Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 308 (2003) 145 – 197

Theoretical Computer Science

www.elsevier.com/locate/tcs

Clustering for Petri nets

Walter Keller

Steinackerweg 33, CH-8304 Wallisellen, Switzerland

Received 16 June 1999; received in revised form 28 July 2002; accepted 19 August 2002
Communicated by U. Montanari

Abstract

This work builds a bridge from

- clustering techniques—merging neighbouring nodes which is a key feature for software engineering and the practical applications of Petri nets—to
- folding techniques—merging only transitions with transitions and places with places, preserving behaviour and allowing theoretical connections to many models of concurrency.

A new category of Petri nets is introduced. Morphisms support clustering, offering attractive properties to software engineering and integrating smoothly with invariants. A computationally reasonable adjunction connects it to folding-based Petri nets, namely, to two new cocomplete and complete categories. The dichotomy of structure and behaviour of Petri nets is expressed as compatible adjunctions to behavioural categories. Finally reachability and process semantics are attached categorically and a new variant of occurrence nets is proposed as a purer image of causality and branching. This framework offers categorical support for practical applications of Petri nets.

Keywords: Petri nets; Folding; Clustering; Structure; Behaviour; Semantics; Category theory

1. Introduction

This paper is a concentrated version of the Petri net theoretical part of the author's Ph.D. thesis [10] that explores the potential of Petri nets as a tool for reverse engineering. The focus on conventional applications means using Petri nets in a field where they cannot play their natural strength to model concurrency.

E-mail address: wkeller@tiscalinet.ch (W. Keller).

0304-3975/03/$ - see front matter © 2002 Elsevier B.V. All rights reserved.
PII: S0304-3975(02)00597-2

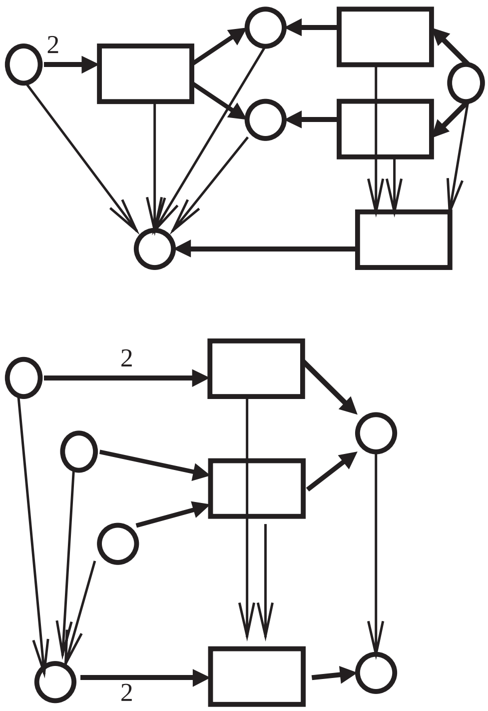


Fig. 1. Clustering (top) and folding (bottom). Fat arrows show net-arcs, (bunches of) thin arrows morphisms.

The idea to use Petri nets for software engineering was—to say the least—a bit outside of mainstream research. The task of reverse engineering is to recover software structures—ideally reusable components. But the lack of a commonly accepted compositional semantics (e.g. how do semantics change if two nets are combined or a token is added to the initial marking?) is generally felt as a weakness of Petri nets.

In fact the research yielded some surprising results. A manner to express this is by elaborating the differences between two different kinds of modelling: folding and clustering (see Fig. 1).

In clustering, directly connected entities are grouped together. Neighbourhoods are collapsed into a single object which also swallows relationships. Foldings, on the other hand, merge similar objects and, separately, similar relationships. They group functionality and not adjacency. A model object stands for similar objects and hence inherits many properties.

Clustering respects vicinity and is the usual method of breaking complex systems down into subsystems. This is typical for software engineering (e.g. [21, p. 48]). A web of collaborating active and passive elements is abstracted as a unit capable of delivering certain services. Such a unit is called (sub)program, module, object, package, etc. and is used as a building block to compose larger systems. But for Petri nets, a clustered node is neither a transition nor a place and, thus, not a Petri net concept. Folding is, therefore, preferred for theoretical purposes but clustering is indispensable in practice and especially for reverse engineering. Furthermore, coloured Petri nets are a very convenient representation of foldings [10].

In Petri nets both principles are used (Section 1.1) but in separated areas. For our application of Petri nets to reverse engineering we wanted to combine the practical relevance of clustering-based approaches with the theoretical attractiveness of folding-based methods. But we could not find such a combination in literature. Now, the main results of our Ph.D. thesis are:

- categorical means to bridge from clustering to folding in Petri nets,
- folding-based Petri net algorithms for reverse engineering and Petri nets as an engineering metaphor.

This paper reports results from the first point.

Folding-based morphisms transfer behaviour very nicely and a rich set of powerful categorical connections to many models of concurrency have been detected—giving deep theoretical insights (e.g. [27,16,25,19]). Such connections are missing in clustering-based categories. The unfolding of a coloured or hierarchical net into the flat net (e.g. [9]) is not a contradiction to this statement—it is an internal or definitional transformation rather than semantics.

Hence, the categorical bridge built from clustering to folding in this work is new and widens the applicability of categorical Petri net theory because it allows the combination of folding techniques with clustering techniques. First, CPNs (a well-known kind of coloured nets [9]) are tightly related to comma categories of special foldings. Hence, we propose to enhance existing Petri Net tools (e.g. using CPN) with the categorical machinery presented here, e.g. morphisms for simulation or implementation/design relationships, universal constructions for flexible compositions of subsystems and functors for the transfer of semantics from one environment to another. Secondly, we use the presented Petri net categories for reverse engineering [11]. An iteration of couniversal constructions extracts a high-level design from a net which represents low-level information of an existing system.

Most Petri net classes use multisets. This is so widely accepted as natural that it is seldom put in question. But multisets lack universal constructions, which normally propagate to categories built on top of them. As multiset morphisms, net morphisms usually map a place to a marking (e.g. [26]). Categories with such morphisms allow a powerful transfer of behaviour, e.g. by the functors and adjunctions in [19]. However, they neglect a principle of net theory—the interplay of structure and behaviour. Such a morphism abstracts places to markings. But for practical work the operations on the graph (e.g. the places) are essential for graph transformations, for visualisations, etc.. Furthermore, these morphisms map places to markings, but transition to transitions. This asymmetry is a problem for the clustering of places with transitions. To remedy these inconveniences we introduce the new category $\underline{1S}$ of one-sets (Definition 2.2). It has the same objects as multisets but morphisms are restricted.

Place–transition nets—the objects of the category $\underline{PTNET}$ (Definition 3.8)—are defined in the standard way but morphisms contain novel features. They are a combination of

- 1S morphisms,
- vicinity preserving graph morphisms,
- foldings commuting with the pre- and post-function.

Interpretation of a morphism between graphs as an implementation/specification relation leads to the interpretation of the origins of a node as a subsystem. For place–transition morphisms the origins of a transition form a subnet with proper port transitions—similar to many net classes with compositional features. The origins of a place form a subnet from which (essentially) only places are connected to the outside. Hence, there are clustering capabilities useful for both software engineering and net theory.

Net invariants are an important tool for Petri nets because they allow us to deduce behavioural properties from the structure. A morphism transfers a place invariant backward from the destination to the source net and a transition invariant forward. A semi-positive place invariant gets simply a morphism to a net consisting of a single place while a semi-positive transition invariant is the image of a T-system. Thus, morphisms integrate well with linear algebraic techniques and yield a convincing reinterpretation of invariants.

PPNET (Definition 4.19) is the key category of our categorical approach. It supports clustering in a restricted form (places are mapped only to places), and by an adjunction in the more general form of PTNET. But it is rigorous enough to enjoy nice categorical features. All limits and colimits exist (Proposition 4.23). Hence, we may compose nets by universal constructions and use graph transformation techniques, e.g. the simple or double pushout approach [4,24].

PPNET is the subcategory of PTNET with the same objects but only place-preserving morphisms which do not map places to transitions. The connecting adjunction (Proposition 4.21) allows to simulate a net of PTNET in PPNET. This is both simple and computationally reasonable—three nodes are added for every transition in the original net. A morphism in PPNET yields a simulation that maps markings to markings and transition-steps to steps (in a compatible way) whereas, in PTNET the phenomenon of ‘transition in occurrence’ emerges (i.e. a transition has already consumed the inputs but not yet produced the outputs).

In the dichotomy of clustering and folding, PPNET sits in the middle and so the adjunction from PTNET looks like the first half of a bridge from clustering to folding. A similar construction yields the second half of the bridge and the two adjunctions compose to an adjunction from PTNET to FNET the category of nets with foldings (Definition 4.24).

The first step to catch the behaviour of a Petri net is to add an initial marking which enables the token game: transition occurrences consume and produce tokens. Because morphisms should map enabled steps to enabled steps they have to preserve the initial marking. Again we deviate from the usual definition: the image of the initial marking may be less or equal than the initial marking in the destination net—not only equal (categories PTSYS, PPSYS and FSYS in Definition 5.27).

This deviation is not a big change for simulation: some tokens of the initial marking just remain immobile. But it is important for composition. For example, Fig. 2 shows

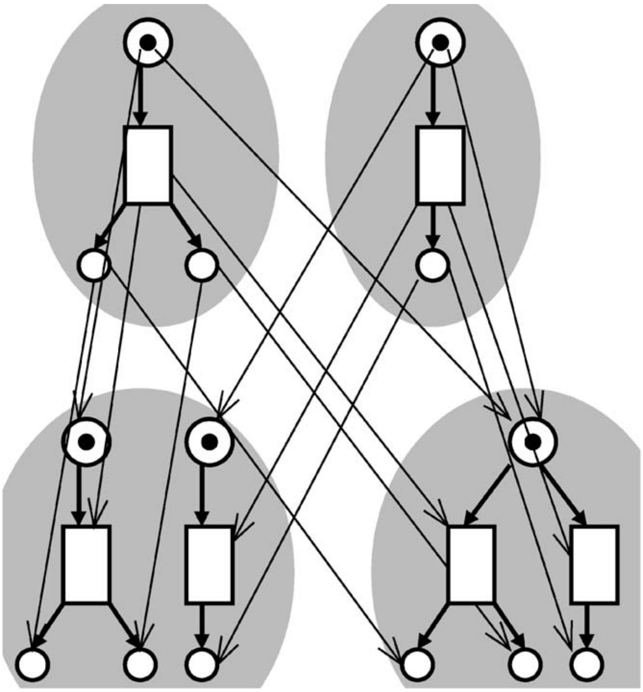


Fig. 2. Two systems (top), their parallel composition (bottom left) and selective composition (bottom right).

that morphisms may now model the parallel composition of two systems whereas the usual definition only allows selection.

Furthermore, this generalisation allows an adjunction between nets and systems, the adjunctions between PTNET, PPNET and FNET lift to adjunctions between the corresponding system categories PTSYS, PPSYS and FSYS and all these adjunctions form a commutative diagram (Fig. 3). Altogether, this shows an impressive web of categorical connections between behaviour oriented systems and structure oriented nets. This is the categorical illustration of a basic Petri net principle: the interplay between static structure and dynamic behaviour. We are not aware of any previous work formulating this categorically. Especially, the net categories of both [26,16], define nets with initial markings, i.e. systems in our terms. This might seem a small difference, but if nets without initial marking are not formalised, they are not first class citizens. E.g. it gets difficult to reason about a net with different initial markings which is a basic task for many applications.

The last two sections show two examples of how the developed framework deals with semantics. First, step reachability gives a functor from place-preserving systems to state machines (category SM, Definition 6.32). It allows us to transfer properties like liveness or boundedness (Proposition 6.35). This functor has no adjoint, the price for our decision that morphisms should keep contact with the underlying graph.

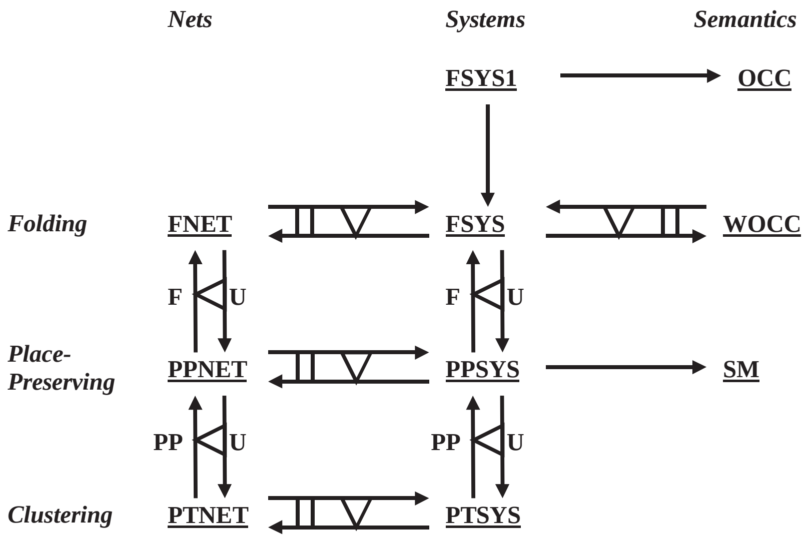


Fig. 3. The synopsis of categories and adjunctions.

Process-based semantics (e.g. [3]) fit well in the framework because a process is a kind of morphism. First the unfolding of a system of FSYS to a new category WOCC of weighted occurrence systems (Definition 7.38) is a functor which is part of a coreflection. The unfolding contains all processes of the system. The unfolding in the category OCC of (safe) occurrence systems (Definition 7.38) yields a functor only for the subcategory FSYS1 (systems with the initial marking a set, Definition 7.44) and yields a coreflection for the subcategory of semi-weighted systems. Ignoring the different definitions of morphisms, this is similar to [19]. But, we have no adjunction to OCC corresponding to that in [14] which requires only a restriction on the morphisms.

A comparison of these three semantics first shows that the expressive power of the unfolding into WOCC is strictly stronger than that of the unfolding into OCC which again is strictly stronger than that of process semantics. Secondly, under the dichotomy of individual and collective token philosophy [8] process semantics and the unfolding into OCC uses individual tokens, i.e. each occurrence of a transition produces new individual tokens. The unfolding into WOCC, however, individualises tokens iff they have a different causal history. This is neither individual nor collective token philosophy, rather, it represents a position between these two poles that reflects causality and branching. In this sense, the unfolding to WOCC gives the pure semantics of causality and branching for anonymous tokens whereas the unfolding to OCC represents individual token philosophy.

The diagram of categories, functors and adjunctions in Fig. 3 summarises this work. Clearly visible are

- the bridge from clustering to folding (PTNET over PPNET to FNET),

- the dichotomy of structure ($\underline{*\mathrm{NET}}$) and behaviour ($\underline{*\mathrm{SYS}}$),
- the connections to semantics ($\underline{\mathrm{SM}}$, $\underline{\mathrm{OCC}}$ and $\underline{\mathrm{WOCC}}$).

1.1. Literature review

There are many definitions of net morphisms in the literature. Let us start with definitions maintaining a strong relationship to the underlying graph and supporting clustering. Genrich et al. [7] define morphisms as a map $f:X\rightarrow X'$ mapping neighbours in X to neighbours in X' and retaining the transition/place property, if the nodes are not collapsed. This definition is usual for graphs and it is a special case of our place–transition morphism. Christodoulakis and Moritz [2] sketch the use of such morphisms for software engineering—we explore another facet of this approach by the application of place–transition morphisms to clustering. In the outlook of [2] the relationship between morphisms and invariants is mentioned—a question which we answer for our flavour of morphism.

A major inspiration for our work was Fehling [6]. Here, the same vicinity respecting homomorphisms are used to define refinement and abstraction. Refinement is the source of a morphism which is surjective on the arcs and the nodes whereas the image is the abstraction. A graph of such refinements builds a hierarchical net and a collection of net morphisms forms a hierarchical morphism, turning hierarchical nets into a category.

Lakos [12] also combines clustering with behaviour transfer. He uses much more complicated definitions for morphisms—partially due to the explicit handling of colours. There are some similarities with the present work, e.g. the transfer of place invariants, but the focus is different: Lakos [12] adds many requirements to the morphism definitions to finally reach the envisioned properties—whereas the current work uses simple definitions and investigates categorical relationships between them.

The other line of morphism—that we call folding-based—considers a Petri net as a two sorted algebra—with multisets of places and multisets of transitions—pre and post as unary operators and the initial marking as a constant. A net morphism then is simply an algebra morphism which preserves the operators. Winskel [26] introduced such a definition together with safe occurrence systems as the semantics, connecting the two categories by a coreflection. This needs some restrictions most notably nets have to be safe. This restriction was softened in [14]. Note, that the definition disallows a node to collapse with its neighbours. Hence clustering is not supported. Further adjunctions are described in [18,15,25] (see [19] for an overview). They exemplify the far reaching connections such folding-based morphism allow—very attractive from a theoretical point of view.

Mukund [16] strengthens the morphism definition of Winskel [26]. The multiset morphism on places must be induced by a partial function in the reverse direction of the morphism (whereas our morphisms retract to a partial function in forward direction). This allows a coreflection between Petri nets and step reachability which lacks in our approach.

But the categorical framework also starts to attack more concrete tasks. For example Nielsen and Cheng [17] gives an elegant categorical description of many variations of bisimulation and other forms of behavioural equivalence. This opens the possibility to prove such an equivalence of concrete nets in the language of category theory.

Folding-based morphisms easily extend to algebraic high level nets. Padberg et al. [20] use categorical constructions in such nets to prove safety properties. They are applied in a real world software project using a combination of Petri nets, algebraic specification and algebraic graph transformation [4,24].

As predecessors of our bridge from folding to clustering we are only aware of the following two generalities:

- Although clustering (e.g. decomposition in modules with high cohesion and low coupling) is predominant in software engineering, foldings are also used, e.g. in layered models or non-functional requirements.
- Most Petri net tools support different forms of clustering (e.g. hierarchies) and foldings (e.g. colours)—it is simply unavoidable for engineering.

However, most (newer) Petri net categories use morphisms that are pure foldings. The widespread feeling that category theory is too theoretical for practical work is supported by the lack of clustering in most Petri net categories. This work aims to reduce both, this lack and this feeling.

1.2. Notation

For a general introduction to Petri nets refer to [23] for category theory to [13]. The following summarises the used notation:

Numbers

$\mathbb{N}$	the natural numbers including zero
$\mathbb{N}^+$	the naturals excluding zero
$\mathbb{Z}$	the integers
$[x, y]$	the interval $\{x, x+1, x+2, \ldots, y\}$ of integers from x to y
$\|s\|$	the cardinality of a set or a multiset for example $\|x+2y\| = 3$

Functions

$f : X \to Y$	a function, morphism, functor or natural transformation from X to Y
$g\, f\, x$	multiplication, function composition and functor application may be written with or without operator and or brackets. Functions are lifted to multisets or power sets without special notation.
$\mathrm{src}(f)$	the source X for $f : X \to Y$
$\mathrm{dst}(f)$	the destination Y
$\mathrm{def}(f)$	the elements of $\mathrm{src}(f)$ on which the partial function f is defined
$\mathrm{im}(f)$	the image of f, a subset of $\mathrm{dst}(f)$
Δ	the diagonal operator $\Delta x = (x, x)$

Sets, relations, multisets

$\exists!$	the unique existence quantifier: there is one and only one
$A \times B$	the cartesian product of two sets A and B
eRf	$(e, f) \in R$ for a relation $R \subseteq E \times F$
$[x]_R$	the equivalence class of x for the equivalence relation R

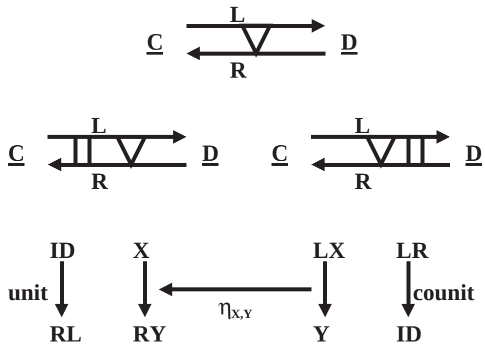


Fig. 4. Symbols for adjunction (top) and coreflection (middle left), reflection (middle right) and mnemonics (bottom).

eR	$\{f \in F \mid e R f\}$ for a relation $R \subseteq E \times F$ or $\{e\} \times R$ or $\{er \mid r \in R\}$ for a set R
R^*	the reflexive and transitive closure of a relation R
<u>SETP</u>	the category of sets and partial functions
Multiset	a finite linear combination of elements of a base set B, written as $\sum_i \lambda_i b_i$ with $\lambda_i \in \mathbb{N}$ and $b_i \in B$. We use the usual addition and multiplication with a natural. In this paper, the base set may be infinite, however, the multisets are always finite sums
$m(b)$	the cardinality of the element b in a multiset m in function notation
supp(m)	$\{b \mid m(b) > 0,\ b \in B\}$ the support of a multiset m over B
Linear	a function between two semigroups (e.g. appropriate sets of multisets) is linear if it fulfils $f(m + m') = f\,m + f\,m'$ and consequently $f\,\lambda m = \lambda f\,m$ for every natural λ
<u>MS</u>	the category of multisets. An object consists of all finite multisets over a given base set. A morphism is a linear function $M \to M'$.
MS	The functor <u>SETP</u> $\to$ <u>MS</u> maps a set to the multisets over it and a partial function f to its linear extension by $(\mathrm{MS}\,f)(\sum_i \lambda_i b_i) = \sum_i \lambda_i f(b_i))$

Petri nets

${}^\bullet x$	the pre-set of a node x
$x^\bullet$	the post-set of a node x
$m[\sigma > m'$	the occurrence of a step σ leading from marking m to m'

Categories: An adjunction is symbolised by two functor arrows connected by a triangle (Fig. 4 top). The triangle points from the left adjoint L to the right adjoint R and is also the direction of unit, counit and morphisms which are transferred by the natural equivalence η.

For a coreflection the adjunction symbol is decorated with a vertical equal sign at the side of the unit which consists of all isomorphisms (Fig. 4). A reflection gets an equal sign at the side of the counit.

$$\begin{tikzcd}
\mathbf{M} \arrow[r, "\mathbf{f'}"] \arrow[d, "\mathbf{f''}"'] & \mathbf{M'} \arrow[d, "\mathbf{p'}"'] \arrow[ddr, "\mathbf{q'}"] & \\
\mathbf{M''} \arrow[r, "\mathbf{p''}"] \arrow[drr, "\mathbf{q''}"'] & \mathbf{P} \arrow[dr, "\exists ! \mathbf{q}"'] & \\
 & & \mathbf{Q}
\end{tikzcd}$$

$$\mathbf{f'} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad \mathbf{f''} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \mathbf{f^*} = \begin{pmatrix} \mathbf{f'} \\ \mathbf{f''} \end{pmatrix}$$

$$\mathbf{p'} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad \mathbf{p''} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 2 \end{pmatrix} \quad \mathbf{p^*} = \begin{pmatrix} \mathbf{p'} & \mathbf{p''} \end{pmatrix}$$

$$\mathbf{q^*} = \begin{pmatrix} \mathbf{q'} & \mathbf{q''} \end{pmatrix}$$

Fig. 5. Although P is the pushout for $\mathbb{Z}$-modules, the pushout does not exist in $\underline{\text{MS}}$. Morphisms are given by their matrices and elements are column-vectors of appropriate dimension.

2. One-sets

This section defines a subcategory of multisets, called one-sets, in which all (co)universal constructions exist and morphisms keep contact with the underlying graph. Neither is true for multisets:

Lemma 2.1. *Let $M \cong \mathbb{N}^1$, $M' \cong M'' \cong \mathbb{N}^2$. Then f' and f'' defined in Fig.* 5 *lack a colimit in* $\underline{\text{MS}}$.

Proof. With $M \cong \mathbb{Q}^1$, $M' \cong M'' \cong \mathbb{Q}^2$ and $P \cong \mathbb{Q}^3$ Fig. 5 is a diagram in the category of vector spaces. The proof proceeds in three steps:
(i) (p', p'') is the colimit of (f', f'') for vector spaces,
(ii) a colimit of (f', f'') in $\underline{\text{MS}}$ lifts to vector spaces,
(iii) (p', p'') does not transform to $\underline{\text{MS}}$.

For (i) it suffices to consider morphisms q' and q'' to a one-dimensional Q. With the matrices defined in Fig. 5 the universal property translates as follows: For any (four-dimension row vector) q^* with

$$q^*(1, 2, -2, -1)^{\mathrm{T}} = 0 \tag{1}$$

there exists one and only one (four-dimensional row vector) q with

$$q^* = q\, p^*. \tag{2}$$

For vector spaces this holds by linear algebra because the rows of p^* are a base of the solutions q^* of Eq. (1).

For (ii) we show that if $(r' : M' \to R, r'' : M'' \to R)$ is a pushout for (f', f'') in $\underline{\mathrm{MS}}$, then it is also a pushout for vector spaces. Let r^* be the matrix (r', r''). The rows of p^* correspond to commutative morphism pairs from M' and M'' to Q. Thus, there is a (unique) 3×3 matrix L with

$$p^* = L\, r^*.$$

Any commutative vector space morphism pair $q^* = (q', q'')$ is linear dependent on the rows of p^*, i.e. $q^* = c\, p^*$ for a three-dimensional row vector c. $q^* = c\, p^* = (c\, L)\, r^*$ shows that $(c\, L)$ is a factorisation of q^* over R.

To show the uniqueness of the factorisation of q^*, assume by contradiction $q^* = l' r^* = l''\, r^*$ with $l' \neq l''$. Let l_1 be the factorisation of $(1,0,0,1) = l_1\, r^*$. For $\lambda, \mu \in \mathbb{N}$ follows by linearity:

$$\lambda(1,0,0,1) = \lambda\, l_1\, r^* = (\lambda\, l_1 + \mu(l' - l''))r^*.$$

For an appropriate choice of λ and μ there are two different factorisations of $\lambda(1,0,0,1)$ in $\underline{\mathrm{MS}}$. Thus, a pushout in $\underline{\mathrm{MS}}$ lifts to vector spaces.

Now, choose the connecting morphisms $s^* = (2,0,1,0) = (2,1,-1)\, p^*$. This factorisation is valid for $\mathbb{Z}$-modules but invalid in $\underline{\mathrm{MS}}$. For (iii) it remains to show that this cannot be fixed by a base transformation T of P. T would transform Eq. (2) to

$$q^* = (qT)(T^{-1}\, p^*).$$

If q^* varies over the rows of p^* then q varies over the base vectors $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. Hence, if an entry of T is negative then one of the morphism pairs (of a row of p^*) does not allow a factorisation in $\underline{\mathrm{MS}}$. An invertable matrix with non-negative entries may only permute (and stretch) the base. Therefore, T cannot transform $(2,1,-1)$ to a non-negative vector. □

$\mathbb{Z}$-modules are (co)complete, but to use them for Petri nets is a dead end. Hiding the difference between an available resource $+r$ and a missing resource $-r$ in a tiny sign compromises the essence of net theory.

Definition 2.2. $\underline{1\mathrm{S}}$, called one-sets, is the subcategory of $\underline{\mathrm{MS}}$ with multisets over a base set as objects. A morphism $f \in \underline{1\mathrm{S}}[M, M']$ is the linear expansion of the product of two functions, namely the base component f_β and the coefficients f_γ with

$$f\left(\sum_i \lambda_i s_i\right) = \sum_{i \text{ with } s_i \in \operatorname{def} f_\beta} \lambda_i (f_\gamma\, s_i)(f_\beta\, s_i) \quad \text{for a multiset } \sum_i \lambda_i s_i \lambda_s \in M$$

with $f_\beta \in \underline{\mathrm{SETP}}[S, S']$, $f_\gamma : S \to \mathbb{N}$, $\operatorname{def} f_\beta = \{s \in S \mid f_\gamma\, s \neq 0\}$, S the base set of M, and S' the base set of M'.

In other words $\underline{\mathrm{1S}}$ is a subcategory of $\underline{\mathrm{MS}}$ with the same objects and the restriction on morphisms that a base element is mapped to a multiple of a base element. This is a generalisation over partial functions but a specialisation from multisets morphisms. It is easy to see that the morphisms are compositional. A morphism f uniquely determines the two factors f_γ and f_β and we will often use these symbols to designate these factors of a morphism. Further, we reuse $_\gamma$ for the linear expansion of f_γ to $M \to \mathbb{N}$. Let us investigate the relationships of the new category.

Definition 2.3. $1\mathrm{S} : \underline{\mathrm{SETP}} \to \underline{\mathrm{1S}}$ is the functor mapping a set S to the one-set with base S and a partial function f to a $\underline{\mathrm{1S}}$ morphism with

$$\forall s \in S\colon\ (1\mathrm{S}\, f)s = \begin{cases} f\, s \text{ (interpreted as a multiset)} & \text{if } s \in \operatorname{def} f, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

This functor builds the one-set over a given set. It is well-defined and compositional because the $\underline{\mathrm{1S}}$ morphisms with coefficients 1 are a direct translation of partial functions on the base sets.

Definition 2.4. $B : \underline{\mathrm{1S}} \to \underline{\mathrm{SETP}}$ is the functor that maps an object of $\underline{\mathrm{1S}}$ to its base set (i.e. $B\ (1\mathrm{S}\, S) = S$) and retracts a morphism f to f_β.

B is the inverse to 1S on objects by mapping a one-set to its base set (for morphisms see Proposition 2.6). This is possible for $\underline{\mathrm{1S}}$-morphisms but not in general for $\underline{\mathrm{MS}}$-morphisms. Compositionality follows because undefined values propagate in partial function composition in the same way as zeros do in products.

Definition 2.5. $BN : \underline{\mathrm{1S}} \to \underline{\mathrm{SETP}}$ is the functor mapping an object M to the set $\mathbb{N}^+ \times (BM)$ and a morphism $f \in \underline{\mathrm{1S}}[M, M']$ to

$$\operatorname{def}(BN\, f) = \mathbb{N}^+ \times (\operatorname{def} f_\beta),$$

$$(BN\, f)(\lambda, s) = (\lambda\, f_\gamma(s), f_\beta(s)) \quad \text{for } (\lambda, s) \in \operatorname{def}(BN\, f).$$

BN maps a one-set to the base skeleton, which is the set of non-zero multiples of base elements. By the same argument as above BN is well defined. The three functors are related as follows:

Proposition 2.6. 1S *is left adjoint to* BN. *There is a natural equivalence* $\varepsilon : \mathrm{Id}_{\mathrm{SETP}} \to B$ 1S *and there are isomorphisms* $\delta_M : M \to 1\mathrm{S}\, B\, M$ *with* $B\, \delta_M = \varepsilon_{\beta M}$.

$$\begin{tikzcd} \underline{\mathbf{SETP}} \arrow[r, shift left=1.5, "\mathbf{1S}"] \arrow[r, phantom, "\nabla"] & \underline{\mathbf{1S}} \arrow[l, shift left=1.5, "\mathbf{BN}"] \end{tikzcd}$$

Proof. Let S be a set and M be an object of $\underline{\mathrm{1S}}$. $\phi_{S,M} : \underline{\mathrm{1S}}[1\mathrm{S}\, S, M] \to \underline{\mathrm{SETP}}[S, BN\, M]$ given by $\phi\, f = (f_\gamma, f_\beta)$. ϕ is obviously a bijection between the two sets of morphisms. Clearly, $s \in \underline{\mathrm{SETP}}[S', S]$, $m \in \underline{\mathrm{1S}}[M, M']$ and $f \in \underline{\mathrm{1S}}[1\mathrm{S}\, S, M]$

implies

$$\phi_{S',M'}(m\, f\, 1\mathrm{S}\, s) = (BN\, m)(\phi_{S,M}\, f)s.$$

This is the necessary naturality that turns ϕ into the claimed adjunction.

The functor 1S maps a set $\{s \in S\}$ to $\{\sum_i \lambda_i s_i \mid s_i \in S\}$ which B maps back to $\{s \in S\}$. Hence $\varepsilon_S = Id_S : S \to B\, 1\mathrm{S}\, S$ is a natural isomorphism. On the other hand $\delta_M = \mathrm{Id}_M : (M) \to 1\mathrm{S}\, B\, M$ is an isomorphism of an object M of $\underline{1\mathrm{S}}$. But δ is not a natural transformation $1\mathrm{S}\ B \to \mathrm{ID}_{\underline{1\mathrm{S}}}$ because it changes morphisms $(\mathrm{im}(1\mathrm{S}\, B\, f)_\gamma \subseteq \{0,1\})$. □

Proposition 2.7. $\underline{1\mathrm{S}}$ *is cocomplete and complete.*

Proof. To construct the colimit $p : D \to P$ of a given diagram D in $\underline{1\mathrm{S}}$ we use the colimit $r : BN\, D \to R$ in $\underline{\mathrm{SETP}}$. R exists because it is well-known that $\underline{\mathrm{SETP}}$ is cocomplete. To find the base elements of P let $\sim$ be the equivalence relation on the elements of R generated by

$$\{(q_C(1,c), q_C(\lambda,c)) \mid C \text{ an object of } D,\ c \in BC,\ \lambda \in \mathbb{N}^+\}.$$

Let $\Theta \subseteq R$ an equivalence class of $\sim$ and $\Phi(\Theta)$ the set of functions compatible with multiplication:

$$\Phi(\Theta) = \{f : \Theta \to \mathbb{N}^+ \mid f(q_C(\lambda,c) = \lambda\, f(q_C(1,c))$$
$$\text{for all } \lambda \in \mathbb{N}^+,\ c \in BC \text{ with } q_C(1,c) \in \Theta \text{ and } C \text{ an object of } D\}$$

and let $\phi(\Theta)$ be the minimal function in $\Phi(\Theta)$ i.e.

$$\phi(\Theta) = \begin{cases} f \in \Phi(\Theta) \text{ with } \Phi(\Theta) = \mathbb{N}^+ f & \text{if such an } f \text{ exists,} \\ \text{undef,} & \text{otherwise.} \end{cases}$$

Now let $P_B = \mathrm{def}\ \phi$ be the equivalence classes of $\sim$ with such a function, P the one-set with basis P_B and

$$p_C : C \to P \text{ linear with } \forall c \in BN\, C\text{:}\ \ p_C\, c = \text{if } [c]_\sim \notin P_B$$
$$\text{then } \mathbf{0} \text{ else } (\phi[c]_\sim)(c)[c]_\sim.$$

Clearly these are well defined $\underline{1\mathrm{S}}$ morphisms to P. Moreover they are natural transformations because the restrictions to the base skeletons are.

To show that P has the required universal property let $q : D \to Q$ be another natural transformation as shown in Fig. 6. By the universal property of R there is a unique $q_R : R \to BN\, Q$ making the diagram in $\underline{\mathrm{SETP}}$ commutative. Because each q_C is a one-set morphism the elements of an equivalence class Θ of $\sim$ are mapped either all to $\mathbf{0}$ or to the multiples of a base element b_Θ of Q and then the function

$$q_\Theta : \Theta \to \mathbb{N} \quad \text{with } q_\Theta\, x = (q_R\, x) b_\Theta$$

is in $\Phi(\Theta)$ and q_Θ is a multiple of $\phi(\Theta)$. Hence q_R lifts in a unique way to a $\underline{1\mathrm{S}}$ morphism $q_P : P \to Q$. This finishes the proof that P is the claimed colimit.

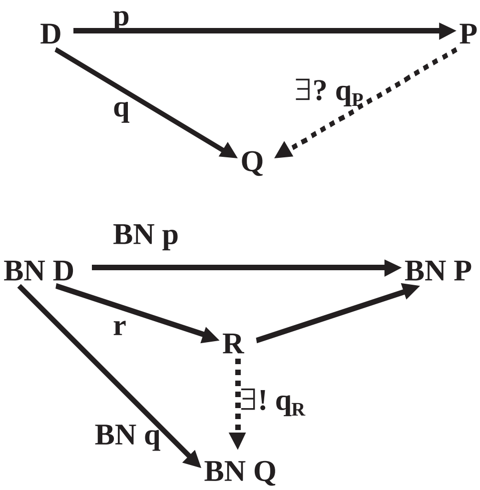


Fig. 6. The colimit in 1S.

Because BN has a left adjoint it preserves limits. Hence the skeleton of the limit of a diagram D has to coincide with the limit $r : R \to BN\,D$. To find a multiplication of an element $x \in R$ with a $\lambda \in \mathbb{N}^+$ use

$$\forall \text{ objects } C \text{ of } D: r_C(\lambda x) = \begin{cases} \lambda\, r_C\, x & \text{if } x \in \operatorname{def} r_C, \\ \text{undef}, & \text{otherwise.} \end{cases}$$

By the universality of R such a λx is unique because the elements of R are in bijection with the natural transformations from a singleton set to R. On the other hand, if x is an element of R it corresponds to such a natural transformation and by the linearity of the morphisms in D also λx as defined above. Hence λx uniquely exists and by the same argument there is at most one s with $\lambda s = x$ for any $x \in R$ and $\lambda \in \mathbb{N}^+$. The base in R is simply the set of minimal elements:

$$BR = \{b \in R \mid \forall s \in R,\ \lambda \in \mathbb{N}^+\colon \text{ if } b = \lambda s \text{ then } \lambda = 1).$$

We claim that any $x \in R$ is the multiple of exactly one $b \in BR$ and hence $R = \mathbb{N}^+ \times BR$. Assume $r = b\,\lambda = b'\,\lambda'$. If $\lambda = \lambda'$ then we showed already that $b = b'$. If $\lambda \neq \lambda'$ then r is divisible by the least common multiple of λ and λ'. This contradicts that $b, b' \in BR$. Thus, x is a multiple of at most one $b \in BR$. Such a b exists if the set $D = \{s \mid \exists \lambda \in \mathbb{N}^+\colon \lambda s = x\}$ is finite. But there is a r_C that is defined on x and it maps D bijectively to $r_C\, D \subseteq \{s' \mid \exists \lambda \in \mathbb{N}^+\colon \lambda s' = r_C\, x\}$ and the latter set is finite. □

The computation of a colimit is computationally reasonable—the size of the product does not exceed the size of the disjoint union that is the size of the input. But this is not true for limits. Already the product of two one-sets with bases consisting of a single element has an infinite base. Although 1S is complete products are of limited computational value.

3. Place–transition nets

This section builds the category $\underline{\mathrm{PTNET}}$ of place–transition nets. Its morphisms are one-set morphisms and combine features from vicinity respecting and behaviour preserving morphisms. They allow clustering, folding and mixtures. From vicinity respecting morphism $\underline{\mathrm{PTNET}}$ inherits a strong relationship with the underlying graph, which is expressed in a coreflection with $\underline{1\mathrm{S}}$. The interpretation of a morphism as an abstraction relation shows powerful clustering capabilities. Place–transition morphisms integrate with net-invariants and allow a simple interpretation of semi-positive invariants.

3.1. Basic properties

Definition 3.8. The category $\underline{\mathrm{PTNET}}$ of place–transition nets consists of objects

$$N = (\mathrm{pre}_N, \mathrm{post}_N \in \underline{\mathrm{MS}}[\mathrm{MS}\, T_N, \mathrm{MS}\, P_N])$$

with disjoint sets T_N of transitions and P_N of places.

$$X_N = T_N \cup P_N$$

are the nodes of the net N and f is a morphism $N \to N'$ iff

$$f \in \underline{1\mathrm{S}}[1\mathrm{S}\, X_N, 1\mathrm{S}\, X_{N'}],$$

$$((f_\beta\, t \in T_{N'} \text{ and } f\, \mathrm{pre}_N\, t = \mathrm{pre}_{N'}\, f t) \text{ or } f\, \mathrm{pre}_{N'}\, t = f t) \tag{3}$$

and

$$((f_\beta\, t \in T_{N'} \text{ and } f\, \mathrm{post}_N\, t = \mathrm{post}_{N'}\, f t) \text{ or } f\, \mathrm{post}_{N'}\, t = f t) \tag{4}$$

for all $t \in T_N$. f is called

- a folding iff $f_\beta\, P_N \subseteq P_{N'}$ and $f_\beta\, T_N \subseteq T_{N'}$,
- unitary iff $f_\gamma\, X_N \subseteq \{1\}$, and
- binary iff $f_\gamma\, X_N \subseteq \{0, 1\}$.

As usual we will replace the index N by other forms of sub- or superscripting or drop it completely, abbreviate $X_{N'}$ to X', etc. We define the pre-set and post-set of a node x by

$$^\bullet x = \begin{cases} \mathrm{supp}(\mathrm{pre}\, x) & \text{if } x \in T \\ \{t \in T \mid x \in \mathrm{supp}(\mathrm{post}\, t)\}, & \text{otherwise} \end{cases}$$

$$x^\bullet = \begin{cases} \mathrm{supp}(\mathrm{post}\, x) & \text{if } x \in T \\ \{t \in T \mid x \in \mathrm{supp}(\mathrm{pre}\, t)\}, & \text{otherwise.} \end{cases}$$

The application of a pre or post-function will imply that the argument is a transition multiset.

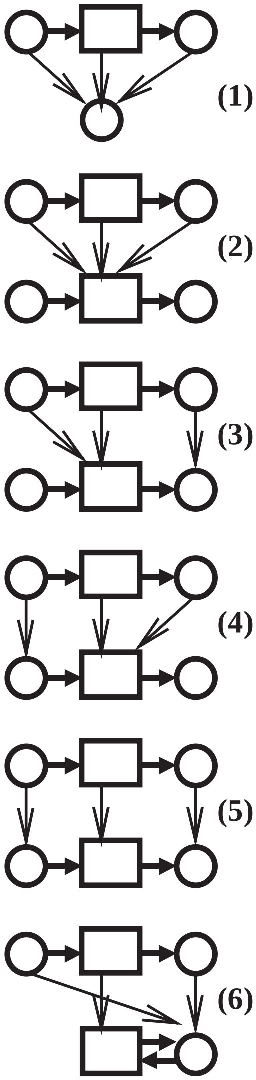


Fig. 7. Examples of PT-Morphisms.

The pre- and post-functions map a transition to a multiset of places. A morphism maps a node of the source net to a multiple of a node of the destination net and the pre (post) multiset of a transition is either mapped to the pre (post) multiset of the image or to the image of the transition itself, as shown in Fig. 7. This means that the occurrence of a transition t is faithfully reflected by an occurrence of its image if $\text{pre}\, f\, t = f\, \text{pre}\, t$ and $\text{post}\, f\, t = f\, \text{post}\, t$. If, on the other hand, $\text{pre}\, f\, t = f\, t = \text{post}\, f\, t$ then an occurrence of the transition is completely invisible in the destination net. The definition further allows that only 'half' of a transition occurrence is visible: either the removal of the tokens from the pre-places or the creation of tokens on to post-places. Such a split of a transition occurrence violates the principle of atomicity of transition occurrence but is often a helpful concept (e.g. complex markings in [1]). For simpler definitions, Definition 4.19, Lemma 4.20 or Definition 4.24.

The objects of $\underline{\mathrm{PTNET}}$ are nets with weighted arcs that often occur in the literature, e.g. in [19]. The morphisms, however, represent a novel combination of commutative and vicinity respecting definitions. With the former (see [19] for an overview) a morphism is the union of a (partial) function from T to T' and a multiset morphism from $\mathrm{MS}\,P \to \mathrm{MS}\,P'$ which respects pre and post (e.g. $f\,\mathrm{pre} = \mathrm{pre}'\,f$) plus possibly some further restrictions. Hence, our definition is more restrictive on the mapping of places to places, but more general by allowing the mapping of transitions to multiples of transitions and 'cross mappings' from transitions to places and vice versa. The requirement to respect pre and post is reflected in our definition by the condition before the '*or*' in formulas (3) and (4).

Vicinity respecting morphisms (e.g. [7,6]) apply to elementary or condition/event nets (with pre- and post-sets instead of multisets, i.e. bipartite graphs without multiple arcs). Here a morphism is a graph morphism that is allowed to map an arc and its two incident nodes either into a single destination node or to an arc and the incident transition to the incident transition and the incident place to the incident place (e.g. $f^\bullet t = f\,t$ or $f^\bullet t = {}^\bullet f t$). The base component f_β of our morphism is such a morphism and f is a generalisation dealing with multiplicities. It is worth mentioning that a morphism is allowed to map a node to $\mathbf{0}$.

We still have to show that

Lemma 3.9. $\underline{\mathrm{PTNET}}$ *is a well defined category with zero object.*

Proof. For the composition of two morphisms $f : N \to N'$ and $g : N' \to N''$ we get:

$$g\,f\,\mathrm{pre}_{N'}\,t = g\left(\begin{Bmatrix} f(t) \\ \mathrm{pre}_{N'}\,f\,t \end{Bmatrix}\right) = \begin{Bmatrix} g\,f\,t \\ g\,f\,t \\ \mathrm{pre}_{N''}\,g\,f\,t \end{Bmatrix}$$

all possible combinations end up in $g\,f\,t$ or $\mathrm{pre}_{N''}\,g\,f\,t$ as required. The proof is complete because of the mentioned convention that the application of a pre function implies the argument to be a transition multiset. With the analogous calculation for post this shows that the composition is yet again a morphism. It is associative because it is in $\underline{1\mathrm{S}}$. The identities are obvious and the empty net with neither transitions nor places is the zero object of the category. □

Let us first characterise the basic type of morphisms:

Proposition 3.10. *A* $\underline{\mathrm{PTNET}}$ *morphism* $f : N \to N'$ *is*
epimorphic iff f_β *is surjective*
monomorphic iff $\mathrm{def}(f_\beta) = X$ *and* f_β *injective*
an isomorphism iff f *is unitary*, f_β *a bijection between* X *and* X' *and isolated places are mapped to places.*

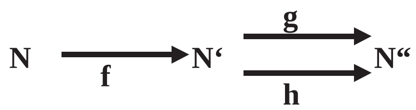


Fig. 8. f is epimorphic iff $gf = hf$ implies $g = h$.

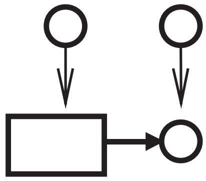


Fig. 9. Mono and epi but not iso.

Proof. If f_β is surjective then clearly f is epimorphic. In the other direction we assume that f_β is not surjective and construct $g, h : N' \to N''$ which contradict that f is epimorphic (see Fig. 8). In a first case let $t' \in T' \backslash \mathrm{im}(f_\beta)$. N'' is constructed from N' by "doubling t" as follows:

- $T'' = T' \cup \{t''\}$ for a $t'' \notin X'$,
- $\mathrm{pre}''(t) = \text{if } t \in T' \text{ then } \mathrm{pre}'(t) \text{ else } \mathrm{pre}'(t')$,
- $\mathrm{post}''(t) = \text{if } t \in T' \text{ then } \mathrm{post}'(t) \text{ else } \mathrm{post}'(t')$,
- $N'' = (\mathrm{pre}'', \mathrm{post}'')$,
- $g(x') = x'$,
- $h(x') = \text{if } x' = t' \text{ then } t'' \text{ else } x'$.

Clearly $g \neq h$ but $gf = hf$ which is the necessary contradiction to f epimorphic. The second case is a place p' instead of t'. If p' is not isolated it is connected to a transition t' that neither can be in $\mathrm{im}(f_B)$ and the first case applies. Otherwise, take N'' with two places and no transitions. g and h are defined only on p' and map it to the first respectively the second place of N''. As above this contradicts f to be epimorphic.

The second claim is proved similarly. For the last claim if f is iso it must be epi and mono. Furthermore f_γ must equal 1, because this is the only natural with a reciprocal. But this is not sufficient: Fig. 9 shows a morphism that fulfils all these conditions but its inverse is not a morphism. If a place is mapped to a transition there exists no inverse because

- if the transition is not isolated then by bijectivity and commutativity with pre and post,
- if the transition is isolated then the inverse must map it to an isolated transition too.

For the reverse direction any non-isolated transition must be mapped to a transition—otherwise, the pre- and post-sets would also map to the image of the transition which contradicts bijectivity. Consequently, non-isolated places map to places. By the above and the last condition this is also true for isolated elements. All together this ensures that the inverse of f also commutes with pre and post and hence is a <u>PTNET</u> morphism. This proves the last claim. □

Remark. The proposition still holds in subcategories of <u>PTNET</u> allowing only foldings (<u>FNET</u>) or disallowing morphisms to map places to transitions (<u>PPNET</u>).

An isomorphism in <u>PTNET</u> is a bijective unitary folding or equivalently a pair of bijections, one between the places and one between the transitions. Hence our Petri net isomorphisms are exactly the same as in clustering-based [7] or folding-based [19] approaches. On the other hand, even our foldings are in general no morphisms of [19] because they are allowed to multiply transitions.

Definition 3.11. $\mathrm{PTNET} : \underline{1\mathrm{S}} \to \underline{\mathrm{PTNET}}$ is the functor that sends an one-set M to the net with places BM, no transitions and hence empty pre and post functions. A morphism is mapped to the same morphism. U is the underlying functor which sends a net N to $1\mathrm{S}\,X_N$.

PTNET interprets the base elements of an object of $\underline{\mathrm{MS}}$ as the places of a net whereas U forgets the net structure and only keeps the one-set properties.

Proposition 3.12. $\mathrm{PTNET} : \underline{1\mathrm{S}} \to \underline{\mathrm{PTNET}}$ *is the left adjoint of* U. *Moreover, they form a coreflection.*

$$\underline{\mathbf{1S}} \underset{\mathbf{U}}{\overset{\mathbf{PTNET}}{\rightleftarrows}} \underline{\mathbf{PTNET}}$$

Proof. The bijection $\underline{\mathrm{PTNET}}\ [\mathrm{PTNET}\,M, N'] \to \underline{1\mathrm{S}}[M, U\,N']$ translates to

$$\underline{\mathrm{PTNET}}[\mathrm{PTNET}\,1\mathrm{S}\,P, N'] \to \underline{1\mathrm{S}}[1\mathrm{S}\,P, 1\mathrm{S}\,X']$$

by setting $M = 1\mathrm{S}\,P$ and $U\,N' = 1\mathrm{S}\,X'$. Obviously the interpretation of the same morphism once in $\underline{\mathrm{PTNET}}$ and once in $\underline{1\mathrm{S}}$ is natural. The unit given by

$$\mathrm{Id} \in \underline{\mathrm{PTNET}}[\mathrm{PTNET}\,M, \mathrm{PTNET}\,M] \to \varepsilon_M \in \underline{1\mathrm{S}}[M, U\,\mathrm{PTNET}\,M]$$

is clearly isomorphic and hence the adjunction is a coreflection. □

Together with Proposition 2.6 this yields that $BN\,U : \underline{\mathrm{PTNET}} \to \underline{\mathrm{SETP}}$ has a left adjoint. But probably more important is the functor $U\,U : \underline{\mathrm{PTNET}} \to \underline{\mathrm{SETP}}$ which retracts a net morphism to a partial function of the nodes of the net. This is not far from an appropriate graph morphism. Such a relationship is an important property of a clustering-based morphism. It underlines an important Petri net principle: the interplay of structure and behaviour. This allows to deduce behavioural properties from the structure, furthermore, for software engineering and visualisation a tight relationship to the underlying graph is beneficial. Multiset-based morphism do not retract so directly to the nodes, which degrades the net structure to a 'second class citizen'.

3.2. Clustering

A simple way to use category theory in software engineering is to interpret a morphism $f : C \to D$ as the design D of the implementation C. The origins of a node of D are a subsystem of C. This section investigates such clustering properties of $\underline{\mathrm{PTNET}}$ morphisms. The naive idea that the origins of a place form a super place and the origins of a transition a super transition is not bad.

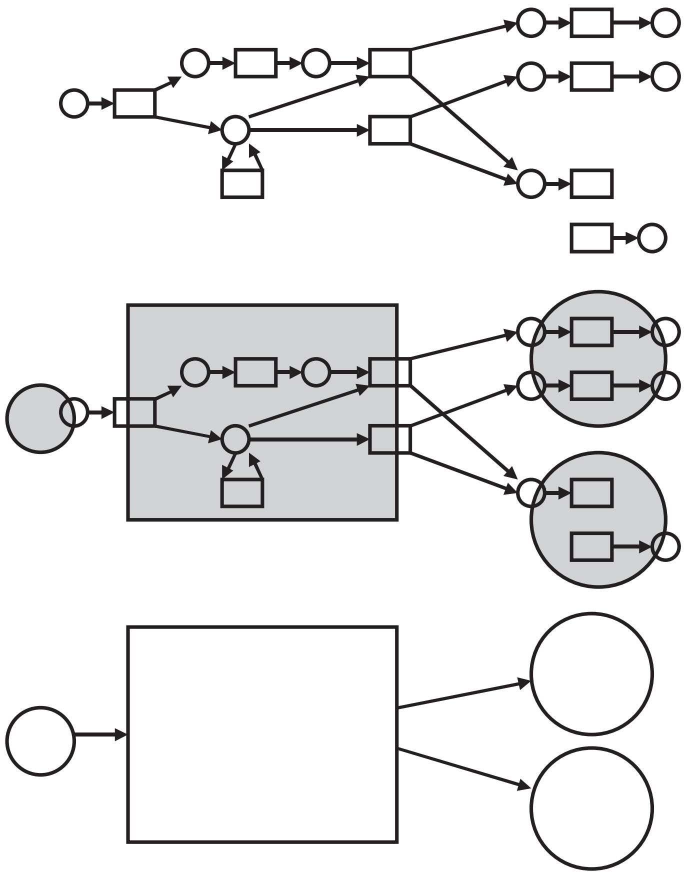


Fig. 10. The origins of a node form a super node.

Definition 3.13. A subset S of nodes of a net is called transition-bordered iff any node in S with an arc to a node not in S is a transition. Place-bordered is defined correspondingly as shown in Fig. 10.

Proposition 3.14. *Let* $f: N \to N'$, $K_f = X_N \setminus \mathrm{def}(f_\beta)$ *and* $S' \subseteq X'$. *Then*
$\mathrm{def}(f_\beta)$ *is transition-bordered and* K_f *place-bordered,*
if S' *is transition-bordered then also* $f_\beta^{-1}(S')$ *and*
if S' *is place-bordered then also* $f_\beta^{-1}(S') \cup K_f$.

Proof. A straightforward application of the definitions. E.g. if $t \in K_f \cap T$ then $f\,t = \mathbf{0} = f\,\mathrm{pre}\,t = f\,\mathrm{post}\,t$. Hence, ${}^\bullet t \subseteq K_f \supseteq t^\bullet$ and K_f is place-bordered. □

Hence, ignoring K_f—the ‘garbage component’ consisting of the nodes mapped to $\mathbf{0}$—the origin of a node is a super node of the same type as shown in Fig. 10.

Looking closer at Fig. 10 one detects that super transitions have proper port transitions. That is, the post set of a transition is either completely inside or completely outside of its super node (i.e. the origins of its image). This is easily formalised, generalised from single nodes to subnets and proved to be correct 'up to the garbage component' (refer to [10]).

In conclusion, PTNET morphisms directly offer clustering capabilities that are very attractive for software engineering and are tightly related to different compositional net classes.

3.3. Invariants

Invariants play an important role in net theory, because they allow you to deduce behavioural properties from the net structure by linear algebraic techniques.

Definition 3.15. A place invariant of net N is a

a map $i : P_N \to \mathbb{Z}$ fulfilling $i\,\mathrm{pre} = i\,\mathrm{post}$

with $i\,\mathrm{pre} : T_N \to \mathbb{Z}, i\,\mathrm{pre}(t) = \sum_{p \in P} i(p)\mathrm{pre}(t)(p)$

and analogously for post

and a transition invariant a

linear map $j : T_N \to \mathbb{Z}$ fulfilling pre $j = \mathrm{post}\, j$.

with pre $j : P_N \to \mathbb{Z}, \mathrm{pre}\, j(p) = \sum_{t \in T} \mathrm{pre}(t)(p) j(t)$

and each such sum finite.

A transition or place invariant is called positive if its range is in $\mathbb{N}^+$ and semi-positive if its range is in $\mathbb{N}$.

For functions to $\mathbb{Z}$ we freely switch between function notation and formal sum notation and use linear expansion, as we do for multisets. E.g. in the second equation j must be read as a formal linear combination of transitions. The equations express the standard properties: a place invariant does not change under transition occurrence and a transition invariant produces and consumes the "same tokens". Morphisms transfer place invariants in the reverse direction:

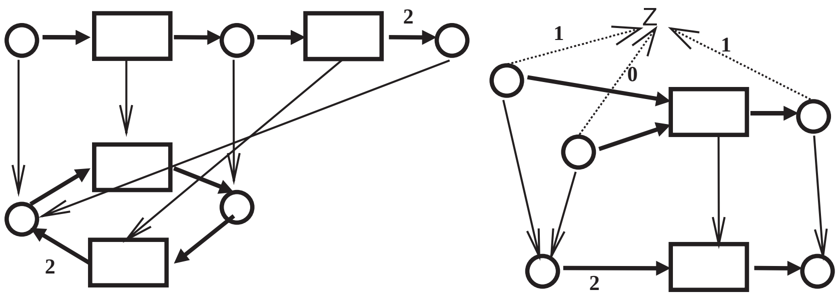


Fig. 11. A place invariant may not transfer because of global (left) or local (right) incompatibility.

Proposition 3.16. *Let $f:N'\to N$, i a place invariant of N and*

$$\underline{i}: X \to \mathbb{Z} \text{ with } \underline{i}\,x = \begin{cases} i\,x & \text{if } x \in P \\ i\,\mathrm{pre}\,x, & \text{otherwise.} \end{cases}$$

The restriction of $\underline{i}\,f$ to P is a place invariant of N'. Restriction to P is a bijection from morphisms from N to PTNET $\mathbb{N}$ *and semi-positive place invariants of N.*

Proof. Because i is a place invariant for each transition $t\in T$ holds $i\,\mathrm{pre}\,t = i\,\mathrm{post}\,t = \underline{i}\,\mathrm{pre}\,t = \underline{i}\,\mathrm{post}\,t = \underline{i}\,t$.

Hence, if i is semi-positive $\underline{i}$ is a PTNET-morphism $N\to$ PTNET $\mathbb{N}$. On the other hand, given such a morphism $\underline{i}$ reading the above proof backwards shows that the restriction of $\underline{i}$ to P is a semi-positive place invariant of N. This proves the second claim.

The composition $\underline{i}\,f$ is a morphism from N' to PTNET $\mathbb{N}$. Thus, by the above, we get the first claim for semi-positive place invariants. But this implies the general case because compositionality with semi-positive invariants (i.e. functions to $\mathbb{N}$) generalises to place invariants (i.e. functions to $\mathbb{Z}$). $\square$

Thus, a P-invariant travels from the destination to the source. Surprisingly, the propagation in the direction of the morphism may fail as Fig. 11 shows.

Definition 3.17. A T-system is Petri net in which the pre-multiset as well as the post-multiset of any place is a set consisting of a single transition, i.e.

$$\forall p \in P\colon \sum_{t\in T} \mathrm{pre}(t)(p) = 1 = \sum_{t\in T} \mathrm{post}(t)(p).$$

Proposition 3.18. *Let $f:N\to N'$ and j a transition invariant of N. If*

$$j' = \sum_{\substack{t\in T \\ f\,\mathrm{pre}\,t\neq f\,t}} j(t)f(t) = \sum_{\substack{t\in T \\ f\,\mathrm{post}\,t\neq f\,t}} j(t)f(t)$$

is defined (i.e. $\forall t' \in T'$: $j'(t')$ is finite) then it is a transition invariant of N'. A semi-positive transition invariant corresponds to a unitary folding from a T-system.

Proof. Let f and j as above and interpret j as linear combination $j = \sum_{t \in T} j(t)t$. In the following derivation the $t \in T$ under each summation symbol is omitted:

$$\mathbf{0} = (\text{post-pre})j$$

$$\mathbf{0} = f(\text{post-pre})j$$

$$\begin{aligned}\mathbf{0} = f(\text{post-pre})\Bigg(& \sum_{\substack{f\,\text{pre}\,t = f\,t \\ f\,\text{post}\,t = f\,t}} j(t)t + \sum_{\substack{f\,\text{pre}\,t = \text{pre}\,f\,t \\ f\,\text{post}\,t = f\,t}} j(t)t \\ & + \sum_{\substack{f\,\text{pre}\,t = f\,t \\ f\,\text{post}\,t = \text{post}\,f\,t}} j(t)t + \sum_{\substack{\text{pre}\,f\,t = f\,t \\ f\,\text{post}\,t = \text{post}\,f\,t}} j(t)t \Bigg)\end{aligned}$$

$$\begin{aligned}\mathbf{0} = & \sum_{\substack{f\,\text{pre}\,t = f\,t \\ f\,\text{post}\,t = f\,t}} j(t)(f\,t - f\,t) + \sum_{\substack{f\,\text{pre}\,t = \text{pre}\,f\,t \\ f\,\text{post}\,t = f\,t}} j(t)(f\,t - \text{pre}\,f\,t) \\ & + \sum_{\substack{f\,\text{pre}\,t = f\,t \\ f\,\text{post}\,t = \text{post}\,f\,t}} j(t)(\text{post}\,f\,t - f\,t) \\ & + \sum_{\substack{f\,\text{pre}\,t = \text{pre}\,f\,t \\ f\,\text{post}\,t = \text{post}\,f\,t}} j(t)(\text{post}\,f\,t - \text{pre}\,f\,t)\end{aligned}$$

$$\begin{aligned}\mathbf{0} = & \sum_{\substack{f\,\text{pre}\,t = \text{pre}\,f\,t \\ f\,\text{post}\,t = f\,t}} j(t)f\,t - \sum_{\substack{f\,\text{pre}\,t = f\,t \\ f\,\text{post}\,t = \text{post}\,f\,t}} j(t)f\,t \\ & - \sum_{\substack{f\,\text{pre}\,t = \text{pre}\,f\,t \\ f\,\text{post}\,t = f\,t}} j(t)\,\text{pre}\,f\,t + \sum_{\substack{f\,\text{pre}\,t = f\,t \\ f\,\text{post}\,t = \text{post}\,f\,t}} j(t)\,\text{post}\,f\,t \\ & + \sum_{\substack{f\,\text{pre}\,t = \text{pre}\,f\,t \\ f\,\text{post}\,t = \text{post}\,f\,t}} j(t)(\text{post}\,f\,t - \text{pre}\,f\,t).\end{aligned}$$

The term $f(\text{post-pre})\sum_{\substack{f\,\text{pre}\,t=f\,t\\ f\,\text{post}\,t=\text{post}\,f\,t}} j(t)$ vanishes ((1) and (2) in Fig. 7) and f maps the transitions in the remaining terms to transitions only. Thus, in the last equation above, the first two summands are transition multisets whereas the last three are place multisets. Hence already the sum of the first two terms vanishes which may be rewritten as

$$\sum_{\substack{f\,\text{pre}\,t=\text{pre}\,f\,t\\ f\,\text{post}\,t=f\,t}} j(t)f\,t = \sum_{\substack{f\,\text{pre}\,t=f\,t\\ f\,\text{post}\,t=\text{post}\,f\,t}} j(t)f\,t$$

$$\sum_{\substack{f\,\text{pre}\,t=\text{pre}\,f\,t\\ f\,\text{post}\,t=f\,t}} j(t)f\,t + \sum_{\substack{f\,\text{pre}\,t=\text{pre}\,f\,t\\ f\,\text{post}\,t=\text{post}\,f\,t}} j(t)f\,t$$

$$= \sum_{\substack{f\,\text{pre}\,t=\text{pre}\,f\,t\\ f\,\text{post}\,t=\text{post}\,f\,t}} j(t)f\,t + \sum_{\substack{f\,\text{pre}\,t=\text{pre}\,f\,t\\ f\,\text{post}\,t=\text{post}\,f\,t}} j(t)f\,t$$

$$\sum_{f\,\text{pre}\,t\neq f\,t} j(t)f\,t = \sum_{f\,\text{post}\,t\neq f\,t} j(t)f\,t = j'$$

which shows that j' is well defined. Similarly the sum of the last three terms vanishes which yields that j' is indeed a transition invariant:

$$\sum_{f\,\text{pre}\,t=\text{pre}\,f\,t} j(t)\,\text{pre}\,f\,t = \sum_{f\,\text{post}\,t=\text{post}\,f\,t} j(t)\,\text{post}\,f\,t$$

$$\text{pre}\sum_{f\,\text{pre}\,t=\text{pre}\,f\,t} j(t)f\,t = \text{post}\sum_{f\,\text{post}\,t=\text{post}\,f\,t} j(t)f\,t$$

$$\text{pre}\,j' = \text{post}\,j'.$$

For a T-system N holds $\text{pre}\,T = P = \text{post}\,T$, thus T is a transition invariant and by the above for any folding $f: N \to N'$ follows that $f\,T$ is a transition invariant of N'. If, on the other hand, j' is a transition invariant of a net N' a transition system $N = (\text{pre}, \text{post})$ and a unitary folding $f: N \to N'$ may be constructed by 'unfolding j' and the arcs':

$$T = \{(t', \alpha) \in T' \times \mathbb{N}^+ \mid 1 \leqslant \alpha \leqslant j'(t')\},$$

$$P = \{((t', \alpha), p', \beta') \in T \times P' \times \mathbb{N}^+ \mid 1 \leqslant \beta' \leqslant (\text{pre}'\,t')(p')\},$$

$$\text{pre}(t) = \sum_{(t,p',\beta')\in P} (t, p', \beta').$$

To construct the post-function the output arcs must be unfolded and a bijection b between output and input places must be chosen. Such a bijection exists because j' is

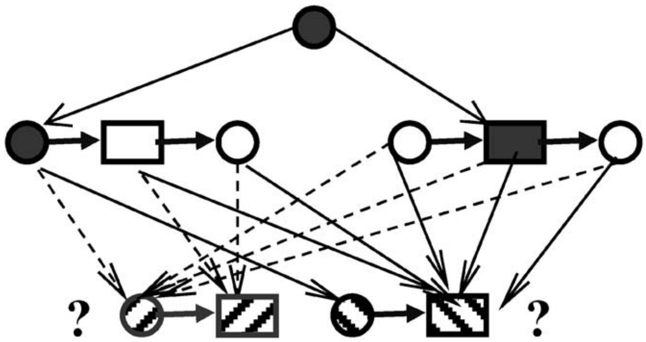


Fig. 12. The pushout is lacking in the diagram with the three nets in the upper half. Fat arrows show net-arcs, (bunches of) thin arrows morphisms.

a transition invariant:

$$Q = \{((t', \alpha'), p'', \beta'') \in T \times P' \times \mathbb{N}^+ \mid 1 \leqslant \beta \leqslant (\text{post}'\, t')(p'')\},$$

$$b : P \to Q \text{ a bijection with if } b(t, p', \beta') = (u, p'', \beta'') \text{ then } p' = p'',$$

$$\text{post}(t) = \sum_{(t, p'', \beta'') \in Q} b(t, p'', \beta''),$$

$$f : N \to N' \text{ with } f(t, \alpha) = t \text{ and } f(t, p', \beta') = p'.$$

The construction ensures that $f\,\text{pre} = \text{pre}'\, f$ and $f\,\text{post} = \text{post}'\, f$ thus f is a morphism and even a unitary folding. Because b is bijective N is a T-system and by construction $f\,T = j'$ which finishes the proof of the last claim. □

The asymmetry between place and transition invariants is a consequence of the break in transition/place duality in the definition of a morphism. A place invariant has a very concise characterisation: it is just a morphism to a single place net. For a transition invariant the procedure is more complex—it has to get unfolded in a T-system—which is neither unique nor small.

3.4. Limitations

A major limitation of PTNET are missing universal constructions. Pushouts exist only for foldings but not in general as the example in Fig. 12 shows. By commutativity the black elements map to the same node. Should it be a transition or a place? For a factorisation to the left bottom net to a place but for the right bottom net to a transition! Hence the pushout cannot exist. Because there are no restrictions for morphisms to interchange places and transitions the transfer of the bipartite structure fails. For pullbacks additionally the pre- and post-function may collapse. Furthermore, the transfer of behaviour is rather complicated. For the token game this may be seen from the formula in Proposition 3.18 which preserves the balance of produced and consumed tokens.

4. From clustering to folding

The simple restriction that morphisms have to map places to places yields the category $\underline{\text{PPNET}}$ of place-preserving nets. Here, all (co)universal constructions exist and additional relationships to one-sets hold. A crucial point is that a place–transition net can be simulated by a place–preserving net, namely by a functor PP which is part of an adjunction (Proposition 4.21).

Place-preserving nets are the middle pillar of the bridge from clustering to folding which is represented by the category $\underline{\text{FNET}}$. The second half of the bridge is constructed similarly. This yields an adjunction to $\underline{\text{FNET}}$ which composes to a computationally reasonable adjunction from $\underline{\text{PTNET}}$ to $\underline{\text{FNET}}$.

4.1. Place-preserving nets

Definition 4.19. A PTNET morphism $f: N \to N'$ is called a place-preserving morphism iff $f_\beta(P) \subseteq P'$. The category $\underline{\text{PPNET}}$ is the subcategory of $\underline{\text{PTNET}}$ with the same objects and place-preserving morphisms.

Obviously, the PP in $\underline{\text{PPNET}}$ stands for place-preserving. But the following lemma gives it another but equivalent sense.

Lemma 4.20. *Let $\Delta x = (x, x)$ and*

$$\mathrm{pp} = (\mathit{pre}, \mathit{post}) \in (\underline{\mathrm{MS}}[\mathrm{MS}\, T, \mathrm{MS}\, P])^2 \cong \underline{\mathrm{MS}}[\mathrm{MS}\, T, (\mathrm{MS}\, P)^2]$$

and N and N' be Petri nets. For a $\underline{1\mathrm{S}}$ morphism $f: 1\mathrm{S}\,X \to 1\mathrm{S}\,X'$ the following are equivalent in pairs

- *f is a place-preserving net morphism*
- *$\forall t \in T$ holds:* $f\,\mathrm{pp}\,t = \begin{cases} \mathrm{pp}\ f\,t & \text{if } t \in \mathrm{def}\, f_\beta \text{ and } f_\beta\, t \in T', \\ f\,t, & \text{otherwise} \end{cases}$
  and $\forall p \in P \cap \mathrm{def}\, f_\beta$ holds: if ${}^\bullet p = p^\bullet = \{\}$ then $f_\beta\, p \in P'$,
- $(\forall t \in T\colon\ f\,\mathrm{pp}\,t = \mathrm{pp}'\, f\, t$ *or* $f\,\mathrm{pp}\,t = \Delta f\, t)$
  and $(\forall p \in P \cap \mathrm{def}\, f_\beta\colon \exists p' \in P$ *with* $f_\beta\, p' \in P'$ *such that p and p' are connected by an undirected path with all nodes in* $\mathrm{def}(f_\beta))$.

Proof. A place-preserving morphism disallows examples (2)–(4) from Fig. 7, and the remaining cases fulfil the second condition. From the second condition follows the third, by selecting $p = p'$ and the path consisting of p and no edges.

Assume the third condition. It implies that f is a $\underline{\text{PTNET}}$ morphism. Furthermore, f_β cannot map a place to a transition, otherwise, on the postulated path from such a place one could find a transition t connected to places p and q with $f_\beta\, p \in T'$ and $f_\beta\, q \in P'$. But this contradicts both $f\,\mathrm{pp}\,t = \Delta f\, t$ and $f\,\mathrm{pp}\,t = \mathrm{pp}'\, f\, t$. $\square$

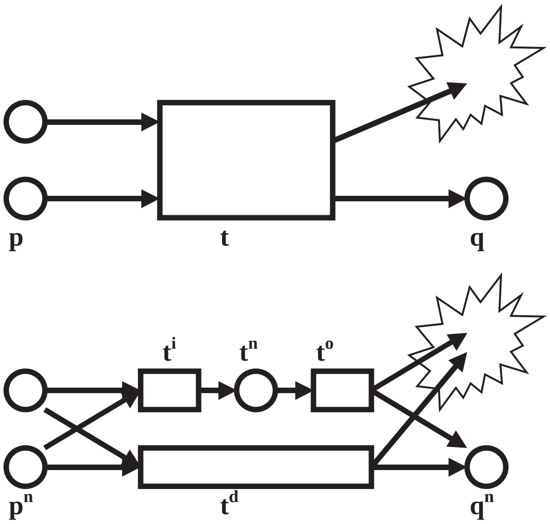


Fig. 13. PP maps a transition to 4 nodes.

This lemma allows us to replace pre and post by the combined function pp. This simplifies notation and proofs. For example we may abbreviate a net $N = (\text{pre}, \text{post})$ by $N = \text{pp}$.

4.2. From PTNET to PPNET

Clearly, there is a forgetful functor $U : \underline{\text{PPNET}} \to \underline{\text{PTNET}}$ simply forgetting the place-preserving restriction on morphisms. We construct a functor PP in the reverse direction. The basic idea is, to replace every transition by three transitions and a place as shown in Fig. 13. This deals with the four combinations of the image of pre and post of a transition for a PTNET morphism, shown as cases (2)–(5) in Fig. 7. Formally, a net N_{PP} is derived from a given net N by:

4 new symbols d (direct), i (input), o (output) and n (internal or inherited)

$$P_{\text{PP}} = \{x^n \mid x \in X\},$$

$$T_{\text{PP}} = \{t^y \mid t \in T,\ \ y \in \{d,i,o\}\}$$

the x^y notation is extended to multisets in the obvious way

$$\text{pre}_{\text{PP}}\, t^y = \begin{cases} t^n & \text{if } \ y = o \\ (\text{pre}\, t)^n, & \text{otherwise,} \end{cases}$$

$$\text{post}_{\text{PP}}\, t^y = \begin{cases} t^n & \text{if } \ y = i \\ (\text{post}\, t)^n, & \text{otherwise,} \end{cases}$$

$$\text{PP}\, N = (\text{pre}_{\text{PP}}, \text{post}_{\text{PP}} : \text{MS}\, T_{\text{PP}} \to \text{MS}\, P_{\text{PP}}).$$

Clearly PPN is a net. The linear function ε_N with $\varepsilon_N\, x^y = x$ is a $\underline{\mathrm{PTNET}}$ morphism $\varepsilon_{\mathrm{N}} : N \to N_{\mathrm{PP}}$, by construction of $\mathrm{pre}_{\mathrm{PP}}$ and $\mathrm{post}_{\mathrm{PP}}$. Furthermore, for any $\underline{\mathrm{PTNET}}$ morphism $f : N' \to N$ there is a unique place-preserving morphism $g : N' \to N_{\mathrm{PP}}$ making the triangle commutative (Fig. 14). g is defined on $x' \in X'$ by

$$g(x') = \begin{cases} (f\,x')^n & \text{if } x' \in P' \text{ or } f\ \mathrm{pre}'\,x' = f\,x' = f\,\mathrm{post}'\,x' \\ (f\,x')^o & \text{if } f\,\mathrm{pre}'\,x' = f\,x' \text{ and } f\,\mathrm{post}'\,x' = \mathrm{post}\,f\,x' \\ (f\,x')^i & \text{if } f\,\mathrm{pre}'\,x' = \mathrm{pre}\,f\,x' \text{ and } f\,\mathrm{post}'\,x' = f\,x' \\ (f\,x')^d & \text{if } f\,\mathrm{pre}'\,x' = \mathrm{pre}\,f\,x' \text{ and } f\,\mathrm{post}'\,x' = \mathrm{post}\,f\,x'. \end{cases}$$

Clearly g is place-preserving. The proof that it is a morphism needs to check each case of the definition. For an input transition t':

$$g\,t' = (f\,t')^i$$

$$g\,\mathrm{pre}\,t' = (f\,\mathrm{pre}'\,t')^n = (\mathrm{pre}\,f\,t')^n = \mathrm{pre}\,g\,t'$$

$$g\,\mathrm{post}\,t' = (f\,\mathrm{post}\,t')^n = (f\,t')^n = \mathrm{post}\,g\,t'$$

and similarly in the three other cases.

To prove the uniqueness of g notice that the commutativity of the triangle requires $g\,x' = (f\,x')^y$. If x' is a place $y = n$ because g is place-preserving. If x' is a transition there is only one possibility to select y such that g fulfils the place-preserving-morphism-property on $\{x'\} \cup {}^\bullet x' \cup x'^\bullet$, namely the one given above in the definition of g. $\square$

Proposition 4.21. PP *is a functor from* $\underline{\mathrm{PTNET}}$ *to* $\underline{\mathrm{PPNET}}$ *and is right adjoint to the underlying functor* U *with* ε *the counit.*

$$\begin{tikzcd} \underline{\mathbf{PPNET}} \arrow[r, shift left=1.5, "\mathbf{U}"] \arrow[r, phantom, "\top"] & \underline{\mathbf{PTNET}} \arrow[l, shift left=1.5, "\mathbf{PP}"] \end{tikzcd}$$

Proof. Redrawing Fig. 14 as Fig. 15 yields that ε_N is a universal arrow from U to N which by a result from category theory [13, Theorem 2, p. 81] yields that PP is a functor and is right adjoint to U with the counit ε. $\square$

Notice that the two adjunctions between $\underline{\mathrm{PTNET}}$ and $\underline{1\mathrm{S}}$, respectively $\underline{\mathrm{PPNET}}$ do not compose as adjunctions because left and right do not match. But there are other adjunctions. This has consequences for universal construction in $\underline{\mathrm{PPNET}}$ (Section 4.3).

Proposition 4.22. *Let* $\mathrm{PL} : \underline{\mathrm{PPNET}} \to \underline{\mathrm{PPNET}}$ *be the functor dropping all transitions and keeping the places of a net.* $U\,\mathrm{PL} : \underline{\mathrm{PPNET}} \to \underline{1\mathrm{S}}$ *forms a coreflection with the left adjoint* PP PTNET *and forms a reflection with the right adjoint* $\mathrm{MM2} : \underline{1\mathrm{S}} \to \underline{\mathrm{PPNET}}$.

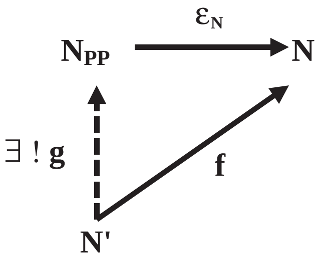


Fig. 14. The unique factorisation g.

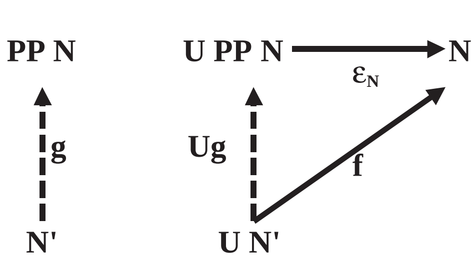
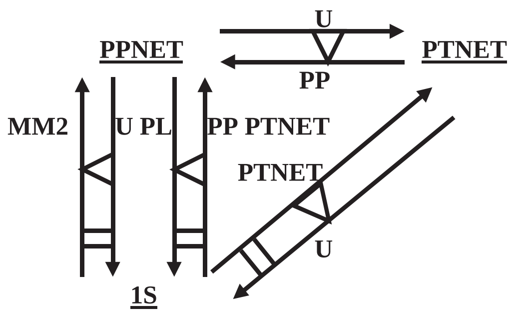


Fig. 15. The unique factorisation g as universal arrow from U to N.

The diagram below summarises the situation. Notice that it is not commutative, e.g. $U\,U \neq U\,\mathrm{PL}$ and the cyclic triangles compose to identities only in special cases (e.g. $U\,\mathrm{PL}\,\mathrm{PP}\,\mathrm{PTNET} = \mathrm{Id}_{1\mathrm{S}}$) but not in general.



Proof. Let M be an object of $\underline{1\mathrm{S}}$ and N of $\underline{\mathrm{PPNET}}$. PTNET M has only places, so PP has no transitions to expand and produces an isomorphic object. Furthermore,

$$\underline{\mathrm{PPNET}}[\mathrm{PP}\,\mathrm{PTNET}\,M, N] \cong \underline{1\mathrm{S}}[M, U\,\mathrm{PL}\,N]$$

is just a reinterpretation of the same function once as a place-preserving morphism and once as $\underline{1\mathrm{S}}$ morphism between M and the multiset over the places of N. Clearly the units of this adjunction are isomorphisms in $\underline{1\mathrm{S}}$.

The construction of the claimed functor MM2 uses the same categorical technique as Proposition 4.21 (see Fig. 15). To a given one-set $M = 1\mathrm{S}\,S$ we build a net $N = \mathrm{MM2}\,M$ which contains a node for each minimal combination of pre and post

$$X = \{(m', m'') \in M \times M \mid \forall l', l'' \in M,\ \lambda \in \mathbb{N}^+:$$
$$\text{if } (m', m'') = (\lambda l', \lambda l'') \text{ then } \lambda = 1 \text{ or } m' = m'' = \mathbf{0}\}.$$

The set of places is an embedding of S in X

$$P = \{(1s, 1s) \in X \mid s \in S\}$$

and pp is defined on a transition $(m', m'') \in T = X \backslash P$ by

$$\mathrm{pp}(m', m'') = \left(\sum_{s \in S} m'(s)(1s, 1s), \sum_{s \in S} m''(s)(1s, 1s) \right).$$

The counit $\varepsilon_M : U\,\mathrm{PL}\,\mathrm{MM2}\,M \to M$ is simply the projection with

$$\varepsilon_M(1s, 1s) = s \quad \text{for } (1s, 1s) \in P.$$

Now, let $f : U\,\mathrm{PL}\,N' \to M$ be a $\underline{1\mathrm{S}}$ morphism. If there exists a factorisation $f = \varepsilon_M U\mathrm{PL}\,g$ with a place-preserving morphism g it fulfils

$$g\,p' = (f_\gamma\,p')(1\,f_\beta\,p', 1\,f_\beta\,p') \quad \text{for } p' \in P',$$

$$g\,t' = \lambda_{t'}((1/\lambda_{t'})g\,\mathrm{pp}'\,t') \quad \text{for } t' \in T' \text{ and an appropriate } \lambda_{t'} \in \mathbb{N}^+.$$

By the definition of X such a $\lambda_{t'}$ is uniquely determined unless $f\,\mathrm{pp}'\,t' = \mathbf{0}$. These equations completely determine g, hence, g is unique. Conversely, these equations define a linear function g which maps places to places and yields the wanted factorisation. That g is indeed a $\underline{\mathrm{PPNET}}$ morphism is derived by

$$g\,\mathrm{pp}'\,t' = \left(\sum_{s \in S} (f\,\mathrm{pre}'\,t')(s)(1s, 1s), \sum_{s \in S} (f\,\mathrm{post}'\,t')(s)(1s, 1s) \right)$$

$$= \begin{cases} \Delta g\,t' \\ \mathrm{pp}(f\,\mathrm{pp}'\,t') = \mathrm{pp}(g\,t'). \end{cases}$$

Thus $\varepsilon_M : U\,\mathrm{PL}\,\mathrm{MM2}\,M \to M$ is an universal arrow from $U\,\mathrm{PL}$ to M which yields the claimed adjunction. □

4.3. Universal constructions

Proposition 4.23. $\underline{\mathrm{PPNET}}$ *is cocomplete and complete.*

Proof. Because $U : \underline{\mathrm{PPNET}} \to \underline{1\mathrm{S}}$ has a right adjoint it preserves colimits [13, Freyd's adjoint functor theorem]. Thus, if a diagram D in $\underline{\mathrm{PPNET}}$ has a colimit it can be constructed by adding a net structure on the colimit $v : U\,D \to V$ in $\underline{1\mathrm{S}}$ which exists by Proposition 2.7. First a bipartite structure is defined such that the v_C maps places to places:

$$P = \{v_{C,\beta}\,p \in B\,V \mid p \in P_C \text{ for an object } C \text{ of } D\},$$

$$T = B\,V \backslash P.$$

In order to build pp we start with functions px_C for each object C of D (see Fig. 16):

$$px_C \in \underline{1\mathrm{S}}[U\,C, 1\mathrm{S}\,P \times 1\mathrm{S}\,P]$$

$$px_C(x) = \begin{cases} v_C(\mathrm{pp}_C\,x) & \text{if } x \in T_C, \\ v_C\,\Delta x, & \text{otherwise.} \end{cases}$$

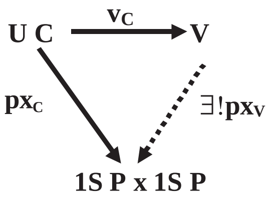


Fig. 16. Construction of pp for a colimit.

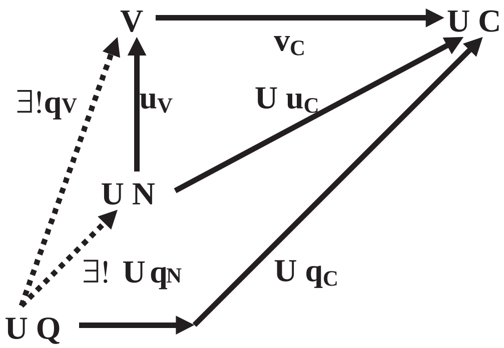


Fig. 17. The limit in PPNET.

The px_C are well defined 1S morphisms, furthermore, they are a natural transformation from D to $1\mathrm{S}\,P \times 1\mathrm{S}\,P$ because the arrows in D are PPNET morphisms and v is natural in 1S. By the universal property of V this yields a unique 1S morphism

$$px_V : V \to 1\mathrm{S}\,P \times 1\mathrm{S}\,P \quad \text{with} \quad px_C = px_V\, v_C$$

and we get pp_V as the restriction of px_V to T. The construction yields for $t \in T_C$

$$v_C\, \mathrm{pp}_C\, t = px_C\, t = px_V\, v_C\, t = \left\{ \begin{array}{c} \mathrm{pp}_V\, v_C\, t \\ \Delta v_C\, t \end{array} \right\}.$$

Thus, pp_V turns V into a net $N = \mathrm{pp}_V$ and the v_C and v in PPNET-morphisms. The universal property of $v : D \to N$ in PPNET follows from the universal property of V in 1S, the construction of pp ensures that the connecting 1S morphism is a PPNET morphism.

To show completeness a limit is constructed by modifying a limit in 1S. Let $v : V \to U\,D$ be the limit in 1S of a $U\,D$ for a diagram D in PPNET as depicted in Fig. 17. Define a net $N = \mathrm{pp}$ with places the base elements from V that are mapped to places and ‘all necessary transitions’:

$$P = \{ p \in B\,V \mid \exists \text{ an object } C \text{ of } D\text{: } p \in \mathrm{def}\, v_{C,\beta} \text{ and } v_{C,\beta}\, p \in P_C \},$$

$$\begin{aligned} T = \{ &(t, y) \in (B\,V \backslash P) \times (1\mathrm{S}\,P) \times (1\mathrm{S}\,P) \mid \\ &\forall \text{ objects } C \text{ of } D\text{: } v_C\, y = (\text{if } v_{C,\beta}\, t \in T_C \text{ then } v_C \text{ pp}\, t \text{ else } \Delta v_C\, t) \} \end{aligned}$$

$$pp(t, y) = y.$$

Clearly $N = \mathrm{pp}$ is a net. If the projection $u_V \in 1\mathrm{S}[UN, V]$ is defined by

$$u_V\, x = \text{if } x \in P \text{ then } x \text{ else if } x = (t, y) \in T \text{ then } t$$

then by construction the compositions $u_C = v_C\, u_V$ become PPNET morphisms from N to C and the u_C form a natural transformation $u : N \to D$ in PPNET.

	[7]	PTSYS	PPSYS	FSYS	[28]	[17]
○ ⇒ 2* ○	∅	√	√	√	√	∅
○ ⇒ ○ ○	∅	∅	∅	∅	√	√
○→□→○ ⇒ ○	√	√	√	∅	∅	∅
○→□→○ ⇒ □	√	√	∅	∅	∅	∅
□ ⇒ 2* □	∅	√	√	√	∅	∅

Fig. 18. Allowed (√) and disallowed (∅) mappings for different morphisms.

It remains to show that N has the universal property. For that, let $q: Q \to D$ be a natural transformation. The universal property of V yields a unique morphism $q_V \in \underline{1S}[UQ, V]$ as shown in Fig. 17. If the required connection $q_N : Q \to N$ exists it must fulfil for any $x \in X_Q$:

$$q_N\, x = \text{if } q_V\, x = \mathbf{0} \text{ or } q_{V\beta}\, x \in P_Q \text{ then } q_V\, x \text{ else } (q_V\, x, q_V\, \mathrm{pp}\, x).$$

This implies the uniqueness of q_N. The existence of q_N follows by a translation of the $\underline{\text{PPNET}}$ morphism definition of $q_C = v_C\, q_V$ to the construction of N. □

It is interesting to compare this proposition with the adjunctions in Proposition 4.22. $U\,P : \underline{\text{PPNET}} \to \underline{\text{1S}}$ has a left adjoint and preserves limits. That it vanishes on transitions gives the freedom to multiply the transitions for all possible combinations of pre- and post-sets. This would be impossible if the adjunctions over $\underline{\text{PTNET}}$ would compose and $U\,U : \underline{\text{PPNET}} \to \underline{\text{1S}}$ had a left adjoint. We can reiterate the remark after the universal constructions in $\underline{\text{1S}}$: colimits are computationally reasonable, their size does not exceed the disjoint union, but limits have a tendency for infiniteness.

4.4. Foldings

Definition 4.24. $\underline{\text{FNET}}$ is the subcategory of $\underline{\text{PTNET}}$ with the same objects but only foldings as morphisms.

Fig. 18 shows different mappings of nodes and tabulates in which categories they are allowed for morphisms. It lists our three categories, a typical clustering-based one [7] and folding-based ones [26,16]. However, this table reflects symptoms only. It cannot explain the underlying principles that, e.g. in [16], are quite different from ours.

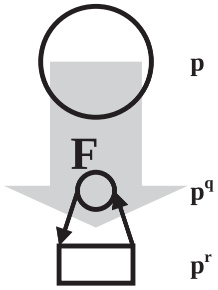


Fig. 19. The functor F.

Proposition 4.25. *There is a functor F from* <u>PPNET</u> *to* <u>FNET</u> *which is right adjoint to the underlying functor U.*

$$\begin{tikzcd} \underline{\mathbf{FNET}} \arrow[r, shift left, "U"] \arrow[r, phantom, "\perp"] & \underline{\mathbf{PPNET}} \arrow[l, shift left, "F"] \end{tikzcd}$$

Proof. Basically, F replaces every place by one place and one transition as shown in Fig. 19. This may be thought of as moving outside the transition hidden inside a place in <u>PPNET</u>. Formally, N_F is constructed from a given net N by two new symbols q and r

$$P_F = \{p^q \mid p \in P_N\}$$

$$T_F = \{x^r \mid x \in X_N\}$$

the p^q and x^r notation is extended to (pairs of) multisets in the obvious way

$$\mathrm{pp}_F\, x^r = \begin{cases} (\mathrm{pp}_N\, x)^q & \text{if } x \in T_N, \\ (x^q, x^q), & \text{otherwise.} \end{cases}$$

$$\varepsilon_N(x^y) = x \quad \text{for } x^y \in X_F.$$

Clearly N_F is a net and ε_N extends to a place-preserving morphism. For any <u>PPNET</u> morphism $f : N' \to N$ there is a unique factorisation into a folding g and ε_N (analogous to Fig. 14). g is defined on $x' \in X'$ by

$$g(x') = \begin{cases} (f\,x')^q & \text{if } x' \in P,' \\ (f\,x')^r, & \text{otherwise.} \end{cases}$$

This is the only possibility to get a connecting folding. On the other hand, g is a well defined folding for any <u>PPNET</u> morphism f. The same argumentation as in the proof of Proposition 4.21 yields that there is a functor F with $FN = N_F$ and that it is right-adjoint to U. $\square$

	[7]	PTSYS	PPSYS	FSYS	[28]	[17]
$\forall p \in P$: f p	$\in X'$	$\in \mathbb{N}\, X'$	$\in \mathbb{N}\, P'$	$\in \mathbb{N}\, P'$	$\in$ MS P'	$\subseteq P'$
$\forall t \in T$: f t	$\in X'$	$\in \mathbb{N}\, X'$	$\in \mathbb{N}\, X'$	$\in \mathbb{N}\, T'$	$\in T' \cup$undef	$\in T' \cup$undef
f I	-	$\leq I'$	$\leq I'$	$\leq I'$	$= I'$	$= I'$

Fig. 20. Comparison of a morphism $f: S \to S'$ in different categories.

Proposition 4.26. FNET *is cocomplete and complete.*

Proof. The colimit of a diagram is the same as in PPNET: the universal connection automatically becomes a folding in FNET. Limits are different: transitions have to get paired only with transitions. With this change the same construction and proof works as for Proposition 4.23. □

5. Structure and behaviour

This section introduces categories for net systems and shows the relationships to the structural categories from the last sections. As usual, a net system is defined as a net with an initial marking but morphisms are new. As announced in the introduction, this yields modelling power and an impressive web of adjunctions. Systems are not cocomplete or complete as the underlying nets are. But we will characterise the existence and construction of (co)universal constructions in an intuitive way (Proposition 5.31).

5.1. Net systems

Definition 5.27. A system S is a pair $S=(N_S, I_S)$ with a net N_S and an initial marking $I_S \in 1\mathrm{S}\, P_S$. The categories PTSYS, PPSYS and FSYS respectively have systems as their objects. A morphism $f: S \to S'$ is a morphism $N_S \to N_S'$ of PTNET, PPNET and FNET respectively which fulfils $f\, I_S \leqslant I_S'$. ∗SYS and ∗NET symbolise any of these pairs of corresponding categories.

As usual the initial marking is a multiset of places. The idea behind the definition is that the image of an enabled step sequence in the source system is enabled in the destination system (for foldings, the situation is more complicated for general morphisms). For this it is sufficient that the image of the initial marking is contained in the initial marking of the destination net. This generalisation over the usual equal yields several benefits. First, the empty net with marking **0** becomes the zero object of the category. Secondly, as shown in Fig. 2, it allows to simulate subsystems, parallel compositions, etc.

The objects in the categories ∗SYS, in [26,16] are the same up to technical restrictions such as finiteness or emptiness. Genrich et al. [7] provides no initial markings and no arc weights. For morphisms the situation is tabulated in Fig. 20.

Definition 5.28. Let $S=(N_S,I_S)$ and S' be systems, $f:S\to S'$ be a morphism and define the following functors:

NET: $\underline{*\mathrm{SYS}}\to\underline{*\mathrm{NET}}$ maps S to N_S and f to f.
IP: $\underline{*\mathrm{SYS}}\to\underline{*\mathrm{NET}}$ maps S to the net (PTNET $\mathrm{supp}\, I_S$) and f to the restriction of f to IP S.
IM: $\underline{*\mathrm{SYS}}\to\underline{*\mathrm{SYS}}$ maps S to $(\mathrm{IP}\, S, I_S)$ and f to IP f
SYS0: $\underline{*\mathrm{NET}}\to\underline{*\mathrm{SYS}}$ maps a net N to $(N,\mathbf{0})$ and a morphism to itself.

NET forgets the initial marking, IP retains only the places marked by the initial marking, IM retains only the initial marking and SYS0 adds a zero initial marking.

Proposition 5.29. SYS0: $\underline{*\mathrm{NET}}\to\underline{*\mathrm{SYS}}$ *forms a coreflection with* NET. PP *and F lift to systems giving the commutative adjunction diagram alongside.*

$$\begin{tikzcd}
\underline{\mathbf{FNET}} \arrow[r, shift left, "\mathbf{SYS0}"] \arrow[d, shift left, "\mathbf{U}"] & \underline{\mathbf{FSYS}} \arrow[l, shift left, "\mathbf{NET}"] \arrow[d, shift left, "\mathbf{U}"] \\
\underline{\mathbf{PPNET}} \arrow[u, shift left, "\mathbf{F}"] \arrow[r, shift left, "\mathbf{SYS0}"] \arrow[d, shift left, "\mathbf{U}"] & \underline{\mathbf{PPSYS}} \arrow[u, shift left, "\mathbf{F}"] \arrow[l, shift left, "\mathbf{NET}"] \arrow[d, shift left, "\mathbf{U}"] \\
\underline{\mathbf{PTNET}} \arrow[u, shift left, "\mathbf{PP}"] \arrow[r, shift left, "\mathbf{SYS0}"] & \underline{\mathbf{PTSYS}} \arrow[u, shift left, "\mathbf{PP}"] \arrow[l, shift left, "\mathbf{NET}"]
\end{tikzcd}$$

Proof. Clearly

$$\eta = \mathrm{id} : [\mathrm{SYS0}\, N, S] \to [N, \mathrm{NET}\, S]$$

is natural and injective because it is the interpretation of the same morphism once in a system and once in a net. Because $f\,\mathbf{0}=\mathbf{0}\leqslant I_S$ it is also surjective and hence a bijection as required for the claimed adjunction. The unit ID $\to$ NET SYS0 adds and removes an initial marking from a net yielding an isomorphism and turning the adjunction into a coreflection.

Let $S'=(N',I')$ a place-preserving system. Interpret for a moment N' as an object of $\underline{\mathrm{PPNET}}$. The unit $\varepsilon'_N : N'\to \mathrm{NET}\,\mathrm{PP}\,N'$ gives a place-preserving morphism $\varepsilon' : N'\to \mathrm{PP}\,N'$. Hence $\varepsilon' I'$ is a marking in $\mathrm{PP}\,N'$ and $\mathrm{PP}\,S'=(\mathrm{PP}\,N',\varepsilon' I')$ is well defined. Although $\varepsilon' : \mathrm{Id}\to\mathrm{PP}$ is not natural, it is 'natural on places' and this suffices for

$$(\mathrm{PP}\, f)I_{\mathrm{PP}\, S} = (\mathrm{PP}\, f)\varepsilon I_S = \varepsilon' f\, I_S \leqslant \varepsilon' I'_S$$

hence PP is well defined on morphisms.

Let $\eta_{N,N'}:\underline{\text{PTNET}}[U\,N,N']\to\underline{\text{PPNET}}[N,\text{PP}\,N']$ the natural equivalence given by the adjunction between NET and PP. η lifts to systems by

$$\eta_{N,N'}:\underline{\text{PTSYS}}[U(N,I),(N',I')]\to\underline{\text{PPSYS}}[(N,I),(\text{PP}\,N',\varepsilon'\,I')]$$

for the same reason, namely, that ε is 'natural for the initial marking I'. F is lifted in the same way. Commutative adjunctions means that left adjoints are composed with left adjoints and right adjoints with right adjoints. That these compositions commute follows directly from the construction. □

Proposition 5.30. IM : $\underline{*\text{SYS}}\to\underline{*\text{SYS}}$ *has a right adjoint, namely* MM2I *for* $\underline{\text{PTSYS}}$ *and* $\underline{\text{PPSYS}}$ *and* FMM2I *for* $\underline{\text{FSYS}}$.

$$\begin{tikzcd} \underline{\textbf{*SYS}} \arrow[r, shift left, "\textbf{IM}"] \arrow[r, phantom, "\nabla"] & \underline{\textbf{*SYS}} \arrow[l, shift left, "\textbf{*MM2I}"] \end{tikzcd}$$

Proof. We have to lift the functor MM2 and the appropriate part of Proposition 4.22 to systems. For that define the object function and counit ε_S by

$$\text{MM2}I\,S=(\text{MM2 IP}\,S,I)$$

$\varepsilon_S=\iota_S\varepsilon_{\text{IP}\,S}:\text{IM MM2}I\,S\to S$ with $\varepsilon_{\text{IP}\,S}:U\,\text{PL MM2 IP}\,S\to\text{IP}\,S$ the counit of the adjunction of U PL and MM2 and $\iota_S:\text{IP}\,S\to\text{NET}\,S$ the natural embedding.

This works fine on the initial marking and the universal property follows from that in $\underline{\text{PPNET}}$. This proves the claim for $\underline{\text{PPSYS}}$. It implies the claim for $\underline{\text{PTSYS}}$ because IM f is place-preserving for any place–transition morphism $f:S\to S'$ by

$$f_\beta\,\text{supp}\,I\subseteq\text{supp}\,I'\subseteq P'.$$

Finally, for $\underline{\text{FSYS}}$ we must define

$$\text{FMM2I}\,S=(F\,\text{MM2 IP}\,S,\varepsilon_{\text{MM2 IP}\,S}\,I)$$

IM FMM2I S equals IM MM2I S, thus, the counit ε_S may be defined as above and F lifts the universality of MM2I to $\underline{\text{FSYS}}$ by Proposition 4.25. □

5.2. Universal constructions

Proposition 5.31. *A diagram D in* $\underline{\text{PPSYS}}$ *or* $\underline{\text{FSYS}}$ *has a colimit respectively a limit iff* IM D *has.*

Proof. The only if direction for colimits follows from Proposition 5.30: because IM has a right adjoint it preserves colimits [13]. For limits let $u:U\to D$ be the limit of D in $\underline{*\text{SYS}}$ as shown in Fig. 21. A natural transformation $q:Q\to\text{IM}\,D$ yields a natural transformation $\iota\,q:Q\to D$ with $\iota:\text{IM}\to\text{ID}$ the obvious natural transformation embedding an initial marking in its system. The universality of U yields a unique $r_U:R\to U$. But this r_U maps R to the initial marking of U hence it uniquely retracts to a unique $q':Q\to\text{IM}\,U$ with $j_U\,q'=q_U$. Thus IM U is the limit of IM D.

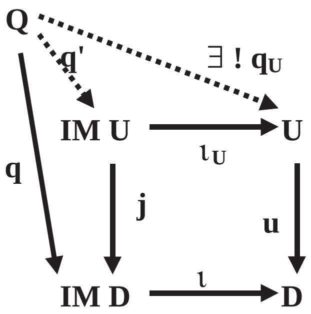


Fig. 21. IM on limits.

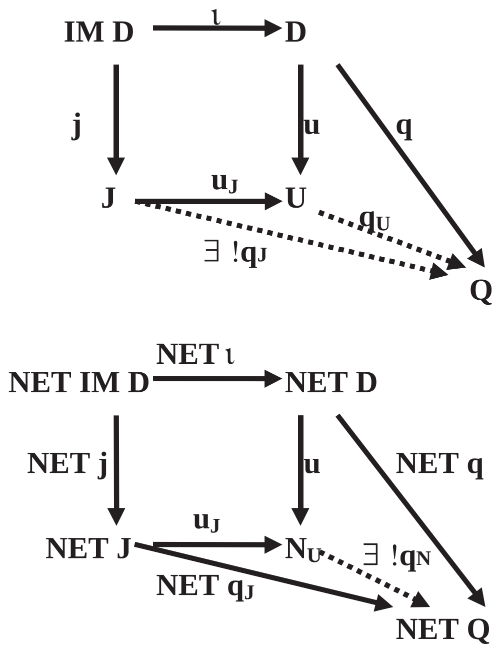


Fig. 22. Construction of a colimit.

To construct the colimit of a diagram D let $j:\mathrm{IM}\,D \to J$ be the colimit of the initial marking. The combined diagram $(j:\mathrm{IM}\,D \to J, \iota:\mathrm{IM}\,D \to D)$ has a colimit in $\underline{*\mathrm{NET}}$

$$u : \mathrm{NET}(j : \mathrm{IM}\,D \to J, \iota : \mathrm{IM}\,D \to D) \to N_U$$

as shown in Fig. 22. To lift this colimit to $\underline{*\mathrm{SYS}}$ let

$$U = (N_U, u_J\, I_J).$$

For any object C of Du_C preserves the initial marking by

$$u_C\, I_C = u_C\, \iota_C\, I_{\mathrm{IM}\,C} = u_J\, j_{\mathrm{IM}\,C}\, I_{\mathrm{IM}\,C} \leqslant u_J\, I_J = I_U.$$

To show that U has the necessary universal property let $q:D \to Q$ be a natural transformation in $\underline{*\mathrm{SYS}}$. The universal property of J yields a unique $q_J:J \to Q$. (NET q) together with NET q_J yields a unique $q_N:N_U \to \mathrm{NET}\,Q$ by the universal property of N_U. q_N uniquely retracts to a $q_U:U \to Q$ with NET $q_U = q_N$ because it preserves the initial marking by

$$q_U\, I_U = q_U\, \iota_J\, I_J = q_J\, I_J \leqslant I_Q.$$

q_U is unique because it is determined by NET $q_U = q_N$ which is unique. Thus U is the claimed colimit.

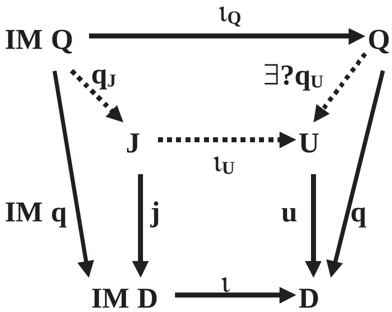


Fig. 23. Construction of a limit.

NET has a left adjoint and preserves limit. If $u : U \to D$ is the limit of a diagram D in $\underline{*\mathrm{SYS}}$ then NET $u : N_U \to \mathrm{IM}\, U$ is the limit in $\underline{*\mathrm{NET}}$. Thus to construct the limit U of D we have to find an appropriate initial marking which lifts N_U to $U = (N_U, I_U)$. For this we use the limit $j : J \to \mathrm{IM}\, D$. First, $\iota\, j : J \to D$ is natural and the universal property of U yields a $\iota_U : \mathrm{NET}\, J \to N_U$. The definition $I_U = \iota_U\, I_J$ makes U a system and ι_U a morphism in $\underline{*\mathrm{SYS}}$. This also holds for u_C by

$$u_C\, I_U = u_C\, j_U\, I_J = \iota_C\, j_{\mathrm{IM}\, C}\, I_J \leqslant I_C \text{ for any object } C \text{ of } D.$$

Given a natural transformation $q : Q \to D$ in $\underline{*\mathrm{SYS}}$ there are unique q_J and q_U by the universality of J and N_U respectively, as shown in Fig. 23. Clearly, $J = \mathrm{IM}\, U$, $q_J = \mathrm{IM}$ q_U and by $\iota : \mathrm{IM}\, Q \to \mathrm{ID}$ the whole diagram is the natural transformation of the commutative left-hand triangle to the right-hand triangle. Hence, q_U preserves the initial marking and is the required unique morphism in $\underline{*\mathrm{SYS}}$. Thus, U is the limit of D. $\square$

Remark. The prerequisite in $\mathrm{IM}\, D$ may be weakened to naturality whereas uniqueness may be dropped. A transfer of this proposition to $\underline{\mathrm{PTNET}}$ must explicitly consider the existence of the specific colimit (limit respectively) in $\underline{\mathrm{PTNET}}$ which has been used in the proof.

6. Reachability and liveness

This section looks at reachability semantics—one of the simplest and most frequently used ones. Mapping a system to its step reachability is shown to be functorial for place-preserving nets. This was to be expected—the intuition behind the definition of morphisms was the transfer of step occurrences. But in place–transition nets morphisms allow the mapping of ‘half transitions’ by handling the pre- and post-functions of a transition separately. In contrast to other authors which formalise the split of a transition in an input and an output occurrence we use the adjunction to place-preserving nets. Reachability semantics becomes functorial there—with transition occurrences remaining atomic.

Definition 6.32. $\underline{\text{SM}}$ the category of state machines is the full subcategory of $\underline{\text{PPSYS}}$ with the objects fulfilling

all arc weights are one and each transition has exactly one input and one output place

$I = 1\,p_I$: the initial marking consists of a single token on a single place p_I,

$X = p_I F^*$ with F^* the transitive closure of the flow-relation F: neither dead transition nor never-marked places.

In our framework it is easier to handle step than sequential reachability:

Definition 6.33. SM$\,S$ for a system $S = (N, I)$ is the state machine (N', I') with

$P' =$ all markings of S reachable from I including I itself

$T' = \{(m, \sigma) \mid m \in P'\ \sigma \in 1\mathrm{S}\,T$ with $m[\sigma >\}$

$\mathrm{pp}(m, \sigma) = (m, m')$ with $m[\sigma > m'$.

$I' = I$.

In short, SM unfolds a system to its step reachability graph.

Proposition 6.34. *SM extends to a functor* $SM : \underline{\text{PPSYS}} \to \underline{\text{SM}}$.

Proof. To define SM on a morphism $f : S \to S'$ let $m[\sigma > m'$ be a step occurrence in S and

$$(\mathrm{SM}\, f)\, m = f\, m + (I' - fI),$$

$$\underline{\sigma} = \sum_{t \in T \text{ and } f_\beta t \in T} \sigma(t)t$$

$$(\mathrm{SM}\, f)(m, \sigma) = \begin{cases} fm + I' - fI & \text{if } \underline{\sigma} = 0, \\ (f\, m, f\, \underline{\sigma}), & \text{otherwise.} \end{cases}$$

The first line maps markings to markings because f is place-preserving. That SM$\,f$ maps enabled step sequences to enabled step sequences is shown by induction over the length of enabled step sequences. Induction start for the initial marking is obvious. For the induction step let m be reachable in S, $(\mathrm{SM}\, f)\ m$ be reachable in S' and $m[\sigma > m'$ be an enabled step sequence in S. Then

$$(Sf)\, m \geqslant \mathrm{pre}\, f\, \sigma \geqslant f\, \mathrm{pre}\, \underline{\sigma} = \mathrm{pre}\, f\, \underline{\sigma}$$

shows that $(Sf)\, m[(Sf)\, \underline{\sigma} >$ is enabled in S' and

$$\begin{aligned}(\mathrm{SM}\, f)m' - (fI - I') = f\, m' &= f\, m + f(-\mathrm{pre} + \mathrm{post})\sigma = f\, m + f(-\mathrm{pre} + \mathrm{post})\underline{\sigma} \\ &= f\, m + (-\mathrm{pre}' + \mathrm{post}')f\, \underline{\sigma} = f\, m + (-\mathrm{pre}' + \mathrm{post}')(\mathrm{SM}\, f)\sigma\end{aligned}$$

shows that $(\mathrm{SM}\, f)m[(\mathrm{SM}\, f)\sigma > (\mathrm{SM}\, f)m'$ is enabled in S' and that $(\mathrm{SM}\, f)m'$ is a reachable marking of S'. Hence, SM$\,f$ is a place-preserving morphism from SM$\,S$ to SM$\,S'$. SM is compositional because the source morphism are place-preserving. □

SM clarifies how a morphism transfers properties which may be deduced from the reachability graph.

Proposition 6.35. *For a place-preserving morphism $f: S \to S'$, a transition $t \in T_S$ with $f_\beta\, t \in T'$ and reachable markings m and m' of S hold*:
- *if m' is reachable from m then also fm' from fm,*
- *if $f_\beta\, t$ (even $f\,t$ suffices) is dead at $f\,m$ then also t at m,*
- *if S is live and f epimorphic then also S' is live,*
- *if S' is bounded at $f_\beta\, p$ by k' then S is bounded at p by $k'/(f_\gamma\, p)$.*

Proof. Use SM to map the pertinent step sequences. □

Proposition 6.36. $\mathrm{SM}: \underline{\mathrm{PPSYS}} \to \underline{\mathrm{SM}}$ *has neither a left nor a right adjoint.*

Proof. Let $N_{i\text{-}k}$ be the net consisting of a single transition with i input places and k output places, all arc weights one and all $i+k$ places disjoint. SM sends any $N_{i\text{-}k}$ with i not zero to $N_{1\text{-}1}$. Let L be a left adjoint of SM. What is $L\,N_{1\text{-}1}$? Consider $[L\,N_{1\text{-}1}, N_{3\text{-}1}] \cong [N_{1\text{-}1}, N_{1\text{-}1} = \mathrm{SM}\,N_{3\text{-}1}]$. If the initial marking of $L\,N_{1\text{-}1}$ is zero there are either 1 or infinitely many morphisms in the left morphism set. Hence there must be at least one place with a single token. But the permutations of the three input-places are automorphisms of $N_{3\text{-}1}$. Hence if there is not only the 0 morphism in the considered set there are at least 4 morphisms. Contradiction to a bijection between the two morphism sets.

For a right adjoint R consider $[N_{1\text{-}1} = \mathrm{SM}\,N_{3\text{-}1}, N_{1\text{-}1}] \cong [N_{3\text{-}1}, R\,N_{1\text{-}1}]$. Again the initial marking cannot be zero and with the permutations of the input-places of $N_{3\text{-}1}$ there are more morphisms in the right morphism set. □

This proposition shows that SM has its limitations. Other authors, e.g., [19,16] get adjunctions here. Their morphisms may map a place to a marking. Our definition enforces that a place is mapped to (the multiple of) a place in order to enforce a tight relationship with the underlying graph. The missing adjoint for SM is a disadvantage of this decision. In our setting reachability semantics abstracts too much from the graph of a system to allow an adjunction.

7. Process semantics

The last subject of this paper is process based semantics in the style of, e.g., [15]. It models the interplay between causality and branching in the form of occurrence systems. It differentiates between two occurrences of a transition if they consume tokens produced by different transition occurrences. One could understand it as the labelling of tokens and transition occurrences with their history.

We introduce a functor WOCC which unfolds a system into a (weighted) occurrence system that contains all processes of the system. WOCC is part of a coreflection between $\underline{\mathrm{FSYS}}$ and $\underline{\mathrm{WOCC}}$ the category of weighted occurrence systems. These represent a new variant of process semantics—known from the literature are (safe)

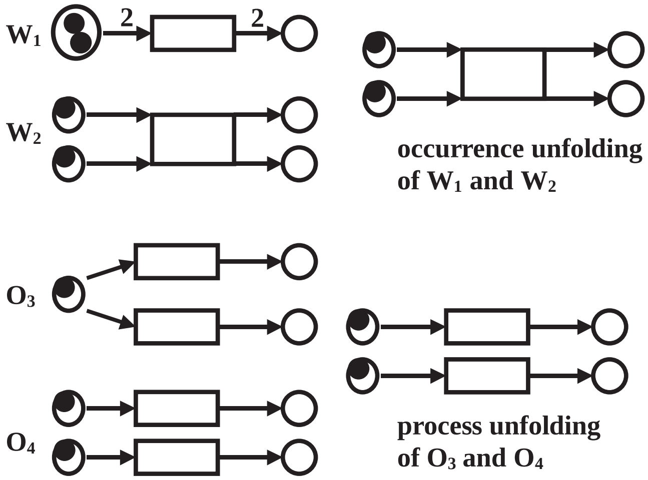


Fig. 24. A weighted occurrence system W_1, a safe occurrence system O_3, processes W_2 and O_4 and their unfoldings.

occurrence nets (e.g. [14,3], etc.). Fig. 24 shows examples for the different systems and unfoldings.

It is remarkable that WOCC is functorial and an adjunction on the whole category FSYS. For unfoldings to safe occurrence systems this has been achieved only for subcategories (e.g. [26,14]). This will be discussed in the last two sections for known methods from literature as well as for our categories. Such unfoldings require restrictions that limit the freedom of how to map tokens with the same history. Such tokens are not separated by WOCC.

7.1. Processes

Definition 7.37. A system is safe iff all arc weights are one and any reachable marking is a set (rather than a multiset).

Definition 7.38. The category WOCC of weighted occurrence systems is the full subcategory of FSYS whose objects O fulfil

F is acyclic and $\forall x \in X_O$: $F^* x$ is finite with F the flow-relation of O
$|^\bullet p| \leqslant 1$ for any place $p \in P_O$ and
$\operatorname{supp} I_O = \{p \in P_O \mid {}^\bullet p = \{\}\}$
O has no dead transitions.

The category OCC of (safe) occurrence systems is the full subcategory of WOCC of safe systems and the category PROC of processes is the full subcategory of OCC with

$$\forall p \in P_O: \ |p^\bullet| \leqslant 1.$$

A process of a system S is a process O together with a unitary folding $o: O \to S$. Similarly a (weighted, safe) occurrence system of a system is defined. The depth of a node of a weighted occurrence system is

0	for places with empty pre-set and 1 for transitions with empty pre-set,
$i+1$	for a transition t and its post-places if i is the maximal depth of a pre-place of t.

Fig. 24 shows examples for the definitions. Processes are acyclic safe systems. Each place not marked by the initial marking has a unique input transition able to deliver a single token on it and a unique output transition that may consume this token later. Our definition of processes coincides with the standard definition and as usual the image of a maximal place cut of a process is a reachable marking of the system (refer to [22]). Notice that we allow transitions with empty pre-set in $\underline{\text{WOCC}}$ but such transitions are not allowed in safe systems.

7.2. Weighted occurrence systems

The set of all processes is a bit awkward. It would be nicer to fold the set of processes into a single net. This is exactly the purpose of occurrence systems. First we need a generalisation of a maximal place cut:

Definition 7.39. A cut step of a system S is a multiset γ of transitions with $\operatorname{pre}\gamma \leqslant I_S + \operatorname{post}\gamma$.

Lemma 7.40. *The image of the transitions of a finite process of a weighted occurrence system is a cut step and each cut step is such an image. Hence, a marking is reachable iff it equals* $(I - \operatorname{pre}\gamma + \operatorname{post}\gamma)$ *for a cut step* γ. *A process of a weighted occurrence system of a system* S *extends to a process of* S.

Proof. Let $o: O \to W$ be a process of a weighted occurrence system and $T_i \subseteq T_O$ be the transitions of O of depth i. The definition of a process and induction over i yields

$$\operatorname{pre} T_i \leqslant I_O + \sum_{k=1}^{i-1} -\operatorname{pre} T_k + \operatorname{post} T_k$$

and because o is a folding

$$\operatorname{pre} o\, T_i \leqslant I_W + \sum_{k=1}^{i-1} -\operatorname{pre} o\, T_k + \operatorname{post} o\, T_k$$

Thus $T_1, T_2, T_3, \ldots$ is an enabled step sequence in $O, o\,T_1, o\,T_2, o\,T_3, \ldots$ is an enabled step sequence in W and $o\,T$ is a cut step in W. To find a process for a given cut step γ of a weighted occurrence system W let

$$T_O = \{(t,k) \in T_W \times \mathbb{N}^+ \mid k \leqslant \gamma(t)\},$$

$$P_O = \{(p,j) \in P_W \times \mathbb{N}^+ \mid j \leqslant I_W(p)\}$$
$$\cup \{((t,k), p, j) \in T_O \times P_W \times \mathbb{N}^+ \mid j \leqslant (\operatorname{post}_W t)(p)\},$$

$$\operatorname{post}_O(t,k) = \sum_{((t,k),l) \in P_O} ((t,k), l).$$

In order to construct the pre function we first write γ as the sum

$$\gamma = \sum_{i=1}^{n} \gamma_i \text{ with } \gamma_i \text{ a multiset of transitions of depth } i.$$

Because $\gamma_r^{\bullet}$ is disjoint from $\gamma_S^{\bullet}$ for any $r \neq s$ and by induction over i it follows that

$$\mathrm{pre}\, \gamma_i \leqslant I_W + \sum_{k=1}^{i-1} -\mathrm{pre}\, \gamma_k + \mathrm{post}\, \gamma_k.$$

Thus, again by induction over the depth i, for any place $p \in P_W$ there is an injection β_p which maps the consumed tokens to the produced tokens P_O:

$$\beta_p : \{((t,k), p, j) \in T_O \times \{p\} \times \mathbb{N}^+ \mid j \leqslant (\mathrm{pre}_W t)(p)\} \to P_O$$

with $\beta_p((t,k), p, j) = (p, j')$ or $= ((t',k'), p, j')$

$$\mathrm{pre}_O(t,k) = \sum_{((t,k),p,j) \in \mathrm{def}\, \beta_p} \beta_p((t,k), p, j),$$

$$O = \left((\mathrm{pre}_O \mathrm{post}_O), \sum_{(p,j) \in P_O} (p,j) \right).$$

By construction O is a process and the projection

$$o : X_O \to X_W \text{ with } o(t,k) = t, \quad o(p,k) = p \quad \text{and} \quad o((t,k), p, j) = p$$

extends to a unitary folding $o : O \to W$. Hence it is the claimed process of W.

Finally, given $w : W \to S$ a weighted occurrence system of a system and a process $o : O \to W$ then $wo : O \to S$ is a process of S. $\square$

This lemma shows how to compute the processes of a weighted occurrence system. This also works for a system in general if an appropriate occurrence system is available. This is accomplished by:

Proposition 7.41. *There is a functor* WOCC : $\underline{\text{FSYS}} \to \underline{\text{WOCC}}$ *which is right adjoint to the underlying functor* U *forming a coreflection.*

$$\begin{tikzcd} \underline{\mathbf{FSYS}} \arrow[r, shift left, "\mathbf{WOCC}"] \arrow[r, phantom, "\perp"] & \underline{\mathbf{WOCC}} \arrow[l, shift left, "\mathbf{U}"] \end{tikzcd}$$

Proof. As in the proof of Proposition 4.21, it is sufficient to unfold a system S into a weighted occurrence system WOCC S, to construct an universal arrow δ_S from U to S and to verify that the units are isomorphisms. The proof translates constructions from [14] in our framework.

WOCC S is constructed as the colimit O_ω of the infinite diagram in Fig. 25. There, O_ω is approximated by a sequence of weighted occurrence systems O_i of maximal depth i. The diagram is constructed by induction. Induction start: ε_0 is the natural embedding from $O_0 = \mathrm{IM}\, S$ into S.

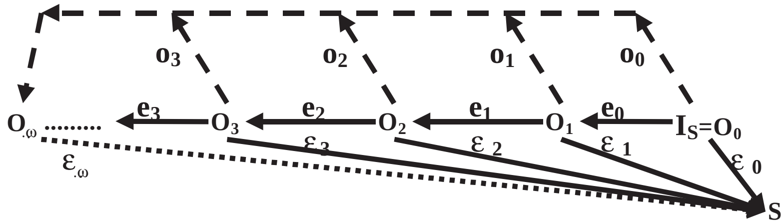


Fig. 25. The infinite diagram and its colimit O_ω.

Induction step from O_i to O_{i+1}. O_i is expanded as follows:

$$T_{i+1} = T_i \cup \{(m,t) \mid t \in T_S \text{ is a transition of depth } i+1 \text{ and } m \text{ is a marking of } O_i \text{ with } \varepsilon_i\, m = \text{pre}\, t \text{ and } m \leqslant m' \text{ for an } m' \text{ reachable in } O_i\}$$

$$P_{i+1} = P_i \cup \{(m,t,p) \mid (m,t) \in T_{i+1} \backslash T_i \text{ and } p \in t^\bullet\}$$

$$\text{pp}_{i+1}(t') = \text{if } t' \in T_i \text{ then } \text{pp}_i\, t' \text{ else if } t' = (m,t) \text{ then}$$

$$\left(m, \sum_{p \in t^\bullet} ((\text{post}\, t)(p))(m,t,p) \right).$$

This defines an occurrence system $O_{i+1} = (\text{pp}_{i+1}, I_i)$ and the morphism $e_i : O_i \to O_{i+1}$ is the embedding used in the construction. ε_{i+1} is the unique extension of ε_i with

$$\varepsilon_{i+1}(m,t) = t \quad \text{for } (m,t) \in T_{i+1} \backslash T_i,$$

$$\varepsilon_{i+1}(m,t,p) = p \quad \text{for } (m,t,p) \in P_{i+1} \backslash P_i.$$

It is straightforward to see that with ε_i also $\varepsilon_{i+1} : O_{i+1} \to S$ is a weighted occurrence system of S and e_i is a monomorphism with $\varepsilon_i = \varepsilon_{i+1} e_i$.

This defines an infinite diagram $D = (e_0, e_1, e_2, \ldots)$ and IM S is isomorphic to all initial markings hence by Proposition 5.31 the colimit $o : D \to O_\omega$ exists. By the universal property of O_ω there is a unique $\varepsilon_\omega : O_\omega \to S$ making the combined diagram commutative.

We claim that $\varepsilon_\omega : O_\omega \to S$ is a weighted occurrence system of S. ε_ω and o_i are unitary because $\varepsilon_\omega\, o_i = \varepsilon_i$ which is unitary by construction. The o_i are monomorphic because X_ω is simply the union of the X_i. This implies that the pre-set of each place of P_ω consists of a single transition, that F_ω is acyclic, $F_\omega^* x$ is finite for any node x of X_ω and, thus, O_ω is an occurrence system.

Next, we prove that $\varepsilon_\omega : O_\omega \to S$ is a universal arrow from U to S as depicted in Fig. 26. Such an $f : U\, O' \to S$ may be expanded to the diagram in Fig. 27. Its upper part redraws the construction of $O_\omega = \text{WOCC}\, S$. In the lower part each O_i' is the subnet of $U\, O'$ which consists of the nodes of depth at most i. The e_i' and ε_i' are the natural embeddings. O' equals the colimit O_ω' of the diagram $(e_0', e_1', e_2', \ldots)$

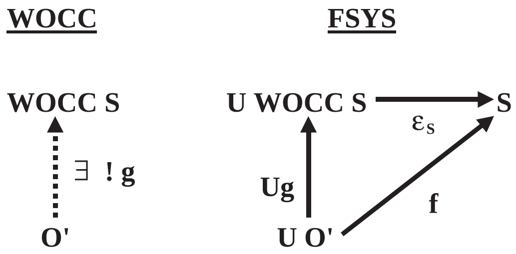


Fig. 26. WOCC by an universal arrow from U to S.

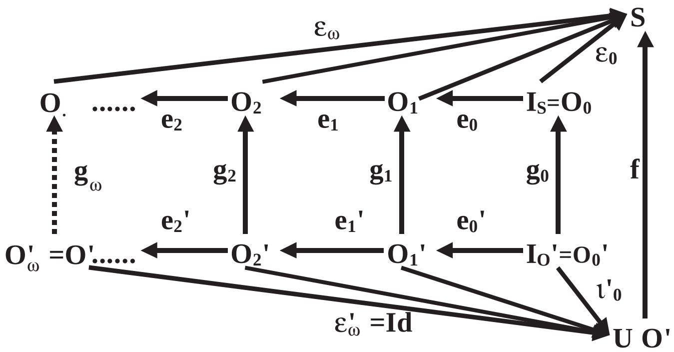


Fig. 27. Universality of $o_\omega : D_\omega \to O_\omega$.

and the monomorphisms $o_i' = \varepsilon_i' : O_i' \to O_\omega'$ make the diagram $(e_0', \varepsilon_0', o_0', e_1', \varepsilon_1', o_1', e_2', \ldots)$ commutative.

By induction over i we show that there exist unique g_i which make the combined diagram (without g_ω) commutative. Induction start: g_0 must equal IM f. Step from i to $i+1$. If such a g_{i+1} exists it is defined on all nodes of depth less or equal i by $g_{i+1} e_i' = e_i g_i$ because e_i' is monomorphic. For any transition $t' \in T_{i+1}'$ of depth $i+1$ it must fulfil

$$\text{pre}\, g_{i+1}\, t' = g_{i+1}\, \text{pre}\, t' = e_i\, g_i\, e_i'^{-1}\, \text{pre}\, t' \ \text{ and } \varepsilon_{i+1}\, g_{i+1}\, t' = f\, \varepsilon_{i+1}'\, t'.$$

This implies by the construction of O_{i+1}

$$\begin{aligned} & g_{i+1}\, t' = (f_\lambda\, \varepsilon_{i+1}'\, t')(m, t) \text{ with} \\ & \quad t = f_\beta\, \varepsilon_{i+1}'\, t' \text{ and} \\ & \quad m = (1/(f_\gamma\, \varepsilon_{i+1}'\, t'))g_i\, e_i'^{-1}\, \text{pre}\, t' \end{aligned}$$

which uniquely determines the image of t'. For the postplaces of t' follows similarly:

$$g_{i+1}\, p' = (f_\lambda\, \varepsilon_i'\, p')(m, t, f_\beta\, \varepsilon_i'\, p') \quad \text{for} \ \ p' \in t'^{\bullet}$$

Together, this uniquely determines the morphism $g_{i+1} : O_{i+1}' \to O_{i+1}$.

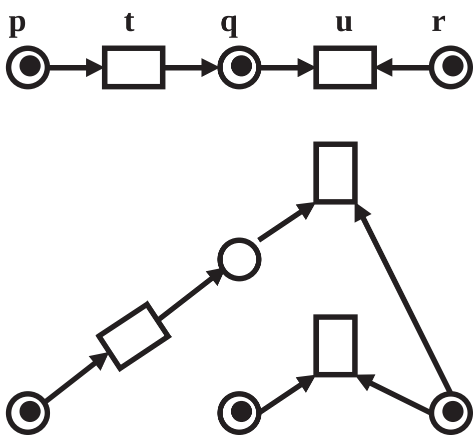


Fig. 28. A system (top) and its unfolding (bottom).

To verify the existence of g_{i+1} notice that there is a step sequence σ' of $U\,O'$ enabling t' by the definition of an occurrence system. $\varepsilon_i\, g_i\, \sigma'$ is a step sequence of S which enables $f\,\varepsilon'_{i+1}\,t'$ and thus t as defined above. Hence, (m,t) is indeed a transition of O_{i+1} and g_{i+1} is well defined by the above equations.

The $o_i g_i$ form a natural transformation from the diagram $(e'_0, e'_1, e'_2, \dots)$ to O_ω. The universal property of O'_ω yields a unique connecting morphism $g_\omega : O'_\omega \to O_\omega$ making the combined diagram commutative and $g = \varepsilon'^{\,-1}_{\omega}\, g_\omega$ is the required connecting morphism.

Thus, the existence of a connecting g is proved. For another g'' which factorises ε_ω, the image of $o'_i\, g''$ consists of nodes of depth less or equal i and $o'_i\, g''$ retracts to a g''_i : $O'_i \to O_i$. This yields again the diagram of Fig. 27 for which we proved that the g_i are unique. Hence, g''_i equals g_i and the universal property of O'_ω implies that g'' equals g.

Finally, a unit ε_O : $\mathrm{WOCC}\, U\, O \to O$ of a weighted occurrence system is obtained in Fig. 25 by setting $S = U\,O$. But as already mentioned, in this situation the O_i are simply the subnets of $U\,O$ of nodes of depth less or equal i, and both $U\,O$ and O_ω are the colimit of the diagram. Hence, they are isomorphic and the unit $\varepsilon_\omega = \varepsilon_S$ is isomorphic. This finishes the proof that U and WOCC form a coreflection. □

As a corollary the processes of a system S are in one to one correspondence with the processes of $\mathrm{WOCC}\,S$. A process $o : O \to \mathrm{WOCC}\,S$ yields a process $\varepsilon_S\, o : O \to S$ and because ε_S is a universal arrow this relationship is bijective. Thus the proposition allows to reduce the computation of the processes of a system to the computation of the processes of its weighted occurrence system.

Fig. 28 illustrates the unfolding by WOCC at an example taken from [8]. An occurrence of transition t creates a second token on place q. According to the individual token philosophy there are two possibilities for the occurrence of u:

- either the initial token on q is consumed: such an occurrence is concurrent with t,
- or the token produced by t is consumed: such an occurrence is causal dependent from t.

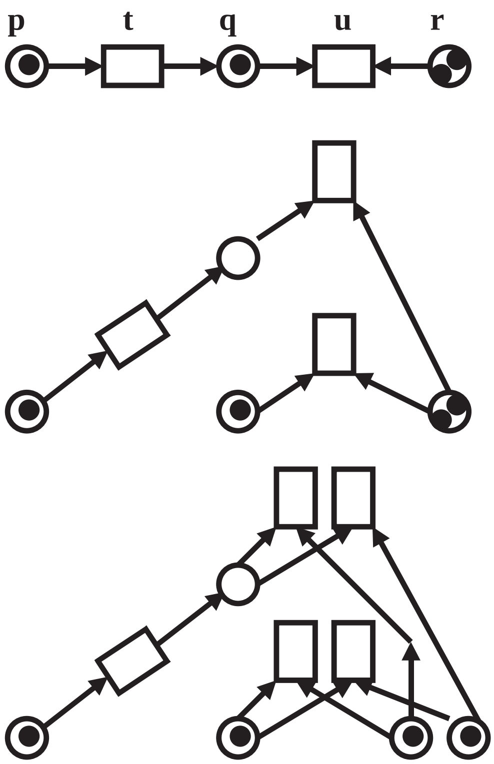


Fig. 29. A system (top), its WOCC-unfolding (middle) and its safe-unfolding (bottom).

This is reflected by the unfolding in Fig. 28: there is one transition for each of these two possibilities. Furthermore, this unfolding coincides with the unfolding into safe occurrence nets (as defined, e.g., in [15]). The two unfoldings coincide for safe and for semi-weighted nets (systems with the weight of any output-arc of any transition equal 1 and the initial marking a set [20]). But they differ on any system that is not semi-weighted.

This is shown in Fig. 29. It uses the same net as the previous example, but, there are two initial tokens on place r. The WOCC unfolding yields the same net as before, only the initial marking is adapted. However, the safe unfolding puts the additional initial token in an additional place and consequently needs to differentiate between further types of occurrences of transition u. This means that

- the unfolding into safe occurrence systems radically individualises tokens, they have a unique identity,
- the WOCC-unfolding differentiates only tokens that are located in different places or have a different causal history,
- in the collective token philosophy tokens on the same place are indistinguishable.

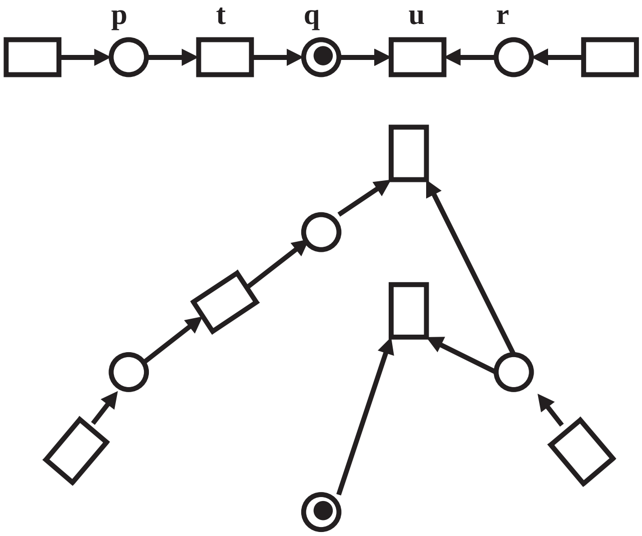


Fig. 30. A system (top) with variable initial marking on input-places p and r and its WOCC-unfolding.

The radical individualisation of tokens by safe unfoldings introduces inconveniences. First, the unfolding yields conflicts between tokens on the same place with the same history which seems strange in the case of anonymous tokens. Secondly, Section 7.4 will argue that there might be no 'reasonable' adjunction between systems and safe occurrence systems. Finally, safe unfoldings do not easily cope with variable initial markings. Such markings often occur when a net is used as a function transforming tokens from 'input-places' to tokens on'output-places' (e.g. [6,5]).

This is handled elegantly by our WOCC-unfolding. Fig. 30 shows how to convert p and r in our example net to input-places. Here, p and r may receive an arbitrary number of tokens from two initial transitions with empty presets. Such transitions are disallowed for safe unfoldings. But they are no problem for WOCC.

Safe occurrence systems realise a radical individual token philosophy. However, if the main interest is in causality and branching it is reasonable to look for the position between the two poles of individual and collective token philosophy that best serves this interest. It is exactly this intermediate position that weighted occurrence systems realise: they individualise tokens iff they have a different causal history, not generally as safe unfoldings do. In this sense, weighted occurrence systems represent the pure semantics of causality and branching for anonymous tokens.

7.3. Decorated occurrence systems

Definition 7.42. A place decoration Φ of a safe system S is a function $\Phi : P_S \to \mathbb{N}^+$ with

$$\forall t \in T_S: \ \Phi(t^\bullet) = [1, |t^\bullet|]$$

$$\forall p \in \operatorname{supp} I_S\colon\ \Phi(p) = 1,$$

Thus, a place decoration numbers the output places of each transition of a system.

Definition 7.43. $D = (\mathrm{pp}_D, I_D, \Phi_D)$ is a decorated occurrence system if (pp_D, I_D) is a safe occurrence system and Φ_D a decoration of the places of $S.\ P_D$. $\underline{\mathrm{DEC}}$ is the category of decorated occurrence systems. A morphism $f: D \to D'$ of $\underline{\mathrm{DEC}}$ is a morphism of the underlying safe occurrence systems which fulfils:

$$\forall p, q \in P_D \cap \operatorname{def} f_\beta \text{ with } {}^\bullet p = {}^\bullet q \neq \{\}\colon\ (\Phi_D\, p < \Phi_D\, q \text{ iff } \Phi'_D\, f\, p < \Phi'_D\, f\, q).$$

Definition 7.44. $\underline{\mathrm{FSYS1}}$ is the full subcategory of $\underline{\mathrm{FSYS1}}$ of systems with the initial marking being a set. Similarly, $\underline{\mathrm{WOCC1}}$ is the full subcategory of $\underline{\mathrm{WOCC}}$ of occurrence systems with the initial marking being a set.

Proposition 7.45. *The underlying functor* $U: \underline{\mathrm{DEC}} \to \underline{\mathrm{WOCC1}}$ *has a right adjoint* DEC $: \underline{\mathrm{WOCC1}} \to \underline{\mathrm{DEC}}$.

$$\begin{tikzcd} \underline{\mathbf{WOCC1}} \arrow[r, shift left, "\mathbf{DEC}"] \arrow[r, phantom, "\perp"] & \underline{\mathbf{DEC}} \arrow[l, shift left, "\mathbf{U}"] \end{tikzcd}$$

Proof. The proof is similar to that of Proposition 7.41. Let S be a weighted occurrence system of $\underline{\mathrm{WOCC1}}$. Again, DEC S is constructed as the colimit of the infinite diagram from Fig. 25. O_0 is simply IM S. Induction step from O_i to O_{i+1}. To the transitions of O_i we add all combinations of presets of multiples of transitions of depth $i+1$ and decorations of output-places:

$$T_{i+1} = T_i \cup \left\{ (m, \lambda t, \phi) \mid m \in 1\mathrm{S}\, P_i, \lambda, \phi_m \in \mathbb{N}^+, t \in T_S, \phi\colon [1, \phi_m] \to \mathbb{N}^+ t^\bullet \text{ with} \right.$$
$$t \text{ has depth } i+1,\ \lambda t \text{ is not dead in } S, \varepsilon_i\, m = \operatorname{pre} \lambda t \text{ and}$$
$$\left. \sum_{k=1}^{\phi_m} \phi(k) = \operatorname{post}(\lambda t) \right\}$$

$$P_{i+1} = P_i \cup \{((m, \lambda t, \phi), k) \in T_{i+1} \times \mathbb{N}^+ \mid k \leqslant \phi_m\}.$$

pp_i and ε_i are extended by

$$\mathrm{pp}_i(m, \lambda t, \phi) = \left(m, \sum_{((m,\lambda t,\phi),k) \in P_{i+1}} ((m, \lambda t, \phi), k) \right)$$

$$\varepsilon_{i+1}(m, \lambda t, \phi) = \lambda t$$

$$\varepsilon_{i+1}((m, \lambda t, \phi), k) = \phi(k)$$

and e_i is the obvious embedding of O_i in O_{i+1}. Then the colimit $o: D \to O_\omega$ exists. A decoration Φ on O_ω is defined by

$$\Phi\, o_0(p) = 1 \text{ and}$$

$$\Phi\, o_i((m, \lambda t, \phi), k) = k \quad \text{for } i > 0$$

which turns $\mathrm{DEC}\, S = (O_\omega, \Phi)$ into a decorated occurrence system.

To show the universality of $\varepsilon_\omega : O_\omega = U\, \mathrm{DEC}\, S \to S$ Fig. 27 is reused. Induction start: $\mathrm{IM}\, g$ yields g_0 because I_S is a set. Induction step from g_i to g_{i+1}. If such a g_{i+1} exists it must fulfil for any transition t' of O'_{i+1} of depth $i+1$

$$\mathrm{pre}\, g_{i+1}\, t' = g_{i+1}\, \mathrm{pre}\, t' = e_i\, g_i\, e_i'^{-1}\, \mathrm{pre}\, t',$$

$$\varepsilon_{i+1}\, g_{i+1}\, t' = f\, \varepsilon'_{i+1}\, t'.$$

Furthermore, g_{i+1} must correspond to a decorated morphism, especially it must be binary. This yields by the construction of O_{i+1}

$$g_{i+1}\, t' = (g_i\, e_i'^{-1}\, \mathrm{pre}\, t', f\, \varepsilon'_{i+1}\, t', \sigma) \text{ with}$$

$$\sigma : [1, |t^{\neq}|] \to \mathbb{N}^+\, T' \text{ with } t^{\neq} = \{p' \in t'^{\bullet} \mid f\, \varepsilon_{i+1}\, p' \neq \mathbf{0}\},$$

$$\sigma(k) = f\, \varepsilon_{i+1}\, p' \text{ for the } p' \in t^{\neq} \text{ with } k = |\{p'' \in t^{\neq} \mid \Phi'\, p'' \leqslant \Phi'\, p'\}|$$

which uniquely determines the image of t' and its post-places. Thus, g_{i+1} is unique.

The existence of g_{i+1} follows from the facts that $g_i\, e_i'^{-1}\, \mathrm{pre}\, t'$ is reachable in S because pre $\mathrm{pre}\, t'$ is reachable in O'_i and $\sigma([1, |t^{\neq}|]) = \mathrm{post}\, f\, \varepsilon'_{i+1}\, t'$. It is easy to see, that the colimit of the ε_i yields the claimed morphism g. On the other hand, such a g' yields a similar diagram with g'_i. But, we have shown that g_i equals g'_i which implies the equality of g and g'. □

7.4. Safe occurrence systems

Proposition 7.46. *The functor pair U,* DEC *restricts to an adjunction between* <u>DEC</u> *and* <u>OCC</u>.

$$\begin{tikzcd} \underline{\mathbf{DEC}} \arrow[r, shift left=2, "\mathbf{U}"] \arrow[r, phantom, "\vee"] & \underline{\mathbf{OCC}} \arrow[l, shift left=2, "\mathbf{DEC}"] \end{tikzcd}$$

Proof. The underlying occurrence system of any decorated occurrence system is safe. □

Thus, the composition $\mathrm{OCC} = U\, \mathrm{DEC}\, \mathrm{WOCC}$ is a functor from <u>FSYS1</u> to <u>OCC</u>. This is interesting, although we could not find an adjunction between the two categories. It is not easy to devise a functor from <u>FSYS</u> to <u>OCC</u>. It seems, that, in general, compositionality breaks down (permutations of places symbolising tokens on the same place). From the literature adjunctions are known for safe systems [26], semi-weighted systems ([19], here post t must be a set for any transition t) and nets with restricted morphisms ([14], here the images of the places in the postset of a transition must be disjoint).

Although our definition of a morphism is different, there are similarities:

- WOCC unfolds safe and semi-weighted systems to safe occurrence systems.
- OCC is a functor on general morphisms. The restriction concerns only the objects, namely, that the initial making must be a set. Although OCC is not part of an adjunction, there is a morphisms $\varepsilon_S : U\,\mathrm{OCC}\,S \to S$ which factorises any process of S (but not uniquely).

Our definition of decorated occurrence systems is a modification of that in [14]. We number all output-places of a transition whereas Meseguer et al. [14] enforces the order only on subsets corresponding to one output-place of a transition in the weighted net. This implies different restrictions and different functors. This method works also in our framework. But in [14] an adjunction is achieved what we could not do here, again, because we disallow to map places to markings.

In conclusion, we may compare three semantics for a system S, namely, $\mathrm{WOCC}\,S$, $\mathrm{OCC}\,S$ and $\mathrm{PROC}\,S = \{o : O \to S \mid o \text{ is a process of } S\}$.

Proposition 7.47. *The expressive power of* WOCC *is strictly stronger than that of* OCC *which again is strictly stronger than that of* PROC.

Proof. *Stronger* follows from $\mathrm{OCC} = U\,\mathrm{DEC}\;\mathrm{WOCC}$ and that every process of a system S factorises through $\varepsilon_S : \mathrm{OCC}\,S \to S$. The semantics differ on the examples of Fig. 24. Together this yields *strictly stronger*. □

Meseguer et al. [14] compared process semantics and safe unfoldings. The finding that "the unfolding contains several copies of the same process which, ..., are needed to provide a fully causal explanation of the behaviour" fully applies to the current setting. But it applies also to the $\underline{\mathrm{WOCC}}$ and $\underline{\mathrm{OCC}}$ unfolding and one would expect that $\underline{\mathrm{WOCC}}$ semantics lies between the process and $\underline{\mathrm{OCC}}$ semantics. The same expectation comes from the discussion that WOCC individualises tokens only partially. But the previous proposition falsifies this expectation! The explanation is that WOCC remembers more about the distribution of tokens on nodes of the net (and similar of occurrences to transitions)—a phenomenon countering the above argumentation.

8. Conclusion

This papers presents adjunctions between Petri net categories standing for the dichotomies of

- clustering and folding,
- structure and behaviour

and hence building a bridge from applications of Petri nets using clustering-based techniques for software engineering to folding-based categorical methods offering a rich set of theoretical relationships and insights. The hope is that this bridge allows the two paradigms to benefit more easily from each other. However many interesting questions remain open. We mention only a few:

- Research further relationships between morphisms and linear algebraic techniques, starting from our results about invariants.
- Use comma categories to build coloured, hierarchical, algebraic, etc. nets. This could be used to enhance an existing Petri net tools with categorical machinery, e.g. universal construction or an integration of graph transformation systems.
- Explore the categorical relationships between this work and other models of Petri nets. Search for links to other semantics.
- The second part of the author's Ph.D. thesis—not reflected in this paper—shows how to use universal construction in reverse engineering. But there must be more such novel applications.

Summarising, this paper is a beginning—there are many tasks for further research.

Acknowledgements

We wish to thank the anonymous referees for numerous important suggestions.

References

[1] E. Best, R. Devillers, M. Koutny, The box algebra—a model of nets and process expressions, in: S. Donatelli, J. Kleijn (Eds.), Application and Theory of Petri Nets, Lecture Notes in Computer Science, Vol. 1639, Springer, Berlin, 1999, pp. 344–363.

[2] D. Christodoulakis, M. Moritz, Net Morphisms and software engineering, in: C. Girault, W. Reisig (Eds.), Selected Papers from the 1 and 2 European Workshop on Application and Theory of Petri Nets, 1982, IF 52, Springer, Berlin, 1980, pp. 111–117.

[3] J. Engelfriet, Branching processes of Petri nets, Acta Informatica 28 (1991) 575–591.

[4] C. Ermel, M. Gajewsky, Expanding the use of structuring: formal justification for working on subnets, in: Petri Nets in System Engineering (PNSE'97), University of Hamburg, Informatik, Bericht No. 205, 1997.

[5] R. Esser, An object oriented petri net approach to embedded system design, vdf, Hochschulverlag an der Eth Zürich, 1997.

[6] R. Fehling, A concept of hierarchical petri nets with building blocks, in: G. Rozenberg (Ed.), Advances of Petri Nets, Lecture Notes in Computer Science, Vol. 674, Springer, Berlin, 1993, pp. 148–168.

[7] H.J. Genrich, K. Lautenbach, P.S. Thiagarajan, Elements of general net theory, in: W. Brauer (Ed.), Net Theory and Applications, Lecture Notes in Computer Science, Vol. 84, Springer, Berlin, 1979, pp. 21–163.

[8] R.J. van Glabbeek, G.D. Plotkin, Configuration structures, IEEE Symp. on Logic in Computer Science, 1995, pp. 199–209.

[9] K. Jensen, Coloured Petri Nets: Basic Concepts. Analysis Methods and Practical Use, Vol. 1, Springer, Berlin, 1992.

[10] W. Keller, Petri nets for reverse engineering, Ph.D. Thesis, University of Zurich. Available at www.ifi.unizh.ch/Forschung/dissertationen.html or home.tiscalinet.ch/wkeller

[11] W. Keller, Reverse engineering with Petri nets, in: 4th European Conf. on Software Maintenance and Reengineering, IEEE, 2000, pp. 235–238.

[12] C. Lakos, On the abstraction of coloured Petri nets, in: P. Azema, G. Balbo (Eds.), Application and Theory of Petri Nets, Lecture Notes in Computer Science, Vol. 1248, Springer, Berlin, 1997, pp. 42–61.

[13] S. Mac Lane, Categories for the Working Mathematician, Springer, Berlin, 1971.
[14] J. Meseguer, U. Montanari, V. Sassone, On the semantics of Petri Nets, in: W.R. Cleaveland (Ed.), Concur'92. Internat. Conf. on Concurrence Theory, 1992, pp. 286–301.
[15] J. Meseguer, U. Montanari, V. Sassone, On the model of computation of place/transition petri nets, in: R. Valette (Ed.), Application and Theory of Petri Nets, Lecture Notes in Computer Science, Springer, Berlin, 1994, pp. 16–38.
[16] M. Mukund, Petri nets and step transition systems, Internat. J. Foundations Comput. Sci. 3 (4) (1992) 443–478.
[17] M. Nielsen, A. Cheng, Observing behaviour categorically, in: P.S. Thiagarajan (Ed.), Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science, Vol. 1026, Springer, Berlin, 1995, pp. 263–278.
[18] M. Nielsen, G. Rozenberg, P.S. Thiagarajan, Behavioural notions for elementary net systems, in: Distributed Computing, Vol. 4, Springer, Berlin, 1990, pp. 45–57.
[19] M. Nielsen, V. Sassone, Petri nets and other models of concurrency, in: W. Reisig, G. Rozenberg (Eds.), Lecture on Petri Nets I: Basic models, Lecture Notes in Computer Science, Vol. 1491, Springer, Berlin, 1998, pp. 587–642.
[20] J. Padberg, M. Gajewsky, C. Ermel, Rule-based refinement of high-level nets preserving safety properties, in: E. Asteiano (Ed.), Fundamental approaches to software engineering, First International Conference, FASE '98, Springer, Berlin, 1998, pp. 221–238.
[21] G. Pomberger, B. Blaschek, Software Engineering, Carl Hanser Verlag, München, 1993.
[22] W. Reisig, Petri Nets, An Introduction, Springer, Berlin, 1985.
[23] W. Reisig, G. Rozenberg (Eds.). Lectures on Petri Nets I: Basic Models, Lecture Notes in Computer Science, Vol. 1491, 1998.
[24] G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing Graph Transformation, World Scientific, Singapore, 1997.
[25] V. Sassone, M. Nielsen, G. Winskel, Models for concurrency: towards a classification, Theoret. Comput. Sci. 170 (1996) 297–348.
[26] G. Winskel, Categories of models for concurrency, in: S.D. Brookes, A.W. Roscoe, G. Winskel (Eds.), Seminar on Concurrency, Pittsburgh, Lecture Notes in Computer Science, Vol. 197, Springer, Berlin, 1984, pp. 246–267.
[27] G. Winskel, Event structures, in: G. Goos, J. Hartmanis (Eds.), Petri Nets: Applications and Relationships to Other Models of Concurrency, Lecture Notes in Computer Science, Vol. 225, Springer, Berlin, 1986, pp. 325–392.