The Universal Net Composition Operator

W. Reisig

Humboldt-Universität zu Berlin

Abstract. Petri nets are frequently composed of given nets. Literature suggests a lot of different composition operators, for different purposes and different classes of Petri nets.
Formal definitions are frequently surprisingly technical, not matching the intuitively very elegant composition of Petri nets in the framework of their graphical representation.
This paper suggests the *universal net composition operator.* This operator allows to specify any specific composition variant by very simple means, leaving all technical details to the operator, where they are treated once and for all. General properties of composition, in particular associativity, are inherited by all instantiations of the operator. We show the practical advantage of the universal composition operator by means of a lot of examples from various areas of Petri nets.

1 Introduction

Petri nets are frequently composed from given nets. Literature suggests a lot of different composition operators, for different purposes and different classes of nets. Definitions of composition operators are frequently surprisingly technical, not matching the operators' intuitive and graphical elegance.

We want to overcome this problem, once and for all, by the general definition of a reasonable, universal notion of *composition* of Petri nets. In detail:

A composition operator for nets is "reasonable" if it is sufficiently expressive, covering, e.g. all examples of forthcoming Sec. 2;

A composition operator is "universal" if it can be applied to any two nets N_1 and N_2, returning a net $N_1 \cdot N_2$;

"proper" composition means that the composition operator is associative, i.e. for all nets N_1, N_2, N_3:

$$(N_1 \cdot N_2) \cdot N_3 = N_1 \cdot (N_2 \cdot N_3).$$

This means intuitively that a composed net should not carry around the "history" of its composition process. In the rest of this paper we derive such a notion.

The rest of this paper is organized as follows: Chapter 2 presents a choice of very different examples where classes of nets are to be composed in very different manners. These examples set standards for the generality and variability of the operator. The operator itself will then be defined in Sec. 3. In Sec. 4 we return to the examples of Sec. 2 and show that they can very simply be conceived as instances of the universal composition operator.

2 A Choice of Examples

Literature suggests a lot of different composition operators, for different purposes and different classes of nets. We select different areas where Petri nets are composed: Concurrent runs, service nets, branching processes and nets with high-level places or transitions in their interface. Additionally we consider *non-deterministic* composition of nets.

2.1 Composition of Concurrent Runs

A concurrent run of an elementary system net N is itself a net, labelled by the elements of N. For the sake of simplicity we stick to *finite* concurrent runs, and a concurrent run of N may start at any reachable marking of N. Fig. 2.1 shows an example of a 1-bounded elementary system net N. Typical distributed runs of N are shown in Fig. 2.2.

We are interested in the *composition* of such nets. Fig. 2.3 shows two examples of runs, composed from the runs of Fig. 2.2. Intuitively and graphically this kind of composition is fairly simple. The formal counterpart of a composition operator that would allow to write the runs of Fig. 2.3 as

$$r_1 \cdot r_2 \cdot r_1 \text{ and } r_1 \cdot r_3 \cdot r_2 \tag{2.1}$$

should likewise be simple.

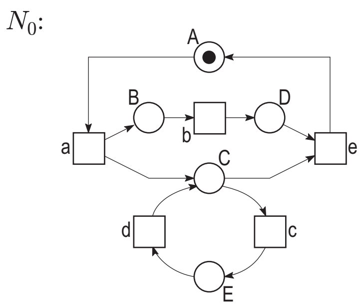


Fig. 2.1. an elementary system net, N

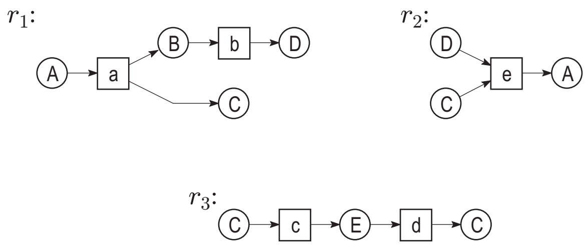


Fig. 2.2. three concurrent runs of the net N of Fig. 2.1

2.2 Composition of Service Nets

As an entirely different kind of example, Fig. 2.4 shows a *service net*, buyer. This net shows how a buyer component orders goods, expects to receive an invoice, will pay the invoice, and – concurrently to those actions – receive the goods. The four places order, invoice, payment and goods build the component's *environment*.

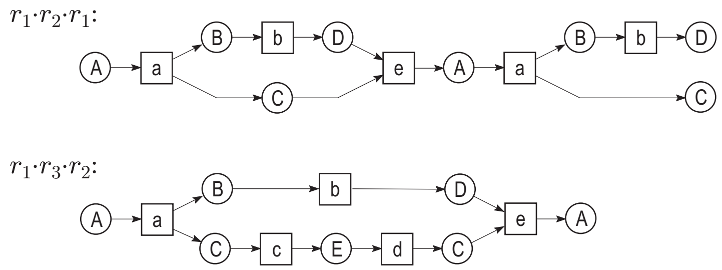


Fig. 2.3. two concurrent runs, composed from the runs of Fig. 2.2

We may assume a retailer company, running a sales department (seller) to accept the buyer's orders, to send an invoice back to the buyer, to receive the payment, and to advice the companies' warehouse to deliver the ordered goods. Fig. 2.5 shows this behaviour. The job of the warehouse is to just deliver goods, upon the sales department's request, as shown in Fig. 2.6.

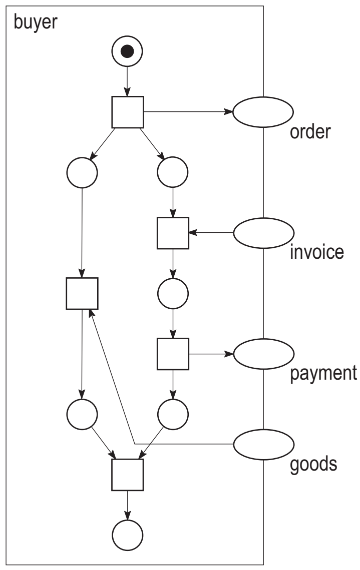


Fig. 2.4. Service net of the buyer

An *interface net* such as in Fig. 2.4, Fig. 2.5 or Fig. 2.6 is a marked net with a distinguished set of *interface places*. Graphically, an interface net is drawn inside a box, with the interface place on its surface.

Composition $N_1 \cdot N_2$ of two interface nets N_1 and N_2 "glues" N_1 and N_2 along their interface places: Some places are turned into internal places, the rest constitutes the interface of $N_1 \cdot N_2$. The internal structure of $N_1 \cdot N_2$ is just the union of the internal structure of N_1 and N_2, and the newly gained internal places.

As an example, Fig. 2.6 describes the retailer company as the composition of the seller and the warehouse. At the end of the day we will write

$$\mathsf{retailer} =_{def} \mathsf{seller} \cdot \mathsf{warehouse}. \tag{2.2}$$

The place order that links the seller and the warehouse is turned into an internal place. However, the goods place of the warehouse is an interface place of the retailer, as it is required to communicate with the buyer.

Fig. 2.7 shows the resulting interfaces of some more compositions. As a – somewhat unusual – composition, buyer·seller "glues" the three places at the left side of seller with the corresponding places of buyer and turns them into

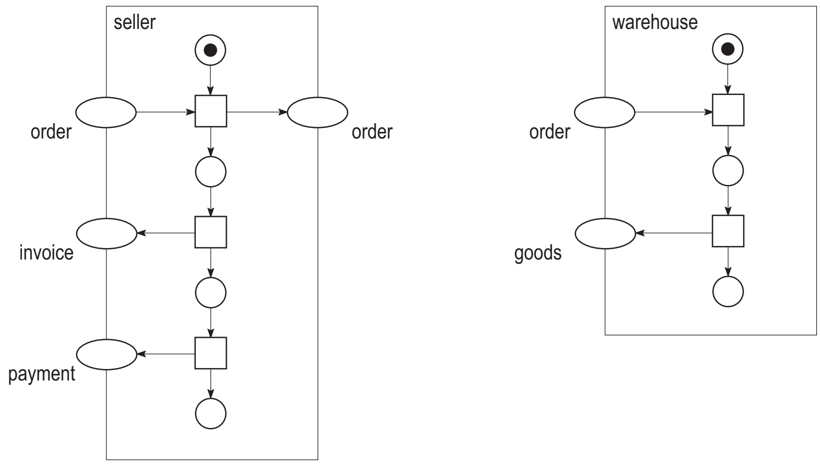


Fig. 2.5. Service nets of the seller and the warehouse

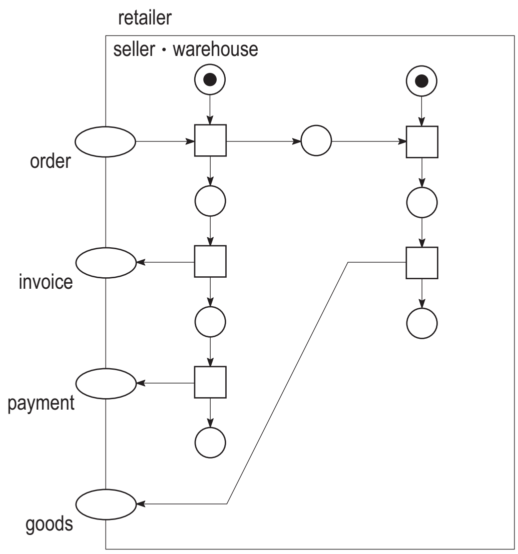


Fig. 2.6. service net seller · warehouse

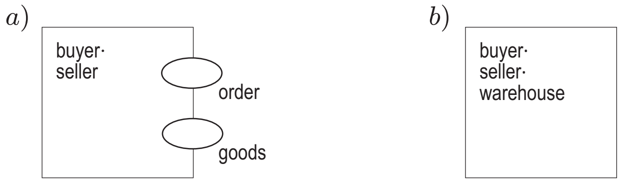


Fig. 2.7. Some more compositions of components

inner places. There remain two places, the seller's order on its right side, and the buyer's goods. This net provides a perfect environment for the warehouse. Finally, composing all four components, the environment remains empty.

As a variant, the retailer company may run a *second* warehouse, warehouse', that alternatively to the given one may deliver goods. The two warehouses share their environment places order and goods, as indicated in Fig. 2.8. Consequently, the interface of the composition

warehouse' · warehouse

should be identical with the interface of warehouse alone.

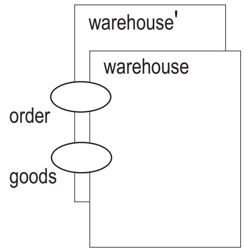


Fig. 2.8. alternative warehouses

2.3 Composition of Branching Processes

With the alternative warehouse in Fig. 2.8 we have seen already a case where two nets N_1 and N_2 are composed such that in $N_1 \cdot N_2$, some inner elements of N_1 and of N_2 access the same interface places. This kind of composition is also required whenever *branching processes* are to be composed.

For example, Fig. 2.9 shows an initial part of the branching process of the system net N of Fig. 2.1. The problem here is to identify a small number of concurrent runs and a composition operator such that Fig. 2.9 can be written as a composition of concurrent runs.

2.4 Composition at Mixed Interfaces

Here we consider a business organization where each employee by mail may demand to see the director. The director's staff either denies or grants the demand. If granted, the director prepares the meeting and meets with the employee by handshake communication.

Fig. 2.10 models this system by means of two components, workforce and management, together with their joint interface. The interface consists of high-level places as well as transitions. An adequate composition operator also covers

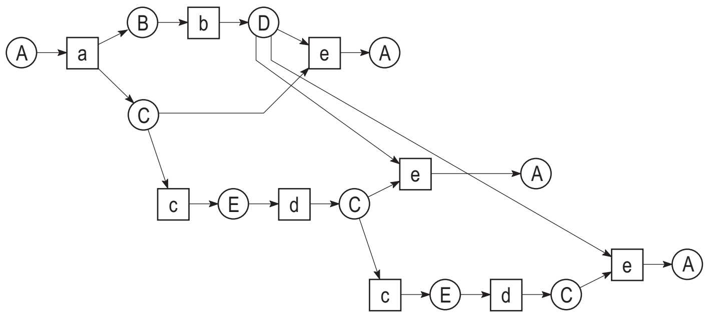


Fig. 2.9. Prefix of the branching process of the net N as in Fig. 2.1

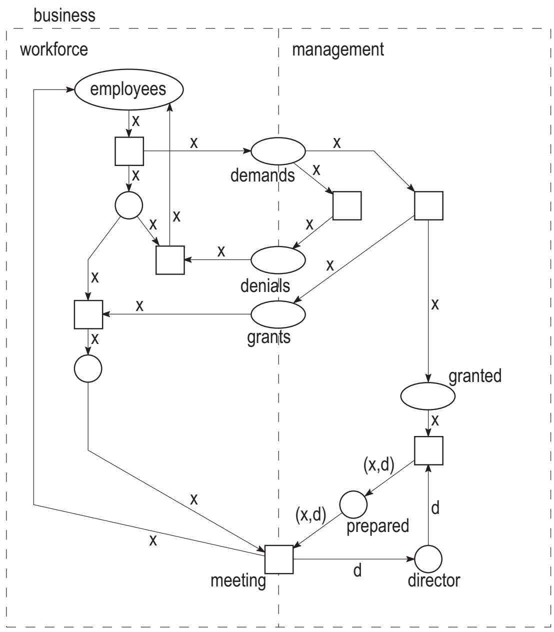


Fig. 2.10. employee meets director

this kind of an interface. As a variant, the version of Fig. 2.11 composes the management from a director- and a staff component.

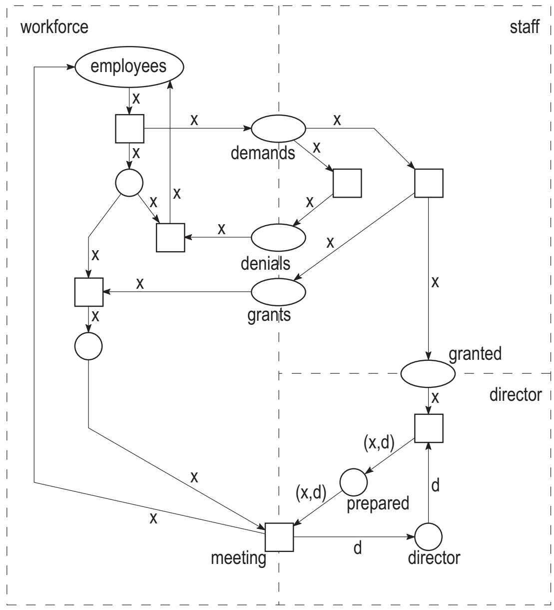


Fig. 2.11. management, composed of staff and director

Interfaces including high-level places and -transitions should be manageable by a proper composition operator as well.

2.5 Nondeterministic Composition of Nets

In the examples considered so far, places of interfaces of different nets may be equally labelled. But different places of the interface of *one* net were never equally labelled. This is however not what we require. Typical examples are concurrent runs of marked nets that are not 1-bounded. Fig. 2.12 shows an example of a system net N with initially *two* tokens at place at place A. Fig. 2.13 shows two concurrent runs r_1 and r_2 of N. The environment of r_1 contains two equally laballed places. This gives rise to *two* different composed runs, r_3 and r_4, as shown in Fig. 2.14.

This is an example of *non-deterministic* composition. This case should be covered by a universal net composition operator, too.

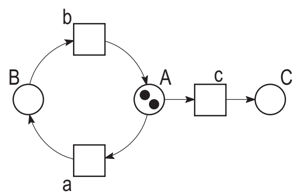


Fig. 2.12. 2-bounded system net N

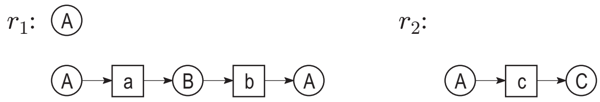


Fig. 2.13. Two concurrent runs of N as in Fig. 2.12

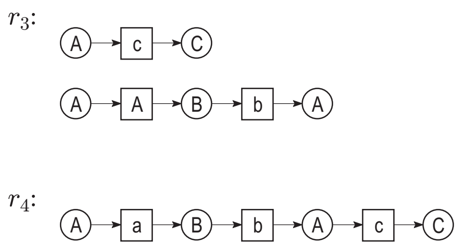


Fig. 2.14. composition $r_1 \cdot r_2$, yielding two concurrent runs, r_3 and r_4

2.6 Resumé

Each of the above examples stands for a *class* of nets. It should be intuitively obvious how nets of each such class are to be composed. Together those classes cover a wide range of nets and corresponding composition operators found in literature. Those composition operators turn out as special cases of the operator discussed in this paper. This justifies its denotation as *universal*.

The next Section provides the formal background of the universal composition operator for nets. In Section 4 we return to the above examples, showing that they indeed are just instances of the universal operator.

3 The universal composition operator

3.1 The idea of ports

The above examples show that composition and decomposition of nets in many cases is intuitively fairly simple. The graphical representation of nets supports this claim decisively.

A formal definition of a composition operator should cover all aspects and variants as they occurred above, and coincidently remain as simple as intuition suggests.

Here we take a closer look at what the above examples do have in common. We furthermore outline the notion of *left* and *right port* of a net. These ports govern the composition of nets.

First of all we notice that the composition $N_1 \cdot N_2$ of two nets N_1 and N_2 always is the *union* of N_1 and N_2, with some pairs of equally labelled elements of N_1 and N_2 replaced by *one* element in $N_1 \cdot N_2$. However, not *all* equally labelled elements of N_1 and N_2 turn into one place in $N_1 \cdot N_2$, nor is each place of N_1 and N_2 necessarily labelled at all. To cope with this observation, a net N that is intended to be composed with other nets has two kinds of elements: Elements of the *interface* of N are those directly affected by the composition (i.e. those where upon composition, a new arrow will start or end). All other elements are *inner* elements of N.

For example, in r_1 of Fig. 2.2, composition affects the places labelled A, C and D. They constitute the interface of r_1. The place B and both transitions a and b are the inner elements.

The decisive concept is the notion of *port*: The interface of a net N is the union of *two* subsets L_N and R_N of elements of N, denoted as the *left* and *right port* of N. The two ports can reflect various different aspects of real systems. Typical examples include

- front end and back end,
- input and output,
- standard case and exception,
- buy side and sell side,
- customers and suppliers.

Composition of nets along their ports motivate the denotation of "left" and "right" port: In the composed net $N_1 \cdot N_2$, elements of the right port of N_1 and elements of the left port of N_2 are "glued" and turned into inner elements of $N_1 \cdot N_2$.

In concurrent runs such as r_1 in Fig. 2.2, the notions of "left port" and "right port" are particularly intuitive: The left port of r_1 contains the A-labelled place. Its right port consists of the two places labelled D and C. Consequently, for r_3, the left port as well as the right port consists of one place. Both are labelled by C.

Generally formulated, the ports of a concurrent run such as r_1, r_2 or r_3 in Fig. 2.2 are canonically defined: The left port comprizes all places with empty preset. The right port comprizes all places with empty postset. This principle is likewise applied in case of nondeterministic composition of nets, as discussed in Sec. 2.5.

Service nets, as considered in Sec. 2.2, have their interface places on their surrounding box. The places of its left and right port are placed on the left and right edge of the box, respectively. This likewise applies to the mixed interface elements of Sec. 2.4.

Other classes of nets do not exhibit canonical ports. The designer of an interface net has the freedom to determine them according to his or her needs and interests. In particular, the two ports of an interface net are not necessarily assumed to be disjoint. This is in particular exploited when branching processes are composed as in Sec. 2.3. Taken to extreme, if a net is assumed to have just a unique interface I instead of two ports, I is conceived as the left as well as the right port.

3.2 Composing Nets along their Ports

As a general setting, supported by all examples of Chapter 2, a port of a net N is a subset of *labelled* elements of N. With L_i and R_i denoting the left and right port of two nets N_i ($i = 1, 2$), the following rule of thumb yields the composition $N_1 \cdot N_2$ of N_1 with N_2:

- Identify equally labelled elements in R_1 and L_2, glue them, and make them inner elements of $N_1 \cdot N_2$.
- The remaining elements of L_2 and R_1 go to the left and right port L_{12} and R_{12} of $N_1 \cdot N_2$, respectively.
- L_1 becomes a subset of L_{12} and R_2 a subset of R_{12}.

Fig. 3.1 outlines this construct. The symbol — $\sim$ — links equally labelled elements of R_1 and L_2. The symbol — = — depicts their identification.

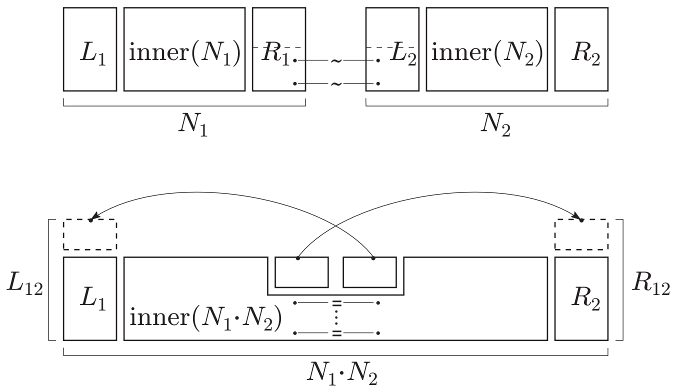


Fig. 3.1. A first idea to compose nets

A closer look reveals however that composition can not be constructed as simple as this. In particular, we have to cope with the case where a port has two (or more) identically labelled elements.

One might be tempted to simply exclude such ports. But this would exclude important classes of nets too, e.g. those considered in Sec. 2.5. Even worse, there is no reasonable, universal and proper composition operator without such ports: They may arise as the result of composing two nets.

We construct a composition operator for nets where a port may very well have identically labelled elements. We do so with the mild assumption that identically labelled elements of a port are *ordered*.

Nevertheless we may start more generally with a net where one (or both) of its ports has has identically labelled but unordered elements. Obviously, they can be ordered in different ways. With each such order we can proceed as described above. All together, they provide the means to express *nondeterministic* composition, as discussed in Sec. 2.5.

Ordering of identically labelled elements of a port yields a lot of important properties and consequences, to be studied next.

3.3 Index labelled Sets

As discussed in Sec. 3.2, different elements of a port may be labelled identically. Identically labelled elements are assumed to be *ordered*. This yields a special kind of order, as studied in the sequel. First of all, the order is *strict*:

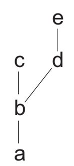
Definition 1. *Let A be a set. A relation $< \subseteq A \times A$ is a* strict partial order *on A iff*

- *for no $a \in A$ holds $a < a$, and*
- *for all $a, b, c \in A$, if $a < b$ and $b < c$ then $a < c$.*

Formulated differently, a binary relation is a strict partial order iff it is irreflexive and transitive.

We alway assume sets A that are at most countable. Then each order $<$ on A defines for each $a \in A$ unique *successors* i.e. elements b with $a < b$ and for no $c \in A$, $a < c < b$. A strict order on a finite set can then be depicted by its *Hasse diagram*.

For example, with $A = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$, the order $<$ with $\mathsf{a} < \mathsf{b}$, $\mathsf{a} < \mathsf{c}$, $\mathsf{a} < \mathsf{d}$, $\mathsf{a} < \mathsf{e}$, $\mathsf{b} < \mathsf{c}$, $\mathsf{b} < \mathsf{d}$, $\mathsf{b} < \mathsf{e}$, $\mathsf{d} < \mathsf{e}$ can be depicted as



(1)

Smaller elements are depicted below larger elements. Each element is linked by an upwards line with its successors.

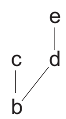
Definition 2. *Let A be a set with a strict partial order $<$, and let $a \in A$. The* index *of a in A, written $index(A, a)$, is the number of elements in A that are smaller than or equal to A.*

For example, the order of (1) yields $index(A, a) = 1$, $index(A, b) = 2$, $index(A, c) = index(A, d) = 3$ and $index(A, e) = 4$. This notation is unique as long as we consider only *one* order on any given set. We will always observe this requirement.

Observations

- In general, the indices of two elements of a strict partial order may very well coincide.
- An element involved in two different strictly partially ordered sets may exhibit a different index in each set.

For example, with the set A ordered as in (1) and the set $B = \{\mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$, the order



(2)

yields for each $x \in B$: $order_B(x) = order_A(x) - 1$.

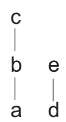
We now consider particular orders on sets with *labelled* elements: An *index order* strictly orders the identically labelled elements:

Definition 3. *Let A and L be sets, let $\lambda : A \to L$ be a mapping and let $<$ be a strict partial order on A, where for all $a, b \in A$:*

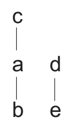
$$a < b \text{ or } b < a \quad \text{if and only if} \quad \lambda(a) = \lambda(b).$$

Then $<$ is an index-order on A and λ*, and A is said to be* index labelled *by λ and $<$.*

As an example, with $A = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$, $\lambda(\mathsf{a}) = \lambda(\mathsf{b}) = \lambda(\mathsf{c}) = l_1$ and $\lambda(\mathsf{d}) = \lambda(\mathsf{e}) = l_2$, typical index orders on A and λ are



and



(3)

The order in (1) is no index order on A and λ. Even more, there exists no labelling λ at all, that would make (1) an index order. Intuitively formulated, the Hasse diagram of an index order is a set of "columns", one for each label.

To fully understand index labelled sets, consider the extreme cases: if λ is injective (i.e. different elements carry different labels), the only index order on A w.r.t. λ is the empty order, i.e. no two elements are ordered, and the index of each element is "1". In the other extreme case, all elements of A carry the same label. Then a strict order on A is an index order iff it is total. If A has n elements, they are indexed $1, \ldots, n$.

Notations

For the rest of this paper we assume a "universal" set L of labels. For each index labelled set A we denote the labelling and the order on A by λ_A and $<_A$, respectively. We furthermore assume the index of each element of A to be finite.

The terms *index order* and *index labelled set* are motivated by the observation that each element a of A is uniquely determined by its label and its index:

Lemma 1. *Let A be a set, index labelled by λ and $<$. Let $a, b \in A$. If $\lambda(a) = \lambda(b)$ and $index(A, a) = index(A, b)$ then a=b.*

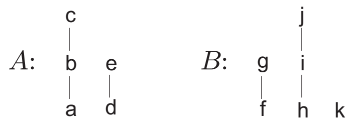
For example, the labelling λ and the left order of (3) determines b by the label l_1 and the order 2. In the right order of (3), the order of b is 1.

This property of index orders gives rise to a canonical *equivalence* on the elements of different index labelled sets:

Definition 4. *Let A and B be sets, index labelled by λ_A, $<_A$ and λ_B, $<_B$. Two elements $a \in A$ and $b \in B$ are* AB-equivalent *written $a \ {}_A{\sim}_B \ b$ iff*

- $\lambda_A(a) = \lambda_B(b)$, *and*
- $index(A, a) = index(B, b)$.

As an example, let $A = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$ and $B = \{\mathsf{f}, \mathsf{g}, \mathsf{h}, \mathsf{i}, \mathsf{j}, \mathsf{k}\}$ be sets, labelled by $\lambda_A(\mathsf{a}) = \lambda_A(\mathsf{b}) = \lambda_A(\mathsf{c}) = \lambda_B(\mathsf{f}) = \lambda_B(\mathsf{g}) = l_1$ and $\lambda_A(\mathsf{d}) = \lambda_A(\mathsf{e}) = \lambda_B(\mathsf{h}) = \lambda_B(\mathsf{i}) = \lambda_B(\mathsf{j}) = l_2$ and $\lambda_B(\mathsf{k}) = l_2$, index ordered as follows:



(4)

(A is as in the left order of (3)). Then $\mathsf{a} \ {}_A{\sim}_B \ \mathsf{f}$, $\mathsf{b} \ {}_A{\sim}_B \ \mathsf{g}$, $\mathsf{d} \ {}_A{\sim}_B \ \mathsf{h}$ and $\mathsf{e} \ {}_A{\sim}_B \ \mathsf{i}$. No other pair of elements is AB-equivalent.

This equivalence gives rise to *isomorphic* index labelled sets:

Definition 5. *Two index labelled sets A and B are* isomorphic *(written $A \approx B$) iff there exists a bijection $f : A \to B$ such that for all $a \in A$,*

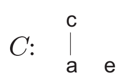
$$a \ {}_A{\sim}_B \ f(a).$$

The index order of an index labelled set A canonically translates to its subsets:

Definition 6. *Let A be an index labelled set, let $B \subseteq A$. The* canonical index labelling of B *is defined for all $a, b \in B$ by*

- $\lambda_B(a) = \lambda_A(b)$
- $a <_B b$ *iff* $a <_A b$.

As an example, for A as in (4), $C = \{\mathsf{a}, \mathsf{c}, \mathsf{e}\} \subseteq A$ yields the index order



(5)

i.e. $l_C(\mathsf{a}) = l_C(\mathsf{c}) = l_A(\mathsf{a}) = l_A(\mathsf{c}) = l_1$, $l_C(\mathsf{e}) = l_A(\mathsf{e}) = l_2$, $index_C(\mathsf{a}) = 1 = index_A(\mathsf{a})$, $index_C(\mathsf{c}) = 2 < 3 = index_A(\mathsf{c})$ and $index_C(\mathsf{e}) = 1 < 2 = index_A(\mathsf{e})$.

Hence for $\mathsf{b} \in B$, the label remains (i.e. $\lambda_B(\mathsf{b}) = \lambda_A(\mathsf{b})$), but the index may decrease (i.e. $\lambda_A(\mathsf{a}) \geq \lambda_B(\mathsf{a})$).

A subset B of an index labelled set A may preserve the indices. Then B is a prefix of A.

Definition 7. *Let A be an index labelled set, let $B \subseteq A$ such that for all $b \in B$, $index_B(b) = index_A(b)$. Then B is a* prefix *of A, written $B \sqsubseteq A$.*

For example, with A as in (4), $\{\mathsf{a}, \mathsf{b}, \mathsf{d}\}$ and $\{\mathsf{a}, \mathsf{d}, \mathsf{e}\}$ are prefixes of A. In contrast, C as in (5) is no prefix of A.

A prefix can be conceived as a subset that respects the index order. Similarly we define index order respecting *intersection*, *complement* and *union*:

Definition 8. *Let A and B be index labelled sets. Let $A' \sqsubseteq A$ and $B' \sqsubseteq B$ be the largest isomorphic prefixes of A and B. Then A' is the* intersection of A with B, *written $A \sqcap B$.*

For example, with A and B as in (4), $A' = \{\mathsf{a}, \mathsf{b}, \mathsf{d}, \mathsf{e}\}$ and $A' = \{\mathsf{f}, \mathsf{g}, \mathsf{h}, \mathsf{i}\}$ are the largest isomorphic prefixes of A and B. Consequently,

$$A \sqcap B = \begin{array}{cc} \mathsf{b} & \mathsf{e} \\ | & | \\ \mathsf{a} & \mathsf{d} \end{array} \qquad B \sqcap A = \begin{array}{cc} \mathsf{g} & \mathsf{i} \\ | & | \\ \mathsf{f} & \mathsf{h} \end{array} \tag{6}$$

For A, A', B, B' as in the above definition, sheer symmetry implies that $B' = B \sqcap A$ is the intersection of B with A.

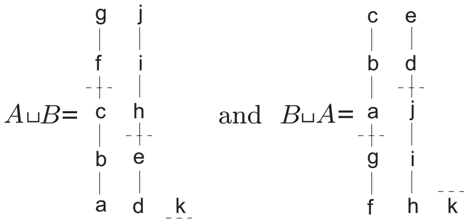
Definition 9. *Let A and B be index labelled sets. Then $A - B$ is the index labelled set generated by $A \backslash (A \sqcap B)$.*

Again with A and B as in (4) (and $A \sqcap B$, $B \sqcap A$ as in (6)), $A - B = \{\mathsf{c}\}$ and $B - A = \{\mathsf{j}, \mathsf{k}\}$. With B as in (4) and C as in (5), $B - C = \{\mathsf{i}, \mathsf{j}, \mathsf{k}\}$ and $C - A = \emptyset$. Finally, the union of index labelled sets:

Definition 10. *Let A and B be disjoint index labelled sets. Then $A \sqcup B$ (the* union *of A and B) is the set $A \cup B$, index labelled as follows:*

- *For each $x \in A$ and $y \in B$, $\lambda_{A \sqcup B}(x) = \lambda_A(x)$ and $\lambda_{A \sqcup B}(y) = \lambda_B(y)$.*
- *For all $x, y \in A \cup B$, $x <_{A \sqcup B} y$ iff $x <_A y$ or $x <_B y$ or ($\lambda(x) = \lambda(y)$, $x \in A$ and $y \in B$).*

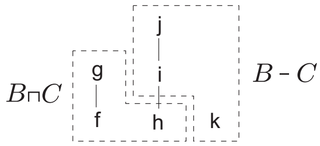
Intuitively formulated, each column of the Hasse diagram of $A \sqcup B$ consists of the corresponding columns of A on its bottom and of B on its top, respectively. For example, with A and B as in (4),



The above operators on index labelled sets behave in analogy to the corresponding operators on ordinary sets.

Lemma 2. *Let A and B be index labelled sets. Then $A = (A \sqcap B) \sqcup (A - B)$.*

For example, with B as in (4) and C as in (5), $B = (B \sqcap C) \sqcup (B - C)$:



This completes the required constructs on index labelled sets.

3.4 Interface Nets

We are now ready to define interface nets and their composition, starting with the usual basic notion of nets:

Definition 11. *Let P and T be disjoint sets, and let $F \subseteq (P \times T) \cup (T \times P)$. Then $N = (P, T, F)$ is a* net. *Elements in P, T and F are denoted as* places, transitions *and* arcs, *respectively. $P \cup T$ is the* carrier *of N. It contains the* elements *of N.*

Graphically we depict places, transitions and arcs as usual by circles, squares and arrows, respectively.

We frequently consider nets with labelled elements. The label of an element e is placed *inside* its graphical symbol, with its identity e on its side. For example,

denotes the A-labelled place p.

Union of two nets N_1 and N_2 is defined component wise. It will usually be applied to nets that share places or transitions. The case of a place of N_1 being identical with a transition of N_2 never occurs.

Definition 12. *For $i = 1, 2$ let $N_i = (P_i, T_i, F_i)$ be nets. Then the net $N_1 \cup N_2 =_{def} (P_1 \cup P_2, T_1 \cup T_2, F_1 \cup F_2)$ is the* union *of N_1 and N_2.*

We consider nets with index labelled subsets, and replace elements by equivalent ones:

Definition 13. *Let $N = (P, T, F)$ be a net. Let $C =_{def} P \cup T$ and let Q, R be index labelled sets with $Q \subseteq C$ and $Q \approx R$. Let $C' =_{def} (C \backslash Q) \cup R$.*
For each $(x, y) \in F$ let
$(x, y) \in F'$ *iff* $x, y \in C$,
$(x', y) \in F'$ *iff* $x \; {}_Q{\sim}_R \; x'$,
$(x, y') \in F'$ *iff* $y \; {}_Q{\sim}_R \; y'$.
Then $N' =_{def} (P \cap C', T \cap C', F')$ is the substitution of Q by R in N, *written* $N[Q \to R]$.

An *interface net* N is now defined as a net together with two index labelled sets, serving as the ports of N:

Definition 14. *Let N be a net with carrier C and let $L, R \subseteq C$ be index labelled sets. Then N together with L and R is an* interface net. *The sets L and R are the* left *and* right port *of N, respectively. The set inner$(N) =_{def} C\backslash(L \cup R)$ contains the* inner *elements of N.*

To compose two nets N_1 and N_2, we assume them to be disjoint (otherwise, copy shared elements). Elements of the left port L_2 of N_2 are replaced by equivalent of the right port R_1 of N_1 and turned into inner elements of $N_1 \cdot N_2$. In technical terms, each element of $L_2 \sqcap R_1$ is replaced by the corresponding equivalent element of $R_1 \sqcap L_2$ and turned into an inner element of $N_1 \cdot N_2$. The remaining elements of L_2 and R_1 go to the left and the right port, respectively, of $N_1 \cdot N_2$. Fig. 3.1 outlined this construct already.

Definition 15. *For $i = 1, 2$ let N_i be disjoint interface nets with left and right ports L_i and R_i. Let $N_2' =_{def} N_2[L_2 \sqcap R_1 \to R_1 \sqcap L_2]$. Then the* composition *$N_1 \cdot N_2$ of N_1 with N_2 is the interface net $N_1 \sqcup N_2'$ with*
the left port $L_{12} =_{def} L_1 \sqcup (L_2 - R_1)$ and
the right port $R_{12} =_{def} R_2 \sqcup (R_1 - L_2)$.

As a technical example, Fig. 3.2 shows two nets N_1 and N_2, the variant N_2' of N_2 as required in the Definition, and the composition $N_1 \cdot N_2$.
In this example, $L_2 \sqcap R_1 = \{\mathsf{f}\}$, $R_1 \sqcap L_2 = \{\mathsf{d}\}$, $L_2 - R_1 = \{\mathsf{g}\}$ and $R_1 - L_2 = \{\mathsf{e}\}$.

As a further example. Fig. 3.3 shows the composition $N_1 \cdot N_1$ of N_1 with itself. As explained above, only disjoint nets can be composed, for which shared elements are copied. In Fig. 3.3, the copied elements are primed, going to the right instance of $N_1 \cdot N_1$. Furthermore, b < b' and d' < d. We follow the convention that the order $x < y$ of two identically labelled elements x and y of a port is expressed by placing x below y.

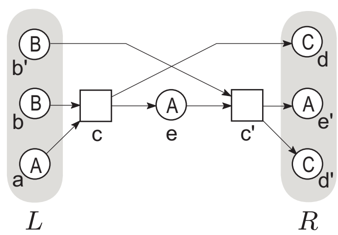


Fig. 3.3. The product $N_1 \cdot N_1$

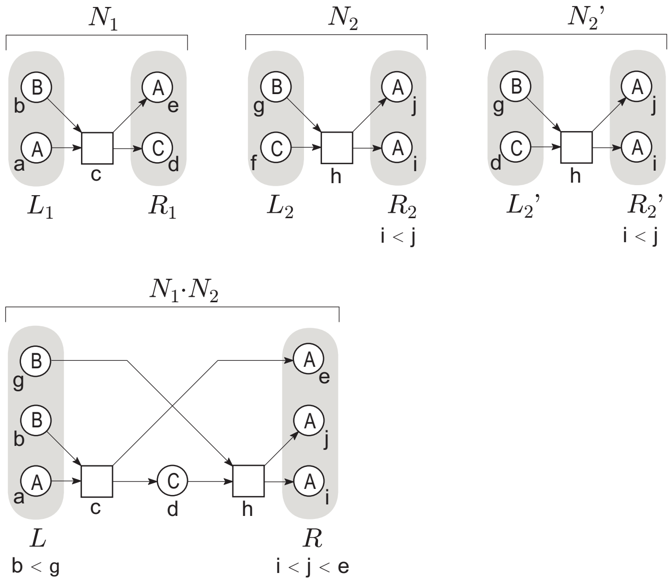


Fig. 3.2. Composition $N_1 \cdot N_2$ of N_1 with N_2

In Figs. 3.2 and 3.3, the two ports of N_1 are disjoint, as well as the two ports of N_2. Examples for intersecting ports will show up in Sec. 4.

In the terminology of the introduction, this composition operator certainly is sufficiently expressive, and definitely applies to *any* two nets (with interfaces). The following Theorem shows that composition is also associative:

Theorem 1. *For $i = 1, 2, 3$ let N_i be interface nets. Then $(N_1 \cdot N_2) \cdot N_3 = N_1 \cdot (N_2 \cdot N_3)$.*

Proof of this Theorem is postponed to the Appendix.

4 Examples Revisited

We turn back to the examples of Sec. 2, showing that the intuitive requirements discussed there are met by the composition operator of Sec. 3. To this end, each net of Sec. 2 is to be equipped by its left and right port.

4.1 Concurrent Runs

For concurrent runs such as in Fig. 2.3 we suggest an intuitively obvious convention for the identification of port elements:

The left port L_r and the right port R_r of a concurrent run r is the set of places p with empty pre- and postset, respectively. So, for each place p of r,

$$p \in L_r \text{ iff } {}^{\bullet}p = \emptyset, \qquad p \in R_r \text{ iff } p^{\bullet} = \emptyset. \tag{4.1}$$

For the runs of Fig. 2.2, L_{r_1} contains the C-labelled and the D-labelled places. The unique place of L_{r_3} and the unique place of R_{r_3} both are C-labelled.

The places of a concurrent run r of a 1-bounded system net N are labelled by the corresponding place of N. This labelling is unique at each port of r. In the composition $r \cdot r'$ of two concurrent runs r and r' we use the labels of R_r and $L_{r'}$ to identify equal elements (c.f. Sec. 3.1).

With these conventions, each concurrent run in fact *is* an interface net. The products $r_1 \cdot r_2 \cdot r_1$ and $r_1 \cdot r_3 \cdot r_2$ of the concurrent runs r_1, r_2 and r_3 in Fig. 2.2 yield what we expected in Sec. 2.1.

4.2 Service Nets

Fig. 2.5 sketched a number of examples of service nets. The reader may have wondered why all places of the interface are drawn to the right at the buyer, to the left at the warehouse and to both sides at the seller. He or she meanwhile may have guessed the graphical convention: The left and the right port of a service net are drawn on the left and right side, respectively, of the box that contains the inner elements.

Hence the buyer's left port is empty, and its right port includes four places, labelled order, invoice, payment and goods.

Likewise, the places of the warehouse's left port are labelled order and goods. Its right port is empty. The seller has no empty ports. Labels of places of the two ports may coincide. Both ports may even consist of identically labelled places, as e.g. the ports of the credit bank.

With these conventions, the service nets sketches in Fig. 2.4 and Fig. 2.5 *are* interface nets as defined in Sec. 3.2. Fig. 2.6 shows the composition of the seller- and the warehouse net.

Refraining from any inner aspects of service nets, the composition of the service nets of Fig. 2.7 follow the definition of Sec. 3.3.

The case of an additional, alternative warehouse W' of Fig. 2.8 deserves special attention. With shorthands A $=_{def}$ buyer $\cdot$ seller and W $=_{def}$ warehouse, what we need is a construct that would leave the right port R_{A} of A available in A $\cdot$ W. This is achieved by the variant W' of W that employs R_{A} not only as its left port, but coincidently also as its right port, i.e.

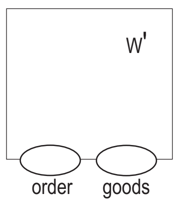


Fig. 4.1. Warehouse variant W' with ports $L_{\mathsf{W'}} = R_{\mathsf{W'}}$.

$R_\mathsf{A} = L_\mathsf{W} = L_\mathsf{W'} = R_\mathsf{W'}$. Consequently, the right port $R_\mathsf{W'}$ of $\mathsf{A} \cdot \mathsf{W'}$ is R_A. Hence, $\mathsf{A} \cdot \mathsf{W'} \cdot \mathsf{W}$ is the wanted service net!

The graphical convention for places p that belong coincidently to the left *and* right port of N is their position on the top- or bottom line of N, as for $\mathsf{W'}$ in Fig. 4.1. (Fig. 2.8 is not adequate in this sense).

Of course, a retailer with *three* warehouses can be achieved by a second instance $\mathsf{W'}$, i.e. the system $\mathsf{A} \cdot \mathsf{W'} \cdot \mathsf{W'} \cdot \mathsf{W}$.

Summing up, we have seen a nice new idea here: The places p of the right port R of a net N are not necessarily the "right edge" of N, with empty postset $p^\bullet$. Instead, as Fig. 2.4 shows for $p = \mathsf{invoice}$ and $p = \mathsf{goods}$, empty preset ${}^\bullet p$ is reasonable, too. More generally, *any* element p may belong to R.

This combines seamlessly with the transmission in a composition $N_1 \cdot N_2$ of a place p in R_{N_1} to $R_{N_1 \cdot N_2}$, in case $r \notin L_{N_2}$.

4.3 Branching Processes

The example of a second and third warehouse in Sec. 4.2 showed already how alternative access of two interface nets N_1 and N_2 to one port is organized: One of the nets (or both of them) extends its right port with its left port.

This idea is the basis to construct the branching process (Fig. 2.9) of the system net N (Fig. 2.1) by means of some concurrent runs of N. In contrast to Sec. 4.1, the ports of those runs do not canonically contain the places p with empty preset ${}^\bullet p$ or empty postset $p^\bullet$, respectively. Instead we define the ports explicitely.

Fig. 4.2 shows three concurrent runs, equipped with ports. The ports of N_0 and N_2 are as we know them for concurrent runs from Sec. 4.1 (though so far we never saw an isolated place such as the D-labelled one of N_2). However, the left and right ports of N_1 are identical! In particular, the right port may not meet everybody's intuition. But remember the right port's role of N_1: It provides the places to attach the successor run, N, in a composition $N_1 \cdot N$.

Fig. 4.3 shows the composition $N_0 \cdot N_1$. The right port of $N_0 \cdot N_1$ is isomorphic to the left port of $N_2 \cdot N_1$, also shown in Fig. 4.3. Consequently, the composition $(N_0 \cdot N_1) \cdot (N_2 \cdot N_1)$ is as shown in Fig. 4.4. To continue the argument, the branching process of Fig. 2.6 can now be written as

$$N_0 \cdot N_1 \cdot N_2 \cdot N_1 \cdot N_2 \cdot N_1. \tag{4.2}$$

The – unique – infinite branching process of the net N in Fig. 2.1 (of which Fig. 2.9 shows a prefix) can informally be written as

$$N_0 \cdot N_1 \cdot (N_2 \cdot N_1)^\infty$$

or, precisely, as the unique smallest solution of the equational system

$$X = N_0 \cdot N_1 \cdot Y$$
$$Y = N_2 \cdot N_1 \cdot Y,$$

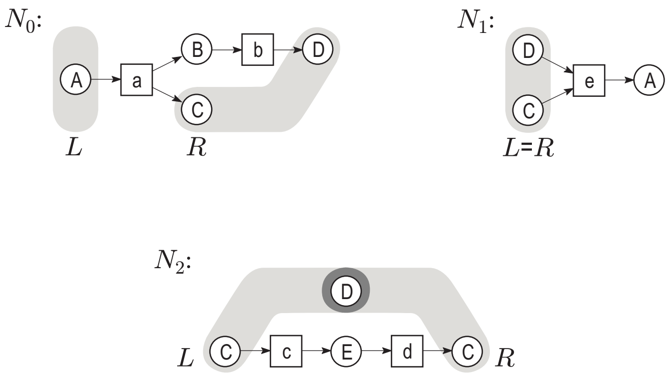


Fig. 4.2. three concurrent runs, equipped with ports

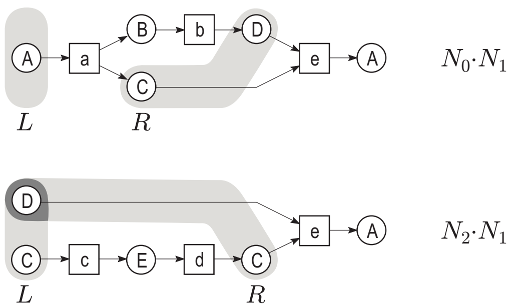


Fig. 4.3. The products $N_0 \cdot N_1$ and $N_2 \cdot N_1$ with their ports

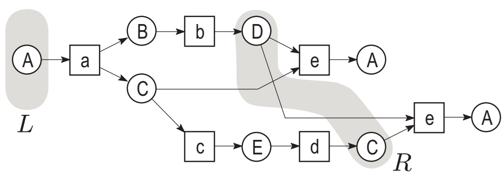


Fig. 4.4. The composition $N_0 \cdot N_1 \cdot N_2 \cdot N_1$ with its ports L and R

where X and Y are variables, ranging over infinite nets.

Summing up, the new idea of this section resembles the transmission of a place p from R_{N_1} to $R_{N_1 \cdot N_2}$, as in Sec. 4.1 and Sec. 4.2. This was achieved with p not belonging to L_{N_2}, hence with no access of N_2 to p. Here, however, p is a member of L_{N_2} *and* R_{N_2}, hence N_2 has access to p, and p still remains available for a net N_3 as in $N_1 \cdot N_2 \cdot N_3$.

4.4 Mixed Interfaces

Fig. 2.10 shows the example of a composed interface net, called business. The interfaces of the involved nets include high-level places, as well as transitions. This is fully covered by the definition of the composition operator in Def. 15. With the workforce and management interface nets as sketched in Fig. 4.5 (abstracting from the inner elements given in Fig. 2.10), the net of Fig. 2.10 reads

$$\mathsf{business} =_{def} \mathsf{workforce} \cdot \mathsf{management}.$$

For the variant of Fig. 2.11, management may be composed as

$$\mathsf{management} =_{def} \mathsf{staff} \cdot \mathsf{director},$$

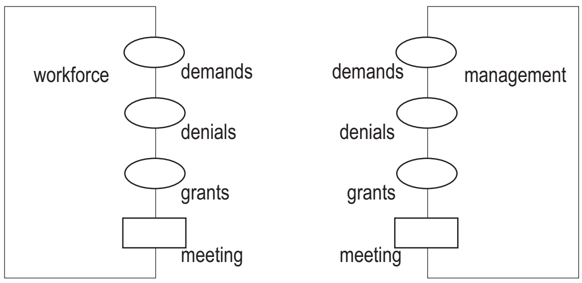


Fig. 4.5. interface nets workforce and management

with staff and director as in Fig. 4.6.

The director's view of the organization of the meeting is

$$\mathsf{workforce} \cdot \mathsf{staff}.$$

This interface net has an empty left port. Its right port contains a granted-labelled place and a meeting-labelled transition.

Summing up, the new idea of this section is based on the almost trivial observation that the composition makes no assumption at all on the nature of the port elements: They are not necessarily places, but also may be transitions, as well as high-level elements.

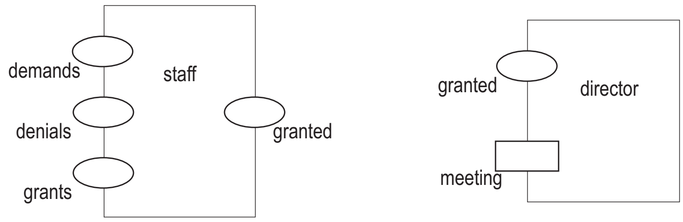


Fig. 4.6. interface nets staff and director

4.5 Nondeterminism

Nondeterministic composition $N_1 \cdot N_2$ of two interface nets N_i with ports L_i and R_i $(i = 1, 2)$, as discussed in Sec. 2.5, can be reduced to the case of more than one feasible identification of elements of R_1 with elements of L_2. This, in turn, occurs whenever the labelling of R_1 or of L_2 (or of both nets) is not unique, i.e. two elements of R_1 or of L_2 are identically labelled. As an example, the label A occurs twice in R_{r_1} of Fig. 2.13.

In technical terms, the port R_{r_1} of Fig. 2.13 is labelled, but no order is given for identically labelled elements. The labelling of R_{r_1} thus apparently gives rise to *two* index orders. Each of them yields a different index labelled net, behaving differently when composed with other nets. Fig. 4.7 employs the index notation

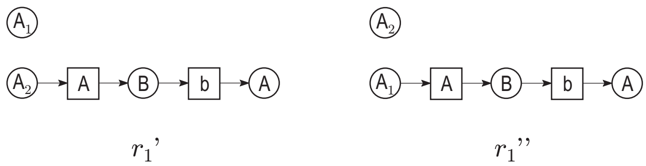


Fig. 4.7. Index labelled versions of r_1

to depict the two nets. Then $r_1' \cdot r_2$ is r_3 of Fig. 2.14, whereas $r_1'' \cdot r_2$ returns r_4 of Fig. 2.14.

5 Conclusion

The biggest obstacle for a definition of a universal composition operator are identically labelled elements: Even if each initially given net has only uniquely labelled elements, composition of the nets may yield equally labelled elements. The central idea of this paper is to impose an *order* onto those elements. This yields a unique definition of composition, which in addition is *associative*. Furthermore, if a port of a composed net gains equally labelled elements, the composition operator provides their required order.

The reader may wonder how the fairly simple notion of interface nets and their composition can blossom the rich expressive power, as demonstrated and exploited in this paper. This is due to the observation that each element of a port of an interface net N_1 may go into one or both ports of $N_1 \cdot N_2$, or into the inner of $N_1 \cdot N_2$. Furthermore, the product is not commutative.

A rich choice of composition operators for Petri Nets have been suggested during the recent decades. Typical examples include the process algebra like Petri Net Algebra of Best, Devillers and Koutny [1], or the composition of Petri Net modules [2]. A survey on composition (and refinement) operators give Gomes and Barros in [3]. Those operators are frequently not associative. Only few of them suggest composition along ports.

References

[1] E. Best, R. Devillers, M. Kounty: *Petri Net Algebra*, Springer Monographs in Theoretical Computer Science (2001)

[2] S. Christensen, L. Petruci: *Modular Analysis of Petri Nets*, Computer Journal Vol 43 No 3, pp 224-242 (2001)

[3] Luis Gomes, João Paulo Barros: *Structuring and Composability Issues in Petri Nets Modeling*, IEEE Transactions on Industrial Informatics Vol 1 No 2, pp 112-123 (2005)

A Proof of Theorem 1

Let N_1, N_2 and N_3 be interface nets. We have to show that both interface nets $(N_1 \cdot N_2) \cdot N_3$ and $N_1 \cdot (N_2 \cdot N_3)$ have the same elements, arcs and ports.

In this proof we employ some shorthands, writing
N_{xy} for $N_x \cdot N_y$,
L_x for L_{N_x} and
R_x for R_{N_x}.

For example, the Theorem's central claim then reads

$$N_{1(23)} = N_{(12)3}.$$

In the rest of this proof, let

$$a \in L_1 \cup L_2 \cup L_3.$$

We frequently skip mentioning a explicitely, writing for a set A of labelled elements

- $index(A)$ instead of $index(A, a)$, and
- $\overline{A}$ for the maximal index of elements $e \in A$ with $label(e) = label(a)$.

Some first observations:

Lemma 3. *Let A and B be labelled sets, let $a \in A$. Then $a \in A \sqcup B$ with $index(A \sqcup B) = index(A)$*

Proof follows immediately from Def. 10.

Lemma 4. *Let A, B and C be labelled sets. Then*
a) $\overline{A \sqcup B} = \overline{A} + \overline{B}$
b) $\overline{A - B} = \overline{A} - \overline{B}$

a) follows from Def. 10
b) follows from Def. 9 and Def. 8
c) follows from *a*) and *b*)

In the sequel, let $x, y \in \{1, 2, 3, 12, 23\}$.

Lemma 5. *Let $a \in inner(N_x)$. Then*
a) $a \in inner(N_{xy})$,
b) $a \in inner(N_{yx})$.

follows from Def. 15.

Lemma 6. *a) If $\overline{R_x} \geq \overline{L_y}$ then $\overline{L_{xy}} = \overline{L_x}$.*
b) If $\overline{R_x} \leq \overline{L_y}$ then $\overline{R_{xy}} = \overline{R_y}$.

a) $\overline{R_x} \geq \overline{L_y}$ implies $index(L_2 - R_1) = 0$ by Def. 8 and Def. 9. Hence the proposition.
b) follows by symmetrical arguments.

Lemma 7. *Let $a \in L_y$.*
a) $a \in inner(N_{xy})$ *if* $\overline{R_x} \geq index(L_y)$.
b) $a \in L_{yx}$ *with* $index(L_{xy}) = \overline{L_x} + index(L_y) - \overline{R_x}$ *if* $\overline{R_x} < index(L_y)$.

follows along the proof of Lemma 6, and Def. 15.

The next Lemma provides the core arguments of the proof of the Theorem.

Lemma 8. $L_{(12)3} = L_{1(23)}$

We distinguish three cases:

Case 1: $a \in L_1$ (A.1)

As a shorthand, let $n =_{def} index(L_1)$.
We draw two conclusions:

a) $a \in L_1 \sqcup (L_2 - R_1)$ with $index(L_1 \sqcup (L_2 - R_1)) = n$, by (A.1) and Lemma 3.
Then $a \in L_{12}$ with $index(L_{12}) = n$, by Def. 15.
Then $a \in L_{12} \sqcup (L_3 - R_{12})$ with $index(L_{12} \sqcup (L_3 - R_{12})) = n$, by Lemma 3.
Then $a \in L_{(12)3}$ with $index(L_{(12)3}) = n$, by Def. 15.

b) $a \in L_1 \sqcup (L_{23} - R_1)$ with $index(L_1 \sqcup (L_{23} - R_1)) = n$, by (A.1) and Lemma 3.
Then $a \in L_{1(23)}$ with $index(L_{1(23)}) = n$, by Def. 15.

Case 2: $a \in L_2$ (A.2)

We distinguish two subcases:

Case 2.1: $\overline{R_1} \geq index(L_2)$ (A.3)

We draw two conclusions:

a) $a \in inner(N_{12})$, by (A.3) and Lemma 7 *a*).
Then $a \in inner(N_{(12)3})$, by Lemma 5 *a*).
Then $a \notin L_{(12)3}$, by Def. 14.

b) $a \in L_2 \sqcup (L_3 - R_2)$ with $index(L_2 \sqcup (L_3 - R_2)) = index(L_2)$, by (A.2) and Lemma 3.
Then $a \in L_{23}$ with $index(L_{23}) = index(L_2)$, by Def. 15.
Then $a \in inner(N_{1(23)})$, by (A.3) and Lemma 7 *a*).
Then $a \notin L_{1(23)}$, by Def. 14.

Case 2.2: $\overline{R_1} < index(L_2)$ (A.4)

As a shorthand, let $n =_{def} \overline{L_1} + index(L_2) - \overline{R_1}$.
We draw two conclusions:

a) $a \in L_{12}$ with $index(L_{12}) = n$, by (A.4) and Lemma 7 *b*).
Then $a \in L_{12} \sqcup (L_3 - R_{12})$ with $index(L_{12} \sqcup (L_3 - R_{12})) = n$, by Lemma 3.
Then $a \in L_{(12)3}$ with $index(L_{(12)3}) = n$, by Def. 15.

b) $a \in L_2 \sqcup (L_3 - R_2)$ with $index(L_2 \sqcup (L_3 - R_2)) = index(L_2)$, by Lemma 3.
Then $a \in L_{23}$ with $index(L_{23}) = index(L_2)$, by Def. 15.
Then $a \in L_{1(23)}$ with $index(L_{1(23)}) = \overline{L_1} + index(L_2) - \overline{R_1}$, by (A.4) and Lemma 7 *b*).
Then $a \in L_{1(23)}$ with $index(L_{1(23)}) = n$.

Case 3: $a \in L_3$ (A.5)

We distinguish three subcases:

Case 3.1: $\overline{R_2} \geq index(L_3)$ (A.6)

We draw two conclusions:

a) $a \in inner(N_{23})$, by (A.6) and Lemma 7 *a*).
Then $a \in inner(N_{1(23)})$, by Lemma 5 *b*).
Then $a \notin L_{1(23)}$, by Def. 14.

b) $\overline{R_2} + \overline{(R_1 - L_2)} \geq index(L_3)$, by (A.6).
Then $\overline{R_2 + (R_1 - L_2)} \geq index(L_3)$, by Lemma 4 *a*).
Then $\overline{R_{12}} \geq index(L_3)$, by Def. 15.
Then $a \in inner(N_{(12)3})$, by (A.5) and Lemma 7 *a*).
Then $a \notin L_{(12)3}$, by Def. 14.

Case 3.2: $\overline{R_2} < index(L_3)$ (A.7)

and $index(L_3) \leq \overline{R_{12}}$ (A.8)

We draw two conclusions:

a) $a \in inner(N_{(12)3})$, by (A.8) and Lemma 7 *a*).
Then $a \notin L_{(12)3}$, by Def. 14.

b) $a \in L_{23}$ with $index(L_{23}) = \overline{L_2} + index(L_3) - \overline{R_2}$, by (A.7) and Lemma 7 *b*).
Then $a \in L_{23}$ with

$$\begin{aligned} index(L_{23}) &\leq \overline{L_2} + \overline{R_{12}} - \overline{R_2} && \text{(by (A.8))} \\ &= \overline{L_2} + \overline{(R_2 \sqcup R_1) - L_2} - \overline{R_2} && \text{(by Def. 15)} \\ &= \overline{L_2} + (\overline{R_2 \sqcup R_1}) - \overline{L_2} - \overline{R_2} && \text{(by Lemma 4 } b)) \\ &= \overline{L_2} + \overline{R_2} + \overline{R_1} - \overline{L_2} - \overline{R_2} && \text{(by Lemma 4 } a)) \\ &= \overline{R_1}. \end{aligned}$$

Then $a \in inner(N_{1(23)})$, by Lemma 7 *a*).
Then $a \notin L_{1(23)}$, by Def. 14.

Case 3.3: $\overline{R_{12}} < index(L_3)$ (A.9)

As a shorthand, let $n =_{def} \overline{L_1} + \overline{L_2} + index(L_3) - \overline{R_1} - \overline{R_2}$.
We draw two conclusions:

a) $a \in L_{(12)3}$ with $index(L_{(12)3}) = \overline{L_{12}} + index(L_3) - \overline{R_{12}}$,
by (A.9) and Lemma 7 *b*). (A.10)

We distinguish two subcases:

aa) $\overline{R_1} \geq \overline{L_2}$. Then

$$\begin{aligned} index(L_{(12)3}) &= \overline{L_1} + index(L_3) - \overline{R_{12}} && \text{(by (A.10) and Lemma 6 } a)) \\ &= \overline{L_1} + index(L_3) - (\overline{R_2} + (\overline{R_1} - \overline{L_2})) && \text{(by Def. 15)} \\ &= n. \end{aligned}$$

ab) $\overline{R_1} < \overline{L_2}$. Then

$$\begin{aligned} index(L_{(12)3}) &= \overline{L_{12}} + index(L_3) - \overline{R_2} && \text{(by (A.10) and Lemma 6 } b)) \\ &= \overline{L_1 \sqcup (L_2 - R_1)} + index(L_3) - \overline{R_2} && \text{(by Def. 15)} \\ &= \overline{L_1} + (\overline{L_2 - R_1}) + index(L_3) - \overline{R_2} && \text{(by Lemma 4 } a)) \\ &= \overline{L_1} + \overline{L_2} - \overline{R_1} + index(L_3) - \overline{R_2} && \text{(by Lemma 4 } b)) \\ &= n. \end{aligned}$$

b)

$$\begin{aligned} \overline{R_2} &\leq \overline{R_2} + (\overline{R_1 - L_2}) \\ &= \overline{R_2 \sqcup (R_1 - L_2)} && \text{(by Lemma 4 } a)) \\ &= \overline{R_{12}} && \text{(by Def. 15)} \end{aligned}$$

Then $\overline{R_2} < index(L_3)$, by (A.9).
Then $a \in L_{23}$ with $index(L_{23}) = \overline{L_2} + index(L_3) - \overline{R_2}$, by Lemma 7 b).
Then $a \in L_{1(23)}$, with

$$\begin{aligned} index(L_{1(23)}) &= \overline{L_1} + \overline{L_2} + index(L_3) - \overline{R_2} - \overline{R_1} && \text{(by Lemma 7 } b\text{))} \\ &= n. \end{aligned}$$

Definition 15 implies $L_{(12)3} \leq L_1 \cup L_2 \cup L_3$ as well as $L_{1(23)} \leq L_1 \cup L_2 \cup L_3$. Hence, the above three cases span all cases to prove Lemma 8.

Lemma 9. $R_{(12)3} = R_{1(23)}$

Proof of this Lemma proceeds by arguments entirely symmetrical to the proof of Lemma 8.

Lemma 10. $inner(N_{(12)3}) = inner(N_{1(23)})$

Proof:

Lemma 5 implies for each $a \in inner(N_1) \cup inner(N_2) \cup inner(N_3)$: $a \in inner(N_{(12)3})$ as well as $a \in inner(N_{1(23)})$.

In addition, Lemma 7 a) implies for $a \in L_2$: $a \in inner(N_{12})$ if $\overline{R_1} \geq index(L_2)$ and for $a \in L_3$: $a \in inner(N_{23})$ if $\overline{R_2} \geq index(L_3)$. Again, Lemma 7 a) then implies those elements to become inner elements of $N_{(12)3}$ and of $N_{1(23)}$.

Furthermore, exactly those elements belong to $L_2 \sqcap R_1$ and to $L_3 \sqcap R_2$, respectively, to be replaced by the corresponding elements of $R_1 \sqcap L_2$ and of $R_2 \sqcap L_3$, respectively, according to Def. 15, in both nets $N_{(12)3}$ and $N_{1(23)}$.

All other port elements go to ports of $N_{(12)3}$ and of $N_{1(23)}$, according to Lemma 8 and Lemma 9. This completes the proof of the Theorem.