Math. Struct. in Comp. Science (2005), *vol.* 15, *pp.* 1–35.
DOI: 10.1017/S0960129504004311 Printed in the United Kingdom

Compositional semantics for open Petri nets based on deterministic processes[†]

PAOLO BALDAN[‡], ANDREA CORRADINI[§],
HARTMUT EHRIG[¶] and REIKO HECKEL[‖]

[‡]*Dipartimento di Informatica, Università Ca' Foscari di Venezia, Italy*
[§]*Dipartimento di Informatica, Università di Pisa, Italy*
[¶]*Computer Science Department, Technical University of Berlin, Germany*
[‖]*Dept. of Math. and Comp. Science, University of Paderborn, Germany*

Received 10 June 2002; revised 22 July 2003

In order to model the behaviour of open concurrent systems by means of Petri nets, we introduce *open Petri nets*, a generalisation of the ordinary model where some places, designated as *open*, represent an interface between the system and the environment. Besides generalising the token game to reflect this extension, we define a truly concurrent semantics for open nets by extending the Goltz–Reisig process semantics of Petri nets.
We introduce a composition operation over open nets, characterised as a pushout in the corresponding category, suitable for modelling both interaction through open places and synchronisation of transitions. The deterministic process semantics is shown to be compositional with respect to such a composition operation. If a net Z_3 results as the composition of two nets Z_1 and Z_2, having a common subnet Z_0, then any two deterministic processes of Z_1 and Z_2 that 'agree' on the common part, can be 'amalgamated' to produce a deterministic process of Z_3. Conversely, any deterministic process of Z_3 can be decomposed into processes of the component nets. The amalgamation and decomposition operations are shown to be inverse to each other, leading to a bijective correspondence between the deterministic processes of Z_3 and the pair of deterministic processes of Z_1 and Z_2 that agree on the common subnet Z_0. Technically, our result is similar to the amalgamation theorem for data-types in the framework of algebraic specification. A possible application field of the proposed constructions and results is the modelling of interorganisational workflows, recently studied in the literature. This is illustrated by a running example.

1. Introduction

Petri nets (Reisig 1985) are a basic model of concurrent and distributed systems. Because of their intuitive graphical representation, Petri nets are widely used both in theoretical and applied research to specify and visualise the behaviour of systems. One important

[†] This research was partially supported by the EC TMR Network GETGRATS (General Theory of Graph Transformation Systems), by the ESPRIT Working Group APPLIGRAPH (Applications of Graph Transformation), and by the MURST project TOSCA (Teoria della Concorrenza, Linguaggi di Ordine Superiore e Strutture di Tipi).

feature of Petri nets, especially when explaining the concurrent behaviour of a net to non-experts, is the possibility of describing their execution within the same visual notation, that is, in terms of processes (Golz and Reisig 1983).

However, when modelling *reactive systems*, that is, concurrent systems with interacting subsystems, Petri nets force us to take a global perspective. In fact, ordinary Petri nets are not adequate for modelling *open* systems, which can interact with their environment or, taking a different viewpoint, which are only partially specified. This makes it difficult to specify a large system as the composition of smaller components, which is a common practice, for example, in software engineering.

We can explain this problem in more detail by means of a typical application of Petri nets, the specification of workflows. A *workflow* describes a business process in terms of tasks and shared resources. Such descriptions are needed, for example, when interoperability of the workflows of different organisations is an issue, which is frequently the case, for example, when applications of different enterprises are to be integrated over the Internet. A *workflow net* (van der Aalst 1998) is a Petri net satisfying some structural constraints, such as the existence of one initial and one final place, and a corresponding *soundness condition*: from each marking reachable from the initial one (one token on the initial place) we can reach the final marking (one token on the final place). An *interorganisational workflow* (van der Aalst 1999) is modelled as a set of such workflow nets connected through additional places for asynchronous communication and synchronisation requirements on transitions.

For instance, Figure 1 shows an interorganisational workflow consisting of two local workflow nets Traveller and Agency related through communication places can, ack, bill, payment and ticket and a synchronisation requirement between the two reserve transitions, modelled by a dashed line. The example describes the booking of a flight by a traveller in cooperation with a travel agency. After some initial negotiations (which are not modelled), both sides synchronise in the reservation of a flight. Then, the traveller may either acknowledge or cancel and re-enter the initial state. In both cases an asynchronous notification (for example, a fax), modelled by the places ack and can, respectively, is sent to the travel agency. Next the local workflow of the traveller forks into two concurrent threads, the booking of a hotel and the payment of the bill. The trip can start when both tasks are completed and the ticket has been provided by the travel agency.

The overall net in Figure 1 describes the system from a global perspective. Hence, the classical notion of behaviour (described, for example, in terms of processes) is completely adequate. However, for a local subnet in isolation (like Traveller), which will only exhibit a meaningful behaviour when interacting with other subnets, this semantics is not appropriate because it does not take into account the possible interactions.

In order to overcome these limitations of ordinary Petri nets, we extend the basic model by introducing *open nets*. An open net is a P/T Petri net with a distinguished set of places, called *open*, which are intended to represent the interface of the net towards the external world. Some similarities exist with other approaches to net composition, like the *Petri box calculus* (Best *et al.* 1992; Koutny *et al.* 1994; Koutny and Best 1999), *Petri nets with interface* (Nielsen *et al.* 1995; Priese and Wimmel 1998) and *Petri net components* (Kindler 1997), which will be discussed later. As a consequence of the (hidden, implicit) interaction

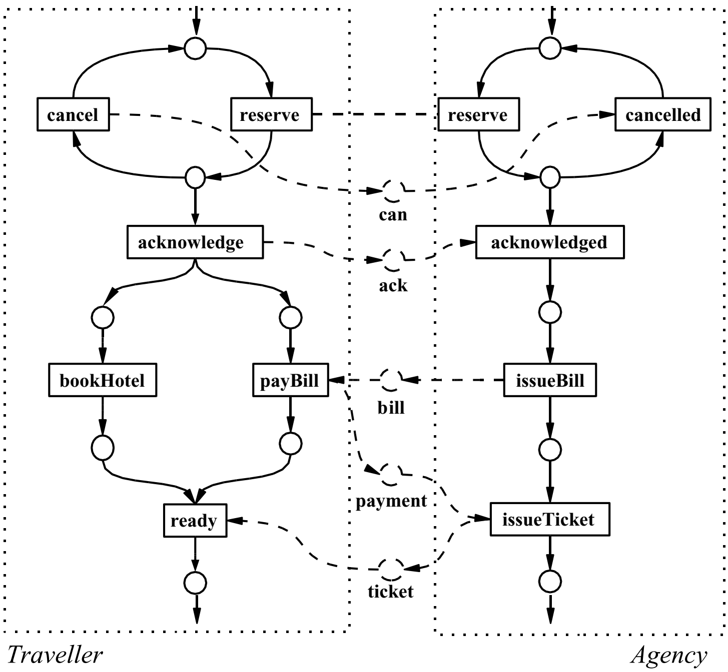


Fig. 1. Sample net modelling an interorganisational workflow.

between the net and the environment, some tokens can 'freely' appear in or disappear from the open places: this will be formalised by generalising the token game. Then we will provide a truly concurrent semantics that extends the ordinary (deterministic) *process semantics* (Golz and Reisig 1983) to open nets.

The embedding of an open net in a context is formally described by an injective morphism in a suitable category of open nets. Intuitively, in the target net new transitions can be attached to open places and, moreover, the interface towards the environment can be reduced by 'closing' open places. Therefore, open net morphisms do not preserve but reflect the behaviour, that is, any computation of the target (larger) net can be projected back to a computation of the source (smaller) net.

A *composition operation* is introduced over open nets. Two open nets Z_1 and Z_2 can be composed by specifying a common subnet Z_0 that embeds both in Z_1 and in Z_2. Then the two nets can be glued along the common part. This is permitted only if the prescribed composition is consistent with the interfaces, that is, only if the places of Z_1 and Z_2 that are used when connecting the two nets are actually open. The composition operation is characterised as a pushout in the category of open nets, where the conditions for the existence of the pushout fit nicely with the mentioned condition over interfaces.

Based on these concepts, the representation of the system of Figure 1 in terms of two interacting open nets is given by the top part of Figure 2, which comprises the two component nets Traveller and Agency, and the net Common that embeds into

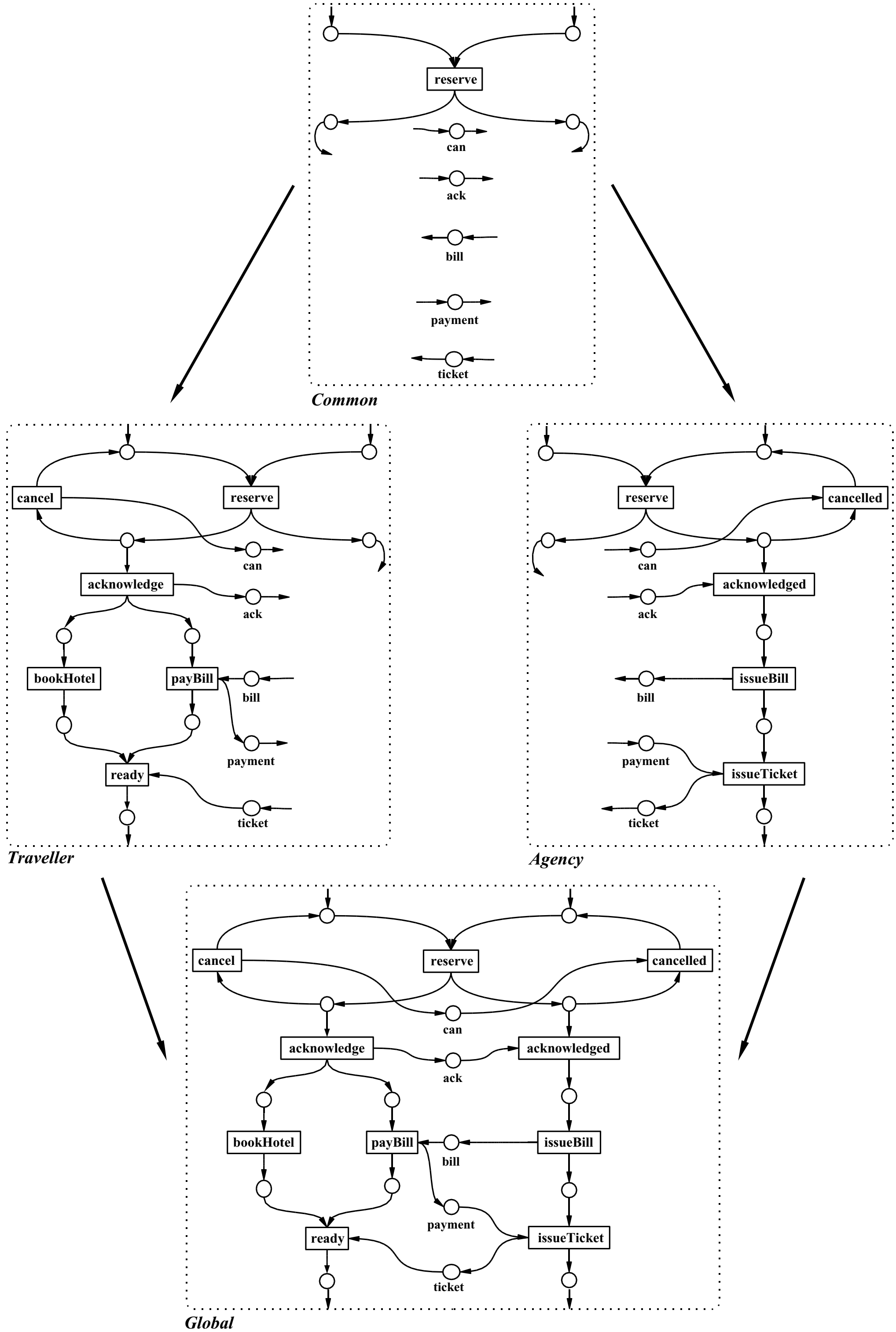


Fig. 2. Interorganisational workflow as the composition of open nets Traveller and Agency.

both components by means of open net morphisms. Places with incoming/outgoing dangling arcs are open. Observe that the common subnet Common of the components Traveller and Agency closely corresponds to the dashed items of Figure 1, which represent the 'glue' between the two components. The net resulting from the composition of Traveller and Agency over the shared subnet Common is shown in the bottom part of Figure 2.

Obviously, one would like to be able to establish a clear relationship between the behaviours of the component nets (in the example, the nets Traveller and Agency) and the behaviour of the composition (in the example, the net Global). We will show that indeed, the behaviour of the latter can be constructed 'compositionally' out of the behaviours of the former, in the sense that two deterministic processes of the component nets that 'agree' on the shared part, can be synchronised to produce a deterministic process of the composed net. Conversely, *any* deterministic process of the global net can be decomposed into deterministic processes of the component nets, which, in turn, can be synchronised to give the original process again. The top part of Figure 3 shows two processes of the nets Traveller and Agency, the corresponding common projections over net Common and the process of Global arising from their synchronisation.

The synchronisation of processes, based on the composition of their underlying nets, resembles the *amalgamation* of data-types in the framework of algebraic specifications, and therefore we will speak of the *amalgamation of processes.* By analogy with the amalgamation theorem for algebraic specifications (Ehrig and Mahr 1985), the main result of this paper shows that the amalgamation and decomposition constructions mentioned above are inverse to each other, establishing a bijective correspondence between the pairs of processes of two nets that agree on a common subnet and the processes of the net resulting from their composition.

The rest of the paper is organised as follows. Section 2 introduces the open Petri net model and the corresponding category. Section 3 extends the notion of process from ordinary to open nets and defines the operation of behaviour projection. Section 4 introduces the composition operation for open nets, based on a pushout in the category of open nets. Section 5 presents the compositionality result of the process semantics of open nets. Finally, Section 6 discusses some related work in the literature and Section 7 draws some conclusions and outlines possible directions for future investigation. An extended abstract of this paper has been published as Baldan *et al.* (2001).

2. Open nets

An *open net* is an ordinary P/T Petri net with a distinguished set of 'open' places that are intended to represent the interface of the net towards the external world (environment). As a consequence of the (hidden, implicit) interaction between the net and the environment, some tokens can freely appear in and disappear from the open places. Concretely, an open place can be either an *input* or an *output* place (or both), meaning that the environment can put or remove tokens from that place.

Given a set X we use $X^{\oplus}$ to denote the free commutative monoid generated by X, and $\mathbf{2}^X$ to denote its powerset. Given $A \in X^{\oplus}$ and $x \in X$, we will write $x \in A$ to mean that

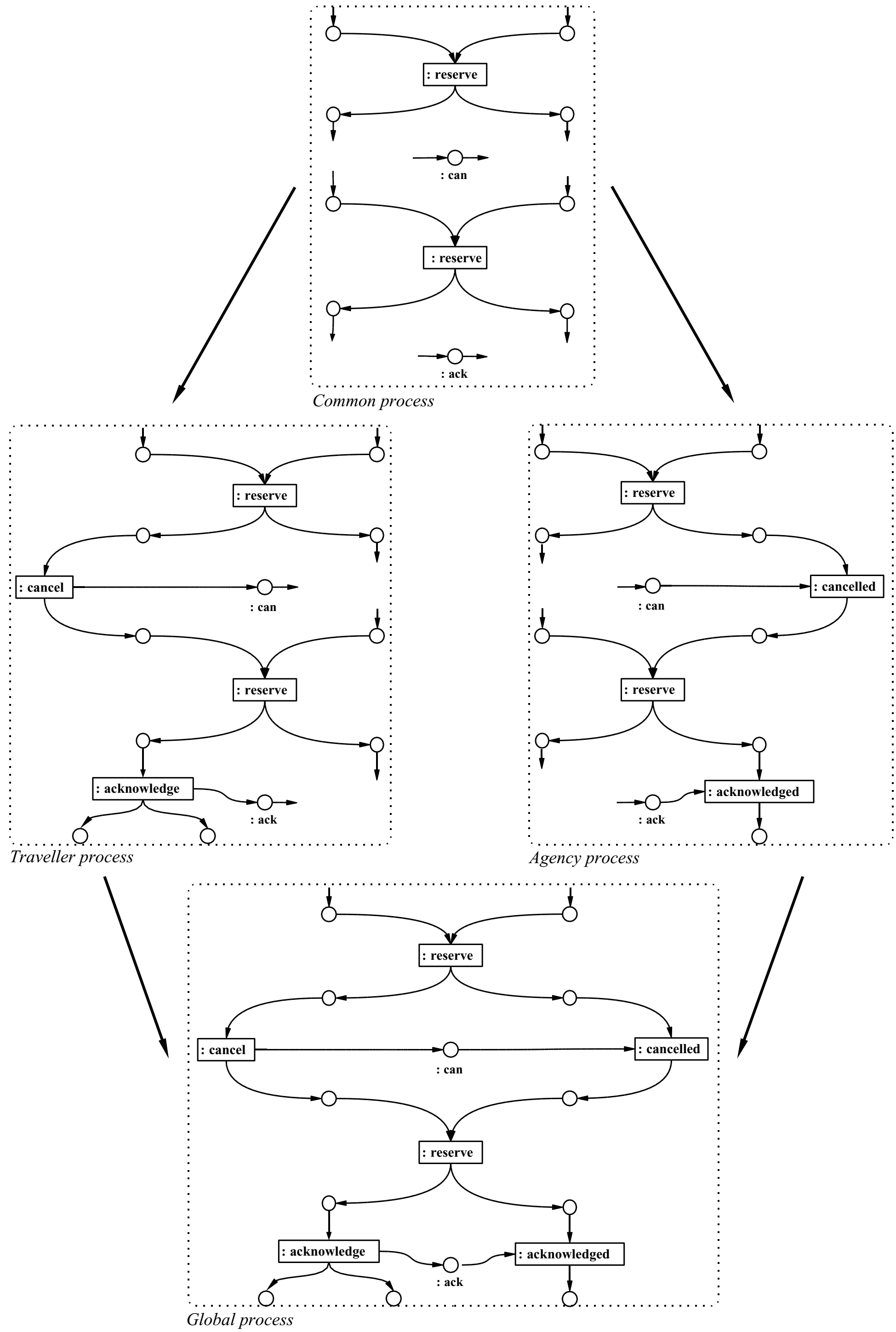


Fig. 3. Amalgamation of processes for the nets Traveller and Agency.

$A = A' \oplus x$ for some $A' \in X^{\oplus}$. Furthermore, given a function $h : X \to Y$ we will use $h^{\oplus} : X^{\oplus} \to Y^{\oplus}$ to denote its monoidal extension, while the same symbol $h : \mathbf{2}^X \to \mathbf{2}^Y$ denotes the extension of h to sets.

Definition 1 (P/T Petri net). A *P/T Petri net* is a tuple $N = (S, T, \sigma, \tau)$ where S is the set of places, T is the set of transitions (with $S \cap T = \varnothing$) and $\sigma, \tau : T \to S^{\oplus}$ are the functions assigning to each transition its pre- and post-set.

In the following we will use $^{\bullet}\_ : T^{\oplus} \to S^{\oplus}$ to denote the monoidal extension of the function $\sigma : T \to S^{\oplus}$. Similarly, $\_^{\bullet}$ denotes the monoidal extension of τ. Furthermore, given a place $s \in S$, the pre- and post-set of s are defined by $^{\bullet}s = \{t \in T \mid s \in t^{\bullet}\}$ and $s^{\bullet} = \{t \in T \mid s \in {}^{\bullet}t\}$.

Definition 2 (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 $t_0 \in T_0$, $^{\bullet}f_T(t_0) = f_S{}^{\oplus}({}^{\bullet}t_0)$ and $f_T(t_0)^{\bullet} = f_S{}^{\oplus}(t_0{}^{\bullet})$ (see the diagram below).

$$\begin{tikzcd}
T_0 \arrow[r, shift left, "\sigma_0"] \arrow[r, shift right, "\tau_0"'] \arrow[d, "f_T"'] & S_0{}^{\oplus} \arrow[d, "f_S{}^{\oplus}"] \\
T_1 \arrow[r, shift left, "\sigma_1"] \arrow[r, shift right, "\tau_1"'] & S_1{}^{\oplus}
\end{tikzcd}$$

The category of P/T Petri nets and Petri net morphisms is denoted by **Net**.

Petri net morphisms are closed under composition. This immediately follows by observing that given $f_0 : N_0 \to N_1$ and $f_1 : N_1 \to N_2$, we have $(f_{S_1} \circ f_{S_0})^{\oplus} = f_{S_1}{}^{\oplus} \circ f_{S_0}{}^{\oplus}$.

Category **Net** is a subcategory of the category **Petri** of Meseguer and Montanari (1990). The latter has the same objects, but more general morphisms, which can map a place into a multiset of places.

We are now ready to introduce the notion of open net.

Definition 3 (Open net). An *open net* is a pair $Z = (N_Z, O_Z)$, where:

— $N_Z = (S_Z, T_Z, \sigma_Z, \tau_Z)$ is an ordinary P/T Petri net, and
— $O_Z = (O_Z^+, O_Z^-) \in \mathbf{2}^{S_Z} \times \mathbf{2}^{S_Z}$ are the input and output open places of the net.

The places in $S - (O_Z^+ \cup O_Z^-)$ will be referred to as *internal* places of Z.

Observe that the sets O_Z^+ and O_Z^- are not necessarily disjoint, hence a place can be both an input and an output open place at the same time.

The notion of enabledness for a transition (or multiset of transitions) of an open net is the usual one, but, besides the changes produced to the state by the firing of the ‘internal’ transitions of the net, the interaction with the environment is also explicitly modelled. This is done by considering a kind of invisible action producing/consuming tokens in the input/output places of the net. The actions of the environment that produce and consume tokens in an open place s are denoted by $+_s$ and $-_s$, respectively.

Definition 4 (Token game). Let Z be an open net. A *sequential move* can be:

(i) the firing of a transition, that is, $m \oplus {}^\bullet t\ [t\rangle\ m \oplus t^\bullet$, with $m \in {S_Z}^\oplus$, $t \in T_Z$;
(ii) the creation of a token by the environment, that is, $m\ [+_s\rangle\ m \oplus s$, with $s \in O_Z^+$, $m \in {S_Z}^\oplus$;
(iii) the deletion of a token by the environment, that is, $m \oplus s\ [-_s\rangle\ m$, with $m \in {S_Z}^\oplus$, $s \in O_Z^-$.

A *parallel move* is of the form

$$m \oplus {}^\bullet A \oplus m^-\ [A \oplus E_- \oplus E_+\rangle\ m \oplus A^\bullet \oplus m^+,$$

where $m \in {S_Z}^\oplus$, $m^+ \in (O_Z^+)^\oplus$, $m^- \in (O_Z^-)^\oplus$, $A \in {T_Z}^\oplus$, and $E_- = \bigoplus_{s \in m_-} -_s$, $E_+ = \bigoplus_{s \in m_+} +_s$.

Alternatively, the token game of an open net can be described as the behaviour of an ordinary net, called the *closure* of Z and denoted by $\bar{Z}$. The net $\bar{Z}$ is obtained by adding transitions connected to open places that can freely produce/remove tokens from input/output places, that is, $\bar{Z} = (T', S_Z, \sigma', \tau')$ where:

— $T' = T_Z \cup \{+_s \mid s \in O_Z^+\} \cup \{-_s \mid s \in O_Z^-\}$;
— $\sigma'(+_s) = 0$ and $\tau'(+_s) = s$ for any $s \in O_Z^+$;
— $\sigma'(-_s) = s$ and $\tau'(-_s) = 0$ for any $s \in O_Z^-$;

and σ', τ' coincide with σ_Z, τ_Z on the other transitions.

Example. The open nets for the local workflows Traveller and Agency of Figure 1 are shown in the middle part of Figure 2. Ingoing and outgoing arcs without source or target designate the input and output places, respectively. Observe that the synchronisation transition reserve is common to both nets. Furthermore, the communication places, like can, become open places.

Definition 5 (Open net morphism). 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

$$\mathsf{in}(f) = \{s \in S_1 \mid {}^\bullet f_S(s) - f_T({}^\bullet s) \neq \emptyset\}$$

and

$$\mathsf{out}(f) = \{s \in S_1 \mid f_S(s)^\bullet - f_T(s^\bullet) \neq \emptyset\},$$

then

(i) $f_S^{-1}(O_2^+) \cup \mathsf{in}(f) \subseteq O_1^+$
(ii) $f_S^{-1}(O_2^-) \cup \mathsf{out}(f) \subseteq O_1^-$.

The morphism f is called an *open net embedding* if both components f_T and f_S are injective.

To simplify the notation in the rest of the paper, given an open net morphism $f = \langle f_S, f_T \rangle : Z_1 \to Z_2$, 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 $f_T(t)$.

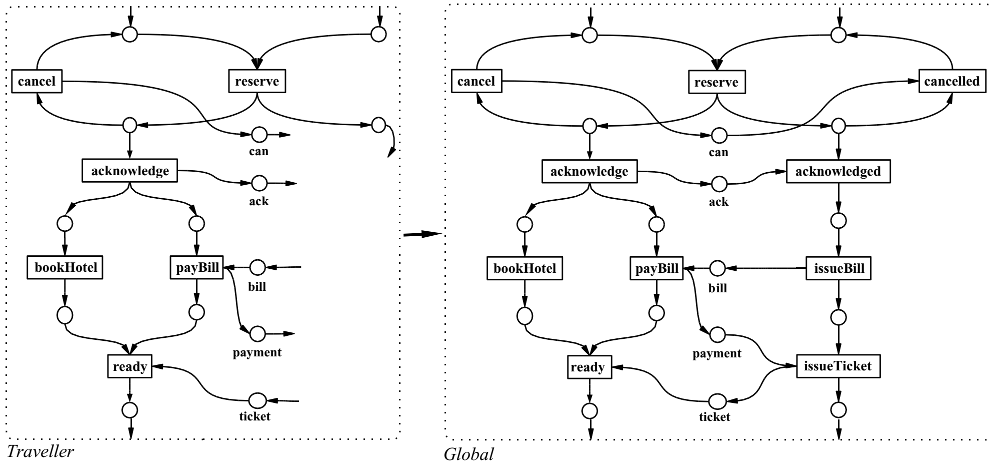


Fig. 4. The open net embedding of net Traveller into net Global.

A morphism $f : Z_1 \to Z_2$ can be thought of as an 'insertion' of the open net Z_1 into a larger net Z_2, which extends Z_1. In other words, Z_2 can be seen as an instantiation of Z_1, where part of the unknown environment gets more specified. Conditions (i) and (ii) first require that open places are reflected, and hence that internal places in Z_1 cannot be promoted to open places in Z_2. Furthermore, the context in which Z_1 is inserted can interact with Z_1 only through the open places. To understand how this is formalised, observe that for each place s in $\mathsf{in}(f)$, its image $f(s)$ is in the post-set of a transition outside the image of ${}^\bullet s$. Hence we can consider that in Z_2 new transitions are attached to s and can produce tokens in such place. This is the reason why condition (i) also requires any place in $\mathsf{in}(f)$ to be an input open place of Z_1. Condition (ii) is analogous for output places.

The above intuition fits better with open net embeddings, and indeed most of the constructions in the paper will be defined for this subclass of open net morphisms.

Example. As an example of open net morphism, consider the embedding of net Traveller into net Global of Figure 4 (extracted from Figure 2). 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. For instance, without the outgoing dangling arc from place can in net Traveller, that is, if place can were not output open, the mapping in Figure 4 would not have been a legal open net morphism.

Next we show that open net morphisms are closed under composition.

Proposition 6. 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 $f_2 \circ f_1$ is a morphism in **Net**. As for condition (i) of Definition 5, first observe that

$$\mathsf{in}(f_2 \circ f_1) \subseteq \mathsf{in}(f_1) \cup f_1^{-1}(\mathsf{in}(f_2)). \tag{1}$$

In fact,

$$\begin{aligned}
\mathsf{in}(f_2 \circ f_1) &= \{s \in S_1 \mid {}^\bullet f_2(f_1(s)) - f_2(f_1({}^\bullet s)) \neq \varnothing\} \\
&= \{s \in S_1 \mid {}^\bullet f_2(f_1(s)) - f_2({}^\bullet f_1(s)) \neq \varnothing\} \\
&\quad \cup \{s \in S_1 \mid f_2({}^\bullet f_1(s)) - f_2(f_1({}^\bullet s)) \neq \varnothing\} \\
&\subseteq \{s \in S_1 \mid f_1(s) \in \mathsf{in}(f_2)\} \cup \{s \in S_1 \mid f_2({}^\bullet f_1(s) - f_1({}^\bullet s)) \neq \varnothing\} \\
&= f_1^{-1}(\mathsf{in}(f_2)) \cup \{s \in S_1 \mid {}^\bullet f_1(s) - f_1({}^\bullet s) \neq \varnothing\} \\
&= f_1^{-1}(\mathsf{in}(f_2)) \cup \mathsf{in}(f_1).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathsf{in}(f_2 \circ f_1) &\subseteq \mathsf{in}(f_1) \cup f_1^{-1}(\mathsf{in}(f_2)) \qquad \text{[using (1)]} \\
&\subseteq O_1^+ \cup f_1^{-1}(O_2^+) \qquad \text{[since, by definition of morphism,} \\
&\qquad\qquad\qquad \mathsf{in}(f_1) \subseteq O_1^+ \text{ and } \mathsf{in}(f_2) \subseteq O_2^+] \\
&\subseteq O_1^+ \qquad \text{[since, by definition of morphism, } f_1^{-1}(O_2^+) \subseteq O_1^+].
\end{aligned}$$

Furthermore, $(f_2 \circ f_1)^{-1}(O_3^+) = f_1^{-1}(f_2^{-1}(O_3^+)) \subseteq f_1^{-1}(O_2^+) \subseteq O_1^+$, since f_1 and f_2 are morphisms. Thus, summing up,

$$(f_2 \circ f_1)^{-1}(O_3^+) \cup \mathsf{in}(f_2 \circ f_1) \subseteq O_1^+ .$$

Condition (ii), over output open places, can be proved in a totally analogous way. □

By the previous proposition, we can consider a category of open nets.

Definition 7 (Open nets category). We will use **ONet** to denote the category of open nets and open net morphisms.

We said earlier that open net morphisms are designed to capture the idea of 'insertion' of a net into a larger one. Hence it is natural to expect that they 'reflect' the behaviour in the sense that given $f : Z_0 \to Z_1$, the behaviour of Z_1 can be projected along the morphism to the behaviour of Z_0 (this fact will be formalised later, in Construction 13). Instead, unlike most of the morphisms considered over Petri nets, open net morphisms cannot be thought of as simulations since they *do not preserve* the behaviour. For instance, consider the open nets Z_0 and Z_1 in Figure 5 and the obvious open net morphism between them. Then the firing sequence $0\ [+_s\rangle\ s\ [t\rangle\ 0$ in Z_0 is not mapped to a firing sequence in Z_1.

There is an obvious forgetful functor from the category of open nets to the category of ordinary nets.

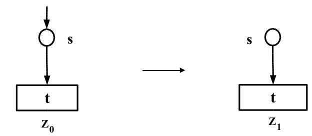


Fig. 5. Open net morphisms do not preserve the behaviour.

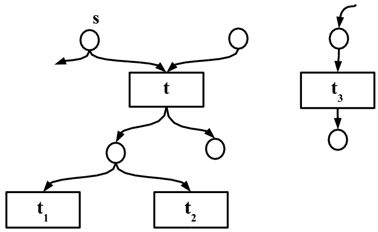


Fig. 6. A (non-deterministic) open occurrence net.

Definition 8. We use $\mathcal{F} : \mathbf{ONet} \to \mathbf{Net}$ to denote the forgetful functor defined by $\mathcal{F}(Z) = N_Z$ for any open net Z and $\mathcal{F}(f : Z_0 \to Z_1) = f : N_{Z_0} \to N_{Z_1}$ for any open net morphism f.

Since functor $\mathcal{F}$ acts on arrows as identity, with abuse of notation, given an open net morphism $f : Z_0 \to Z_1$, we will often write $f : \mathcal{F}(Z_0) \to \mathcal{F}(Z_1)$ instead of $\mathcal{F}(f) : \mathcal{F}(Z_0) \to \mathcal{F}(Z_1)$.

3. Deterministic processes of open nets

In a similar way to what happens for ordinary nets, a process of an open net, providing a truly concurrent description of a (possibly non-deterministic) computation of the net, is an open net itself, satisfying suitable acyclicity and conflict freeness requirements, together with a mapping to the original net.

The open net underlying a process is an open occurrence net, namely an open net K such that N_K is an ordinary occurrence net and satisfying some additional conditions over open places. The open places in K are intended to represent tokens that are produced/consumed by the environment in the computation under consideration. Consequently, every input open place is required to have an empty pre-set, that is, to be minimal with respect to the causal order. In fact, an input open place in the post-set of some transition would correspond to a kind of generalised backward conflict: a token on this place could be generated in two different ways, that is, by the firing of an 'internal' transition or by the environment, and this would prevent one from interpreting the place as a token occurrence.

Observe that, instead, an output open place can be in the pre-set of a transition, as happens for place s in the open occurrence net of Figure 6. The idea is that the token occurrence represented by place s can be consumed either by the environment or by transition t.

Recall that for an ordinary net $N = (S, T, \sigma, \tau)$ the *causal relation* $<_N$ is defined as the least transitive relation over $S \cup T$ such that $x <_N y$ if $y \in x^\bullet$, for $x, y \in S \cup T$. The *conflict relation* $\#_N$ is defined as the least symmetric relation over $S \cup T$ such that:

(i) for any $t, t' \in T$, if ${}^\bullet t \cap {}^\bullet t' \neq 0$ and $t \neq t'$ then $t \#_N t'$ (immediate conflict), and
(ii) if $x \# y$ and $y <_N z$ then $x \#_N z$ (inheritance with respect to causality).

Definition 9 (Open occurrence net). An *open occurrence net* is an open net K such that:

1. N_K is an ordinary occurrence net, in other words in N_K there are no backward conflicts (that is, for any $t, t' \in T_K$, if $t \neq t'$, then $t^\bullet \cap t'^\bullet = \emptyset$), the causal relation $<_K$ is a finitary strict partial order and the conflict relation $\#_K$ is irreflexive.
2. Each input open place is minimal with respect to $<_K$, that is, $\forall s \in O^+_K.\ {}^\bullet s = \emptyset$.

We are now ready to introduce the notion of process for open nets.

Definition 10 (Open net process). A *process* of an open net Z is a mapping $\pi : K \to Z$ where K is an open occurrence net and $\pi : N_K \to N_Z$ is a Petri net morphism, such that

$$\pi_S(O^+_K) \subseteq O^+_Z \quad \text{and} \quad \pi_S(O^-_K) \subseteq O^-_Z.$$

Observe that the mapping from the occurrence net K to the the original net Z is *not*, in general, an open net morphism. In fact, the process mapping, unlike open net morphisms, must be a simulation, that is, it must preserve the behaviour. Furthermore, the image of an open place in K must be an open place in Z, since tokens can be produced (consumed) by the environment only in input (output) open places of Z. Notice that in the case of nets with an empty set of open places, which can be seen as ordinary Petri nets, the notion of process coincides with the classical one.

In the following, when the meaning is clear from the context, we will sometimes identify a process $\pi : K \to Z$ with the corresponding morphism $\pi : N_K \to N_Z$ in the category **Net**.

As usual, a process will be called deterministic if it represents a uniquely determined concurrent computation. First, an open occurrence net is deterministic if the underlying ordinary occurrence net is deterministic, that is, each place is in the pre-set of at most one transition. Furthermore, the output open places must be maximal with respect to the causal order, that is, an output open place cannot be in the pre-set of any transition. In fact, as already observed, an output open place s that is in the pre-set of a transition t represents a token occurrence that can be consumed either by the environment or by transition t. A process will be called deterministic if the underlying open occurrence net is deterministic.

Definition 11 (Deterministic occurrence net and process). An open occurrence net K is called *deterministic* if:

1. The underlying ordinary occurrence net N_K is deterministic, that is, $\forall s \in S_K.\ |s^\bullet| \leqslant 1$.
2. Each output open place is maximal, that is, $\forall s \in O^-_K.\ s^\bullet = \emptyset$.

A process $\pi : K \to Z$ of an open net Z is *deterministic* if K is deterministic.

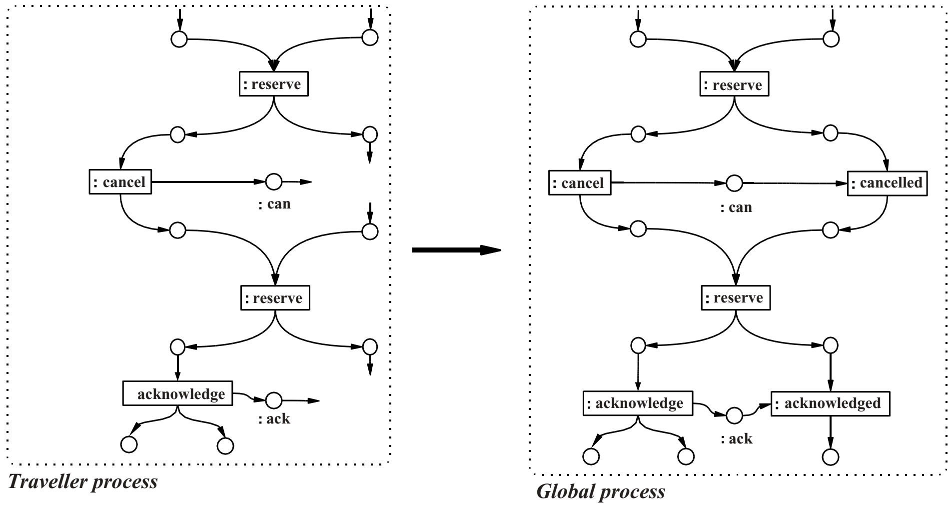


Fig. 7. A process of the open net Global and its projection to the subnet Traveller.

Example. A deterministic process for the open net Traveller is shown in Figure 7 on the left. The morphism back to the original net Traveller is implicitly represented by the labelling. Observe that the requirement that each input place is minimal and each output place is maximal with respect to the causal order of the process has a natural graphical interpretation: the absence of backward and forward conflicts extends to dangling arcs, that is, in total, each place may have at most one ingoing and one outgoing arc.

Next we introduce a category of processes, where the objects are processes and the arrows are pairs of open net morphisms.

Definition 12 (Category of processes). We use **Proc** to denote the category where objects are processes and, given two processes $\pi_0 : K_0 \to Z_0$ and $\pi_1 : K_1 \to Z_1$, an arrow $\psi : \pi_0 \to \pi_1$ is a pair of open net morphisms $\psi = \langle \psi_Z : Z_0 \to Z_1, \psi_K : K_0 \to K_1 \rangle$ such that the following diagram (indeed the underlying diagram in **Net**) commutes

$$\begin{tikzcd} K_0 \arrow[r, "\psi_K"] \arrow[d, "\pi_0"'] \arrow[rd, phantom, "\psi"] & K_1 \arrow[d, "\pi_1"] \\ Z_0 \arrow[r, "\psi_Z"'] & Z_1 \end{tikzcd}$$

3.1. *Projecting processes along embeddings*

Let $f : Z_0 \to Z_1$ be an open net morphism. As mentioned earlier, it is natural to expect that each computation in Z_1 can be 'projected' to Z_0, by considering only the part of the computation of the larger net that is visible in the smaller net. The above intuition is formalised, in the case of an open net embedding $f : Z_0 \to Z_1$, by showing how a process of Z_1 can be projected along f giving a process of Z_0.

Construction 13 (Projection of a process). Let $f : Z_0 \to Z_1$ be an open net embedding and let $\pi_1 : K_1 \to Z_1$ be a process of Z_1. A *projection of* π_1 *along* f is a pair $\langle \pi_0, \psi \rangle$ where $\pi_0 : K_0 \to Z_0$ is a process of Z_0 and $\psi : \pi_0 \to \pi_1$ is an arrow in **Proc**, constructed as follows. Consider the pullback of π_1 and f in **Net**, thus obtaining the net morphisms π_0 and ψ_K.

$$\begin{tikzcd} N_{K_0} \arrow[r, dotted, "\psi_K"] \arrow[d, dotted, "\pi_0"'] & N_{K_1} \arrow[d, "\pi_1"] \\ N_{Z_0} \arrow[r, "f"'] & N_{Z_1} \end{tikzcd}$$

Then K_0 is obtained by taking N_{K_0} with the smallest sets of open places that make $\psi_K : N_{K_0} \to N_{K_1}$ an open net morphism, namely

$$O^+_{K_0} = \psi_K{}^{-1}\big(O^+_{K_1}\big) \cup \mathsf{in}(\psi_K) \quad \text{and} \quad O^-_{K_0} = \psi_K{}^{-1}\big(O^-_{K_1}\big) \cup \mathsf{out}(\psi_K)$$

and $\psi = \langle \psi_K, f \rangle$.

The next proposition shows that the notion of projection is well-defined, and restricts to deterministic processes.

Proposition 14. The process $\pi_0 : K_0 \to Z_0$, as introduced in Construction 13, is well defined. Furthermore, the projection of a deterministic process is still a deterministic process.

Proof. First observe that K_0 is an open occurrence net. In fact, since f is injective, ψ_K is injective also, and thus N_{K_0} is isomorphic to the subnet of N_{K_1} in the codomain of ψ_K, which is clearly an ordinary occurrence net. Furthermore, we must show that each open input place is minimal. Let $s \in O^+_{K_0}$. Then we have two possibilities:

(i) $\psi_K(s) \in O^+_{K_1}$.
Observe that ${}^\bullet s \subseteq \psi_K^{-1}({}^\bullet\psi_K(s))$. Since K_1 is an open occurrence net, ${}^\bullet\psi_K(s) = \emptyset$ and thus ${}^\bullet s = \emptyset$.

(ii) $s \in \mathsf{in}(\psi_K)$.
In this case ${}^\bullet\psi_K(s) - \psi_K({}^\bullet s) \neq \emptyset$. Recalling that K_1 is an occurrence net and thus $|{}^\bullet\psi_K(s)| \leqslant 1$, we conclude that $\psi_K({}^\bullet s) = \emptyset$. Hence, as desired, ${}^\bullet s = \emptyset$.

Now, observe that π_0 is clearly a morphism in **Net**. Hence, to conclude that π_0 is a well-defined process, it only remains to show that it also satisfies

$$\pi_0\big(O^+_{K_0}\big) \subseteq O^+_{Z_0} \quad \text{and} \quad \pi_0\big(O^-_{K_0}\big) \subseteq O^-_{Z_0}.$$

Let us show, for instance, the first inclusion. Consider $s \in O^+_{K_0}$. Since, by construction, $O^+_{K_0} = \psi_K^{-1}(O^+_{K_1}) \cup \mathsf{in}(\psi_K)$, we distinguish two possibilities:

1. $s \in \psi_K^{-1}(O^+_{K_1})$
   We have $f(\pi_0(s)) = \pi_1(\psi_K(s)) \in \pi_1(O^+_{K_1})$ and, by definition of a process, $\pi_1(O^+_{K_1}) \subseteq O^+_{Z_1}$. Hence $\pi_0(s) \in f^{-1}(O^+_{Z_1}) \subseteq O^+_{Z_0}$, since f is an open net morphism.

2. $s \in \mathsf{in}(\psi_K)$
   In this case, ${}^\bullet\psi_K(s) - \psi_K({}^\bullet s) \neq \varnothing$. Since K_1 is an occurrence net, this means that there exists $t \in {}^\bullet\psi_K(s)$ and $\psi_K({}^\bullet s) = \varnothing$, that is, ${}^\bullet s = \varnothing$. Now observe that $\pi_1(t) \in {}^\bullet\pi_1(\psi_K(s)) = {}^\bullet f(\pi_0(s))$. Moreover, since the square in Construction 13 is a pullback, $\pi_1(t) \notin f({}^\bullet\pi_0(s))$. In fact, if $\pi_1(t) \in f({}^\bullet\pi_0(s))$, there would be t' in N_{K_0} such that $f(\pi_0(t')) = \pi_1(t)$, hence $t' \in {}^\bullet s$ and thus $\psi_K(t') \in {}^\bullet\psi_K(s)$, which should be empty. Summing up, $\pi_1(t)$ belongs to ${}^\bullet f(\pi_0(s)) - f({}^\bullet\pi_0(s))$, which thereby is non-empty. Hence $\pi_0(s) \in \mathsf{in}(f)$.

Let us prove now that the projection of a deterministic process is still a deterministic process. Assume that $\pi_1 : K_1 \to Z_1$ is a deterministic process of Z_1. As in the general case, the net N_{K_0} is isomorphic to the subnet of N_{K_1} in the codomain of ψ_K, and thus it is an ordinary deterministic occurrence net. We already know that $\forall s \in O^+_{K_0}.\ {}^\bullet s = \varnothing$, and $\pi_0(O^+_{K_0}) \subseteq O^+_{Z_0}$, $\pi_0(O^-_{K_0}) \subseteq O^-_{Z_0}$. Thus we only need to show that $\forall s \in O^-_{K_0}.\ s^\bullet = \varnothing$. Let $s \in O^-_{K_0}$. To prove that $s^\bullet = \varnothing$, just distinguish the cases:

1. $s \in \psi_K^{-1}(O^-_{K_1})$, and
2. $s \in \mathsf{out}(\psi_K)$.

Then proceed exactly as in points (i) and (ii) above but substituting '$-$' and $\mathsf{out}(\cdot)$ for $+$ and $\mathsf{in}(\cdot)$, respectively. □

The process π_0 of Z_0 is uniquely determined up to isomorphism. Observe that after fixing a representative in the isomorphism class of π_0, we can still have different choices for ψ_K (obtained one from the other by composition with an automorphism over N_{K_0}).

Example. The embedding of Traveller into Global in Figure 4 induces a projection of open net processes in the opposite direction. For instance, the right-hand part of Figure 7 shows a process of Global. Its projection along the embedding of Traveller into Global is shown on the left-hand part of the same figure. Notice how transition acknowledged, which consumes a token in place ack, is replaced in the projection by a dangling output arc: an internal action in the larger net becomes an interaction with the environment in the smaller one.

Remark 15. The construction of category **Proc** strictly resembles the construction of an arrow category. We use $\mathcal{N} : \mathbf{Proc} \to \mathbf{ONet}$ to denote the projection functor that maps each process $\pi : K \to Z$ to Z and each process arrow $\langle \psi_Z, \psi_K \rangle$ to ψ_Z. Then, given an embedding $f : Z_0 \to Z_1$ and a process $\pi_1 : K_1 \to Z_1$, a projection of π_1 along f, as defined above, is a cartesian arrow for π_1 and f.

If we restrict our attention to open net embeddings, thus obtaining the subcategories $\mathbf{ONet}^*$ and $\mathbf{Proc}^*$, the corresponding functor $\mathcal{N}^*$ is a fibration (see, for example, Jacobs (1999)) with total category $\mathbf{Proc}^*$ and base category $\mathbf{ONet}^*$. Furthermore, the fibration $\mathcal{N}^*$ is *split*. In fact, the injectivity of the arrows in $\mathbf{ONet}^*$ provides a choice of the pullbacks that are used for projections. Look at the diagram in Construction 13. When f is injective, ψ_K is injective also, and thus we have a canonical choice $\langle K'_0, \psi'_K, \pi'_0 \rangle$ for the construction, that is:

— *occurrence net K'_0*:
$N_{K'_0}$ is the subnet of N_{K_1} identified as the image of ψ_K; the open places of K'_0 are the open places in K_1 that belong to K'_0 and the 'interface places', namely the places in K'_0 whose precondition is outside K'_0, that is:

$$O^+_{K'_0} = \left(O^+_{K_1} \cap S_{K'_0}\right) \cup \left\{s \in S_{K'_0} : {}^\bullet s \cap \left(T_{K_0} - T_{K'_0}\right) \neq \emptyset\right\},$$

and $O^-_{K'_0}$ is defined in similar way.

— *arrows ψ'_K and π'_0*:
ψ'_K is the the inclusion of K'_0 into K_1, and π'_0 is uniquely determined by the requirement of commutativity.

The cleavage $c(f, \pi_1) = \langle \pi'_0, \langle f, \psi'_K \rangle \rangle$ defined in this way is splitting.

4. Composing open nets

In this section we introduce a basic mechanism for composing open nets, which will be characterised as a pushout construction in the category of open nets. 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. Consider, for instance, the open nets for the local workflows Traveller and Agency in the middle of Figure 2. The two nets share the subnet Common depicted in the top of the same figure, which represents the 'glue' between the two components. The net Global resulting from the composition of Traveller and Agency over the shared subnet Common is shown in the bottom part of Figure 2. This composition is only defined if the embeddings of the components into the resulting net satisfy the constraints of open net morphisms. For example, if we remove the ingoing dangling arc of the place ticket in the net Traveller, the embedding of Common into Traveller would still represent a legal open net morphism. However, in this case the embedding of Traveller into Global would become illegal because of the new arc from issueTicket (see condition (i) of Definition 5).

Formally, given two nets Z_1 and Z_2 and a span of open net embeddings $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$, the composition operation constructs the corresponding pushout in **ONet**. Category **ONet** does not have all pushouts, while category **Net** does. We will see that this corresponds to the intuition that the composition operation can be performed in **Net** and then lifted to **ONet**, but only when it respects the interfaces specified by the various components, for example, a new transition can be attached to a place only if the place is open. For instance, it is possible to verify that there is no pushout for the arrows in Figure 8, since, intuitively, the construction should merge all the places named s, attaching transition t to a place in Z_2 that is not (output) open.

We start by recalling a characterisation of pushouts in category **Net**.

Proposition 16 (Pushout in Net). Let $N_1 \xleftarrow{f_1} N_0 \xrightarrow{f_2} N_2$ be a span in **Net**. Then its pushout always exists, and can be defined as $N_1 \xrightarrow{\alpha_1} N_3 \xleftarrow{\alpha_2} N_2$, where the sets of places and transitions of N_3 are computed as the pushout in **Set** of the corresponding components:

$$S_3 = S_1 +_{S_0} S_2 \qquad \text{and} \qquad T_3 = T_1 +_{T_0} T_2,$$

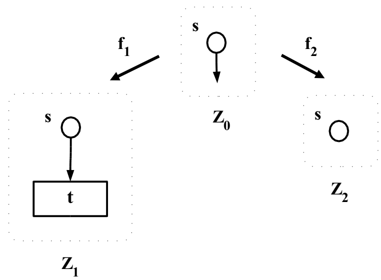


Fig. 8. Category **ONet** does not have all pushouts.

$$\begin{tikzcd}[column sep=small] & Z_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\ Z_1 \arrow[dr, "\alpha_1"'] & & Z_2 \arrow[dl, "\alpha_2"] \\ & Z_3 & \end{tikzcd}$$

Fig. 9. Pushout in **ONet**.

with source and target functions defined by: for all $t \in T_3$, if $t = \alpha_i(t_i)$ with $t_i \in T_i$ and $i \in \{1,2\}$, then ${}^\bullet t = {\alpha_i}^{\oplus}({}^\bullet t_i)$ and $t^\bullet = {\alpha_i}^{\oplus}({t_i}^\bullet)$.

Next we formalise the condition that ensures the composability of a span in **ONet**.

Definition 17 (Composable span). Let $Z_1 \stackrel{f_1}{\leftarrow} Z_0 \stackrel{f_2}{\rightarrow} Z_2$ be a span of open net embeddings. 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 $f_1(\mathsf{out}(f_2)) \subseteq O^-_{Z_1}$.

In words, f_1 and f_2 are composable if the places that are used as interfaces by f_1, namely the places $\mathsf{in}(f_1)$ and $\mathsf{out}(f_1)$, are mapped by f_2 to input and output open places in Z_2, and also the symmetric condition holds. If, and only if, this condition is satisfied, the pushout of f_1 and f_2 can be computed in **Net** and then lifted to **ONet**.

Proposition 18 (Pushouts in ONet). Let $Z_1 \stackrel{f_1}{\leftarrow} Z_0 \stackrel{f_2}{\rightarrow} Z_2$ be a span of embeddings in **ONet** (see the diagram in Figure 9). Compute the pushout of the corresponding diagram in the category **Net** obtaining the net N_{Z_3} and the morphisms α_1 and α_2, and then take as open places, for $x \in \{+,-\}$,

$$O^x_{Z_3} = \left\{ s_3 \in S_3 \middle| \alpha_1^{-1}(s_3) \subseteq O^x_{Z_1} \ \wedge \alpha_2^{-1}(s_3) \subseteq O^x_{Z_2} \right\}.$$

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.

Proof. **(If part)** Let us show that, when f_1 and f_2 are composable, $Z_1 \stackrel{\alpha_1}{\rightarrow} Z_3 \stackrel{\alpha_2}{\leftarrow} Z_2$ is a pushout in **ONet**.

We first prove that α_1 and α_2 are open net morphisms. The proof is given explicitly only for α_1, since the case of α_2 is completely analogous. First notice that

$$\mathsf{in}(\alpha_1) = f_1(\mathsf{in}(f_2)).$$

In fact, let $s_1 \in \mathsf{in}(\alpha_1)$. Hence there exists a transition $t_3 \in {}^\bullet\alpha_1(s_1) - \alpha_1({}^\bullet s_1)$. Since the square in Figure 9 is a pushout in **Net**, there exists $s_2 \in S_2$ such that $\alpha_1(s_1) = \alpha_2(s_2)$ and, also, $t_2 \in {}^\bullet s_2$ such that $\alpha_2(t_2) = t_3$ and $t_2 \notin f_2(T_0)$. By using the properties of pushouts again, we deduce the existence of $s_0 \in S_0$ such that $f_1(s_0) = s_1$ and $f_2(s_0) = s_2$. Now, $t_2 \in {}^\bullet f_2(s_0) - f_2(T_0) \subseteq {}^\bullet f_2(s_0) - f_2({}^\bullet s_0)$. Hence $s_0 \in \mathsf{in}(f_2)$ and thus $f_1(s_0) = s_1 \in f_1(\mathsf{in}(f_2))$. This proves that $\mathsf{in}(\alpha_1) \subseteq f_1(\mathsf{in}(f_2))$. The converse inclusion can be proved by reversing the proof steps.

Now, α_1 is clearly a morphism in **Net** by construction. Furthermore, it satisfies the condition $\alpha_1^{-1}(O^+_{Z_3}) \cup \mathsf{in}(\alpha_1) \subseteq O^+_{Z_1}$ and $\alpha_1^{-1}(O^-_{Z_3}) \cup \mathsf{out}(\alpha_1) \subseteq O^-_{Z_1}$. For instance, the condition over input places is proved by noticing that $\alpha_1^{-1}(O^+_{Z_3}) \subseteq O^+_{Z_1}$ by construction, and, $\mathsf{in}(\alpha_1) = f_1(\mathsf{in}(f_2)) \subseteq O^+_{Z_1}$ by condition (2) of composability (Definition 17). Thus α_1 is an open net morphism.

Moreover, for any pair of open net morphisms $\beta_1 : Z_1 \to Z_4$ and $\beta_2 : Z_2 \to Z_4$ such that $\beta_1 \circ f_1 = \beta_2 \circ f_2$, since $N_{Z_1} \stackrel{\alpha_1}{\to} N_{Z_3} \stackrel{\alpha_2}{\leftarrow} N_{Z_2}$ is a pushout in **Net**, there exists a unique arrow $h : Z_3 \to Z_4$ in **Net** such that the diagram below commutes.

$$\begin{tikzcd}
 & Z_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\
Z_1 \arrow[dr, "\alpha_1"] \arrow[ddr, bend right, "\beta_1"'] & & Z_2 \arrow[dl, "\alpha_2"'] \arrow[ddl, bend left, "\beta_2"] \\
 & Z_3 \arrow[d, dotted, "h"] & \\
 & Z_4 &
\end{tikzcd}$$

We only need to prove that h is an open net morphism by showing that it satisfies the condition over open places of Definition 5. Let us prove, for instance, that $h^{-1}(O^+_4) \cup \mathsf{in}(h) \subseteq O^+_3$. We divide the proof into two parts:

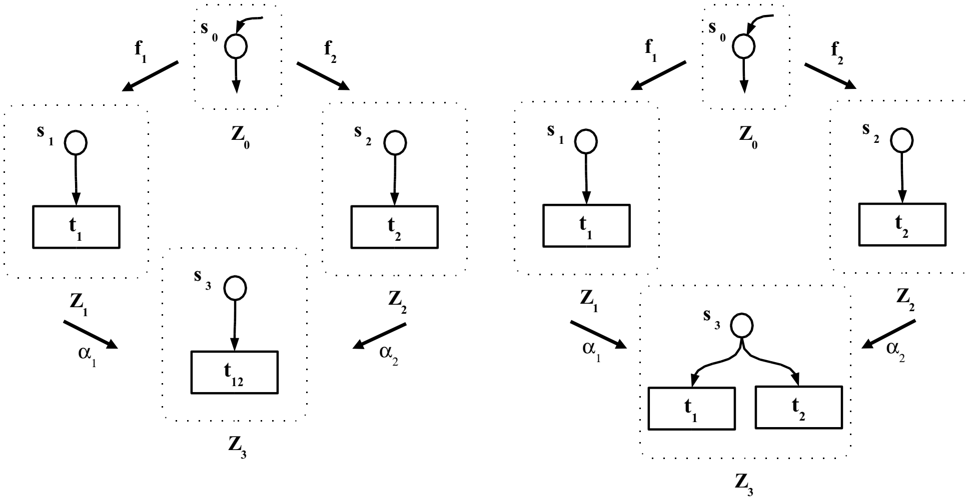
— $h^{-1}(O^+_4) \subseteq O^+_3$

Let $s_3 \in h^{-1}(O^+_4)$, that is, $s_3 \in S_3$ and $h(s_3) \in O^+_4$. Let $s_i \in \alpha_i^{-1}(s_3)$ for some $i \in \{1, 2\}$. By $h \circ \alpha_i = \beta_i$, we have $\beta_i(s_i) = h(s_3) \in O^+_4$. Thus, since β_i is an open net morphism, $s_i \in O^+_i$. In other words, $\alpha_1^{-1}(s_3) \subseteq O^+_1$ and $\alpha_2^{-1}(s_3) \subseteq O^+_2$. Hence, by definition of O^+_3, we have $s_3 \in O^+_3$.

— $\mathsf{in}(h) \subseteq O^+_3$

Let $s_3 \in \mathsf{in}(h)$, namely ${}^\bullet h(s_3) - h({}^\bullet s_3) \neq \varnothing$. Observe that if $s_3 = \alpha_i(s_i)$ for some $i \in \{1, 2\}$, we have

$$\begin{aligned}
\varnothing \neq\ & {}^\bullet h(s_3) - h({}^\bullet s_3) \\
=\ & {}^\bullet h(\alpha_i(s_i)) - h({}^\bullet \alpha_i(s_i)) \\
=\ & {}^\bullet \beta_i(s_i) - h({}^\bullet \alpha_i(s_i))
\end{aligned}$$



(*a*) (*b*)

Fig. 10. (a) A pushout in **ONet** of two non-composable arrows. (b) The pushout of the same arrows in **Net**.

$$
\begin{aligned}
&= \subseteq {}^{\bullet}\beta_i(s_i) - h(\alpha_i({}^{\bullet}s_i)) \qquad [\text{since } {}^{\bullet}\alpha_i(s_i) \supseteq \alpha_i({}^{\bullet}s_i)] \\
&= {}^{\bullet}\beta_i(s_i) - \beta_i({}^{\bullet}s_i)\,.
\end{aligned}
$$

Therefore, $s_i \in \mathsf{in}(\beta_i)$, and thus, since β_i is an open net morphism, $s_i \in O_i^+$. Summing up, we deduce that $\alpha_1^{-1}(s_3) \subseteq O_1^+$ and $\alpha_2^{-1}(s_3) \subseteq O_2^+$. Hence, by definition of O_3^+, $s_3 \in O_3^+$.

(Only if part) To prove that the composability of f_1 and f_2 is also necessary for ensuring that the pushout computed in **Net** is lifted to a pushout in **ONet**, suppose, for instance, that there exists $s_2 \in f_2(\mathsf{in}(f_1))$ and $s_2 \notin O_2^+$. Hence, there is $s_0 \in \mathsf{in}(f_1)$ such that $s_2 = f_2(s_0)$.

Suppose, to give a contradiction, that the described construction gives a pushout $Z_1 \xrightarrow{\alpha_1} Z_3 \xleftarrow{\alpha_2} Z_2$ in **ONet**. Hence, the places $s_1 = f_1(s_0)$ and $s_2 = f_2(s_0)$ have a common image $s_3 = \alpha_1(s_1) = \alpha_2(s_2)$. Since $s_0 \in \mathsf{in}(f_1)$, there exists $t_1 \in {}^{\bullet}f_1(s_0) - f_1({}^{\bullet}s_0)$. Thus $s_3 = \alpha_1(s_1) \in \alpha_1(t_1)^{\bullet}$. Moreover, from the fact that $s_2 \notin O_2^+$, by definition of open net morphism, we have $s_2 \notin \mathsf{in}(\alpha_2)$. Hence there exists $t_2 \in {}^{\bullet}s_2$ such that $\alpha_2(t_2) = \alpha_1(t_1)$. Therefore there is $t_0 \in T_0$ such that $f_1(t_0) = t_1$ and $f_2(t_0) = t_2$. But this contradicts the fact that $t_1 \in {}^{\bullet}f_1(s_0) - f_1({}^{\bullet}s_0)$. $\square$

It is worth stressing that the pushout in **ONet** might also exist when two embeddings f_1 and f_2 are not composable. This is the case for the diagram in Figure 10 (a), which is a pushout in **ONet**, although the underlying diagram in **Net** is *not* a pushout. Indeed, f_1 and f_2 are *not* composable since, for instance, $f_2(\mathsf{out}(f_1)) = f_2(\{s_0\}) = \{s_2\} \nsubseteq O_2^-$. In this case the construction described in Proposition 18 does not work: it leads to the diagram in Figure 10 (b), where the mappings $\alpha_i : Z_i \to Z_3$ are not open net morphisms, since, for instance, $s_1 \in \mathsf{out}(\alpha_1)$, but $s_1 \notin O_1^-$.

One could be tempted to assume a different notion of composable span, that is, to define f_1 and f_2 composable whenever their pushout exists in **ONet**. However, according to our intuition, morphisms f_1 and f_2 define a kind of 'composition plan', which specifies that the images of Z_0 in Z_1 and Z_2 must be fused. The effect of the composition operation should be local, in the sense that nothing more than the images of Z_0 should be affected by the fusion. This fact is formalised by requiring that the pushout in **ONet** is obtained by lifting the pushout in **Net**. Observe that, instead, in the pushout depicted in Figure 10 (a), transitions t_1 and t_2, which are not in the common subnet Z_0, also get fused.

To conclude this section, let us comment on the expressiveness of the composition operation based on pushouts. Observe that any ordinary Petri net N in **Net** without self-loops can be obtained from basic transitions and single places by iterating our composition operation. More precisely, given a net $N = (S, T, \sigma, \tau)$, for any $t \in T$, let B_t be the open net consisting of the single transition t with its pre- and post-set, where all places are both input and output open, and let B_1 be the net consisting of a single place, which is both input and output open. Then it is not difficult to see that iterating our construction on the B_t's and on a finite number of copies of the B_1's, one can obtain an open net Z such that $\mathcal{F}(Z) \simeq N$.

5. Amalgamating deterministic processes

Let $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ be a composable span of open net embeddings and consider the corresponding composition, that is, the pushout in **ONet**, as depicted in Figure 9. We would like to establish a clear relationship among the behaviours of the involved nets. Roughly speaking, we would like the behaviour of Z_3 to be constructed 'compositionally' out of the behaviours of Z_1 and Z_2.

In this section we show how this can be done for deterministic processes. Given two deterministic processes π_1 of Z_1 and π_2 of Z_2 that 'agree' on Z_0, we construct a deterministic process π_3 of Z_3 by 'amalgamating' π_1 and π_2. Conversely, each deterministic process π_3 of Z_3 can be projected over two deterministic processes π_1 and π_2 of Z_1 and Z_2, respectively, which can be amalgamated to produce π_3 again. Hence, all and only the deterministic processes of Z_3 can be obtained by amalgamating the deterministic processes of the components Z_1 and Z_2. This is formalised by showing that, working up to isomorphism, the amalgamation and decomposition operations are inverse to each other. This leads to a bijective correspondence between the processes of Z_3 and the pair of processes of the components Z_1 and Z_2 that agree on the common subnet Z_0.

5.1. *Pushout of deterministic occurrence open nets*

As a first step towards the amalgamation of processes, we identify a suitable condition that ensures that the pushout of deterministic occurrence open nets exists and produces a net in the same class. This condition will be used later to formalise the intuitive idea of processes of different nets that 'agree' on a common part.

First, given a span $K_1 \xleftarrow{f_1} K_0 \xrightarrow{f_2} K_2$, we introduce the notion of causality relation induced by K_1 and K_2 over K_0. When the two nets are composed the corresponding causality relations get 'fused'. Hence, to avoid the creation of cyclic causal dependencies in the resulting net, the induced causality will be required to be a strict partial order.

Definition 19 (Induced causality and consistent span). Let $K_1 \xleftarrow{f_1} K_0 \xrightarrow{f_2} K_2$ be a span of embeddings in **ONet**, where K_i ($i \in \{0,1,2\}$) are occurrence open nets. The relation of causality $<_{1,2}$ induced over K_0 by K_1 and K_2, through f_1 and f_2, is the least transitive relation such that for any x_0, y_0 in K_0, if $f_1(x_0) <_1 f_1(y_0)$ or $f_2(x_0) <_2 f_2(y_0)$, then $x_0 <_{1,2} y_0$.

We say that the span is *consistent*, written $f_1 \uparrow f_2$, if f_1 and f_2 are composable and the induced causality $<_{1,2}$ is a finitary strict partial order.

We next show that the composition operation in **ONet**, when applied to a consistent span of deterministic occurrence nets, produces a deterministic occurrence net. We first need a preliminary result.

Lemma 20. Let $K_1 \xleftarrow{f_1} K_0 \xrightarrow{f_2} K_2$ be a composable span of embeddings in **ONet**, where K_i ($i \in \{0,1,2\}$) are deterministic occurrence open nets. Let $K_1 \xrightarrow{\alpha_1} K_3 \xleftarrow{\alpha_2} K_2$ be the following pushout:

$$\begin{tikzcd} & K_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\ K_1 \arrow[dr, "\alpha_1"'] & & K_2 \arrow[dl, "\alpha_2"] \\ & K_3 & \end{tikzcd}$$

For any x_0, y_0 in K_0, if we let $x_3 = \alpha_1(f_1(x_0)) = \alpha_2(f_2(x_0))$ and $y_3 = \alpha_1(f_1(y_0)) = \alpha_2(f_2(y_0))$, then

$$x_0 <_{1,2} y_0 \quad \text{iff} \quad x_3 <_3 y_3.$$

Proof. Below we will freely use the fact that open net morphisms, and thus, in particular α_1 and α_2, preserve the causality relation, in the sense that if $x_i <_i y_i$ in K_i ($i \in \{1,2\}$), then $\alpha_i(x_i) <_3 \alpha_i(y_i)$.

($\Rightarrow$) Suppose that $x_0 <_{1,2} y_0$. There are two possible cases:

— The causal dependence is directly induced by a causal dependence in K_1 or K_2, namely $f_i(x_0) <_i f_i(y_0)$ for some $i \in \{1,2\}$. Since α_i preserves causality, $\alpha_i(f_i(x_0)) <_3 \alpha_i(f_i(y_0))$, namely $x_3 <_3 y_3$.

— Otherwise, the causal dependence is generated by the transitive closure, in other words, there is z_0 such that $x_0 <_{1,2} z_0 <_{1,2} y_0$. Hence, an inductive reasoning allows us to conclude that $x_3 <_3 \alpha_i(f_i(z_0)) <_3 y_3$ and thus $x_3 <_3 y_3$.

($\Leftarrow$) Let $\prec_i$ denote the immediate causality in K_i, that is, $x \prec_i y$ if $x <_i y$ and there is no z such that $x <_i z <_i y$. It is easy to see that for any x_3, y_3 in K_3,

$$x_3 \prec_3 y_3 \quad \text{iff} \quad \exists\, i \in \{1,2\},\ \exists\, x_i, y_i \text{ in } K_i \text{ such that } x_3 = \alpha_i(x_i),\ y_3 = \alpha_i(y_i),\ x_i \prec_i y_i.$$

Assume that $x_3 <_3 y_3$. Then there is a $\prec_3$-chain $x_3 = x_3^1 \prec_3 x_3^2 \prec_3 \ldots \prec_3 x_3^n = y_3$. Let $C = \{x_3^1, \ldots, x_3^n\}$. By the remark above, if C is included in $\alpha_i(S_i \cup T_i)$ for some $i \in \{1, 2\}$, then $f_i(x_0) <_i f_i(y_0)$, and thus $x_0 <_{1,2} y_0$. More generally, since K_3 is obtained as the pushout of K_1 and K_2, the chain C can be divided into $h+1$ segments $x_3, \ldots, x_3^{k_1}, \ldots, x_3^{k_2}, \ldots, x_3^{k_h}, \ldots y_3$ such that each segment is included in $\alpha_i(S_i \cup T_i)$ for some $i \in \{1, 2\}$ and any 'border' element $x_3^{k_j}$ is in $\alpha_1(S_1 \cup T_1) \cap \alpha_2(S_2 \cup T_2)$. By general properties of pushouts, for any j we can find $x_0^j \in S_0 \cup T_0$, such that $\alpha_i(f_i(x_0^j)) = x_3^{k_j}$ for $i \in \{1, 2\}$.

Therefore, by the remark about immediate precedence in K_3, surely, for any j there is some $i \in \{1, 2\}$, such that

$$f_i(x_0^j) <_i f_i(x_0^{j+1}) \tag{2}$$

and, similarly, $f_i(x_0) <_{i_x} f_i(x_0^1)$ and $f_i(x_0^h) <_{i_y} f_i(y_0)$ for suitable $i_x, i_y \in \{1, 2\}$. But recalling the definition of induced causality, we deduce that $x_0 <_{1,2} x_0^1 <_{1,2} x_0^2 <_{1,2} \ldots x_0^k <_{1,2} y_0$, and thus $x_0 <_{1,2} y_0$. □

Proposition 21. Let $K_1 \xleftarrow{f_1} K_0 \xrightarrow{f_2} K_2$ be a composable span of embeddings in **ONet**, where K_i $(i \in \{0, 1, 2\})$ are deterministic occurrence open nets, and let $K_1 \xrightarrow{\alpha_1} K_3 \xleftarrow{\alpha_2} K_2$ be the following pushout in **ONet**:

$$\begin{tikzcd}
 & K_0 \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\
K_1 \arrow[dr, dotted, "\alpha_1"'] & & K_2 \arrow[dl, dotted, "\alpha_2"] \\
 & K_3 &
\end{tikzcd}$$

Then $f_1 \uparrow f_2$ if and only if the pushout object K_3 is a deterministic occurrence open net.

Proof. $(\Rightarrow)$ We know that K_3 is a well-defined open net. To prove that K_3 is a deterministic open occurrence net, we start by showing that the underlying net N_{K_3} is a deterministic occurrence net.

(1.a) *causality $<_3$ is a strict partial order.*
We prove this by contradiction. Assume that $<_3$ is not irreflexive. Hence, we can find a cycle of immediate causality in K_3, that is, $x_3^1 \prec_3 x_3^2 \prec_3 \ldots \prec_3 x_3^n \prec_3 x_3^1$, and let $C = \{x_3^1, \ldots, x_3^n\}$. The cycle C cannot be included in $\alpha_i(S_i \cup T_i)$ for some $i \in \{1, 2\}$, otherwise $\prec_i$ would be cyclic in K_i. Hence there exists an item $x_3 \in C \cap \alpha_1(S_1 \cup T_1) \cap \alpha_2(S_2 \cup T_2)$. Consider x_0 in K_0 such that $\alpha_i(f_i(x_0)) = x_3$. Since $x_3 <_3 x_3$, by Lemma 20, we have $x_0 <_{1,2} x_0$, contradicting the hypothesis that the span is consistent.

(1.b) *causality $<_3$ is finitary.*
The proof can be carried out as in (1.a) above by exploiting the finitariness of causality in K_1 and K_2, and Lemma 20. Assuming the existence of an infinite descending chain of $<_3$ in K_3, we deduce that $<_{1,2}$ has an infinite descending chain in K_0, contradicting the assumption that the span is consistent, and thus that $<_{1,2}$ is finitary.

(1.c) *K_3 does not have forward conflicts.*
We prove this by contradiction. Suppose that there exists a place $s_3 \in S_3$ such that $|s_3{}^\bullet| > 1$. Let $t_3, t_3' \in s_3{}^\bullet$ such that $t_3 \neq t_3'$. Then, without loss of generality, we may

assume that $t_3 \in \alpha_1(T_1) - \alpha_2(T_2)$ and $t'_3 \in \alpha_2(T_2) - \alpha_1(T_1)$, otherwise we would have a forward conflict in one of K_1 or K_2. Therefore, $s_3 \in \alpha_1(S_1) \cap \alpha_2(S_2)$. Let $s_1 \in S_1$ such that $\alpha_1(s_1) = s_3$. Then $s_1 \in \mathsf{out}(\alpha_1)$. But, since $s_1{}^\bullet \neq \varnothing$, this contradicts the assumption that K_1 is a deterministic open net.

(1.d) K_3 *does not have backward conflicts.*
This case is analogous to (1.c).

To conclude, it remains to show the validity of the conditions over open places.

(2.a) $\forall s \in O_3^-.\ s^\bullet = \varnothing$.
The proof is the same as for point (1.c).

(2.b) $\forall s \in O_3^+.\ {}^\bullet s = \varnothing$.
The proof is the same as for point (1.d).

($\Leftarrow$) Let $K_1 \overset{f_1}{\leftarrow} K_0 \overset{f_2}{\rightarrow} K_2$ be a composable span of embeddings in **ONet**, where K_i ($i \in \{0, 1, 2\}$) are deterministic occurrence open nets, and assume that the pushout K_3 is an open deterministic net. We must show that induced causality $<_{1,2}$ is a finitary strict partial order. Let $f_3 = \alpha_1 \circ f_1 = \alpha_2 \circ f_2$. To conclude, just recall that $<_3$ is a finitary strict partial order and then use the fact that, by Lemma 20, $x_0 <_{1,2} y_0$ iff $f_3(x_0) <_3 f_3(y_0)$. $\square$

5.2. *Amalgamating deterministic processes*

As mentioned earlier, two deterministic processes π_1 of Z_1 and π_2 of Z_2 can be amalgamated only when they agree on the common subnet Z_0, an idea that is formalised by resorting to the notion of a consistent span of deterministic occurrence open nets. In the rest of this section we will refer to a fixed pushout diagram in **ONet**, as represented in Figure 9, where f_1 and f_2 are a composable span of *open net embeddings.*

Definition 22 (Agreement of deterministic processes). We say that two deterministic processes $\pi_1 : K_1 \to Z_1$ and $\pi_2 : K_2 \to Z_2$ *agree* on Z_0 if there are projections $\langle \pi_0, \psi^i \rangle$ along f_i of π_i for $i \in \{1, 2\}$ such that $\psi_K^1 \uparrow \psi_K^2$ (that is, the span $K_1 \overset{\psi_K^1}{\leftarrow} K_0 \overset{\psi_K^2}{\rightarrow} K_2$ is consistent). In this case $\langle \pi_0, \psi^1 \rangle$ and $\langle \pi_0, \psi^2 \rangle$ are called *agreement projections* for π_1 and π_2.

Before introducing the notion of amalgamation, we need to recall a simple technical result.

Lemma 23.
1. Consider the diagram in **Set** depicted in Figure 11 (a). If the diagram is a pushout and f is injective, then the diagram is also a pullback.
2. Consider a commuting diagram in a category **C**, as depicted in Figure 11 (b). If the internal square, marked by PB, and the external square are pullbacks, then the other internal square is a pullback as well.

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "g"] \\
C \arrow[r, "i"'] & D
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
A \arrow[d] \arrow[dr, phantom, "\mathsf{PB}"] & B \arrow[l] \arrow[d] & C \arrow[l] \arrow[d] \\
D & E \arrow[l] & F \arrow[l]
\end{tikzcd}$$

(a) (b)

Fig. 11. Figures for Lemma 23.

$$\begin{tikzcd}
 & & K_0 \arrow[dll, "\psi^1_K"'] \arrow[drr, "\psi^2_K"] \arrow[d, "\pi_0"] & & \\
K_1 \arrow[d, "\pi_1"] \arrow[drr, dotted, "\phi^1_K"] & & Z_0 \arrow[dll, "f_1"'] \arrow[drr, "f_2"] & & K_2 \arrow[d, "\pi_2"] \arrow[dll, dotted, "\phi^2_K"'] \\
Z_1 \arrow[drr, "\alpha_1"'] & & K_3 \arrow[d, dotted, "\pi_3"] & & Z_2 \arrow[dll, "\alpha_2"] \\
 & & Z_3 & &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & K_0 \arrow[dll, "\psi^1_K"'] \arrow[drr, dotted, "\psi^2_K"] \arrow[d, "\pi_0"] & & \\
K_1 \arrow[d, "\pi_1"] \arrow[drr, "\phi^1_K"] & & Z_0 \arrow[dll, "f_1"'] \arrow[drr, "f_2"] & & K_2 \arrow[d, "\pi_2"] \arrow[dll, "\phi^2_K"'] \\
Z_1 \arrow[drr, "\alpha_1"'] & & K_3 \arrow[d, "\pi_3"] & & Z_2 \arrow[dll, "\alpha_2"] \\
 & & Z_3 & &
\end{tikzcd}$$

(a) (b)

Fig. 12. Amalgamation of open net processes.

Definition 24 (Amalgamation of processes). Let $\pi_i : K_i \to Z_i$ ($i \in \{0, 1, 2, 3\}$) be deterministic processes and let $\langle \pi_0, \psi^1 \rangle$ and $\langle \pi_0, \psi^2 \rangle$ be agreement projections of π_1 and π_2 along f_1 and f_2 (see Figure 12 (a)). We say that π_3 is an *amalgamation* of π_1 and π_2, written $\pi_3 = \pi_1 +_{\psi^1,\psi^2} \pi_2$, if there exist projections $\langle \pi_1, \phi^1 \rangle$ and $\langle \pi_2, \phi^2 \rangle$ of π_3 over Z_1 and Z_2, respectively, such that the upper square is a pushout in **ONet**.

We next give a more constructive characterisation of process amalgamation, which also proves that the result is unique up to isomorphism.

Theorem 25 (Amalgamation construction). Let $\pi_1 : K_1 \to Z_1$ and $\pi_2 : K_2 \to Z_2$ be deterministic processes that agree on Z_0, and let $\langle \pi_0, \psi^1 \rangle$ and $\langle \pi_0, \psi^2 \rangle$ be corresponding agreement projections. Then the *amalgamation* $\pi_1 +_{\psi^1,\psi^2} \pi_2$ is a process $\pi_3 : K_3 \to Z_3$, where the net K_3 is obtained as the pushout in **ONet** of $\psi^1_K : K_0 \to K_1$ and $\psi^2_K : K_0 \to K_2$ and the process mapping $\pi_3 : K_3 \to Z_3$ is uniquely determined by the universal property of the underlying pushout diagram in **Net** (see Figure 12 (a)). Hence $\pi_1 +_{\psi^1,\psi^2} \pi_2$ is unique up to isomorphism.

Proof. We first show that π_3, defined as above, is a well-defined process of Z_3. Since, by hypothesis, $\psi^1_K \uparrow \psi^2_K$, we know by Proposition 21 that K_3 is a deterministic occurrence open net.

Furthermore, π_3 is an arrow in **Net**. To conclude that π_3 is a deterministic open net process, we prove that $\pi_3(O^+_{K_3}) \subseteq O^+_{Z_3}$ and $\pi_3(O^-_{K_3}) \subseteq O^-_{Z_3}$.

To this end, we first observe that in the diagram of Figure 12 (a), the square with vertices K_1, K_3, Z_3, Z_1 is a pullback. Let us show, for instance, that the place component of the morphisms form a pullback. Actually, it suffices to show that given $s_1 \in S_{Z_1}$ and $s_3' \in S_{K_3}$ such that $\alpha_1(s_1) = \pi_3(s_3')$, there exists $s_1' \in S_{K_1}$ such that $\phi_K^1(s_1') = s_3'$. In fact, by commutativity of the diagram, this implies that $\alpha_1(\pi_1(s_1')) = \alpha_1(s_1)$, and thus, by injectivity of α_1, we have $\pi_1(s_1') = s_1$. Furthermore, uniqueness of s_1' follows from the injectivity of ϕ_K^1. Hence, let us consider $s_1 \in S_{Z_1}$ and $s_3' \in S_{K_3}$ such that $\alpha_1(s_1) = \pi_3(s_3') = s_3$. Assume, to show a contradiction, that $s_3' \neq \phi_K^1(s_1')$ for all $s_1' \in S_{K_1}$. Since the upper square is a pushout, necessarily, $s_3' = \phi_K^2(s_2')$ for some $s_2' \in S_{K_2}$. Then $\alpha_2(\pi_2(s_2')) = s_3 = \alpha_1(s_1)$. Since the square Z_0, Z_1, Z_2, Z_3 is a pushout, this implies that there exists s_0 in Z_0 such that $f_1(s_0) = s_1$ and $f_2(s_0) = \pi_2(s_2')$. But, since the square Z_2, K_2, K_0, Z_0 is a pullback, there must be $s_0' \in S_{K_0}$ such that $\psi_K^2(s_0') = s_2'$. Hence, if we take $s_1' = \psi_K^1(s_0')$, we have $\phi_K^1(s_1') = \phi_K^1(\psi_K^1(s_0')) = \phi_K^2(\psi_K^2(s_0')) = \phi_K^2(s_2') = s_3'$ yielding the desired contradiction.

Now, take $s_3' \in O_{K_3}^+$ and consider $\pi_3(s_3')$. We distinguish the following (non-exclusive) cases:

— $\pi_3(s_3') = \alpha_1(s_1)$ for some $s_1 \in S_{Z_1}$.
Since, as observed above, the square K_1, K_3, Z_3, Z_1 is a pullback, there is $s_1' \in S_{K_1}$ such that $\phi_K^1(s_1') = s_3$ and $\pi_1(s_1') = s_1$. From the first equality, since ϕ_K^1 is an open net morphism, we deduce that $s_1' \in O_{K_1}^+$, and thus, by the second equality, since π_1 is a process, $s_1 \in O_{Z_1}^+$.

— $\pi_3(s_3') = \alpha_2(s_2)$ for some $s_2 \in S_{Z_2}$.
As above, we can conclude $s_2 \in O_{Z_2}^+$.

Summing up the two cases, we have that $\alpha_1^{-1}(\pi_3(s_3)) \subseteq O_{Z_1}^+$ and $\alpha_2^{-1}(\pi_3(s_3)) \subseteq O_{Z_2}^+$. Therefore, by construction of the pushout in **ONet** (see Proposition 18), $\pi_3(s_3) \in O_{Z_3}^+$. Thus $\pi_3(O_{K_3}^+) \subseteq O_{Z_3}^+$. The other inclusion, that is, $\pi_3(O_{K_3}^-) \subseteq O_{Z_3}^-$, can be shown in a completely symmetric way.

The final thing to observe is that $\langle \pi_i, \phi^i \rangle$ is a projection of π_3 along α_i for $i \in \{1, 2\}$. But this fact immediately follows from the above observations, since the squares K_i, K_3, Z_3, Z_i are pullbacks in **Net**. Furthermore, $O_i^+ = {\phi_K^i}^{-1}(O_3^+) \cup \mathsf{in}(\phi_K^i)$. In fact, ϕ_K^i is an open net morphism, and thus ${\phi_K^i}^{-1}(O_3^+) \cup \mathsf{in}(\phi_K^i) \subseteq O_i^+$. To prove the other inclusion, for instance, when $i = 1$, let $s_1 \in O_{K_1}^+$. If $\phi_K^1(s_1) \in O_{K_3}^+$, we have that $s_1 \in {\phi_K^1}^{-1}(O_{K_3}^+)$. Otherwise, by recalling how the open places of the pushout object are defined (see Proposition 18), we deduce that there exists $s_2 \in S_{K_2}$ such that $\phi_K^2(s_2) = \phi_K^1(s_1)$ and $s_2 \notin O_{K_2}^+$. Since the upper square is a pushout, there must be $s_0 \in S_{K_0}$ such that $\psi_K^1(s_0) = s_1$ and $\psi_K^2(s_0) = s_2$. Since ψ_K^1 is an open net morphism, this implies that $s_0 \in O_{K_0}^+$. Since $s_2 \notin O_{K_2}^+$ and π_0 is a projection of π_2, we have that $s_0 \in \mathsf{in}(\psi_K^2)$. Therefore, since the upper square is a pushout in **Net**, $s_1 \in \mathsf{in}(\phi_K^1)$, as desired. □

The amalgamation construction can be given a more elegant (although less constructive) characterisation, in terms of a pushout in **Proc**.

Proposition 26. Let $\pi_1 : K_1 \to Z_1$ and $\pi_2 : K_2 \to Z_2$ be deterministic processes that agree on Z_0, and let $\langle \pi_0, \psi^1 \rangle$ and $\langle \pi_0, \psi^2 \rangle$ be corresponding agreement projections. Then the *amalgamation* $\pi_1 +_{\psi^1,\psi^2} \pi_2$ and the corresponding process morphisms $\langle \phi^1, \alpha_1 \rangle$ and $\langle \phi^2, \alpha_2 \rangle$ can be obtained as the pushout in **Proc** of the arrows $\psi^1 : \pi_0 \to \pi_1$ and $\psi^2 : \pi_0 \to \pi_2$ (see Figure 12 (a)).

The next result shows how each deterministic process of a composed net can be constructed as the amalgamation of deterministic processes of the components.

Theorem 27 (Decomposition of processes). Let $\pi_3 : K_3 \to Z_3$ be a deterministic process of Z_3 and, for $i \in \{1, 2\}$, let $\langle \pi_i, \phi^i \rangle$ be projections of π_3 along α_i. Then process π_3 can be recovered as a suitable amalgamation of π_1 and π_2.

Proof. Let $\langle \pi_i, \phi^i \rangle$ be projections of π_3 along α_i for $i \in \{1, 2\}$. Take any projection $\langle \pi_0, \psi^1 \rangle$ of π_1 along f_1. The non-dotted part of the diagram in Figure 12 (b) summarises the situation.

Then projection $\langle \pi_0, \psi^2 \rangle$ of π_2 along f_2 is obtained by defining ψ^2_K as the arrow determined by the universal property of the pullback with vertices K_3, Z_3, Z_2 and K_2. To show that the projection is well-defined, first observe two facts:

1. The square with vertices K_0, Z_0, Z_2, K_2 is indeed a pullback in **Net**.
   In fact, by construction, the diagram commutes. Furthermore, in category **Net** the square with vertices K_0, K_3, Z_3, Z_0 is a pullback (since it can be viewed as the composition of two pullbacks K_0, K_1, Z_1, Z_0 and K_1, K_3, Z_1, Z_3). However, the same square is composed out of K_0, K_2, Z_2, Z_0 and K_2, K_3, Z_3, Z_2. Hence, by Lemma 23, the square K_0, Z_0, Z_2, K_2 is also a pullback in **Net**.
2. The upper square with vertices K_0, K_1, K_3, K_2 is a pushout in **Net**.
   In fact, the vertical faces of the cube are pullbacks and the lower face is a pushout, hence, by the 3-cube lemma (Corradini *et al.* 1996), we can conclude that the upper square is a pushout.

Let us prove that $\langle \pi_0, \psi^2 \rangle$ is a well-defined projection of π_2 along f_2 by showing that

$$O^+_{K_0} = {\psi^2_K}^{-1}(O^+_{K_2}) \cup \mathsf{in}(\psi^2_K) \text{ and } O^-_{K_0} = {\psi^2_K}^{-1}(O^-_{K_2}) \cup \mathsf{out}(\psi^2_K).$$

We restrict our attention to the first equality (the second one is proved by symmetric reasoning), and we show the two inclusions separately.

($\subseteq$) Let $s_0 \in O^+_{K_0}$. Since $\langle \pi_0, \psi^1 \rangle$ is a projection of π_1, we have $s_0 \in {\psi^1_K}^{-1}(O^+_{K_1}) \cup \mathsf{in}(\psi^1_K)$. We distinguish two cases:

— Let $s_0 \in {\psi^1_K}^{-1}(O^+_{K_1})$, that is, $\psi^1_K(s_0) \in O^+_{K_1}$. Then, since $\langle \pi_1, \phi^1 \rangle$ is a projection, again, $\psi^1_K(s_0) \in {\phi^1_K}^{-1}(O^+_{K_3}) \cup \mathsf{in}(\phi^1_K)$. If $\phi^1_K(\psi^1_K(s_0)) \in O^+_{K_3}$, then, observing that $\phi^2_K(\psi^2_K(s_0)) = \phi^1_K(\psi^1_K(s_0))$ and recalling that ϕ^2_K is an open net morphism, we conclude that $\psi^2_K(s_0) \in O^+_{K_2}$, and thus $s_0 \in {\psi^2_K}^{-1}(O^+_{K_2})$. If, instead, $\psi^1_K(s_0) \in \mathsf{in}(\phi^1_K)$, then ${}^\bullet\phi^1_K(\psi^1_K(s_0)) - \phi^1_K({}^\bullet\psi^1_K(s_0)) \neq \varnothing$. Since K_1 is an occurrence open net and $\psi^1_K(s_0)$ is input open, we have that ${}^\bullet\psi^1_K(s_0) = \varnothing$. Thus, since the upper square is a pushout,

$$^\bullet\psi^2_K(s_0) - \psi^2_K({}^\bullet s_0) \neq \varnothing.$$

Hence $s_0 \in \mathsf{in}(\psi_K^2)$.

— Let $s_0 \in \mathsf{in}(\psi_K^1)$. Thus there exists $t_1 \in {}^\bullet\psi_K^1(s_0) - \psi_K^1({}^\bullet s_0)$. Since the upper square is a pushout, $\phi_K^1(t_1) \in {}^\bullet\phi_K^2(\psi_K^2(s_0)) - \phi_K^2({}^\bullet\psi_K^2(s_0))$, hence $\psi_K^2(s_0) \in \mathsf{in}(\phi_K^2) \subseteq O_{K_2}^+$, since ϕ_K^2 is an open net morphism. Hence $s_0 \in {\psi_K^2}^{-1}(O_{K_2}^+)$. Observe that, in particular, we have shown that $\psi_K^2(\mathsf{in}(\psi_K^1)) \subseteq O_{K_2}^+$.

($\supseteq$) Let $s_0 \in {\psi_K^2}^{-1}(O_{K_2}^+) \cup \mathsf{in}(\psi_K^2)$. We distinguish two cases:

— Let $s_0 \in {\psi_K^2}^{-1}(O_{K_2}^+)$, that is, $\psi_K^2(s_0) \in O_{K_2}^+$. Since $\langle \pi_2, \phi^2 \rangle$ is a projection of π_3, we have that $\psi_K^2(s_0) \in {\phi_K^2}^{-1}(O_{K_3}^+) \cup \mathsf{in}(\phi_K^2)$. If $\phi_K^2(\psi_K^2(s_0)) \in O_{K_3}^+$, then, since ϕ_K^1 is an open net morphism, $\psi_K^1(s_0) \in O_{K_1}^+$, and thus $s_0 \in O_{K_0}^+$. If, instead, $\psi_K^2(s_0) \in \mathsf{in}(\phi_K^2)$, then ${}^\bullet\phi_K^2(\psi_K^2(s_0)) - \phi_K^2({}^\bullet\psi_K^2(s_0)) \neq \emptyset$. Since the upper square is a pushout, this implies that ${}^\bullet\psi_K^1(s_0) - \psi_K^1({}^\bullet s_0) \neq \emptyset$, and thus $s_0 \in \mathsf{in}(\psi_K^1) \subseteq O_{K_0}^+$.

— Let $s_0 \in \mathsf{in}(\psi_K^2)$. Then ${}^\bullet\psi_K^2(s_0) - \psi_K^2({}^\bullet s_0) \neq \emptyset$. Since the upper square is a pushout, we have that ${}^\bullet\phi_K^1(\psi_K^1(s_0)) - \phi_K^1({}^\bullet\psi_K^1(s_0)) \neq \emptyset$. Since ϕ_K^1 is an open net morphism, $\psi_K^1(s_0) \in O_{K_1}^+$, and thus $s_0 \in O_{K_0}^+$.

To conclude the proof, we need only show that $\psi_K^1 \uparrow \psi_K^2$. We observe that the upper square, which is known to be a pushout in **Net**, is also a pushout in **ONet**. To this end, we prove that, for $x \in \{+,-\}$,

$$O_{K_3}^x = \{s_3 \in S_{K_3} \mid {\phi_K^1}^{-1}(s_3) \subseteq O_{K_1}^x \ \wedge\ {\phi_K^2}^{-1}(s_3) \subseteq O_{K_2}^x\}.$$

Let us consider the condition on input places ($x = +$). Let $s_3 \in O_{K_3}^+$. Then, $\phi_i^{-1}(s_3) \subseteq O_{K_i}^+$ for $i \in \{1,2\}$, since ϕ_i is an open net morphism. For the converse inclusion, assume that

$$ {\phi_K^1}^{-1}(s_3) \subseteq O_{K_1}^+ \quad \text{and} \quad {\phi_K^2}^{-1}(s_3) \subseteq O_{K_2}^+ . \tag{3}$$

Since the upper square is a pushout in **Net**, there is $s_i \in S_i$ (for some $i \in \{1,2\}$) such that $\phi_i(s_i) = s_3$. Assume, without loss of generality, that there exists $s_1 \in S_1$ such that $\phi_K^1(s_1) = s_3$. Hence, by (3), $s_1 \in O_{K_1}^+$. Since π_1 is a projection of π_3, $O_{K_1}^+ = {\phi_K^1}^{-1}(O_{K_3}^+) \cup \mathsf{in}(\phi_K^1)$. If $s_1 \in {\phi_K^1}^{-1}(O_{K_3}^+)$, we conclude. Otherwise, if $s_1 \in \mathsf{in}(\phi_K^1)$, there exists $t_3 \in {}^\bullet\phi_K^1(s_1) - \phi_K^1({}^\bullet s_1)$. Since the upper square is a pushout in **Net**, there are s_2 in K_2 and $t_2 \in {}^\bullet s_2$ such that $\phi_K^2(s_2) = \phi_K^1(s_1) = s_3$ and $\phi_K^2(t_2) = t_3$. Since $s_2 \in {\phi_K^2}^{-1}(s_3)$, by (3) we have that $s_2 \in O_{K_2}^+$, which contradicts the assumption that K_2 is an occurrence net since ${}^\bullet s_2 \neq \emptyset$.

The condition over output places ($x = -$) is dealt with in a symmetric way by exploiting the fact that the occurrence net K_3 is deterministic. This allows us to conclude that $\psi_K^1 \uparrow \psi_K^2$. In fact, this is a necessary condition to ensure that the pushout, computed in **Net** and lifted to **ONet**, gives a deterministic occurrence open net (see Proposition 21). $\square$

The amalgamation and decomposition results for open net processes are summarised in a theorem that establishes a bijective correspondence between the processes of Z_1 and Z_2 that agree on Z_0 and the processes of Z_3. To formulate this result we need some preliminary observations.

Notice that an isomorphism $f : Z_0 \to Z_1$ in **ONet** is an isomorphism $f : \mathcal{F}(Z_1) \to \mathcal{F}(Z_2)$ in **Net** such that $f(O_0^+) = O_1^+$ and $f(O_0^-) = O_1^-$. Let Z be an open net. We say that two deterministic processes of Z, $\pi : K \to Z$ and $\pi' : K' \to Z$ are *isomorphic*, and we write $\pi \simeq \pi'$, if there exists an isomorphism $\rho : K \to K'$ in **ONet** such that $\pi \circ \rho = \pi'$ (in **Net**). In this case we will say that $\rho : \pi \to \pi'$ is a process isomorphism. Observe that this notion of isomorphism is stricter than isomorphism in **Proc**. In fact, $\rho : \pi \to \pi'$ is a process isomorphism iff $\langle \rho, id_Z \rangle$ is an isomorphism in **Proc**.

Let $\pi : K \to Z$ be a process. We use $[\pi]$ to denote the set of processes of Z isomorphic to π, that is, $[\pi] = \{\pi' : K' \to Z \mid \pi' \simeq \pi\}$. Then the set of (isomorphism classes of) processes of Z is denoted by $\mathbf{DProc}(Z)$, that is,

$$\mathbf{DProc}(Z) = \{[\pi] \mid \pi : K \to Z \text{ is a deterministic process}\}.$$

Given a span $Z_1 \xleftarrow{f_1} Z_0 \xrightarrow{f_2} Z_2$ in **ONet**, the isomorphism classes of deterministic processes of Z_1 and Z_2 that agree on Z_0, denoted by $\mathbf{DProc}(Z_1 \xleftarrow{f_1} Z_0 \xrightarrow{f_2} Z_2)$, are the set

$$\{[\pi_1 \xleftarrow{\psi^1} \pi_0 \xrightarrow{\psi^2} \pi_2] \mid \psi^1, \psi^2 \text{ agreement projections for } \pi_1, \pi_2 \text{ along } f_1, f_2\},$$

where isomorphism of process spans is defined by $(\pi_1 \xleftarrow{\psi^1} \pi_0 \xrightarrow{\psi^2} \pi_2) \simeq (\pi_1' \xleftarrow{\phi^1} \pi_0' \xrightarrow{\phi^2} \pi_2')$ if there are process isomorphisms $\rho_i : \pi_i \to \pi_i'$ such that the following diagram commutes:

$$\begin{tikzcd}
 & K_1' \arrow[dl, "\rho_1"'] \arrow[dd, "\pi_1'"] & & K_0' \arrow[ll, "\psi_K^2"'] \arrow[rr, "\psi_K^1"] \arrow[dl, "\rho_0"'] \arrow[dd, "\pi_0'"] & & K_2' \arrow[dl, "\rho_2"'] \arrow[dd, "\pi_2'"] \\
K_1 \arrow[dr, "\pi_1"'] & & K_0 \arrow[ll, "\phi_K^2"'] \arrow[rr, "\phi_K^1"] \arrow[dr, "\pi_0"'] & & K_2 \arrow[dr, "\pi_2"'] & \\
 & Z_1 & & Z_0 \arrow[ll, "f_1"] \arrow[rr, "f_2"'] & & Z_2
\end{tikzcd}$$

Observe that this implies that $\pi_0' \in [\pi_0]$ and that π_1' and π_2' agree on Z_0.

Theorem 28 (Amalgamation theorem). Let Z_0, Z_1, Z_2, Z_3 be as in Figure 9 and assume that the square is a pushout of two composable open net embeddings f_1 and f_2. Then there are composition and decomposition functions:

$$\mathsf{Comp} : \mathbf{DProc}(Z_1 \xleftarrow{f_1} Z_0 \xrightarrow{f_2} Z_2) \to \mathbf{DProc}(Z_3)$$

and

$$\mathsf{Dec} : \mathbf{DProc}(Z_3) \to \mathbf{DProc}(Z_1 \xleftarrow{f_1} Z_0 \xrightarrow{f_2} Z_2)$$

establishing a bijective correspondence between

$$\mathbf{DProc}(Z_3) \text{ and } \mathbf{DProc}(Z_1 \xleftarrow{f_1} Z_0 \xrightarrow{f_2} Z_2).$$

Proof (Sketch). Let us define $\mathsf{Comp} : \mathbf{DProc}(Z_1 \xleftarrow{f_1} Z_0 \xrightarrow{f_2} Z_2) \to \mathbf{DProc}(Z_3)$ by

$$\mathsf{Comp}([\pi_1 \xleftarrow{\psi^1} \pi_0 \xrightarrow{\psi^2} \pi_2]) = [\pi_3],$$

where $\pi_3 = \pi_1 +_{\psi^1,\psi^2} \pi_2$ is the amalgamation of π_1 and π_2 (see Definition 24). Furthermore, $\mathsf{Dec} : \mathbf{DProc}(Z_3) \to \mathbf{DProc}(Z_1 \xleftarrow{f_1} Z_0 \xrightarrow{f_2} Z_2)$ is defined by

$$\mathsf{Dec}([\pi_3]) = [\pi_1 \xleftarrow{\psi^1} \pi_0 \xrightarrow{\psi^2} \pi_2],$$

where $\pi_1 \xleftarrow{\psi^1} \pi_0 \xrightarrow{\psi^2} \pi_2$ is the decomposition of π_3 as defined in Theorem 27. Then it is possible to prove that Comp and Dec are well-defined and inverse to each other. $\square$

Example. The amalgamation theorem is exemplified in Figure 3. Two processes for the component nets $\mathsf{Traveller}$ and Agency that agree on the shared subnet Common, that is, such that their projections over Common coincide, can be amalgamated to produce a process for the composed net Global. Conversely, each process of the net Global can be reconstructed as the amalgamation of compatible processes of the component nets.

Remark. (Amalgamation for ordinary Petri nets). A natural question concerns the possibility of interpreting constructions and results developed for open nets in the setting of ordinary Petri nets. To this end, first consider the full subcategory $\mathcal{A}$ of **ONet** having as objects open nets where each place is both input and output open, that is, open nets of the kind

$$Z = (N_Z = (S_Z, T_Z, \sigma_Z, \tau_Z), O_Z = (S_Z, S_Z)).$$

It can be seen immediately that if Z_1 and Z_2 are open nets in $\mathcal{A}$, then any **Net**-morphism $f : N_1 \to N_2$ is also an open net morphism $f : Z_1 \to Z_2$. Therefore **Net** is isomorphic to $\mathcal{A}$ and, with a little abuse of notation, from now on it will be identified with $\mathcal{A}$ itself. From this point of view, observe that:

— Let $\mathbf{Proc}_{\mathbf{Net}}$ be the full subcategory of **Proc** where objects are of the kind $\pi : K \to Z$ with Z in **Net** and K an open occurrence net where all minimal places are input open and all maximal places are output open. Then it can be shown that $\mathbf{Proc}_{\mathbf{Net}}$ consists exactly of the ordinary Petri net processes (Golz and Reisig 1983).

— Any span $Z_1 \xleftarrow{f_1} Z_0 \xrightarrow{f_2} Z_2$ in **Net** is composable.

— Take a span $f_1 : Z_0 \to Z_1$ and $f_2 : Z_0 \to Z_2$ in **Net** and consider the corresponding composition that, by Proposition 18, is given by the pushout of f_1 and f_2 in **Net** (see Figure 9).

Given two processes $\pi_1 : K_1 \to Z_1$ and $\pi_2 : K_2 \to Z_2$, the notion of agreement over Z_0 reduces to the existence of projections $\langle \pi_0, \psi^i \rangle$ along f_i of π_i, $i \in \{1, 2\}$, such that

- $\mathsf{in}(\psi^1_K) \cap \mathsf{in}(\psi^2_K) = \varnothing$ and $\mathsf{out}(\psi^1_K) \cap \mathsf{out}(\psi^2_K) = \varnothing$;
- the relation $<_{1,2}$ induced over Z_0 is acyclic.

Then, as in the general case, two processes π_1 and π_2 that agree on Z_0 can be composed to produce a process π_3 of Z_3, and, conversely, any process of Z_3 can be obtained as the composition of processes of Z_1 and Z_2.

6. Related work

In the field of Petri nets, several other approaches to net composition have been proposed in the literature. Most of them can be classified as algebraic approaches. One family, which dates back to the papers Nielsen *et al.* (1981) and Winskel (1987a), considers a category of Petri nets where morphisms arise by viewing a Petri net as the signature of a multisorted algebra, the sorts being the places. Then an unfolding semantics is defined, which is characterised categorically as a right adjoint. This fact ensures its compositionality with respect to operations on nets defined in terms of categorical limits (for example, net synchronisation (Winskel 1987b)). The algebraic view is pushed forward in another seminal paper, Meseguer and Montanari (1990), where a Petri net is still seen as a signature, and its computational model (the category of deterministic processes in the sense of Best and Devillers (1987)) is characterised as the free algebra (up to suitable axioms) over such a signature. Being obtained as a free construction, which in categorical terms provides a left adjoint, in this case the semantics is compositional with respect to operations defined in terms of colimits. However, in both cases, unlike what happens in our approach, there is no distinction between open and internal places. Basically, every place of a net N can be seen implicitly as open because it can be used for connecting N to other nets. On the other hand, the semantics (for example, the notions of process in Golz and Reisig (1983) or Meseguer and Montanari (1990)) does not explicitly take into account the interaction with the environment.

A second, more recent class of approaches to Petri net composition aims at defining a 'calculus of nets', where a set of process algebra-like operators allows one to build complex nets starting from a suitable set of basic net components.

For instance, in the Petri Box calculus (Best *et al.* 1992; Koutny *et al.* 1994; Koutny and Best 1999) a special class of nets, called *plain boxes* (safe and clean nets), provides the basic components. Plain boxes are then combined by means of operations that can all be seen as instances of refinements over suitable nets. More precisely, the authors identify a special family of nets, called *operator boxes*. An operator box with n transitions induces an n-ary composition operation over plain boxes. Its effect is to simultaneously refine the n transitions of the operator box with the plain boxes given as argument, thus producing a net that is again a plain box. The calculus is then given a compositional semantics (both interleaving and concurrent). A very interesting aspect of this approach is the fact that it does not concentrate on a specific algebra of Petri nets, but it develops a general theory, which is, in a sense, parametric with respect to the operators and constants of the algebra. These constants and operators, in fact, are not fixed once and for all, but they can be designed according to specific needs, by appropriately choosing the sets of *plain* and *operator boxes*.

Another relevant approach in the second family is presented in the papers Nielsen *et al.* (1995) and Priese and Wimmel (1998), which introduce an algebra of (labelled) Petri nets with interfaces. An interface consists of a set of public places and transitions, where a net can be extended and combined with other nets by means of composition operators. For example, it is possible to add new transitions and places to connect existing public transitions and places by new arcs to hide items in the net, and so on. These operators can be used as basic constructors to build terms corresponding to nets with an interface. The representation of a Petri net *via* a term of the algebra of combinators resembles the encoding of Petri nets into Milner action calculi (Milner 1996). The *pomset semantics* of nets with interfaces, defined by using a notion of universal context for a net, is shown to be compositional with respect to the net combinators (Priese and Wimmel 1998).

The two approaches mentioned certainly share several ideas and technical features with ours, such as the use of interface places (called *entry* and *exit* places, in Best *et al.* (1992)) or the use of a universal context in Nielsen *et al.* (1995) and Priese and Wimmel (1998), which is similar to the closure of an open net, which underlies our open net semantics. However, some basic differences prevent us from making the comparison on a formal level. In our case the basic building blocks of an open system are the *transitions*, with a fixed pre- and post-set. Some places, designated as 'open', represent the system interface towards the environment. Then two systems can be combined by means of a construction that glues them along a common part consistently with open places in a way that does not change the shape of the original transitions. Intuitively, one can also think of the composition operation as a way of making explicit (part of) the unspecified environment of each of the component nets. The composition operation in Best *et al.* (1992) mainly relies on net refinement. Concentrating on a subclass of net components with suitable properties (plain boxes), it offers a powerful way of defining a kind of process algebra over such nets, with operators like sequential and parallel composition, non-deterministic choice, relabelling and synchronisation. The composition is, in a sense, realised at a more semantical level, in that the internal structure of the components (for example, the transitions and their connections) can be changed by the operation that combines their functionalities. As for the approach in Nielsen *et al.* (1995) and Priese and Wimmel (1998), the main difference, besides the focus, which in these papers is more on the Petri net algebra, lies in the fact that 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. Such components are then combined by means of constructors that allow one to connect places and transitions.

Finally, we should mention two approaches to *Petri net components*, that is, Petri nets with distinguished interface places. Kindler (1997) introduced Petri net components with input and output places, which can be combined by means of an operation that connects the input places of a component to the output places of the other, and *vice-versa.* A partial order semantics is introduced for components and it is proved to be compositional. Components can be viewed as particular open nets and, similarly, the composition operation for components can be seen as an instance of the composition operation for open nets. A very interesting idea in Kindler (1997), which could also be worth exploring for open nets, is the introduction of a temporal logic, interpreted over processes, which can be used for reasoning in a modular way over distributed systems.

Basten (1998) considers components of Petri nets with interface places, called pins, of unspecified orientation, where nets can be fused together. A compositional operational semantics of Petri net components is described within a process algebra specifically designed for this purpose. This allows the verification of net components against requirements by means of equational reasoning. Moreover, the algebraic presentation of the operational semantics is used to formalise a notion of behaviour inheritance between components.

7. Conclusions and future work

In this paper we have introduced *open nets* as an extension of ordinary Petri nets that allows one to specify open concurrent systems interacting with an external environment. Open nets are endowed with a composition operation, which is suitable for modelling both interaction through open places and the synchronisation of transitions. The generalisation to open nets of the Goltz–Reisig process semantics has been shown to be compositional with respect to the composition operation over open nets: if two nets Z_1 and Z_2 are composed, producing a net Z_3, then the processes of Z_3 can be obtained as amalgamations of processes of Z_1 and Z_2, and, conversely, any process of Z_3 can be decomposed into processes of the component nets. The amalgamation and decomposition operations are shown to be inverse to each other, leading to a bijective correspondence between the processes of Z_3 and the pair of processes of Z_1 and Z_2 that agree on the common subnet Z_0.

As mentioned in the introduction, the last result appears to be related to the amalgamation theorem for data-types in the framework of algebraic specifications (Ehrig and Mahr 1985). There, an amalgamation construction allows one to 'combine' any two algebras A_1 and A_2 of algebraic specifications $SPEC_1$ and $SPEC_2$ having a common subspecification $SPEC_0$ if and only if the restrictions of A_1 and A_2 to $SPEC_0$ coincide. The amalgamation construction produces a unique algebra A_3 of specification $SPEC_3$, which is the union of $SPEC_1$ and $SPEC_2$. The fact that the amalgamation of algebras is a pushout construction in the Grothendick's category of generalised algebras suggests the possibility of having a similar characterisation for process amalgamation using fibred categories (see also Remark 15).

Open nets have been partly inspired by the notion of *open graph transformation system* (Heckel 1998), which is an extension of graph transformation for specifying reactive systems. In fact, P/T Petri nets can be seen as a special case of graph transformation systems (Corradini 1996), and this correspondence extends to open nets and open graph transformation systems. However, a compositionality result corresponding to Theorem 28 is still lacking in this more general setting.

The notions of projection, agreement, amalgamation and decomposition of processes can be extended in a natural way to general (possibly non-deterministic) processes. However, unlike what happens in the deterministic case, not every non-deterministic process of a composed net can be obtained as the amalgamation of processes of the component nets. For instance, consider net Z_3 in Figure 13, which arises as the composition of Z_1 and Z_2 along Z_0. The process π_3 of net Z_3, depicted in the middle of the picture, cannot be obtained as the amalgamation of processes of the component nets Z_1 and Z_2. In fact, let π_1 and π_2 be processes of Z_1 and Z_2, respectively, such that for $i \in \{1, 2\}$, π_i consists only of a transition t_i. To be able to amalgamate π_1 and π_2 by fusing the pre-sets of t_1 and t_2, both processes must necessarily consider an interaction with the environment, that is, in both processes the pre-set of transition t_i must be an output open place (graphically, π_1 and π_2 would be represented exactly as nets Z_1 and Z_2). Hence, in the process π'_3 resulting from their composition, place s_3 in the pre-set of the t_i's will also be output open (see the right-hand part of Figure 13). Roughly, since both π_1 and π_2 are open to interactions with the environment, the result of their composition is still open. This example suggests that in the non-deterministic case one should expect a weaker compositionality result, stating that for any processes π_3 of a composed net, a suitable amalgamation of the projections of π_3 results in a new process π'_3, which coincides with π_3 except for the fact that π'_3 can exhibit a more general interaction with

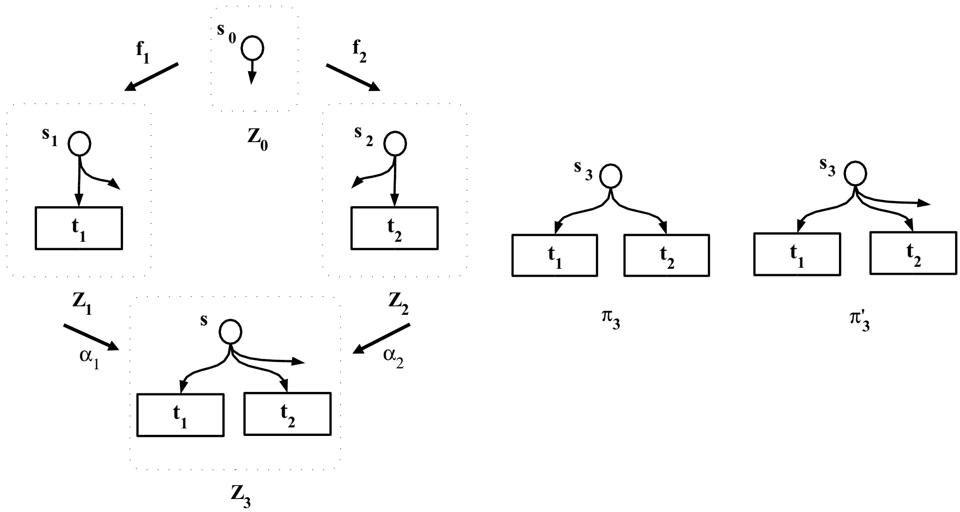


Fig. 13. Composition of non-deterministic processes.

the environment. The generalisation of the amalgamation theorem to non-deterministic processes could represent a first step towards an unfolding semantics for open nets, in the style of Winskel (Nielsen *et al.* 1981; Winskel 1987a), that is still compositional with respect to our composition operation.

It would also be interesting to extend the constructions and results in this paper to open *high level nets*, which have already been studied on a conceptual level in Padberg *et al.* (1998). Part of the technical background is already available – for instance, Padberg *et al.* (1995) shows how to construct pushouts of algebraic high level nets – but a suitable formalisation of high level processes is still missing.

Acknowledgements

We are grateful to the anonymous reviewers for their insightful suggestions on the submitted version of this paper. We are also indebt to Julia Padberg for pointing out that the characterisation of the pushout construction for open nets requires the morphisms in the span to be embeddings, which is on hypothesis that was missing in Baldan *et al.* (2001).

References

Baldan, P., Corradini, A., Ehrig, H. and Heckel, R. (2001) Compositional modeling of reactive systems using open nets. In: Larsen, K. G. and Nielsen, M. (eds.) Proceedings of CONCUR 2001. *Springer-Verlag Lecture Notes in Computer Science* **2154** 502–518.

Basten, T. (1998) *In terms of nets: System design with Petri nets and process algebra*, Ph.D. thesis, Eindhoven University of Technology.

Best, E. and Devillers, R. (1987) Sequential and concurrent behaviour in Petri net theory. *Theoretical Computer Science* **55** (1) 87–136.

Best, E., Devillers, R. and Hall, J. G. (1992) The Petri box calculus: a new causal algebra with multi-label communication. In: Rozenberg, G. (ed.) Advances in Petri Nets. *Springer-Verlag Lecture Notes in Computer Science* **609** 21–69.

Corradini, A. (1996) Concurrent graph and term graph rewriting. In: Montanari, U. and Sassone, V. (eds.) Proceedings of CONCUR'96. *Springer-Verlag Lecture Notes in Computer Science* **1119** 438–464.

Corradini, A., Ehrig, H., Löwe, M., Montanari, U. and Padberg, J. (1996) The category of typed graph grammars and its adjunctions with categories of derivations. In: Cuny, J., Ehrig, H., Engels, G. and Rozenberg, G. (eds.) Proceedings of the 5th International Workshop on Graph Grammars and their Application to Computer Science. *Springer-Verlag Lecture Notes in Computer Science* **1073**.

Ehrig, H. and Mahr, B. (1985) *Fundamentals of Algebraic Specification 1*, Springer-Verlag.

Golz, U. and Reisig, W. (1983) The non-sequential behaviour of Petri nets. *Information and Control* **57** 125–147.

Heckel, R. (1998) *Open Graph Transformation Systems: A New Approach to the Compositional Modelling of Concurrent and Reactive Systems*, Ph.D. thesis, TU Berlin.

Jacobs, B. (1999) *Categorical Logic and Type Theory*, Studies in Logic and the Foundations of Mathematics **141**, North Holland.

Kindler, E. (1997) A compositional partial order semantics for Petri net components. In: Azema, P. and Balbo, G. (eds.) Application and Theory of Petri Nets. *Springer-Verlag Lecture Notes in Computer Science* **1248** 235–252.

Koutny, M. and Best, E. (1999) Operational and denotational semantics for the box algebra. *Theoretical Computer Science* **211** (1–2) 1–83.

Koutny, M., Esparza, J. and Best, E. (1994) Operational semantics for the Petri box calculus. In: Jonsson, B. and Parrow, J. (eds.) Proceedings of CONCUR '94. *Springer-Verlag Lecture Notes in Computer Science* **836** 210–225.

Meseguer, J. and Montanari, U. (1990) Petri nets are monoids. *Information and Computation* **88** 105–155.

Milner, R. (1996) Calculi for interaction. *Acta Informatica* **33** 707–737.

Nielsen, M., Plotkin, G. and Winskel, G. (1981) Petri Nets, Event Structures and Domains, Part 1. *Theoretical Computer Science* **13** 85–108.

Nielsen, M., Priese, L. and Sassone, V. (1995) Characterizing Behavioural Congruences for Petri Nets. In: Proceedings of CONCUR'95. *Springer-Verlag Lecture Notes in Computer Science* **962** 175–189.

Padberg, J., Ehrig, H. and Ribeiro, L. (1995) Algebraic high-level net transformation systems. *Mathematical Structures in Computer Science* **5** (2) 217–256.

Padberg, J., Jansen, L., Heckel, R. and Ehrig, H. (1998) Interoperability in train control systems: Specification of scenarios using open nets. In: *Proc. IDPT*, Society for Design and Process Science 17–28.

Priese, L. and Wimmel, H. (1998) A uniform approach to true-concurrency and interleaving semantics for Petri nets. *Theoretical Computer Science* **206** (1–2) 219–256.

Reisig, W. (1985) *Petri Nets: An Introduction*, EACTS Monographs on Theoretical Computer Science, Springer-Verlag.

van der Aalst, W. (1998) The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computers* **8** (1) 21–66.

van der Aalst, W. (1999) Interorganizational workflows: An approach based on message sequence charts and Petri nets. *System Analysis and Modeling* **34** (3) 335–367.

Winskel, G. (1987a) Event Structures. In: Petri Nets: Applications and Relationships to Other Models of Concurrency. *Springer-Verlag Lecture Notes in Computer Science* **255** 325–392.

Winskel, G. (1987b) Petri nets, algebras, morphisms, and compositionality. *Information and Computation* **72** (3) 197–238.