout complement is not unique in **ONet**.

Additionally, there can be several pushout complements and in this case a canonical choice should be considered.

- The resulting arrow n must be composable with r_p: then we know how to build Z' by Proposition 2.3.

Unfortunately, although a general theory of DPO rewriting has been developed recently in the framework of adhesive categories [19], we cannot exploit it here since the category of open nets falls outside the scope of the theory.

Next we analyse the conditions which ensure the applicability of open net rules. We will first consider the case of general, possibly non-behaviour preserving rules. Then we will instantiate the developed theory to the setting of behaviour preserving rules, which turns out to be simpler and more intuitive. The reader which is not interested in the general case can safely skip it.

6.2.1. *Applying General Rules.* In this section we develop general results concerning the applicability of a rewriting rule to an open net. Given an open net Z, a rule p and a match $m : L_p \to Z$, we first focus on the existence of the pushout complement in **ONet**. As mentioned above, a first necessary condition is a sort of *dangling condition*, which, however, in general, is not sufficient. Consider, for instance, the diagram in Fig. 9(a). It is easy to realise that the only place in D must be input open since an additional transition is attached to such place in Z. However, the resulting diagram is not a pushout in **ONet**: since the places in L_p and in D are input open also their image in Z should be input open. Similarly, the diagram Fig. 9(b) is not a pushout in **ONet**, although the underlying diagram is a pushout in **Net**, since place s of Z should be input open.

Moreover, in the case of general rules, the pushout complement in **ONet** might not be unique. In fact, whenever, as in Fig. 9(c), there is an open place in K_p whose image is not open in L_p (and thus neither in Z), then the corresponding place in D can be either open or not. For instance, the diagram in Fig. 9(c) admits two possible pushout complements consisting of an open net D with a single place s which can be or not input open.

Under additional requirements it is possible to prove the existence of a minimal pushout complement D, i.e., a pushout complement which embeds into any other and which is taken

as a canonical choice. Roughly, the minimal pushout complement is the maximally open one: whenever a place could be either open or not, it is taken to be open (in Fig. 9(c), this corresponds to take the pushout complements D with place s input open).

Lemma 6.4 (existence of the pushout complement)**.** *Let p be a rewriting rule over open nets, let Z be an open net and let $m : L_p \to Z$ be a match. Assume that*

(1) *for all places $s \in L_p - l_p(K_p)$ we have ${}^\bullet m(s), m(s)^\bullet \subseteq m(L_p - l_p(K_p))$;*
(2) *$m(l_p(\mathsf{in}(l_p)) \cap O^+_{L_p}) \subseteq O^+_Z$ and $m(l_p(\mathsf{out}(l_p)) \cap O^-_{L_p}) \subseteq O^-_Z$;*
(3) *$m(O^x_{L_p} - l_p(O^x_{K_p})) \subseteq O^x_Z$ for $x \in \{+, -\}$.*

Then the pushout complement exists in **Net***, defined as $D = Z - m(L_p - l_p(K_p))$, componentwise over the place and transition sets, and it can be lifted to a minimal pushout complement in* **ONet** *by taking as input open places:*

$$O^+_D = d^{-1}(O^+_Z) \cup n(O^+_{K_p} - O^+_{L_p})$$

Output open places are defined analogously. The initial marking $\hat{u}_D$ is defined by $\hat{u}_D(s) = \hat{u}_Z(d(s))$ for any place $s \in S_D$.

Proof. The proof is long, but straightforward. We have already motivated the dangling condition above. In order to understand condition 2, observe that, roughly, a place s of L_p is in $l_p(\mathsf{in}(l_p))$ if applying the rule p the place is preserved but at least one transition in ${}^\bullet s$ is removed. Since the rule deletes an input transition from $m(s)$ – the image of s in Z – the corresponding place in D belongs to $\mathsf{in}(d)$ and thus it must be input open. Therefore if s is open also in L_p, necessarily, by the construction of pushout in **ONet**, $m(s)$ must be open in Z. Similarly, for condition 3, if a place is open in L_p and it is not in the image of K_p then necessarily it will be open in Z.

Formally we have to show that (a) the mappings n and d are well-defined open net morphisms, (b) l_p and m are composable and (c) Z is the pushout. Minimality of the pushout complement then follows by construction.

(a.1) n is a well-defined open net morphism.
Let us prove that $n^{-1}(O^+_D) \cup \mathsf{in}(n) \subseteq O^+_K$ (the condition on output open places is analogous). If $s \in n^{-1}(O^+_D)$ we have two possibilities according to the way O^+_D is defined.
– If $n(s) \in d^{-1}(O^+_Z)$ then $d(n(s)) \in O^+_Z$. Since $d \circ n = m \circ l_p$ and $m \circ l_p$ is a well-defined open net morphism, we deduce that $s \in O^+_K$.
– If $n(s) \in n(O^+_K - O^+_L)$, since n is injective, we have that $s \in O^+_K - O^+_L \subseteq O^+_K$.
If instead $s \in \mathsf{in}(n)$ then $m(l_p(s)) \in \mathsf{in}(m \circ l_p)$. Since $m \circ l_p$ is an open net morphism, we conclude $s \in O^+_{K_p}$, as desired.

Concerning the initial marking, note that for any $s \in S_K$ we have $\hat{u}_K(s) = \hat{u}_Z(m(l_p(s)) = \hat{u}_D(d(n(s)) = \hat{u}_D(s)$, where the last equality holds by construction.

(a.2) d is a well-defined open net morphism.
Also in this case we only prove that $d^{-1}(O^+_Z) \cup \mathsf{in}(d) \subseteq O^+_D$ (the condition on output open places is analogous). If $s \in d^{-1}(O^+_Z)$ then $s \in O^+_D$ by definition. If, instead, $s \in \mathsf{in}(d)$ then it is easy to see that there exists $s' \in S_K$ such that $s' \in \mathsf{in}(l_p) \subseteq O^+_K$. Now, there are two subcases:

– If $l_p(s') \in O_L^+$ we have that $s' \in l_p(\mathsf{in}(l_p)) \cap O_L^+$ and thus $m(s') \in m(l_p(\mathsf{in}(l_p)) \cap O_L^+) \subseteq O_Z^+$ by condition 2. Since $d(s) = m(s')$ we deduce that $s \in d^{-1}(O_Z^+) \subseteq O_D^+$ by construction of D.
– If $l_p(s') \notin O_L^+$ then $s' \in O_K^+ - O_L^+$, and thus $n(s') \in n(O_K^+ - O_L^+) \subseteq O_D^+$, by construction of D.

The condition over the initial marking is trivially satisfied by construction.

(b) n and l_p are composable.
We show the two conditions for composability separately:
– $n(\mathsf{in}(l_p)) \subseteq O_D^+$
In fact, if $s \in \mathsf{in}(l_p)$, then it is easy to see that $m(l_p(s)) \in \mathsf{in}(d) \subseteq O_D^+$. Now, $m(l_p(s)) = d(n(s))$ and, since d is an open net morphism, it must reflect open places, and thus $n(s) \in O_D^+$.
– $l_p(\mathsf{in}(n)) \subseteq O_L^+$
If $s \in l_p(\mathsf{in}(n))$ then, it is easy to see that $s \in \mathsf{in}(m) \subseteq O_L^+$, as desired.

(c) Z is the pushout.
We know that Z is the pushout of n and l_p in **Net**. We have to prove that it is also a pushout in **ONet**.

Concerning the set of open places we have to show that

$$O_Z^x \supseteq \{s \in S_Z : m^{-1}(s) \subseteq O_L^x \ \wedge \ d^{-1}(s) \subseteq O_D^x\}.$$

Then the converse inclusion, and thus equality, follows from the fact that m and d are open net morphisms.

Let $s \in S_Z$ such that there are $s' \in O_L^+$ and $s'' \in O_D^+$ such that $m(s') = s = d(s'')$. Thus, there is $s''' \in S_K$ such that $l_p(s''') = s'$ and $n(s''') = s''$.

Since $s'' \in O_D^+$, then either $s'' \in d^{-1}(O_Z^+)$ or $s'' \in n(O_K^+ - O_L^+)$. Since $s' \in O_L^+$ and $l_p(s''') = s'$, the second possibility cannot arise. In the first case $s = d(s'') \in O_Z^+$, as desired.

When s is only in the image of D, the proof is analogous. When it is only in the image of L_P, we can use condition 3 in the hypothesis. □

Summarizing, condition 1 of Lemma 6.4 is a dangling condition. By the remaining conditions, if a place s in L_p is open, and the rule prescribes either the deletion of incoming/outgoing transitions from such place (condition 2) or the deletion of the place itself (condition 3), then the image of s in Z must be open. Examples of what fails when conditions 2 and 3 are violated can be found in Fig. 9(a) and 9(b).

It is worth observing that in the case of rules p such that morphism l_p preserves open places, i.e., $l_p(O_{K_p}^x) \subseteq O_{L_p}^x$ for $x \in \{+, -\}$, the above result ensures the existence of a unique pushout complement.

Given a match $m : L_p \to Z$ as in the proposition above, the transformation can be completed if $n : K_p \to D$ and $r_p : K_p \to R_p$ are composable. For this we need to suitably restrict matches.

Definition 6.5 (proper match)**.** Let p be a rewriting rule over open nets and let Z be an open net. A match $m : L_p \to Z$ is called *proper* if it satisfies conditions 1, 2, and 3 in Lemma 6.4 and

(4) for any $s \in K_p$, if $s \in \mathsf{in}(r_p) - \mathsf{in}(l_p)$ then $m(l_p(s)) \in O_Z^+$;
(5) $r_p(l_p^{-1}(\mathsf{in}(m))) \subseteq O_{R_p}^+$;

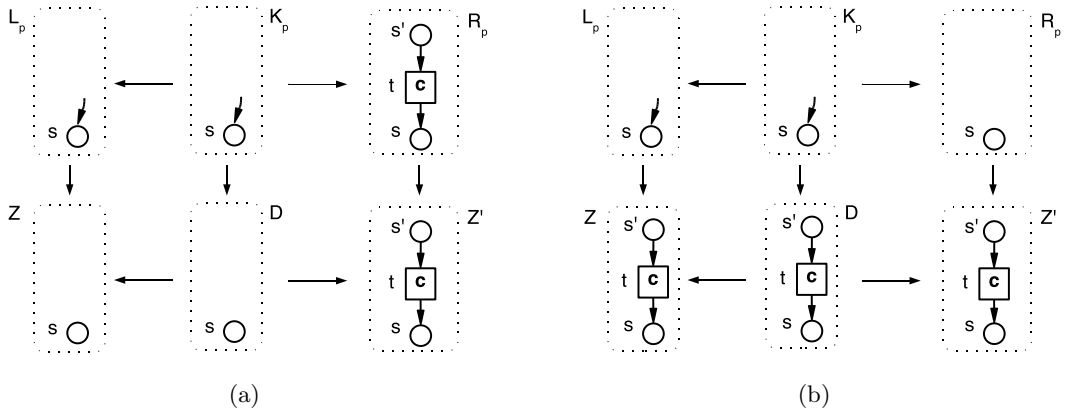


Figure 10: Examples of non-proper matches violating (a) condition 4 and (b) condition 5.

plus the dual conditions on output places.

Intuitively, a match is proper if whenever $s \in l_p(\mathsf{in}(r_p))$, i.e., the rule p creates a new (ingoing) transition connected to place s, then $m(s)$ is (input) open (condition 4). Additionally, input (output) places for the match which are preserved by the rule must be input (output) open in R_p. An example in which condition 4 is violated can be found in Fig. 10(a). For place s in K_p we have $s \in \mathsf{in}(r_p)$, since transition t is added in R_p, but $s \notin \mathsf{in}(l_p)$. Note that the mapping from D to Z' is not a valid open net morphism, since place s in D is not open. In Fig. 10(b) instead is condition 5 which is violated. Place s of L_p is in $\mathsf{in}(m)$, it is preserved by the rule, but the corresponding place in R_p is not open. Again we cannot complete the DPO step since the mapping from R_p to Z' is not a valid open net morphism (place s should be input open in R_p).

We finally arrive at the desired result.

Lemma 6.6 (applying general rules)**.** *Let p be a rule over open nets, let Z be an open net and let $m : L_p \to Z$ be a proper match. Then there exists a transformation $Z \Rightarrow^{p,m} Z'$.*

Proof. Let p be a rule over open nets, let Z be an open net and let $m : L_p \to Z$ be a proper match. Then, by using Lemma 6.4 we can construct the minimal pushout complement of l_p and m, as in Fig. 8(a).

In order to conclude, it suffices to show that n and r_p are composable. To this aim observe that by properness of the match:

- $n(\mathsf{in}(r_p)) \subseteq O_D^+$ (and the same condition holds for $\mathsf{out}(.)$)
  In fact, let $s \in \mathsf{in}(r_p)$ We distinguish two possibilities. If $s \in \mathsf{in}(l_p)$ then necessarily $n(s) \in \mathsf{in}(d)$ and thus $n(s) \in O_D^+$, since n is an open net morphism. If instead, $s \notin \mathsf{in}(l_p)$, then $s \in \mathsf{in}(r_p) - \mathsf{in}(l_p)$, hence, by condition 4 of Definition 6.5, $m(l_p(s)) \in O_Z^+$. Since $m(l_p(s)) = d(n(s))$ and d is an open net morphism, we conclude that also in this case $n(s) \in O_D^+$.
- $r_p(\mathsf{in}(n)) = r_p(l_p^{-1}(\mathsf{in}(m))) \subseteq O_{R_p}^+$ (and the same condition holds for $\mathsf{out}(.)$)
  Immediate by condition 5 of Definition 6.5. □

6.2.2. *Applying Behaviour Preserving Rules.* Sufficient hypotheses which ensure the applicability of behaviour preserving rules are made explicit in the following statement. This is a corollary of the general theory of transformations for open nets developed before.

Corollary 6.7 (applying behaviour preserving rules)**.** *Let p be a* x*-behaviour preserving rule, let Z be an open net and let $m : L_p \to Z$ be a match such that:*
a. for all $s \in L_p - l_p(K_p)$ we have ${}^\bullet m(s) \cup m(s)^\bullet \subseteq m(L_p - K_p)$;
b. for all $s \in K_p$, if $s \in \mathsf{in}(l_p)$ and $l_p(s) \in O^+_{L_p}$ then $m(l_p(s)) \in O^+_Z$;
c. for all $s \in K_p$, if $s \in \mathsf{in}(r_p) - \mathsf{in}(l_p)$ then $m(l_p(s)) \in O^+_Z$;
and the dual of the last two conditions, obtained by replacing in() *by* out() *and $+$ by $-$, hold. Then, there exists a transformation $Z \Rightarrow^{p,m} Z'$.*

Proof. This is an easy consequence of Lemma 6.6. We need to show that conditions (a)-(c) ensure that the match m is proper, i.e., it satisfies conditions 1–5 of Lemma 6.4 and Definition 6.5.

Condition 1 is the same as condition (a), condition 2 is just a compact notation for condition (b) and condition 4 is exactly condition (c). Concerning condition 3, observe that, since p is a behaviour preserving rule then $(r_p \circ l_p^{-1})_{|O_{L_p}}$ is a correspondence between the left- and right-hand side. This means that for any place s in $O^x_{L_p}$ there must be a place s' in K_p such that $l_p(s') = s$, and, by definition of open net morphism s' must be open, i.e., $s' \in O^x_{K_p}$. Therefore $O^x_{L_p} \subseteq l_p(O^x_{K_p})$ and thus condition 3 is trivially satisfied. Similarly, for condition 5, observe that, by definition of open net morphisms, $\mathsf{in}(m) \subseteq O^+_{L_p}$, and, thus

$$r_p(l_p^{-1}(\mathsf{in}(m))) \subseteq r_p(l_p^{-1}(O^+_{L_p})) = O^+_{R_P}.$$

The last equality is justified by the fact that p is behaviour preserving, and thus, as observed above, $(r_p \circ l_p^{-1})_{|O_{L_p}}$ is a correspondence between L_p and R_p. □

The intuition underlying the conditions above is the following. Condition (a) is a typical *dangling condition*, which we have already commented. Condition (b) says that if $s \in \mathsf{in}(l_p)$, i.e., if some (ingoing) transitions are deleted from s then the image of s in Z must be (input) open if so is its image in L_p. Finally, by condition (c), if $s \in \mathsf{in}(r_p) - \mathsf{in}(l_p)$, i.e., the rule p creates a new (ingoing) transition connected to place s, without replacing any old one, then the image of s in Z must be (input) open.

As an example, consider again the DPO diagram in Fig. 8(b). It is not difficult to see that the rule and the match satisfy the conditions of Corollary 6.7. Hence we can complete the double-pushout construction transforming Z into Z', as depicted in the same figure.

6.3. **Modeling Dynamic Reconfigurations of Services.**

Open nets allow us to specify a system as built out of smaller components. Then, its behaviour is captured by the firing or step behaviour of the open net. However, for highly dynamic systems, as mentioned in the introduction, it can be useful to have the possibility of specifying that, under suitable conditions, some structural changes or reconfigurations of the system can take place. For instance the invocation of a service could trigger a rule which provides an implementation of the required service.

The theory of open net reconfigurations can do the job. As an example, consider net Z_0 in Fig. 12 which models the view of a traveller on the journey planning and ticket purchase services offered through a travel agency portal.

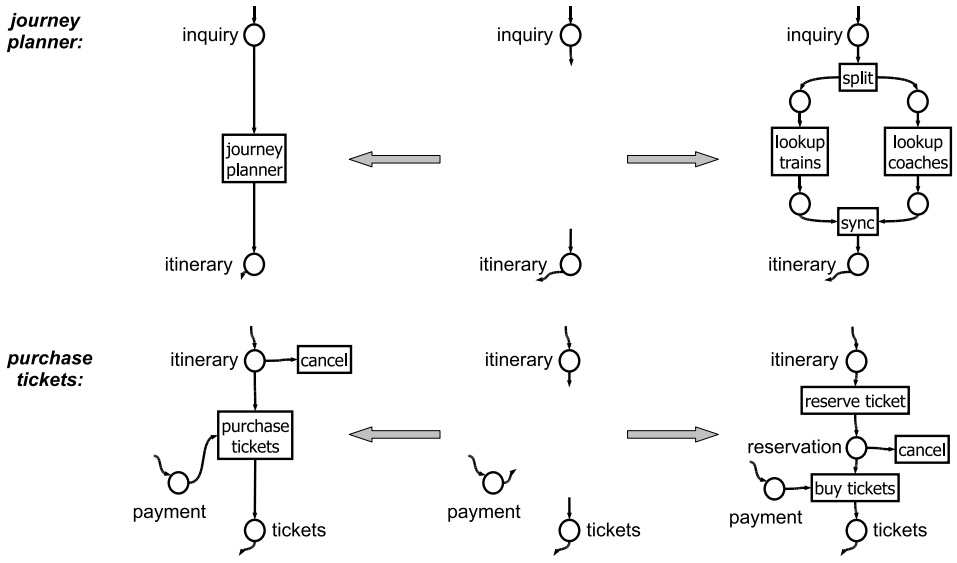


Figure 11: Rules

We distinguish *abstract transitions* representing services that should be provided elsewhere and *concrete transitions* representing local services and control flow actions. The invocation of an external service can be seen at different levels of abstraction. From the point of view of the client process it is just the firing of an abstract transition. At a lower level of abstraction, it is captured by a rule such as the one at the top of Fig. 11. An application of this rule, replacing the abstract transition by a new open net, models the discovery and binding of the concrete services required. The left- and right-hand sides of the rule are weakly firing (actually, also step) bisimilar if we observe only the interactions at the open (interface) places, i.e., if we take $\Lambda_\tau = \Lambda$. This can be seen as a proof of the fact that the bound service meets the requirements: both in the abstract transition and in its concrete counterpart any inquiry will produce a corresponding itinerary.

The rule at the bottom of Fig. 11 represents a case where a simple pattern is replaced by a richer one. On the left we say that, given an itinerary, we can either purchase the required tickets or cancel the processes. On the right the transaction is refined, adding a prior reservation phase, while keeping the option to cancel. As above, the rule has weakly firing (and step) bisimilar left- and right-hand sides, ensuring that the visible effect of the abstract and concrete transitions at the interfaces is the same.

A possible sequence of transformations is shown in Fig. 12. By Theorem 6.3, we are sure that the transformations do not change the observable behaviour of the system, i.e., the start and end nets are weakly bisimilar, a fact that can be interpreted as a proof of conformance of the provided service with respect to the abstract specification.

We have shown only a small example application, however, we believe that this technique can be applied to larger case studies, such as the banking scenario studied in [12]. In order to do this automatically, it would be necessary to implement mechanized bisimulation checking procedures. For finite state spaces, this is quite straightforward, for infinite state

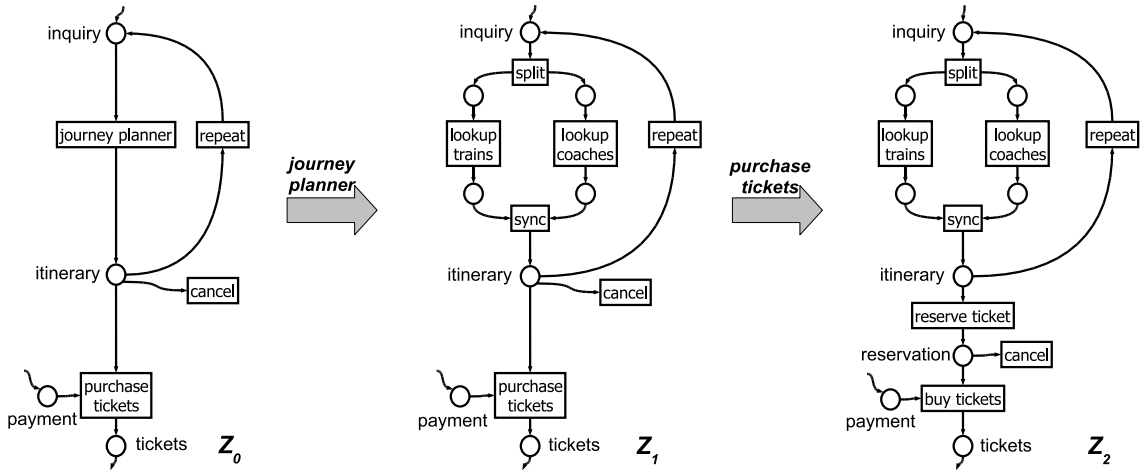


Figure 12: Transformation of open nets representing a travel agent's portal.

spaces we could resort to the techniques presented in [15]. In any case the up-to technique presented in Section 5 will be very useful for practical case studies.

Another relevant question is the following: which kind of bisimilarity should be used? While strong firing bisimilarity is conceptually the simplest behavioural equivalence, practical examples usually require weak bisimulations in order to abstract from internal or silent moves. Finally, step bisimulation is able to distinguish processes that differ with respect to the degree of concurrency. This can be relevant if the observer is able to distinguish different degrees of parallelism or if we take into account efficiency questions.

7. Conclusions and Related Work

Open nets, introduced in [3, 4], are a reactive extension of standard Petri nets which allows to model systems interacting with an unspecified environment.

As mentioned in the introduction there is a vast related literature. A close conceptual relationship exists with the early studies on modular construction and refinement techniques (see, e.g., [37, 36, 28, 41]) and on composition operators and compositional semantics for Petri nets (see, e.g., [1, 9, 6, 46]). The last class comprises also the algebraic approaches to Petri nets which view the class of Petri nets as a category and, characterising the semantics of interest as a universal constructions, automatically deduce the compositionality for suitably defined operators [47, 48, 24].

More recent approaches, which focus more explicitly on the definition of notion of module and interface and where the reactive aspects are taken into account in the semantics can be classified roughly into two classes. Some approaches aim at defining a "calculus of nets", where a set of process algebra-like operators allow one to build complex nets starting from a set of predefined basic components. In this family, the papers [29, 33] propose an algebra of (labelled) Petri nets with interfaces, consisting of public (input) places and (output) transitions, with operators which allow e.g., to add new transitions and places, to connect existing public transitions and places by new arcs, to hide items in the net. We also recall the Petri Box calculus [10, 18, 17], where a special class of safe nets, called *plain boxes*,

provides the basic components, which are then combined by means of (refinement-based) composition operators. Another family of approaches can be classified as "component-oriented": the emphasis, rather than on the algebraic aspects, is put on the mechanisms which allow one to build larger systems by combining nets with clearly identified interfaces. For instance the book [43] proposes a technique for inserting a net, called daughter net, into a so-called host net. The composition is realised by joining the two nets along a predefined set of places, playing the role of open places. The distinction between input and output open places, absent in [43], instead is later considered in [45]. A compositionality result is proved for language equivalence and a notion of bisimilarity, very close to ours, is defined. Interestingly, the same book also focuses on an alternative approach to net composition, based on an operation of synchronised parallel product in the style of [48]. Such operation, roughly speaking, joins two nets by forcing the synchronisation of transitions with the same label. Other members of the "component-oriented" family are, for example, the *Petri net components* [16] and the nets with *pins* [5]. We also recall *workflow nets* [38] which have been proposed as a formal model for the description of workflows, i.e., business processes specified in terms of tasks and shared resources. Workflow nets are special Petri nets satisfying suitable conditions, like the existence of one initial and one final place: tokens in such places characterise the start and the end, respectively, of the represented process. The model has been extended for the specification of *interorganisational workflows* [39], represented as a set of workflow nets connected through additional places for asynchronous communication and synchronisation requirements on transitions. Additional references, as well as a detailed comparison between the approaches to Petri net composition and reactivity just cited and the open net model can be found in [4].

In this paper, firstly we have generalised the theory of open nets, including the characterisation of net composition using pushouts, to the case of marked nets. Next we have introduced several natural notions of bisimilarity over open nets, showing that weak bisimilarities, arising in the presence of unobservable actions, and, as a particular case, also strong bisimilarities are congruences with respect to the colimit-based composition operation over open nets. The considered notions of bisimilarity differ for the choice of the observations. These can be single firings, thus leading to what we called firing bisimilarity, a standard notion of interleaving equivalence, capable of capturing the branching structure of computations. Alternatively, we can observe parallel steps, thus obtaining step bisimilarity, which allows to capture, to some extent, the degree of parallelism that is possible in a component. This can be useful, e.g., when a component is replaced by another one since we might be interested in taking a replacement that exhibits at least the same concurrent behaviour and is hence equally efficient.

In recent years, reactive extensions of Petri nets have been obtained by exploiting a general theory of reactive systems developed for automatically deriving bisimulation congruences. Specifically, an encoding of Petri nets as bigraphical reactive systems has been proposed in [27], while [35] proposes an encoding of nets as reactive systems in the cospan category over an adhesive category. Our results about strong firing bisimilarity can be seen as a generalisation of those in [27, 35], which essentially are developed for a special kind of open nets, where there is no distinction between input and output open places. Furthermore the composition operation used in the cited papers does not allow synchronisation of transitions (technically, the interface net does not contain transitions).

Concerning weak step bisimilarity, some connections seem to exist with the work on action refinement, which goes back to [37]. For example, in [44] (weak) step bisimilarity is

shown to be a congruence with respect to a refinement operation which allows to replace a single event with a deterministic finite event structure. Although the setting is different and a direct comparison is not possible, we observe that, compared to refinement-based approaches, where single transitions are refined by a subnet, the theory presented here works for general reconfigurations, in which both the left- and right-hand sides can be general, arbitrarily large nets.

Weak (step) bisimilarity for Petri nets is studied also in [29]. They observe that such an equivalence is not a congruence in general, but for Petri nets satisfying a suitable condition on the labelling of the public transitions (*well-labelled* nets), a context closure allows one to get a congruence which is then characterised by means of a universal context. The setting is different from ours since the issue of net composition is tackled at a finer level of granularity: the basic components of a net are assumed to be transitions with empty pre- and post-set and single places, which are then combined by means of constructors that allow one to connect places and transitions. Still it would be interesting to understand if a formal relation can be established, e.g., trying to internalise the pushout-based composition operation in the algebra of connectors of [29].

Similarities exist also with the problem studied in [11], where a reactive Petri net model which admits a compositional behavioural equivalence is exploited, in the framework of web-services, to provide a theoretical basis to service composition and discovery. This technique is then used in a case study for checking the correctness of service specifications and the replaceability of services in a banking scenario [12]. Disregarding the technical differences, such as the fact that the mentioned paper deals with C/E nets and the use of read arcs, the kind of nets of interest for this approach are essentially a subclass of open Petri nets, satisfying some structural requirements (all labels are invisible and the interface consists of a single input and a single output place, plus some read places). Generally speaking, compositional Petri net models appears to be promising as a formalism for the specification of control and composition in service oriented architectures as suggested, e.g., in [8, 22, 40, 23]. Investigating possible applications of (reconfigurable) open Petri nets, along the lines of the presented example, in the setting of web-service specification and analysis represent a stimulating direction of future research.

In the second part of the paper we have proposed a rewriting-based framework for Petri nets with reconfigurations. We have shown how our congruence results can be used to identify classes of reconfigurations which do not alter the observational behaviour of the system. This is applied to a small case study of a workflow-like model of a travel agency, where we showed how abstract services can be replaced by more concrete implementations and how we can ensure that the behaviour of the full net is preserved under such operations.

Action refinement of Petri nets (see, e.g., [37, 36, 28, 41]), that we already mentioned above, can be seen as a special form of reconfiguration. The idea of using rewriting techniques for providing a reconfiguration mechanism for Petri nets has been already explored in the literature (see, e.g., reconfigurable nets of [2, 21] and high-level replacement systems applied to Petri nets in [31]). In this approaches, however, the emphasis is more on rewriting as a computational mechanism, rather than on the study of the way the behaviour of the system is affected by the reconfigurations. In future work, besides deepening the relationships between these approaches and ours, we will continue studying the notion of reconfigurable open nets and describe in more detail how reconfigurations can be triggered by the net itself, for example by reaching certain markings or by firing certain transitions, following an intuition similar to that of dynamic nets [13].

Finally, it would be worth studying whether a formal duality can be established between our morphisms and standard simulation morphisms for Petri nets. Viewing our morphisms as inverses of (partial) simulation morphisms would allow to get a precise correspondence between our pushout-based composition and pullback-based synchronisation of Petri nets. Surely by simply taking Winskel's morphisms [47] this does not work (technically because when they are undefined on a transition they must be undefined on the corresponding pre- and post-set). Also more general morphisms for Petri nets, like those proposed in [42, 7], would not provide an immediate solution. Still, it looks feasible to identify generalisations of such morphisms to the context of open Petri nets allowing to develop a dual theory based on simulations.

Acknowledgement: We would like to thank the referees for their insightful and detailed comments.

References

[1] C. André. The behaviour of a Petri net on a subset of transitions. *RAIRO*, 17:5–21, 1983.

[2] E. Badouel, M. Llorens, and J. Oliver. Modeling concurrent systems: Reconfigurable nets. In H. R. Arabnia and Y. Mun, editors, *Proceedings of PDPTA'03*, volume 4, pages 1568–1574. CSREA Press, 2003.

[3] P. Baldan, A. Corradini, H. Ehrig, and R. Heckel. Compositional modeling of reactive systems using open nets. In K.G. Larsen and M. Nielsen, editors, *Proceedings of CONCUR'01*, volume 2154 of *LNCS*, pages 502–518. Springer Verlag, 2001.

[4] P. Baldan, A. Corradini, H. Ehrig, and R. Heckel. Compositional semantics for open Petri nets based on deterministic processes. *MSCS*, 15(1):1–35, 2005.

[5] T. Basten. *In terms of nets: System design with Petri nets and process algebra*. PhD thesis, Eindhoven University of Technology, 1998.

[6] B. Baumgarten. On internal and external characterisation of PT-net building block behaviour. In G. Rozenberg, editor, *Advances in Petri nets*, volume 340 of *LNCS*, pages 44–61. Springer, 1988.

[7] M. A. Bednarczyk and A. M. Borzyszkowski. General morphisms of Petri nets (extended abstract). In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *Proceedings of ICALP'99*, volume 1644, pages 190–199. Springer Verlag, 1999.

[8] B. Benatallah and R. Hamadi. A Petri net-based model for Web service composition. In K.-D. Schewe and X. Zhou, editors, *Australasian Database Conference, Conferences in Research and Practice in Information Technology*, volume 7, pages 191–200. Australian Computer Society, 2003.

[9] G. Berthelot. Transformations and decompositions of nets. In W. Brauer, editor, *Petri Nets: Central models and their properties*, volume 254 of *LNCS*, pages 359–376. Springer, 1987.

[10] E. Best, R. Devillers, and J. G. Hall. The Petri box calculus: a new causal algebra with multi-label communication. In G. Rozenberg, editor, *Advances in Petri Nets*, volume 609 of *LNCS*, pages 21–69. Springer Verlag, 1992.

[11] F. Bonchi, A. Brogi, S. Corfini, and F. Gadducci. A behavioural congruence for web services. In *Proceedings of FSEN '07*, volume 4727 of *LNCS*, pages 240–256. Springer, 2007.

[12] F. Bonchi, A. Brogi, S. Corfini, and F. Gadducci. Compositional specification of web services via behavioural equivalence of nets: A case study. In *Proc. of Petri Nets '08*, pages 52–71, 2008.

[13] M.G. Buscemi and V. Sassone. High-level Petri nets as type theories in the join calculus. In *Proceedings of FoSSaCS'01*, volume 2030 of *LNCS*, pages 104–120. Springer, 2001.

[14] H. Ehrig. Tutorial introduction to the algebraic approach of graph-grammars. In H. Ehrig, M. Nagl, G. Rozenberg, and A. Rosenfeld, editors, *Proceedings of the 3rd International Workshop on Graph-Grammars and Their Application to Computer Science*, volume 291 of *LNCS*, pages 3–14. Springer Verlag, 1987.

[15] D. Hirschkoff. Automatically proving up to bisimulation. In *Proc. of MFCS '98 Workshop on Concurrency*, number 18 in ENTCS, 1998.

[16] E. Kindler. A compositional partial order semantics for Petri net components. In P. Azema and G. Balbo, editors, *Application and Theory of Petri Nets*, volume 1248 of *LNCS*, pages 235–252. Springer Verlag, 1997.
[17] M. Koutny and E. Best. Operational and denotational semantics for the box algebra. *Theoretical Computer Science*, 211(1–2):1–83, 1999.
[18] M. Koutny, J. Esparza, and E. Best. Operational semantics for the Petri box calculus. In B. Jonsson and J. Parrow, editors, *Proceedings of CONCUR '94*, volume 836 of *LNCS*, pages 210–225. Springer Verlag, 1994.
[19] S. Lack and P. Sobociński. Adhesive and quasiadhesive categories. *RAIRO – Theoretical Informatics and Applications*, 39(3):511–555, 2005.
[20] J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In C. Palamidessi, editor, *Proceedings of CONCUR'00*, volume 1877 of *LNCS*, pages 243–258. Springer Verlag, 2000.
[21] M. Llorens and J. Oliver. Introducing structural dynamic changes in Petri nets: Marked-controlled reconfigurable nets. In F. Wang, editor, *Proceedings of ATVA'04*, volume 3299, pages 310–323. Springer Verlag, 2004.
[22] A. Martens. Analyzing Web service based business processes. In M. Cerioli, editor, *Proceedings of FASE'05*, volume 3442 of *LNCS*, pages 19–33. Springer, 2005.
[23] P. Massuthe, W. Reisig, and K. Schmidt. An Operating Guideline Approach to the SOA. *Annals of Mathematics, Computing & Teleinformatics*, 1(3):35–43, 2005.
[24] J. Meseguer and U. Montanari. Petri nets are monoids. *Information and Computation*, 88:105–155, 1990.
[25] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer Verlag, 1980.
[26] R. Milner. Bigraphical reactive systems. In K. G. Larsen and M. Nielsen, editors, *Proceedings of CONCUR'01*, volume 2154 of *LNCS*, pages 16–35. Springer Verlag, 2001.
[27] R. Milner. Bigraphs for Petri nets. In J. Desel, W. Reisig, and G. Rozenberg, editors, *Lectures on Concurrency and Petri Nets*, volume 3098 of *LNCS*, pages 686–701. Springer, 2003.
[28] K. Müller. Constructable Petri nets. *Elektr. Inf. Kybern.*, 21:171–199, 1985.
[29] M. Nielsen, L. Priese, and V. Sassone. Characterizing Behavioural Congruences for Petri Nets. In *Proceedings of CONCUR'95*, volume 962 of *LNCS*, pages 175–189. Springer Verlag, 1995.
[30] M. Nielsen and P. S. Thiagarajan. Degrees of non-determinism and concurrency: A Petri net view. In M. Joseph and R. Shyamasunda, editors, *Proceedings of FSTTCS'84*, pages 89–117. Springer Verlag, 1984.
[31] J. Padberg, H. Ehrig, and L. Ribeiro. High level replacement systems applied to algebraic high level net transformation systems. *Mathematical Structures in Computer Science*, 5(2):217–256, 1995.
[32] B. Pierce. *Basic Category Theory for Computer Scientists (Foundations of Computing)*. MIT Press, 1991.
[33] L. Priese and H. Wimmel. A uniform approach to true-concurrency and interleaving semantics for Petri nets. *Theoretical Computer Science*, 206(1–2):219–256, 1998.
[34] W. Reisig. *Petri Nets: An Introduction*. EATCS Monographs on Theoretical Computer Science. Springer Verlag, 1985.
[35] V. Sassone and P. Sobocinski. A congruence for Petri nets. In *Proceedings of PNGT'04*, volume 127(2) of *Electronic Notes in Theoretical Computer Science*, pages 107–120. Elsevier Science, 2005.
[36] I. Suzuki and T. Murata. A method for stepwise refinement and abstraction of Petri nets. *Journal of computer and system sciences*, 27:51–76, 1983.
[37] R. Valette. Analysis of Petri nets by stepwise refinements. *Journal of Computer and System Sciences*, 18(1):35–46, 1979.
[38] W. van der Aalst. The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
[39] W. van der Aalst. Interorganizational workflows: An approach based on message sequence charts and Petri nets. *System Analysis and Modeling*, 34(3):335–367, 1999.
[40] W.M.P. van der Aalst and A.H.M. ter Hofstede. Yawl: yet another workflow language. *Information Systems*, 30(4):245–275, 2005.
[41] W. Vogler. Behaviour preserving refinement of Petri nets. In G. Tinhofer and G. Schmidt, editors, *Proceedings of WG'86: Graph theoretic concepts in computer sciencea*, volume 246 of *LNCS*, pages 82–93. Springer, 1987.

[42] W. Vogler. Executions: A new partial-order semantics of Petri nets. *Theoretical Computer Science*, 91(2):205–238, 1991.
[43] W. Vogler. *Modular Construction and Partial Order Semantics of Petri Nets*, volume 625 of *LNCS*. Springer, 1992.
[44] W. Vogler. Bisimulation and action refinement. *Theoretical Computer Science*, 114(1):173–200, 1993.
[45] W. Vogler. Efficiency of asynchronous systems that communicate asynchronously. In *Proc. of ICATPN '00*, volume 1825 of *LNCS*, pages 424–444, 2000.
[46] K. Voss. Interface as a basic concept for system specification. In K. Voss, editor, *Concurrency and Nets*, pages 585–604. Springer, 1987.
[47] G. Winskel. Event Structures. In *Petri Nets: Applications and Relationships to Other Models of Concurrency*, volume 255 of *LNCS*, pages 325–392. Springer Verlag, 1987.
[48] G. Winskel. Petri nets, algebras, morphisms, and compositionality. *Information and Computation*, 72(3):197–238, 1987.